20th EuSpRIG Annual Conference
“Spreadsheet Risk Management”

Browns Courtooms, Browns Covent Garden, London.
Thursday 11th & Friday 12th July 2019
European Spreadsheet Risks Interest Group

EuSpRIG 2019 Conference

Spreadsheet Risk Management

Conference Supported By:
ICAEW
Cardiff Metropolitan University
Spreadsheet Engineering
Systems Modelling
Operis

Proceedings Editor:
Dr Simon Thorne

July 2019

Browns Courtrooms, Browns Covent Garden, London.
EuSpRIG 2019 Conference

“Spreadsheet Risk Management”

EUSPRIG 2019 ORGANISING COMMITTEE

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

EUSPRIG 2019 PROGRAMME COMMITTEE

Dr 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 2019 CONFERENCE PROGRAMME

Barrister’s Court, Browns, 82-84 St Martin's Ln, Covent Garden, London WC2N 4AG, UK. Nearest Tube: Leicester square.

Thursday 11 July 2019

09.00-09:30 Arrival, Registration, Coffee and welcome

09:30 – 12:45 Session 1

09:30 Welcome from the chair

09:35 Developing Excel Thought Leadership
    David Lyford-Smith, ICAEW, Chartered Accountants’ Hall

10:15 Practical aspects of applying an End User Computing policy
    Roger Turner, Wesleyan Assurance Society, Birmingham

10:55 coffee break

11:20 A Case Study of Spreadsheet Use within the Finance and Academic Registry units within a Higher Education Institution
    Simon Thorne and Jamie Hancock, Cardiff Metropolitan University

12:00 Abstracting spreadsheet data flow through hypergraph redrawing
    David Birch, Nicolai Stawinoga, Jack Binks
    Bruno Nicoletti, Paul Kelly
    Imperial College London, Filigree Technologies, London

12:40 Lunch

13.30 – 17:30 Session 2

13:30 Implementation Strategies for Multidimensional Spreadsheets
    Paul Mireault, Founder, SSMI International, Honorary Professor, HEC Montréal
14:10 EQUS - helping to see formulae  
*Chris R. Roast, Sheffield Hallam University*

14:50 Tea break

15:10 From webtables to datatables  
*Mária Csernoch, University of Debrecen, Hungary*

15:50 Will Dynamic Arrays finally change the way Models are built?  
*Peter Bartholomew, MDAO Technologies Ltd*

16:30 Cubes an alternative to spreadsheets to mitigate risk and improve performance?  
*Marc BRAUN, Xcubes*

17:05 Does the teaching of spreadsheeting skills have the potential to be a significant global education policy intervention?  
*Morten Siersted, Visiting Research Fellow, Centre for Development Studies, University of Bath*

17:30 Conference Closes

Food and drink for the evening will be found locally, please come along and enjoy the post conference discussions. Details to be announced at the conference.

**Friday 12: Training Sessions**

09:00 – 10:45

Working through Modeloff example problems  
*David Lyford-Smith, ICAEW*

11:00 – 12:45

Dynamic Arrays in Excel  
*Charles Williams, Decision Models Ltd*
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Paper</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maria Csernoch, Piroska Biró</td>
<td>Are digital natives spreadsheet natives?</td>
<td>1</td>
</tr>
<tr>
<td>Paul Mireault</td>
<td>Structured Spreadsheet Modelling and Implementation with Multiple Dimensions - Part 2: Implementation</td>
<td>13</td>
</tr>
<tr>
<td>Roger Turner</td>
<td>Defining and Adopting an End User Computing Policy A Case Study</td>
<td>27</td>
</tr>
<tr>
<td>David Lyford Smith</td>
<td>Developing Excel Thought Leadership</td>
<td>39</td>
</tr>
<tr>
<td>Roger Turner</td>
<td>Practical aspects of applying an End User Computing policy</td>
<td>47</td>
</tr>
<tr>
<td>Simon Thorne and Jamie Hancock</td>
<td>A Case Study of Spreadsheet Use within the Finance and Academic Registry units within a Higher Education Institution</td>
<td>61</td>
</tr>
<tr>
<td>David Birch, Nicolai Stawinoga, Jack Binks, Bruno Nicoletti, Paul Kelly</td>
<td>Abstracting spreadsheet data flow through hypergraph redrawing</td>
<td>79</td>
</tr>
<tr>
<td>Paul Mireault</td>
<td>Implementation Strategies for Multidimensional Spreadsheets</td>
<td>103</td>
</tr>
<tr>
<td>Chris Roast</td>
<td>EQUS - helping to see formulae</td>
<td>115</td>
</tr>
<tr>
<td>Mária Csernoch</td>
<td>From weatables to datatables</td>
<td>127</td>
</tr>
<tr>
<td>Peter Bartholomew</td>
<td>Will Dynamic Arrays finally change the way Models are built?</td>
<td>149</td>
</tr>
<tr>
<td>Marc Braun</td>
<td>Cubes an alternative to spreadsheets to mitigate risk and improve performance?</td>
<td>161</td>
</tr>
</tbody>
</table>
PREFACE

Dear Colleagues,

You are very welcome to the Twentieth Annual Conference of the European Spreadsheet Risks Interest Group. Once again, we are fortunate to have our unique combination of real-world experience and academic research which is the distinguishing feature of our event.

The opening address from the Institute of Chartered Accountants shows how professional bodies think about structuring the learning needed for the safe use of spreadsheets. The case study from Wesley Assurance Society brings their story further forward from last year. Another case study from Cardiff lays the foundation for their journey. Two contributions, on web scraping and multidimensional structures, go into greater technical depth than before on data processing, and a paper and tutorial focus on the new dynamic array features of Excel which could change all this. Two papers focus on visualisations, a tool and a research project. We include in these proceedings three valuable papers from our 2018 conference that we were previously unable to publish.

This will be a packed schedule dealing with some topics that will demand your attention and stimulate debate that will go on both in informal meetings afterwards, and in our online discussion forum on Yahoogroups which has over 950 members. Currently we have about 300 unique visitors to our website per month.

It is my pleasure to once again acknowledge the keen work of our conference and programme organiser Dr Simon Thorne from Cardiff Met. The committee also depends upon the wise counsel and active support of David Colver of Operis, Grenville Croll of Spreadsheet Engineering, and Angela Collins (secretary). Representatives of these organisations have contributed a great amount of expertise in the organising of this conference, the publicity, the proceedings, and much more committee work in the background.

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

Patrick O'Beirne, Chair 2019

http://www.eusprig.org
https://groups.yahoo.com/neo/groups/eusprig/info
Are digital natives spreadsheet natives?

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

ABSTRACT
The present paper reports the results of testing first year students of Informatics on their algorithmic skills and knowledge transfer abilities in spreadsheet environments. The selection of students plays a crucial role in the project. On the one hand, they have officially finished their spreadsheet training – they know everything –, while on the other hand, they do not need any training, since they are digital natives, to whom digital skills are assigned by birth. However, we found that the students had serious difficulties in solving the spreadsheet problems presented; so low were their results that it allowed us to form broad tendencies. Considering computational thinking, algorithmic skills, and knowledge transfer abilities, it is clear that those students performed better who used algorithm-based, multilevel array formulas instead of problem specific, unconnected built-in functions. Furthermore, we can conclude that students, regardless of their birth date and digital generation assigned to them, are in great need of official, high-mathability, algorithm-based training with expert teachers.

1. INTRODUCTION

In the present paper we report the results of testing first year students of Informatics on their algorithmic skills, computational thinking, and knowledge transfer abilities in spreadsheet environments; in general, on their high-mathability problem-solving abilities (Biró & Csernoch, 2015a, 2015b). Furthermore, we questioned how the time spent on digital devices effects the skills and abilities mentioned and the students’ problem-solving strategies.

The introductory section outlines the heated debate about the skills and abilities of digital natives, the age group which is our concern in this paper. Following this the aims of our research are outlined, focusing on the future professionals of Informatics in spreadsheet environments. In the Sample section we provide some background information on the tested students relevant to the present paper, the tasks, and the skills required to solve the problems. The Methods section summarizes the testing methods applied and provides the details of the evaluation process. Finally, the Results and Conclusion sections detail the student’ achievement in the test, how the digital devices influence their problem-solving abilities, and based on these results guidance for the effective training of ‘digital natives’ are discussed.

1.1. Debates

A heated debate seems to have developed between believers and non-believers around the existence of the species known as ‘digital natives’ (Prensky, 2001) (Kirschner & De Bruyckere, 2017). The phenomenon of digital native was introduced by Prensky in his famous speech (Prensky, 2001), when he declared that being born after 1984 “endowed this growing group with specific and even unique characteristics that make its members completely different from those growing up in previous generations.” (Kirschner & De Bruyckere, 2017). However, this conceptualization – attributed to the huge software companies – derives from the assumed privilege enjoyed by the species of end-users – i.e.
users of “user-friendly” environments –, or “user-friendly-users” for short. In this context the software companies claim that being exposed to graphical interfaces and their selection of tools are sufficient to ensure effective computer problem-solving.

However, it is obvious that neither digital natives nor user-friendly-users can live up to their assumed characteristics. As regards spreadsheets, the European Spreadsheet Risk Interest Group (EuSpRIG) is devoted to revealing the consequences of the activities of untrained users and to finding solutions for handling the problems of error-prone documents (Horror Stories, 2017). In accordance with these efforts EuSpRIG introduced the “Twenty principles for good spreadsheet practice” in 2015 (TPDSP, 2015), the “Spreadsheet competency framework” in 2016 (SCF, 2016), and accepted and published the paper entitled “Edu-Edition Spreadsheet Competency Framework” in 2017 (Csernoch & Biró, 2018). All these interconnected research studies and documents focus on training user-friendly-users to understand better, to develop their computational thinking, to improve their problem-solving skills, and, in general, to make them more productive.

2. Aims

In this research, our aim was to measure with quantitative tools and methods how students of Informatics relate to spreadsheets problems, and how they transfer fundamental algorithms between problems and environments. We wanted to find connections between the students’ problem-solving and knowledge-transfer abilities (the sample is detailed in Section 3) in spreadsheet environments, considering both internal and external sources of knowledge and information. In close connection with this aim, we also were interested in finding out how students understand and build algorithms in this unofficial programming environment.

Our other aim was to measure how computer and mobile-device time – in the present context this means mobile devices functioning as computers – influence students’ performance in understanding spreadsheet problems, building algorithms, and coding them with spreadsheet formulas.

3. SAMPLE

3.1. Three Groups of Students of Informatics

Three groups of students of Informatics, during their first week of tertiary studies were tested at the University of Debrecen in the academic year of 2016/2017 as part of the TAaAS (Testing Algorithmic and Application Skills) project launched in 2011 (Csernoch et al., 2015). The three majors taught at the faculty made up the three groups: Software Engineering, System Engineering, and Business Informatics (SOE, SYE, and BIM, respectively) (for details of the majors see Csernoch et al., 2015).

As regards the students’ background knowledge, we know that they finished their high school studies, where Informatics is a compulsory subject with a varying – and untraceable – number of classes (Kerettanterv, 2013), that most of the students passed the maturation exams in Informatics at either the intermediate or/and advanced levels (Érettségi vizsga, 2017), with no significant differences between the three groups (Csernoch et al, 2015), and that many of them have passed the ECDL exams (ECDL Hungary, 2017).

In general, we were testing the future professionals of Informatics and Computer Sciences, who finished their official training in spreadsheets and other birotical environments; in short, the (future) professionals of the subject.
3.2. Tasks

The spreadsheet tasks involved in the research were presented in the form of

- a table with five fields of data and 235 records (Figure 1),
- five open questions to answer with spreadsheet formulas (Figure 2), and
- a spreadsheet array formula to decode, to answer with a complete natural language sentence (Figure 3).

![Table](image)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>Continent</td>
<td>Capital</td>
<td>Area</td>
<td>Population (thousand)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>Kabul</td>
<td>647500</td>
<td>27756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>Tirana</td>
<td>28748</td>
<td>3545</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>Algiers</td>
<td>2357140</td>
<td>32278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>American Samoa</td>
<td>Oceania</td>
<td>Pago Pago</td>
<td>199</td>
<td>69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Andorra</td>
<td>Europe</td>
<td>Andorra la Vella</td>
<td>408</td>
<td>08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Angola</td>
<td>Africa</td>
<td>Luanda</td>
<td>1246700</td>
<td>10590</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anguilla</td>
<td>America</td>
<td>The Valley</td>
<td>102</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yemen</td>
<td>Asia</td>
<td>Sanaa</td>
<td>527970</td>
<td>1670</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yugoslavia</td>
<td>Europe</td>
<td>Belgrade</td>
<td>102350</td>
<td>10957</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zambia</td>
<td>Africa</td>
<td>Lusaka</td>
<td>752614</td>
<td>9999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zimbabwe</td>
<td>Africa</td>
<td>Harare</td>
<td>305650</td>
<td>11377</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Figure 1](image) The sample table of the TaaAS spreadsheet test with the shaded G2 cell referring to the irrelevance of the value stored in it.

a) What is the capital city of the largest country?
b) What is the population density of each country?
c) How many African countries are in the table?
d) What is the average area of those countries whose population is smaller than G2?
e) How many countries have a surface area greater than G2?

![Figure 2](image) Questions of the test to be answered with spreadsheet formulas.

f) What is the result of the following formula?

\[ =\text{SUM}(\text{IF}(\text{B2:B236}="\text{Europe"},\text{IF}(\text{LEFT}(\text{A2:A236}="\text{A"},1)))) \]

![Figure 3](image) The decoding task of the test to be answered with a natural language sentence.

3.3. The Skills Required to Solve the Presented Problems

**Internal skills and knowledge**

- Information retrieval from the table:
  - A thorough data analysis of the table presented in Figure 1. In this case the data source is the table and no further external knowledge is required.
  - Recognizing that the population is stored in thousands.
- Information retrieval from the tasks:
  - Recognizing the input values of the tasks (Figure 1 and Figure 2).
  - Recognizing the output values of the tasks (Figure 1 and Figure 2).
  - Recognizing the G2 cell as a variable (Figure 1 and Figure 2).
– Recognizing the same algorithm in Tasks a–f (Figure 2 and Figure 3).
– Decoding (reading) the presented formula in Task f (Figure 3).

External skills and knowledge

– Understanding the expression “using a formula” to solve problems.
– Understanding the difference between formula-output and constant-output.
– Handling variables in spreadsheets.
– Understanding the difference between a constant and a variable.
– Task a:
  – Building or recalling the algorithm of the capital city problem.
  – Recalling the functions and their arguments to solve the problem.
  – Building multilevel functions.
– Task b:
  – How to calculate population density: building or recalling the algorithm.
  – How to expand the population density of one country: recalling how to a copy formula or create an array formula.
– Tasks c–e, solved with array formulas (AF):
  – Applying the algorithm of conditional summing and its modification according to the criteria.
  – Recalling the functions and their arguments to solve the problem (SUM(), AVERAGE(), IF()).
  – Building multilevel functions.
  – Recognizing that Task e is a two-step-generalization of Task c (inequality and variable).
  – Recognizing that Task d is a three-step-generalization of Task c (inequality, variable, and average instead of sum).
– Task c, solved with build-in formulas (BIF):
  – Recalling at least one of the suitable functions and its arguments (COUNTIF() / COUNTIFS() / DCOUNT() / DCOUNTA() / DGET()).
– Task d, solved with build-in formulas:
  – Recalling at least one of the suitable functions and its arguments (AVERAGEIF() / AVERAGEIFS() / DAVERAGE() / DGET()).
  – Recalling the syntax of handling inequality with variables.
– Task e, solved with build-in formulas:
  – Recalling at least one of the suitable functions and its arguments (COUNTIF() / COUNTIFS() / DCOUNT() / DCOUNTA() / DGET()).
  – Recalling the syntax of handling inequality with variables in the case of the *IF?() functions.
  – Recalling the database functions (DBF).
  – Creating the conditional grid to the database functions.
– Tasks f:
  – Recognizing the functions and their arguments (SUM(), IF(), LEFT()).
  – Recognizing multilevel formulas.
  – Understanding array formulas.

3.4. Methods

To see how the length of time students spend on computers and on mobile devices functioning as computers influences their problem-solving abilities and algorithmic skills in spreadsheets, the periods of time were recorded in the attitude test of the TAaAS project. For both sets of devices three options were offered: at least 5 hours every day (c5 and m5), at least 2 hours every day (c2 and m2), and less than 2 hours every day (c1 and m1), where
c stands for computers and m for mobile devices. Based on these categories, we referred to the students as heavy, moderate, and occasional users, respectively.

**Table 1. The number of students participating in the project and the time they spend on computers and mobile devices (functioning as computers).**

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>c5</th>
<th>c2</th>
<th>c1</th>
<th>m5</th>
<th>m2</th>
<th>m1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOE</td>
<td>120</td>
<td>50</td>
<td>62</td>
<td>7</td>
<td>27</td>
<td>50</td>
<td>41</td>
</tr>
<tr>
<td>SYE</td>
<td>103</td>
<td>47</td>
<td>51</td>
<td>5</td>
<td>23</td>
<td>52</td>
<td>28</td>
</tr>
<tr>
<td>BIM</td>
<td>97</td>
<td>32</td>
<td>52</td>
<td>11</td>
<td>42</td>
<td>36</td>
<td>19</td>
</tr>
</tbody>
</table>

**Table 2. The number of recognizable items of the TAaAS spreadsheet problems (BIF and AF refer to built-in functions and array formulas, respectively).**

<table>
<thead>
<tr>
<th>Task a</th>
<th>Task b</th>
<th>Task c</th>
<th>Task d</th>
<th>Task e</th>
<th>Task f</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIF</td>
<td>AF</td>
<td>BIF</td>
<td>AF</td>
<td>BIF</td>
<td>AF</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
</tbody>
</table>

Considering the students’ algorithm skills and knowledge transfer abilities it was not only the syntactically correct answers which were accepted, but the all identifiable algorithms, regardless of the language used. Any natural language algorithm or pseudo-code was considered and evaluated according to the predefined items.

In cases where either built-in functions or array formulas are acceptable the items were set up according to the skills and knowledge required in the specific cases (Table 2). We must note here that in the TAaAS project, in general, an extremely low number of students selected database functions (DBF), while in the three groups in 2016 none of the students tried them. Consequently, in the present paper the BIF abbreviation refers to the *IF?() functions.

The numbers of recognizable items of Tasks c–e clearly show that applying the array formulas requires the application of the same number of items, while using the built-in functions requires various numbers of items. (In Task c, the number of AF items is 10, compared to 9 in Tasks d and e, due to the string constant in the formula.)

4. RESULTS

The students’ scores were calculated in two different ways. First the correct answers were collected, where their numbers and average numbers were calculated. Table 3 clearly indicates that we can hardly find students who were able to complete these tasks. This method was then further tuned. In the following process the solutions were evaluated according to the predefined items, and the number of the correct items was calculated and used for the wider analyses.

**Table 3. The percentage of the students who solved the problems completely.**

<table>
<thead>
<tr>
<th></th>
<th>Task a</th>
<th>Task b</th>
<th>Task c</th>
<th>Task d</th>
<th>Task e</th>
<th>Task f</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOE</td>
<td>3.33%</td>
<td>1.67%</td>
<td>15.83%</td>
<td>1.67%</td>
<td>2.50%</td>
<td>25.00%</td>
</tr>
<tr>
<td>SYE</td>
<td>0.00%</td>
<td>0.97%</td>
<td>9.71%</td>
<td>0.00%</td>
<td>0.97%</td>
<td>27.18%</td>
</tr>
<tr>
<td>BIM</td>
<td>0.00%</td>
<td>0.00%</td>
<td>8.25%</td>
<td>2.06%</td>
<td>2.06%</td>
<td>14.43%</td>
</tr>
</tbody>
</table>
Table 4. The average results of the students based on the predefined items.

<table>
<thead>
<tr>
<th></th>
<th>Task a (%)</th>
<th>Task b (%)</th>
<th>Task c (%)</th>
<th>Task d (%)</th>
<th>Task e (%)</th>
<th>Task f (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOE</td>
<td>21.06</td>
<td>25.21</td>
<td>35.92</td>
<td>15.53</td>
<td>15.23</td>
<td>31.94</td>
</tr>
<tr>
<td>SYE</td>
<td>16.12</td>
<td>25.49</td>
<td>42.20</td>
<td>15.06</td>
<td>15.22</td>
<td>37.86</td>
</tr>
<tr>
<td>BIM</td>
<td>16.56</td>
<td>25.52</td>
<td>33.40</td>
<td>18.28</td>
<td>14.30</td>
<td>23.37</td>
</tr>
</tbody>
</table>

It is clear from the result tables (Table 3 and Table 4) that students have difficulties in solving these problems, considering the skills required in the test. As is detailed in the list of skills and knowledge, there are fundamentally three approaches to solve Tasks c–e: “hand-made” array formulas, built-in *IF?() functions, or built-in DBF() functions. Aware of these possible solutions and the students’ selections, we also compared the number and the results of the students selecting the BIF or the AF solutions.

4.1. Knowledge transfer

Table 5. The number of students selecting the BIF (*IF?() functions) or the AF solutions in Tasks c–e.

<table>
<thead>
<tr>
<th></th>
<th>Task c</th>
<th>Task d</th>
<th>Task e</th>
<th>BIF</th>
<th>AF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOE</td>
<td>47</td>
<td>14</td>
<td>20</td>
<td>1.875</td>
<td>0.3448</td>
</tr>
<tr>
<td>SYE</td>
<td>47</td>
<td>7</td>
<td>20</td>
<td>1.952</td>
<td>0.3448</td>
</tr>
<tr>
<td>BIM</td>
<td>33</td>
<td>9</td>
<td>11</td>
<td>1.875</td>
<td>0.3448</td>
</tr>
</tbody>
</table>

When we compare the usage of BIF and AF in Tasks c–e, which apply the same algorithm, it is clear from the data of Table 5 that the students’ choice of solution is highly determined by the level of generalization. In Task c, where equivalence is checked in the condition (which can be omitted in the BIF solution) and a string constant is used, the BIF/AF rate is the highest. Furthermore, we found that the higher the generalization level, the lower the BIF/AF rate: Task c > Task d > Task e (1.875, 1.0851, and 0.3448, respectively). This finding indicates that students apply the BIF solution only in simple problems and cannot expand this knowledge to more demanding problems.

Considering the average results of the students who worked with the problems, regardless of the students’ selection of methods – i.e. regardless of the BIF/AF rate – and regardless of the generalization level of the problem, the results of the AF solutions are found to be higher than those of the BIF solutions (Table 6) in all the three groups.

Table 6. The results of the students selecting BIF or AF solutions in Tasks c–e. Since the choice made was tested at this stage of the analysis, this table shows only the results of those who tried the tasks (Table 5).

<table>
<thead>
<tr>
<th></th>
<th>Task c (%)</th>
<th>Task d (%)</th>
<th>Task e (%)</th>
<th>BIF</th>
<th>AF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOE</td>
<td>57.45</td>
<td>30.71</td>
<td>42.50</td>
<td>47.62%</td>
<td>64.29%</td>
</tr>
<tr>
<td>SYE</td>
<td>66.67</td>
<td>34.29</td>
<td>44.32</td>
<td>62.96%</td>
<td>64.29%</td>
</tr>
<tr>
<td>BIM</td>
<td>63.64</td>
<td>37.78</td>
<td>42.48</td>
<td>58.29%</td>
<td>62.96%</td>
</tr>
</tbody>
</table>

Focusing on knowledge transfer abilities, we checked the correlation between the a-c BIF and between the a–c AF solutions. At this point we must note that the extremely low number of students who tried to solve the problems (Table 5) and their low average results (Table 3 and Table 4) requires further statistical analyses on different samples; however, the tendencies are clear and require further attention.
The correlation between the BIF solutions (Table 7) and the AF solutions (Table 8) were calculated and correlation matrixes were built based on the intervals of the weak (W), moderate (M), and strong (S) connections ([0, 0.3), [0.3, 0.7), and [0.7, 1], respectively).

**Table 7. The correlation matrix of the BIF solutions in Tasks c–e.**

<table>
<thead>
<tr>
<th>BIF</th>
<th>Students who tried (ST)</th>
<th>All students (SA)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SOE</td>
<td>SYE</td>
</tr>
<tr>
<td>C-D</td>
<td>W</td>
<td>M</td>
</tr>
<tr>
<td>C-E</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>D-E</td>
<td>W</td>
<td>S</td>
</tr>
</tbody>
</table>

The correlation matrixes can show tendencies relating to how the students are able to transfer their spreadsheet knowledge between problems sharing the same algorithm. The number of weak, middle and strong correlations were counted both in the BIF (Table 7) and the AF (Table 8) solutions. Since the low number of students who dealt with the problems is not enough to form firm conclusions, we will discuss the tendencies. The rate of the three correlation groups are the following, considering those students who at least tried to do something: 3:5:1 and 1:4:4 in the BIF and the AF groups, respectively. This result indicates that those students who use array formulas can transfer the background knowledge required by this type of coding more effectively than those who use the problem specific built-in functions.

**Table 8. The correlation matrix of the AF solutions in Tasks c–e.**

<table>
<thead>
<tr>
<th>AF</th>
<th>Students who tried (ST)</th>
<th>All students (SA)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SOE</td>
<td>SYE</td>
</tr>
<tr>
<td>C-D</td>
<td>W</td>
<td>M</td>
</tr>
<tr>
<td>C-E</td>
<td>M</td>
<td>S</td>
</tr>
<tr>
<td>D-E</td>
<td>M</td>
<td>S</td>
</tr>
</tbody>
</table>

In the case of decoding Task f, in general, we found no correlation between the BIF and AF users. In this respect the weak, middle, and strong correlations of the three groups show such different patterns that this analysis requires further samples to form conclusions. However, on this sample we had to reject our hypothesis that those who write array formulas can decode them, or vice versa. It is remarkable that even those who used AFs in solving Tasks c–e, were not able to transfer their knowledge to decoding in all the groups – no strong correlation was found between the two activities. At this stage of our analyses we can suggest that this result means that the students are code-, rather than algorithm-dependent, which will further be proved by the time spent on computers (Section 4.3). Another explanation would be that writing codes requires different skills than decoding and forming answers in natural language sentences, and students are trained to write codes rather than read them.

Considering Task b, we found that students have problems utilizing both external and internal knowledge. However, internal knowledge, information retrieved from the table (which is our main concern now) caused more problems than the question of how to calculate the population density. Students were not able to recognize that the population is presented in thousands in the table and that an array formula or one formula with copying should be created. This finding reveals that the students tested are not used to data analysis in spreadsheet environments.
Table 9. The results of Task a, considering the items and the order of the three functions of the multilevel function.

<table>
<thead>
<tr>
<th></th>
<th>Task a</th>
<th>Task a without MAX()</th>
<th>The order of INDEX(MATCH(MAX()))</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOE</td>
<td>21.06%</td>
<td>16.35%</td>
<td>18.33%</td>
</tr>
<tr>
<td>SYE</td>
<td>16.12%</td>
<td>11.65%</td>
<td>11.65%</td>
</tr>
<tr>
<td>BIM</td>
<td>16.56%</td>
<td>11.42%</td>
<td>12.37%</td>
</tr>
</tbody>
</table>

As is shown in Table 3, only an extremely low number of students were able to solve Task a correctly. Considering the items, most of the students recognized that the largest area can be calculated with the MAX() function. However, in general this is all they know. They cannot formulate the algorithm, nor build the multilevel function, and are not aware of the arguments of the INDEX() and the MATCH() functions (Table 9). Several students noted that they could solve the problem with computers, something which on the one hand, we cannot prove, while on the other hand it was not the aim of the present analyses. However, our teaching experience clearly shows that students who have passed either the maturation exams or the ECDL exams are not familiar with the algorithm for finding an item in a vector and writing out the corresponding value from another.

We also must note that even though the table structure ruled out the use of both VLOOKUP() and HLOOKUP() several students selected them in their solutions.

4.2. Misconceptions

We cannot ignore the misconceptions revealed, both in our teaching experience and in this test. However, we are glad to report that the number of those who claimed that the formula of Task f is incorrect is significantly lower compared to previous years and compared to the results of the teachers of Informatics, a tendency which is rather promising. In a similar way, in Task a, the number of students who wanted to solve the problem with the BIF HLOOKUP() or VLOOKUP() functions decreased, and in this analysis is below 10%.

At this point, we also must note that the helps available for the MATCH() function, 20 years after MS Excel was published, are still incorrect and inconsistent, stating that the second argument of the MATCH() function can be any range “lookup_array. Required. The range of cells being searched.” (MATCH function, 2017) or an array (Figure 4), instead of a vector (Csernoch, 2017; Csernoch & Biró, 2018). These errors do not help the understanding of the search algorithms and the application of the MATCH() function.

Figure 4. The wizard of the MATCH() function in Excel2016.
4.3. **Time Spent on Computers and Mobile Devices**

Our other major concern was how the use of computers and mobile devices influences the skills in question. As mentioned in the Methods section and shown in Table 1, we formed three groups of students based on the time they spend on computers and mobile devices functioning as computers.

In all the groups in all the tasks – BIF and AF solutions are not separated in this step of the analysis – we compared the results. The average results of the six groups – c5, c2, c1 and m5, m2, m1 – were calculated for each task. Based on the average results, computer and mobile use were separated, and for both categories the ranks of the averages were established. Finally, the sums of these ranks were calculated and mapped in 3D column charts.

We ran a competition and applied the method of those sports where the highest point is the best rank. The highest result in the task is the best and marked with the lowest rank. In the diagram, they are the lowest columns (Figure 5 and Figure 6). The three groups are mapped one by one and finally their summed ranked is mapped as 3G. The ranking reveals minor differences in the comparison of the three groups (with no significant differences), so, we can form our conclusions considering the three groups together.

Considering the use of computers, it is clear that they have positive effects on the results of solving the problems presented by using the skills and knowledge tested (Figure 5), however mainly on creating formulas. The heavy computer user students achieved the highest results with the best ranks, while the occasional users scored the lowest with the worst ranks. The BIM group is somewhat different from the other two groups and from the average, since it is stronger in the moderate users than in the heavy users.

![Figure 5. The summed ranks of the results, in comparison with time spent on computers.](image-url)
Our further question was whether ‘artificial hands’, i.e. mobile devices, have positive effects on the students’ spreadsheet skills, or not. Do they need formal training in the subject or do smart mobile devices provide enough tools, help, and aid?

In this regard, we have found that mobile devices do not improve the spreadsheet skills of the students tested. However, the good news is that we have not encountered any negative effects, either (Figure 6). In general, these devices do not help the development of computational thinking, algorithmic skills, and knowledge transfer abilities among this group as was claimed by Prensky (2001). We can conclude that these students need training (Kirschner et al., 2006), guidance from expert teachers (Hattie, 2003, 2012) who believe in the incremental nature of science (Chen et al., 2015), and are well trained in all aspects of TPCK (Technological, Pedagogical, and Content Knowledge, Mishra & Koehler, 2006); these teachers are not mobile devices, since they do not provide sufficient help.

We also must note that we have to complete further analyses on further samples since the average results of the students were so low that unsolved problems greatly affect the results.

5. CONCLUSION

Our analyses reveal the tendencies regarding the algorithm skills and knowledge transfer abilities of the students tested. It was found in the conditional summing tasks that the use of problem specific built-in functions requires mostly unconnected, non-conventional knowledge, and that the application of these functions constitutes non-transferable knowledge. The application of array formulas, however, requires the knowledge or recognition of the shared algorithm and reveals identifiable connections between the three tasks, which is in accordance with the skills preferred in the E²SCF rather than the original SCF. On the other hand, even the skill of creating array formulas is not enough to decode and read similar formulas.
In a similar way, the results of the students, especially in calculating the population density, proved that information retrieval from tables should be in the focus from the very beginning of spreadsheet studies, as is suggested in the E²SCF. Without this skill, the students’ external knowledge cannot be built into spreadsheet formulas.

Considering the effect of the time spent on computers and mobile devices functioning as computers, we have found that computers have a positive effect on the skills and abilities tested, while mobile devices are neutral. These findings do not prove the ideas of Prensky (2001) that digital natives are born with digital devices and born with the skills required for their intelligent use, but rather support the ideas of Kirschner & De Bruyckere (2017) that digital skills have to be taught and gained through well-structured training. This latter finding is in complete accordance with our findings considering the knowledge transfer, algorithmic skills, and computational skills detected in the solutions of our tested students.

6. REFERENCES


Structured Spreadsheet Modelling and Implementation with Multiple Dimensions - Part 2: Implementation

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

ABSTRACT

In Part 1, we showed how to develop a conceptual model of a problem involving variables of multiple dimensions, like Products, Regions, Sectors and Months. The conceptual model is presented as a Formula Diagram, giving a global view of the interaction between all the variables, and a Formula List, giving a precise view of the interaction between the variables. In this paper, we present precise steps to implement a multi-dimensional problem in a way that will produce a spreadsheet that is easy to maintain.

1 Introduction

Dimensions are an integral part of many models we use every day. Without thinking about it, we frequently use the time dimension: many financial and accounting spreadsheets have columns representing months or years. Representing a second dimension is often done by repeating blocs of formulas in a worksheet or creating multiple worksheets with the same structure. Adding a third or a fourth dimension is a perilous operation, resulting in structures that are hard to understand and maintain.

Figure 1 United Fruits 3-Dimension Spreadsheet

Problems with the same dimensions are implemented with different structures depending on the developer’s preferences. Figure 1, from Brandewindere (2018), shows an implementation with three
dimensions: Product, Location and Quarter. The developer decided to implement the products in different worksheets, the quarters in columns and the locations in repeated blocks where we can see three variables. Figure 2, from the example on the use of the Index function, Microsoft (2018), shows an implementation where the developer decided to implement one variable in two-dimensional tables for the Location and Quarter dimensions, replicated for each Product.

Past research has studied the problem of multiple dimensions in spreadsheet. Cunha, Fernandes, Mendes, Pacheco, and Suraiva (2012) and Cunha, Erwig, Mendes, and Saraiva (2016) have shown how to create an entity-relationship model from an existing spreadsheet.

Other research has focused on building a conceptual model before implementing the spreadsheet. Rajalingham, Chadwick, and Knight (2001) proposed graph-based conceptual model and Mireault (2017c) described a diagram-based conceptual model and introduced a representation of 1-dimensional variables.

In Part 1, Mireault (2017b), we described how to develop a conceptual model of a multidimensional problem using a small case, Acme Techno Widgets, which is reproduced in the Appendix. The conceptual model is composed of a Formula Diagram and a Formula List. In the next section, we describe the important concepts of data warehouse design that we use in the Excel implementation of the conceptual model. In Section 3, we will then describe the steps needed to create the structure spreadsheet. Then, in Section 4, we will illustrate the work necessary to modify the spreadsheet when we add a new element to a dimension. Finally, we will conclude with a discussion of the work involved in implementing multi-dimensional spreadsheets.

2 Data Warehouse Design and Implementation Concepts

We base our structured implementation on principles used in designing data warehouses. One common data structure is called the star schema. A star represents a multidimensional table, called a fact table, linked to other tables, called dimension tables. A dimension table contains a primary key, which identifies a precise row of table, and characteristics, which describe the dimension element itself. A constellation schema is an extension of the star schema that has multiple fact tables sharing dimension tables (Vaisman and Zimányi (2014)).

![Constellation Schema](image)

In Figure 3, we see that the primary keys are underlined. The Sales table has a primary key composed of ProductKey, StoreKey, PromotionKey and TimeKey. Each of the components serves as a foreign key (in the parent table Sales) to refer to the primary keys of children tables Product, Store, Promotion.
and Time. Thus, even though the Sales table only contains an Amount and a Quantity, the foreign keys let us associate them to a specific store, product, promotion and moment in time.

In the SSMI implementation, we will consider each repeating group as either a dimension table, a fact table, or both. It will be considered as a fact if it has at least one calculated variable. Base repeating groups, those having a dimension set of size 1, will correspond to dimension tables.

Fact tables have a primary key and one foreign key for each dimension table they are joined to. In our implementation, we will use the same concepts, as described in Mireault (2017a).

We can reference the parent of a foreign key with an INDEX-MATCH formula:

$$\text{INDEX}(\text{Value}, \text{MATCH}(\text{Foreign Key}, \text{Primary Key}, \text{exact match code}))$$

where Primary Key and Value are in the parent worksheet, and Foreign Key is in the child worksheet.

3 Spreadsheet Implementation

3.1 Naming Convention

In databases, we often see that the foreign key in a child table has the same name as the primary key of the parent table it is linked to. There is no confusion because a name is associated with the table it belongs to. Thus, the two names CLIENT_ID are in fact ORDER.CLIENT_ID and CLIENT.CLIENT_ID, and the full names must be used whenever the two tables are used in the same operation.

In Excel, names can either have a global scope, called Workbook, or a local scope, associated to a specific worksheet, as shown in Figure 4(a). When we use the Create from selection button, Excel always creates a name with the Workbook scope if there isn’t already one, in which case it will create it in the worksheet’s scope.

Since Excel does not provide a way to change the scope of a name after its creation, as shown by the grey field in Figure 4(b), we decided to always use global names in an SSMI implementation. This means that we need to avoid using the same name more than once. Therefore, we use the following naming convention for primary keys and foreign keys:

- A primary key is defined in its principal data worksheet, like Client ID in the Client Data worksheet and Order ID in the Order Data worksheet.
- A foreign key starts with the name of the primary key it refers to, followed by “in” and the name of its worksheet, like Client_ID_in_Order in the Order Data worksheet.

By using this naming convention, all reference formulas have the following form:

$$\text{INDEX}(\text{Variable}, \text{MATCH}(\text{Foreign Key}, \text{Primary Key}, \text{Exact match code}))$$

Variable and Primary Key are in the same dimension set. Foreign Key is in the current dimension set, i.e. the one in which the INDEX reference function is located. Figure 5 shows an example where we extract the name of the client of order 25.
Aggregate functions can use the family of `IF` and `IFS` functions provided in Excel: `AVERAGEIF`, `AVERAGEIFS`, `COUNTIF`, `COUNTIFS`, `MAXIFS`, `MINIFS`, `SUMIF` and `SUMIFS`.

### 3.2 Model Management Variables

To reduce the possibility of having an incorrect number of columns, we can use model management variables.

1. For each dimension set, we can calculate the number of columns it needs. We can also calculate the coordinate of the last column, taking into account the variable labels in column A and any needed initialization columns.
2. For each worksheet, we can verify that it has the correct number of columns with a count of its primary key.
3. For each variable from a different dimension set, since we usually reference its primary key and its foreign keys,

In the structured implementation methodology, we recommend copying whole columns, encompassing all the variables at the same time, we can perform validation 2 once per worksheet. Since the use of aggregate formulas is rarer, we would then perform validation 3 each time it is used.

### 3.3 Structured Implementation

The steps regarding the structured implementation have been developed to facilitate the model's implementation and its maintenance. The basic rule is that a worksheet only contains the definition formulas of variables belonging to the same dimension set. This makes it easier to make sure that we have the correct number of columns in each worksheet.

This doesn't mean that all the variables used in the worksheet have the same number of columns: variables that are used in an aggregate formula are usually from a dimension set with more columns.

#### 3.3.1 Determine the needed worksheets

From the Formula Diagram, we first determine the data sheets by listing all the dimension sets that have at least one data variable or input variable. Dimensionless data and input variables appear in the Data worksheet. The other variables are in worksheets named by their dimension set followed by the word Data.

**Table 1 The Data worksheets**

<table>
<thead>
<tr>
<th>Worksheet</th>
<th>Data or Input Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Base Price and Monthly Fixed Cost</td>
</tr>
<tr>
<td>Product Data</td>
<td>Base Price Multiplier and Unit Production Cost</td>
</tr>
<tr>
<td>Sector Data</td>
<td>Rebate Percentage, DemPar A and DemPar B.</td>
</tr>
<tr>
<td>Region Data</td>
<td>Unit Delivery Cost</td>
</tr>
<tr>
<td>Sector-Product Data</td>
<td>Product Distribution per Sector</td>
</tr>
<tr>
<td>Sector-Region Data</td>
<td>Region Sales Distribution per Sector</td>
</tr>
<tr>
<td>Month-Sector Data</td>
<td>Monthly Sales Distribution per Sector.</td>
</tr>
</tbody>
</table>
Then, we determine the model sheets from the dimension sets that have at least one calculated or output variable. Dimensionless calculated and output variables are defined in the Model worksheet. The other variables are defined in worksheets named by their dimension set.

### Table 2 The Model worksheets

<table>
<thead>
<tr>
<th>Worksheet</th>
<th>Calculated or Output Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sector</td>
<td>Sector Price Factor, Sector Base price and Sector Annual Demand Units.</td>
</tr>
<tr>
<td>Product-Region</td>
<td>PR Unit Cost</td>
</tr>
<tr>
<td>Sector-Product</td>
<td>Annual Sector-Product Unit Sales, and Annual Sector-Product Sales Amount</td>
</tr>
<tr>
<td>Month-Sector-Product</td>
<td>MSP Unit Sales and MSP Sales Amount</td>
</tr>
<tr>
<td>Month-Sector-Product-Region, shortened to MSPR</td>
<td>MSPR Unit Sales and MSPR Variable Cost</td>
</tr>
<tr>
<td>Month</td>
<td>Monthly Variable Cost, Monthly Unit Sales, Monthly Sales Amount, Monthly Costs and Monthly Profit</td>
</tr>
<tr>
<td>Month-Product-Region</td>
<td>MPR Unit Sales</td>
</tr>
<tr>
<td>Month-Product</td>
<td>MP Unit Sales and MP Sales Amount</td>
</tr>
<tr>
<td>Model</td>
<td>Total Profit</td>
</tr>
</tbody>
</table>

#### 3.3.2 Determine the Primary keys and the Foreign keys

To avoid having to manipulate names that are very long, we will use a dimension’s initial when we create primary and foreign key names. Thus, we will use PR in MSPR to refer to Product-Region in Month-Sector-Product-Region and SP in MSP to refer to Sector-Product in Month-Sector-Project.

Every dimension set used needs a primary key: Product, Sector, Region, Month, Sector-Product (SP), Month-Sector (MS), Sector-Region (SR), Month-Product (MP), Product-Region (PR), Month-Sector-Product (MSP), Month-Product-Region (MPR) and Month-Sector-Product-Region (MSPR).

Every time a formula uses a variable from another dimension-set, we need a foreign key. There are two cases: aggregate and non-aggregate formulas.

An aggregate formula, like Monthly Unit Sales = SUM(MSPR Unit Sales), has a resulting dimension set, Month and a starting dimension set, Month-Sector-Product-Region. We will create a Foreign Key with the form Resulting Dimension set in starting Dimension set. (As explained in Part 1, the Resulting Dimension Set is a subset of the Starting Dimension Set.) Table 3 lists the foreign keys created in this step.

### Table 3 Foreign Keys created from aggregate formulas

<table>
<thead>
<tr>
<th>Var No</th>
<th>Foreign Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>M in MSPR</td>
</tr>
<tr>
<td>23</td>
<td>(Same as 22)</td>
</tr>
<tr>
<td>24</td>
<td>M in MSP</td>
</tr>
<tr>
<td>28</td>
<td>MPR in MSPR</td>
</tr>
<tr>
<td>29</td>
<td>MP in MSP</td>
</tr>
<tr>
<td>30</td>
<td>(same as 29)</td>
</tr>
</tbody>
</table>

### Table 4 Foreign Keys created from non-aggregate formulas

<table>
<thead>
<tr>
<th>Var No</th>
<th>Foreign Key (variable requiring the foreign key)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>P in PR (Unit Production Cost) and R in PR (Unit Delivery Cost)</td>
</tr>
<tr>
<td>13</td>
<td>S in SP (Sector Annual Demand Units)</td>
</tr>
<tr>
<td>14</td>
<td>S in SP (Sector Base Price) and P in SP (Base Price Multiplier)</td>
</tr>
<tr>
<td>18</td>
<td>SP in MSP (Annual Sector Product Unit Sales) and MS in MSP (Monthly Sales Distribution per Sector)</td>
</tr>
<tr>
<td>19</td>
<td>(same as 18)</td>
</tr>
<tr>
<td>20</td>
<td>MSP in MSPR (MSP Unit Sales) and SR in MSPR (Region Sales Distribution per Sector)</td>
</tr>
<tr>
<td>21</td>
<td>PR in MSPR (PR Unit Cost)</td>
</tr>
</tbody>
</table>

In a non-aggregate formula, like Price = Sector Base Price * Base Price Multiplier, the resulting dimension set, Sector-Product is the union of the dimension sets of its components, Sector.
for **Sector Base Price** and **Product for Base Price Multiplier**. We will create the foreign keys with the form *Starting Dimension Set in Resulting Dimension Set*. Table 4 lists the Foreign Keys created.

Figure 6 shows the Primary Key for dimension set *Month-Sector-Product*, **MSP**, and all its possible Foreign Keys. In our example, we only need Foreign Keys **MS in MSP** and **SP in MSP**.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP</td>
<td>Jan-G-S</td>
<td>Jan-D</td>
<td>Jan-M-S</td>
<td>Jan-M-D</td>
<td>Jan-P-S</td>
<td>Jan-D</td>
<td>Jan-E-S</td>
<td>Jan-D</td>
<td>Feb-G-S</td>
<td></td>
</tr>
<tr>
<td>M in MSP</td>
<td>Jan</td>
<td>Jan</td>
<td>Jan</td>
<td>Jan</td>
<td>Jan</td>
<td>Jan</td>
<td>Jan</td>
<td>Jan</td>
<td>Feb</td>
<td></td>
</tr>
<tr>
<td>S in MSP</td>
<td>G</td>
<td>G</td>
<td>M</td>
<td>M</td>
<td>P</td>
<td>P</td>
<td>E</td>
<td>E</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td>P in MSP</td>
<td>S</td>
<td>D</td>
<td>S</td>
<td>D</td>
<td>S</td>
<td>D</td>
<td>S</td>
<td>D</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>MS in MSP</td>
<td>Jan-G</td>
<td>Jan-G</td>
<td>Jan-M</td>
<td>Jan-M</td>
<td>Jan-P</td>
<td>Jan-P</td>
<td>Jan-E</td>
<td>Jan-E</td>
<td>Feb-G</td>
<td></td>
</tr>
<tr>
<td>MP in MSP</td>
<td>Jan-S</td>
<td>Jan-D</td>
<td>Jan-S</td>
<td>Jan-D</td>
<td>Jan-S</td>
<td>Jan-D</td>
<td>Jan-S</td>
<td>Jan-D</td>
<td>Feb-S</td>
<td></td>
</tr>
<tr>
<td>SP in MSP</td>
<td>G-S</td>
<td>G-D</td>
<td>M-S</td>
<td>M-D</td>
<td>P-S</td>
<td>P-D</td>
<td>E-S</td>
<td>E-D</td>
<td>G-S</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 6 Primary Key MSP and all its possible Foreign Keys*

Constructing the Primary Keys and Foreign Keys can be as simple as importing them from a database. They can also be constructed using the technique presented in Mireault (2016).

### 3.3.3 The Model Worksheets

Implementing the formulas from the Formula List is straightforward. Following the SMMI implementation methodology, every variable’s definition formula is presented in a block with the top part consisting of references to the variables used in the definition and the bottom part is the formula using the cells just above.

We need to consider the following cases:

- **Non-aggregate formula.**
  - All variables are from the resulting variable’s dimension set. This is the case of calculating *Annual Sector-Product Sales Amount* = *Annual Sector-Product Unit Sales * Price*.
  - All variables are from a subset of the resulting variable’s dimension set. This is the case of **Price** = **Sector Base Price + Base Price Multiplier**.
  - There can be a mix of the two, with some variables from the same dimension set as the result and others from a subset. This is the case of **Annual Sector-Product Unit Sales** = **Sector Annual Demand Units * Product Distribution per Sector**.

- **Aggregate formula.** This is the case of **MP Unit Sales** = **SUM(MSP Unit Sales)**.

The three examples of non-aggregate formulas presented above come from the *Sector-Product* worksheet, illustrated in Figure 7.
In the case of non-aggregate formulas, the reference formula to a variable of the same dimension set is simply the name of the variable, as illustrated in rows 6, 13 and 14 of Figure 8. The reference formulas of variables defined in a subset of the current dimension set, Sector Annual Demand and Sector Base Price in Sector and Base Price Multiplier in Product, use the INDEX–MATCH form presented in section 3.1. This is illustrated in rows 5, 9 and 10 of Figure 8.

Implementing an aggregate calculation follows a similar block structure, as shown in Figure 9. Row 6 is not used in the calculation: it only serves in documenting the dimension set of the variable that is being aggregated, MSP Unit Sales in this case. The coloured highlights show the mechanics of the SUMIF formula being used to perform the calculation: we show the cells in used in the criteria in green and the cells used in the calculation in yellow.

Figure 10 shows the formula view of the calculation. In each column of row 10, the SUMIF function scans row 8 looking for values that are equal to the value of row 9 above and adds the values of row 7 when it finds them.
Figure 9 also illustrates the use of model management formulas discussed in section 3.2. All the variables belonging to dimension set Month-Product extend to column Z and all those belonging to Month-Sector-Product to column CT.

### 3.3.4 Interface Worksheet

The Interface worksheet is where the users will interact with the spreadsheet model. It is where they will enter the value of the Input variables and observe the resulting values of the Output variables. This is where spreadsheet developers can prepare dashboards, tables and charts.

Presenting the values of dimensionless or 1-dimension variables, like **Total Profit** and **Total Monthly Sales**, is straightforward: they can simply be reference formulas.

We need an area to prepare the codes that will be used in one INDEX-MATCH references, in the same structure that will be used the presentation.

With one dimension, it's straightforward.

With Two-dimensions, we need to decide which dimension will be in rows and which one will be in columns.

With Three dimensions, we have no choice but to present them in blocks. We need to decide which dimension will be repeated in blocks: the one with the smallest number of values is usually a good choice because it reduces the number of blocks that need to be repeated.

To prepare the primary keys, we set up the base codes in the visual structure we want, and we build the primary key of the dimension set by concatenating the values of the dimension codes, as illustrated in Figure 11(a). In the presentation area, we use a reference formula using the relative coordinate of the primary key we built in the preparation worksheet: \( = \text{INDEX}(\text{value}, \text{MATCH}([\text{presentation code}], \text{Primary key of Dimension set}, \text{Exact match code})) \), as shown in Figure 11(b).

### 4 Maintaining a structured implementation

In this section, we present how to use the Formula Diagram to determine the exact operations needed to change the number of instances in a structured implementation.
When you change the number of members (instances) in a basic dimension, you know exactly which worksheets to modify.

- The basic dimension worksheet
- All the worksheet with a dimension set containing the base dimension
- All the worksheet variables with an aggregate calculation using a variable from a worksheet above. They are easy to recognize in the Formula Diagram as variables calculated with an arrow coming from repeating sub-model with the base dimension

![Figure 12 Impact of adding a Sector](image)

In our example adding a sector requires the following operations, as illustrated in Figure 12:

- Add one new column in sheets Sector and Sector-Data
- Add the appropriate number of columns in Sector-Products, Month-Sector, Month-Sector-Product, Sector-Region, Month-Sector-Product-Region and their corresponding Data worksheets. (The number columns to add is the product of the cardinalities of the other dimensions. For the dimension set Month-Sector-Product it is: Number of Months × Number of Products. We could also use model management variables that calculate the last column of each dimension set, as explained in section 3.2 above.)
- Add columns to the aggregation formulas of variables MP Sales Amount, MP Unit Sales, MPR Unit Sales, Monthly Variable Cost, Monthly Sales Amount and Monthly Unit Sales. There is no need to change the formulas.

5 Conclusion

In this paper, we use proven concepts from data warehouse design and software engineering to implement spreadsheets that are hard to implement.
As shown in the Introduction, developers tend to develop their spreadsheet according to the way they want to present it to the user. We use the three-tier architecture, from software engineering, to develop the calculations separately from their presentation.

Savage (1997) described two important problems with using dimensions in spreadsheets. First is scalability, which involves changing the cardinality of a dimension. He concluded that spreadsheets rarely scale well. Second is hyper-scalability, which involves changing the dimensions themselves, such as adding more dimensions. His conclusion was, succinctly, “Forget it”.

By using a structure similar to well established data warehouse design concepts, we showed, in section 4, that scalability can be well managed. We also showed, in section 3.3, that hyper-scalability is possible when we use a conceptual model to guide us through the implementation.
Appendix – Case Study

In this section, we present a pedagogical case study to illustrate the concepts presented in this paper. Some numbers are changed from Part 1, but that has no impact on the model.

Acme Techno Widgets Company

The Acme Techno Widgets Company (ATW) produces and sells widgets. Its salesforce is assigned to four major sectors: Government, military, education and private. It produces two products, the Standard widget and the Deluxe widget.

Market research has established that the annual demand for widgets depends on each sector’s Standard widget price. The Pricing Director explains:

We start by setting a global base price. Then, for each sector, we tell our salesforce that they can offer a rebate. For instance, we offer a 70% rebate to the education sector and it’s 10% for the private sector because purchases are usually made by researchers with limited funds. The military sector gets a 20% rebate and the government 40%. This is not made public: all our price lists show the base price, but our clients in each sector are aware of the rebate they can get.

Each sector reacts differently to a change of price. We consulted with a market research expert and she came up with multiple demand functions, one for each sector. The demand function estimates a sector’s annual demand for a given base price. The demand function has the form $B/Price^A$. The parameters $A$ and $B$ are different for each sector, and $Price$ is the sector’s price, after the rebate. This table shows the values the expert gave us:

<table>
<thead>
<tr>
<th>Sector</th>
<th>Government</th>
<th>Military</th>
<th>Private Sector</th>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebate Percentage</td>
<td>40%</td>
<td>20%</td>
<td>10%</td>
<td>70%</td>
</tr>
<tr>
<td>DemParA</td>
<td>3.59</td>
<td>3.46</td>
<td>3.18</td>
<td>4.11</td>
</tr>
<tr>
<td>DemParB</td>
<td>22000000000</td>
<td>22000000000</td>
<td>22000000000</td>
<td>22000000000</td>
</tr>
</tbody>
</table>

The price of the Deluxe widget is 45% higher than the Standard widget.

The Sales Manager explains the sales pattern:

The annual demand of each Sector is split between the Standard and Deluxe products, but the distribution is very different in each sector. For instance, in the education sector, with its limited funds, the split is 80%-20% and it is 25%-75% in the military sector. I guess these guys always go for the best, and they have higher budgets. The distribution is 65%-35% for the government sector and 40%-60% for the private sector. The ratios are then applied to the sector’s annual demand to get the annual demand by product.

Another interesting pattern is the distribution of sales during the year. We noticed that our clients buy more just before the end of their fiscal year, when some want to spend their budget surpluses, and the beginning, when others have new funds allotted. Each sector has a different pattern, and we noticed that it is pretty stable year after year.

<table>
<thead>
<tr>
<th></th>
<th>Government</th>
<th>Military</th>
<th>Private Sector</th>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>9%</td>
<td>8%</td>
<td>12%</td>
<td>6%</td>
</tr>
<tr>
<td>Feb</td>
<td>10%</td>
<td>9%</td>
<td>11%</td>
<td>8%</td>
</tr>
<tr>
<td>Mar</td>
<td>12%</td>
<td>10%</td>
<td>9%</td>
<td>9%</td>
</tr>
<tr>
<td>Apr</td>
<td>12%</td>
<td>12%</td>
<td>7%</td>
<td>10%</td>
</tr>
<tr>
<td>May</td>
<td>11%</td>
<td>13%</td>
<td>6%</td>
<td>12%</td>
</tr>
<tr>
<td>Jun</td>
<td>9%</td>
<td>11%</td>
<td>4%</td>
<td>12%</td>
</tr>
<tr>
<td>Jul</td>
<td>7%</td>
<td>9%</td>
<td>5%</td>
<td>11%</td>
</tr>
<tr>
<td>Aug</td>
<td>6%</td>
<td>7%</td>
<td>6%</td>
<td>9%</td>
</tr>
<tr>
<td>Sep</td>
<td>5%</td>
<td>6%</td>
<td>8%</td>
<td>7%</td>
</tr>
<tr>
<td>Oct</td>
<td>5%</td>
<td>4%</td>
<td>9%</td>
<td>6%</td>
</tr>
<tr>
<td>Nov</td>
<td>6%</td>
<td>5%</td>
<td>11%</td>
<td>5%</td>
</tr>
<tr>
<td>Dec</td>
<td>8%</td>
<td>6%</td>
<td>12%</td>
<td>5%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Sales to a sector are not uniformly distributed by region. For example, there are more universities in the South-West than in the West. The following table shows the distribution of a sector’s sales by region. With it, we can calculate the expected monthly sales per product per region, which helps our Logistics Department do its planning.

<table>
<thead>
<tr>
<th></th>
<th>Government</th>
<th>Military</th>
<th>Private Sector</th>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>25%</td>
<td>52%</td>
<td>22%</td>
<td>24%</td>
</tr>
<tr>
<td>SE</td>
<td>18%</td>
<td>13%</td>
<td>21%</td>
<td>15%</td>
</tr>
<tr>
<td>SW</td>
<td>18%</td>
<td>18%</td>
<td>17%</td>
<td>32%</td>
</tr>
<tr>
<td>E</td>
<td>22%</td>
<td>0%</td>
<td>25%</td>
<td>17%</td>
</tr>
<tr>
<td>W</td>
<td>17%</td>
<td>17%</td>
<td>15%</td>
<td>12%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

The costs of producing a widget are $48 and $72 for the Standard and the Deluxe widget respectively. The monthly fixed costs for this year are $20000. Delivery costs depend solely on the region and are shown in this table:

<table>
<thead>
<tr>
<th>Region</th>
<th>North</th>
<th>South-East</th>
<th>South-West</th>
<th>East</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Delivery Cost</td>
<td>$10.25</td>
<td>$9.73</td>
<td>$9.58</td>
<td>$8.26</td>
<td>$11.02</td>
</tr>
</tbody>
</table>

The company CEO wants to see the following results:

- The monthly unit sales per product per region.
- The monthly sales amount and unit sales per product.
- The monthly unit sales and profit.
- The total profit.

Acme Techno Widget Company Formula Diagram
### Acme Techno Widget Company Formula List

<table>
<thead>
<tr>
<th>Var No</th>
<th>Variable</th>
<th>Type</th>
<th>Dimension Set</th>
<th>Value / Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Base Price</td>
<td>Input</td>
<td></td>
<td>$100</td>
</tr>
<tr>
<td>2</td>
<td>Base Price Multiplier</td>
<td>Data</td>
<td>Product</td>
<td>(1, 1.45)</td>
</tr>
<tr>
<td>3</td>
<td>Unit Production Cost</td>
<td>Data</td>
<td>Product</td>
<td>list of values</td>
</tr>
<tr>
<td>4</td>
<td>Rebate Percentage</td>
<td>Data</td>
<td>Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>5</td>
<td>Sector Price Factor</td>
<td>Calculated</td>
<td>Sector</td>
<td>1-Rebate Percentage</td>
</tr>
<tr>
<td>6</td>
<td>Sector Base Price</td>
<td>Calculated</td>
<td>Sector</td>
<td>Base Price * Sector Price Factor</td>
</tr>
<tr>
<td>7</td>
<td>DemParA</td>
<td>Data</td>
<td>Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>8</td>
<td>DemParB</td>
<td>Data</td>
<td>Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>9</td>
<td>Sector Annual Demand Units</td>
<td>Calculated</td>
<td>Sector</td>
<td>DemParA*DemParB&gt;Sector Base Price</td>
</tr>
<tr>
<td>10</td>
<td>Unit Delivery Cost</td>
<td>Data</td>
<td>Region</td>
<td>list of values</td>
</tr>
<tr>
<td>11</td>
<td>PR Unit Cost</td>
<td>Calculated</td>
<td>Product-Region</td>
<td>Unit Production Cost + Unit Delivery Cost</td>
</tr>
<tr>
<td>12</td>
<td>Product Distribution per Sector</td>
<td>Data</td>
<td>Sector-Product</td>
<td>list of values</td>
</tr>
<tr>
<td>13</td>
<td>Annual Sector-Product Unit Sales</td>
<td>Calculated</td>
<td>Sector-Product</td>
<td>Sector Annual Demand Units * Product Distribution per Sector</td>
</tr>
<tr>
<td>14</td>
<td>Price</td>
<td>Calculated</td>
<td>Sector-Product</td>
<td>Sector Base Price * Base Price Multiplier</td>
</tr>
<tr>
<td>15</td>
<td>Annual Sector-Product Sales Amount</td>
<td>Calculated</td>
<td>Sector-Product</td>
<td>Annual Sector-Product Unit Sales * Price</td>
</tr>
<tr>
<td>16</td>
<td>Region Sales Distribution per Sector</td>
<td>Data</td>
<td>Sector-Region</td>
<td>list of values</td>
</tr>
<tr>
<td>17</td>
<td>Monthly Sales Distribution per Sector</td>
<td>Data</td>
<td>Month-Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>18</td>
<td>MSP Unit Sales</td>
<td>Calculated</td>
<td>Month-Sector-Product</td>
<td>Annual Sector-Product Unit Sales * Monthly Sales Distribution per Sector</td>
</tr>
<tr>
<td>19</td>
<td>MSP Sales Amount</td>
<td>Calculated</td>
<td>Month-Sector-Product</td>
<td>Annual Sector-Product Sales Amount * Monthly Sales Distribution per Sector</td>
</tr>
<tr>
<td>20</td>
<td>MSPR Unit Sales</td>
<td>Calculated</td>
<td>Month-Sector-Product-Region</td>
<td>MSP Unit Sales * Region Sales Distribution per Sector</td>
</tr>
<tr>
<td>21</td>
<td>MSPR Variable Cost</td>
<td>Calculated</td>
<td>Month-Sector-Product-Region</td>
<td>MSP Unit Sales * PR Unit Cost</td>
</tr>
<tr>
<td>22</td>
<td>Monthly Variable Cost</td>
<td>Calculated</td>
<td>Month</td>
<td>SUM(MSPR Variable Cost)</td>
</tr>
<tr>
<td>23</td>
<td>Monthly Unit Sales</td>
<td>Output</td>
<td>Month</td>
<td>SUM(MSPR Unit Sales)</td>
</tr>
<tr>
<td>24</td>
<td>Monthly Sales Amount</td>
<td>Calculated</td>
<td>Month</td>
<td>SUM(MSP Sales Amount)</td>
</tr>
<tr>
<td>25</td>
<td>Monthly Fixed Cost</td>
<td>Data</td>
<td></td>
<td>$20000</td>
</tr>
<tr>
<td>26</td>
<td>Monthly Costs</td>
<td>Calculated</td>
<td>Month</td>
<td>Monthly Fixed Cost + Monthly Variable Cost</td>
</tr>
<tr>
<td>27</td>
<td>Monthly Profit</td>
<td>Calculated</td>
<td>Month</td>
<td>Monthly Sales Amount - Monthly Costs</td>
</tr>
<tr>
<td>28</td>
<td>MPR Unit Sales</td>
<td>Output</td>
<td>Month-Product-Region</td>
<td>SUM(MSPR Unit Sales)</td>
</tr>
<tr>
<td>29</td>
<td>MP Unit Sales</td>
<td>Output</td>
<td>Month-Product</td>
<td>SUM(MSP Unit Sales)</td>
</tr>
<tr>
<td>30</td>
<td>MP Sales Amount</td>
<td>Output</td>
<td>Month-Product</td>
<td>SUM(MSP Sales Amount)</td>
</tr>
<tr>
<td>31</td>
<td>Total Profit</td>
<td>Output</td>
<td></td>
<td>SUM(Monthly Profit)</td>
</tr>
</tbody>
</table>
References


Defining and Adopting an End User Computing Policy
A Case Study
Roger Turner
Wesleyan Assurance Society, Colmore Circus, Birmingham, B4 6AR
roger.turner@wesleyan.co.uk

ABSTRACT
End User Computing carries significant risks if not well controlled. This paper is a case study of the introduction of an updated End User Computing policy at the Wesleyan Assurance Society. The paper outlines the plan and identifies various challenges. The paper explains how these challenges were overcome.

We wrote an End User Computing Risk Assessment Application which calculates a risk rating band based on the Complexity, Materiality and Control (or lack of it) pertaining to any given application and the basis of assessment is given in this paper.

The policy uses a risk based approach for assessing and mitigating against the highest risks first and obtaining the quickest benefit.

1 INTRODUCTION

The paper gives the background and the case history of the introduction of an End User Computing policy at the Wesleyan Assurance Society.

The Wesleyan Assurance Society is a financial service mutual founded in 1841 that provides specialist advice and solutions to doctors, dentists, teachers and lawyers. Wesleyan aims to build life-long relations with its customers, providing them with products and services at every stage of their life from graduation to retirement and beyond.

The Wesleyan group of companies employs approximately 1,500 staff divided between the Head Office in Birmingham, Oswestry, New Malden and Northwich as well as sales staff located throughout the UK.

There are £7 bn assets under management and the Society is successful in passing on good performance to its policyholders through its financial strength and long-term investment policy.

After the Action plan there are the two main sections. These detail the challenges and how they are overcome, then an End User Computing Risk Assessment Application specially written as part of the policy is described in section 6.
2 BACKGROUND

Before the 1980’s all serious computing was done under the control of the organisation’s IT department where it was best practice for strict controls to be in place for the design, development and maintenance of all the organisation’s systems and programs.

Any new systems or changes to systems which were required by the business would frequently be done according to a lengthy development life cycle. Sometimes the requirements would change while the system was being developed so that the new system was not what the customer wanted. Business units doing their own thing was not an option.

Microsoft Excel first became available in 1985 (Wikipedia, 2018) and its gradually increasing functionality and use provided opportunity for computing independently of the organisation’s IT department to take place. End User Computing was born along with its associated risks.

3 CASE HISTORY AT THE WESLEYAN – ASSOCIATED CHALLENGES

The Wesleyan Assurance Society updated its End User Computing policy in 2012 because of Solvency II. Subsequently the Data Governance department reviewed the effectiveness of the policy and determined that changes were required to enhance its use.

The Society had not experienced any particular problems with its spreadsheet and end user applications but was keen to ensure it enhanced its policy to keep pace with best practice and minimise the risk of issues arising in the future.

The business need which brought about the update of the policy was the potential risk of an application causing a substantial loss event. We considered this to be sufficient to warrant at least an investigation into how best to mitigate this risk.

The objective is to establish a clear plan of action, try it using at least one pilot and, once proven, roll it out throughout the Society. Two pilot runs have been successful and the Society has approved a phased roll-out on risk based approach.

The main challenges which were faced were:

a. What to cover in the Policy (the Scope)

b. Where the End User Computing risks are

c. Getting the buy-in to adopt the updated Policy

d. Storage of data pertaining to the applications subject to the policy

e. Regular review.
4 PLAN OF ACTION

We considered the proceedings of a Hellenic American Union conference (Mallikourtis & Papanikolaou, 2010) and attended a workshop run by The Corporate IT Forum (CITF, 2016). Resulting from this background we decided that the following steps should be taken:

a. Produce the first draft of the updated Policy Document which includes a means of assessing applications (spreadsheets) for risk. The scope to cover “any computing which is not supplied by, acquired by or supported by any of the Wesleyan’s formal IT departments”. As to the applications, nearly all of the end user computing applications are spreadsheets. The scope is not complete, however, without including local databases (usually Access), Business Intelligence reports (e.g. SQL, Crystal, Power BI), Mobile apps and some third party apps.

b. Find stakeholders who are willing to co-operate in running at least one pilot.

c. Run the pilot(s) which involves collecting data about each submitted end user computing application, assessing it for risk and storing the details in a repository where it could be accessed when the need arises.

d. Conduct “show and tell” sessions to demonstrate which applications already have satisfactory controls and which might be deemed to fall short.

e. Agree an action plan to fix any errant applications.

f. Apply governance which will then become part of the End User Computing policy.

Several challenges and how they were overcome are provided here.

5 CHALLENGES FACING EUC ROLL-OUT & OVERCOMING THESE

5.1 Defining the Risk Metrics to assess the applications with

The Complexity and the Materiality of an application are the two main contributors to risk. The more complex a spreadsheet (or for that matter any application) is, the greater the risk is of the risk crystallising and creating an issue. Once the risk crystallises, how material is the effect on the Society’s business operation?

Complexity

We used one of the simpler ways to measure Complexity and this is suggested by PwC (PwC, 2004). A spreadsheet with low complexity is just for information logging and tracking. There are no formulae or links. Medium complexity is where simple formulae are used, for example to translate or reformat information. High complexity is the rest, where complex formulae are used, there are links to external sources, macros and modelling.

The more complex an application is the less likely someone other than the author can understand it and the greater is the spreadsheet risk.

Materiality

Materiality could be measured as the impact resulting from the risk crystallising. This could be
a. Inconvenient  
b. Poor Customer Outcomes  
c. Reputational  
d. Loss of Business  
e. Financial  
f. Statutory / Legislative

Different areas of the business rank these in different orders so we used a different approach instead.

Independent research done by Chartis suggests the following classification for materiality (Chartis, 2016):

a. High – Application supports financial or regulatory reporting or private or confidential information.

b. Medium – Application supports management reporting, calculation or input into a core management information system, or used for making key business decisions.

c. Low – internal operations or day to day decisions, or contains outputs from core management information systems.

Control

Following the Complexity and Materiality metrics in this way leads us to the front face of the cube provided that the application is well controlled.

The four colours on the cube indicate the risk rating band, and this identifies what remedial action, if any, is needed ranging from blue (none) to red (urgent action needed). Wesleyan uses the Magique system to record the risks and track them through to a resolution. (Magique, 2018).

Control, when considered, defines how far back we go in the cube.

We wrote a Risk Assessment Application for the End User Computing policy. The SMEs or other experts in the user department are provided with the application (which itself is a spreadsheet) then they use the application to assess the risk. Complexity and Materiality of an application are collected by the SMEs’ self assessment because they have hands-on knowledge of the applications and their context within the business.
By a series of yes / no questions the risk assessment application then gathers information about:

a. How accessible the application is, whether its location is known and whether there are operating instructions

b. Business Continuity, Back-up and Recovery

c. Version controlling, whether it needs reviewing and evidence of having been tested

d. Security, Privacy and Integrity, in other words unauthorised access to the system

e. The ability to fix the application if it breaks, including the existence of a second person able to fix and the existence of technical documentation

f. Finally, whether the system contains personal or sensitive personal information (in the context of GDPR, the General Data Protection Regulation, (IT Governance, 2018)).

The answers to the questions are recorded in the Risk Assessment Application and the application calculates the risk rating band. The user then sends the result back to Data Governance who records the results and ensures that there is an action plan to fix the application if it falls short within the assessment.

5.2 Whether to use a Top Down or Bottom up approach

As regards knowing what to assess for risk, two approaches are available, one being top down and the other being the bottom up. The bottom up approach means scanning the whole of the file store for spreadsheets, databases and such like for likely candidates and then finding owners. Even though there are tools which can scan for spreadsheets (Microsoft (2013), Finsbury (2014)) this is a formidable task if one considers that there could be several million files, only a few of these in current use and a few again requiring assessment.

The other way is to use the top down approach where managers and subject matter experts know where their applications are and can use the Risk Assessment Application to assess their applications and return the results. This is what we believe to be a more practical method.

5.3 Where and how to store the assessment results of EUC applications

Wesleyan’s Group Reference Architecture provides for the use of Orbus’ iServer as a repository for all the assets, whether an IT system or part of End User Computing. (Orbus, 2018).

Each application, (spreadsheet, other EUC application or IT supported system) can be stored in a way whereby its relationships with others can be visualised, for example in terms of the processes the application is used by, which department runs the process and which technology or platform the application runs on.

Its use within End User Computing is to be able to report on applications which require remedial action and to trigger action when an application needs to be reviewed. The policy states that each application should be reviewed annually.
5.4 Assimilation of the End User Computing policy

The full version of the policy document came to more than 80 pages and reading this is a big ask. We considered that effective communication of the policy is important so we split the document into smaller, more manageable amounts and put these on the intranet to draw the reader’s attention to what action is needed based on their role, being one of the following:

a. Executives
b. Senior Managers
c. Managers
d. Subject Matter Experts (SMEs)
e. Data Stewards

For example, if the reader is a manager the manager is led to this screen:

**END USER COMPUTING ACTION REQUIRED OF MANAGERS**

<table>
<thead>
<tr>
<th>Action Required</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adhere to the End User Computing policy and the Data Governance Policy,</td>
<td>EUC Policy</td>
</tr>
<tr>
<td>Cascade to your direct reports.</td>
<td>Data Governance Strategy</td>
</tr>
<tr>
<td>Understand the Risks and Controls (Magique) for End User Computing</td>
<td>Risks Controls</td>
</tr>
<tr>
<td>Understand the impact of end user computing on data quality.</td>
<td>Data Governance Strategy</td>
</tr>
<tr>
<td>Responsible for Data Quality including fitness for purpose of End User</td>
<td>Taking Remedial Action</td>
</tr>
<tr>
<td>Computing applications, and making these known to Data Governance,</td>
<td></td>
</tr>
<tr>
<td>Responsible for improving the control.</td>
<td></td>
</tr>
<tr>
<td>Provide assurance to your Senior Manager / Executive through an assessment of</td>
<td></td>
</tr>
<tr>
<td>'data' risk expressed within your Operational Risk and Control Assessments</td>
<td></td>
</tr>
<tr>
<td>(ORCA); or, where assurance cannot be given provide an action plan to strengthen</td>
<td></td>
</tr>
<tr>
<td>controls/contain and fix known issues.</td>
<td></td>
</tr>
<tr>
<td>Identify all end user computing applications so that the inventory of all End</td>
<td>EUC Inventory</td>
</tr>
<tr>
<td>User Computing applications kept by the Senior Manager is kept up to date.</td>
<td></td>
</tr>
<tr>
<td>Review each application periodically as required by the next review date.</td>
<td></td>
</tr>
<tr>
<td>Ensure that each end user computing application is assessed to determine its risk</td>
<td>Risk Assessment App</td>
</tr>
<tr>
<td>rating and that testing has been done. Communicate the results to Data</td>
<td>Operating Instructions</td>
</tr>
<tr>
<td>Governance.</td>
<td>Risk Metrics</td>
</tr>
<tr>
<td>Assist in creating and implementing an action plan to fix errant applications as</td>
<td>Action Plan</td>
</tr>
<tr>
<td>indicated by the assessment tool.</td>
<td></td>
</tr>
<tr>
<td>The Complexity and Materiality of each EUC application are each graded 1 to 3</td>
<td>Development Life Cycle</td>
</tr>
<tr>
<td>and if the sum of these is 5 or more then the Development Life Cycle must be</td>
<td></td>
</tr>
<tr>
<td>understood and followed as appropriate.</td>
<td></td>
</tr>
<tr>
<td>Establish a control framework which ensures the above actions are met.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Sample Intranet page - EUC action required of managers
Instructions are on the left and the hyperlinks on the right reference the appropriate part of the policy. The hyperlink for “Risk Assessment App” launches the application in Excel and they save it so that the users can use this risk assessment application to assess their applications.

5.5 How to engage the Stakeholders

It was known that complex spreadsheets can contain many errors (Bregar, 2004) and the challenge was to maintain the buy-in from the stakeholders so as to mitigate against potential spreadsheet risk.

We decided to run pilots with two willing departments, chosen for the likelihood of having material or complex applications. Both of these were in the Finance area, one being Middle Office and the other Financial Accounting so we had to approach the department heads for their cooperation.

It certainly helped to have a well-prepared presentation identifying the risks and benefits surrounding End User Computing applications.

To facilitate buy-in we made and used a “horror slide” to highlight the risks in which some firms have lost billions of dollars, because of a mistake in a spreadsheet. Ample evidence is found on the (EuSpRIG, 2018) page.

We pointed out that writing a spreadsheet can be a quick and easy solution but the costs in the event of a risk crystallising can be substantial.

Think of a situation where only one person knows how to run an application, and that person is not there when the result is needed. The temptation is to get anybody to do the job, not knowing what to do or how to do it. We made the point by using this picture from the (Financial Times, 2013) showing this willing but unknowledgeable user.

![I HAVE NO IDEA WHAT I'M DOING](image_url)

Figure 2- Engaging willing but unknowledgeable users

Think of a situation where only one person knows how to run an application, and that person is not there when the result is needed. The temptation is to get anybody to do the job, not knowing what to do or how to do it. We made the point by using this picture from the (Financial Times, 2013) showing this willing but unknowledgeable user.
Both pilots ran for five weeks during which the departments had each submitted 20 applications. We gave them the Risk Assessment application (see section 6 for the screen shots) and we collected the details of all the applications submitted (they were all spreadsheets). The collection for each application took around 10 minutes.

### 5.6 Assessment Results Returned to Data Governance

Each application returned had a risk rating calculated from details provided by the SMEs and the opportunity was available to challenge some detail if thought needed. For the complexity metric these tools are available:

Excel Inquire (Microsoft, 2016) is the easiest mentioned here and can report on links between spreadsheets and worksheets, and identify errors, hidden sheets and such like.

Finsbury EUCE (Finsbury, 2014) can perform the above and provide a complexity rating, as can the Discovery and Risk Assessment Server (Microsoft, 2013).

A sample of the spreadsheets were checked for the complexity rating provided in the pilots and the ratings agreed.

### 5.7 The Show and Tell sessions

Even the users’ own assessment of the applications in the pilots gave a surprising proportion of applications with a red rating meaning that urgent action is needed to fix to mitigate potential risk. We looked at the reason why a poor rating was being produced and if there were any quick wins to remediate. The applications concerned all had either the materiality or complexity set to 3 with the other parameter at least 2. In these applications, the main concern was expressed in the security section where the functions or data in the spreadsheets could be open to accidental alteration or corruption (although we found no evidence this has actually occurred), and in some cases there was lack of version control.

Fixing these was seen as a quick win because spreadsheets could be baselined and copies made read only and before the next use a comparison could be made with the baseline. Comparison against a baseline can be done using Excel Inquire (Microsoft, 2016).

After the quick wins several red applications became amber and the most frequent reason for them remaining amber was the lack of evidence for testing and sometimes the lack of technical documentation (as opposed to any ongoing concerns).

This is more of an ongoing issue, however, Finance asserted that the results of these applications are subject to audit and there are many professionals who are equipped to challenge the results should any be considered to be suspicious.

The outcome is that the Risk Assessment Application highlights areas where attention to the control of an application ought to be focussed and it is up to the user department as to what action to take. They are responsible for a truthful entry of data into the Risk Assessment application and are accountable for whatever risks there are in the end user computing applications.
5.8 Finding where the EUC risks are

This is part of the roll-out plan. We gave a short presentation to a meeting consisting of executives and managers about what End User Computing is, its associated risks and benefits of control. We had prepared a spreadsheet-based questionnaire in which the managers make their own assessment of the control which exists over their most complex and material applications. The managers were asked to return these to Data Governance and we will roll the policy out first to the areas of highest perceived risk during the coming months.

The Human Resources and Field Support departments continue to show their willingness to take part in this activity.

6 THE END USER COMPUTING RISK ASSESSMENT APPLICATION

The two main screens are given here.

Figure 3- Risk Assessment Application screen - General Details

The top part of the screen is all about the people who interact with the application. We in Data Governance use these people as a point of contact.
The next section is about the application itself, giving its name, description, version and version history, where it is and which platform it runs on. This provides us with more depth to the information we hold about the application. All of these details are recorded in iServer.

We learned from one of the pilots that users like to partly complete a batch of applications and go back to them later to finish off. We needed to provide a “Restore Previous Input” button which allows the user to call back information about any previously entered application for completion.

Clicking on “Next” navigates us to the next screen.

![Figure 4- Risk Assessment Application screen - Assessment](image)

The second screen gathers information about the control of the application. The SME or other expert in the User Department enters the Complexity and Materiality ratings first and then answers mostly “yes” or “no” to the other questions, divided into the categories as described above.

The last (General) box is to include any free-form information about the application which might be useful, for example “We are currently training a second person who can fix this application if it breaks”.

At the end of entering all the data, the user clicks on the “Calculate” button and the Risk Assessment application calculates the risk rating band according to the details entered.

The green box near the bottom is then populated with the next action required, according to whether the rating is blue, green, amber or red. It is coloured appropriately. The possible outcomes are:
a. Blue – No action needed.

b. Green – An awareness of this application is needed. This is the minimum rating band for applications which hold customer data (GDPR) and applications which are Green and above are reported to Data Governance and are subject to annual review.

c. Amber – Falls short of acceptable control and an action plan is needed to fix. An entry is made in the Magique system. We would expect the plan to be implemented within three months.

d. Red – As Amber but urgent action is needed to fix – within a month or before the application is next run if later.

The last box is for free-format information about the risk which might assist in mitigation.

The Risk Assessment application calculates the next review date as being a year from the previous review and the applications which carry the most significant degree of risk are known.

We updated the End User Computing policy having run the pilots.

Already some areas see the policy favourably. In rolling the policy out a risk based approach will be used so that those areas where there is a greater risk will be worked with first.

7 CONCLUSION

Wesleyan started with an End User Computing policy which required updating. After we had written a new policy and the End User Computing Risk Assessment Application we successfully ran pilots in two areas of Finance. The assessment results from these pilots enabled some quick wins to be done and from the learning points gained we were able to improve the policy and the Risk Assessment Application.

The evidence thus provided enabled senior management at the Wesleyan to approve the policy and we are using a risk based approach to roll it out across the Society.

Any models or information contained in this paper are intended for educational purposes only. To the extent permitted by law, the author and Wesleyan Assurance Society shall not be held liable for any liability or loss suffered by a third party who uses the models or information within this document for purposes for which they were not intended.

© Wesleyan Assurance Society May 2018 All Rights Reserved

REFERENCES


Microsoft (2018), ‘Business Intelligence’, Online [available] https://powerbi.microsoft.com/en-us/, accessed 28/03/2018. Power BI is provided by Microsoft and has a desktop application which allows you to connect to data from a multiple of sources, to shape that data through queries and use the results to create reports using a range of standard and bespoke visuals. The resultant report files can be shared like any other or can be uploaded (and shared) on the Power BI Service which is cloud based.


Developing Excel Thought Leadership

David Lyford-Smith
c/o ICAEW, Chartered Accountants’ Hall
1 Moorgate Place
EC2R 6EA
david.lyford-smith@icaew.com

ABSTRACT

Over a period of five years, the Institute of Chartered Accountants in England and Wales (ICAEW) has developed a suite of three ‘thought leadership’ papers surrounding good practice in spreadsheet use and spreadsheet work environments. We will review the history of these three papers, the key lessons which each has to teach, and discuss how the process of making them has helped ICAEW to develop its position in the field.

1 TWENTY PRINCIPLES FOR GOOD SPREADSHEET PRACTICE

1.1 History

ICAEW was established by royal charter in 1880, combining four previous regional accountancy bodies within the UK. Today, ICAEW is a professional membership organisation and regulator, with over 150,000 members worldwide. ICAEW offers a range of accounting qualifications, supports its members throughout their careers, regulates the professional activities of accountants, and produces thought leadership and technical policy guidance in appropriate fields.

Having established an IT Faculty in the early 1990s to advise on key IT matters for accountants, ICAEW noticed that Excel- and spreadsheet-related content was consistently some of the most popular that they produced. As a result, the spreadsheet-related content was spun off into a separate ‘Excel Community’ in 2011. In early 2013, a volunteer advisory group was set up, seeking to get spreadsheet experts’ input and opinions on activities for the community to pursue.

Starting with its first committee meeting in 2013, this group – the Excel Community Advisory Committee (ECAC) – consisted of a variety of spreadsheet experts from the fields of financial modelling, training, academia, and more (the author was a volunteer member at this time). The group met regularly to discuss both the training and other benefits that the Excel Community made available to its subscribers, but also ideas for public-benefit thought leadership work, under ICAEW’s royal charter obligations.

An early theme in the discussions was that of spreadsheet risk. Many contributors felt as though spreadsheet risk was a serious issue affecting world trade, with misuse of spreadsheets being common and a mainly self-taught user-base figuring things out for themselves. But it seemed that little attention was being paid to these issues. So a decision was made to try and tackle the issue by producing a guide to spreadsheet “good practice”.
The group individually tried to come up with their own high-level rules for good practice. The format of these attempts differed significantly – some opted for longer lists, some for shorter; some emphasised specific functions and detailed guidance, while others were more principle-driven; some were generalised, and some were divided into separate recommendations for different user groups or applications.

From this list, ICAEW staff member Paul Booth worked to check for commonalities between the authors, and to trim out recommendations that were uncommon or where authors disagreed. Through this analysis, eventually the hundreds of ideas submitted were pared back to a list of twenty that the group could agree upon. These were published in 2014 as the Twenty Principles for Good Spreadsheet Practice [ICAEW, 2014], now into its third edition.

1.2 Overview

The Twenty Principles for Good Spreadsheet Practice (20P) are twenty high-level guidance statements for spreadsheet users. The list covers general rules for building a good spreadsheet working environment as well as specific rules about common risky practices.

Example Principles include:

2. Adopt a standard for your organisation and stick to it

14. Never embed in a formula anything that might change or need to be changed.

19. Build in checks, controls and alerts from the outset and during the course of spreadsheet design.

The 20P are designed to be high-level, such that they are applicable to a range of spreadsheet users in different fields and of different experience levels. They are also principles rather than rules; in some circumstances deviation might be appropriate.

As well as the 20P document itself, ICAEW produced a series of explanatory blogs, webinars, and related content. A facility was made available for organisations to have their financial modelling standards, templates, training courses, or other similar spreadsheet materials ratified as “Twenty Principles Compliant”. Several notable standards achieved recognition, including the FAST Standard [FAST Standard Organisation, 2016].

The 20P is designed to start a conversation around best practice, and provide some simple and common-sense recommendations that any spreadsheet user can understand and adopt. It is predicated on the belief that the high-level principles of good practice are industry- and user experience-agnostic, and that they can be applied everywhere.

1.3 Lessons learned

The 20P was ICAEW’s first experience in creating spreadsheet-focused thought leadership. It was created by overlapping the suggestions of a disparate set of spreadsheet experts and looking for commonalities. This method helped to find the core of agreed good practice from many differing opinions, and would be used repeatedly within ICAEW’s later work.

The 20P themselves do not contain any revelatory new ideas about spreadsheet risk and best practice – most of the recommendations are common-sense and even commonplace. However, having a single summary covering a variety of aspects that is not tied to one use case or to a certain class of users is novel and valuable.
As previously pointed out at EuSpRIG, no practice is universally beneficial and worthy of the title of “best practice” [David Colver, 2010]. This was very much in the minds of the committee when trying to identify the most broad recommendations possible – and the choice of “good practice” for the title rather than “best practice” was very much intentional.

The work of Ray Panko and others in the field [e.g. Panko, 2016] focuses on both the need for improvements to spreadsheets – given their high error rates – and the difficulty in using practice recommendations alone to weed out those errors. Panko recommends spending significant effort in team-based code reviews of spreadsheets to improve error rates. Conducting peer review and testing are both principles within the 20P; the paper recommends a broader view of a spreadsheet and the business environment within which it sits rather than just focusing on in-workbook recommendations.

The other work of the Excel Community has been based on the 20P as a founding document.

2 SPREADSHEET COMPETENCY FRAMEWORK

2.1 History

Following on from the success of the 20P, the ECAC was looking for another area of difficulty with spreadsheets that they could tackle. The subject of spreadsheet capability and knowledge arose as an area of interest – namely, the issues with terms like “spreadsheet expert” or “super user”, and the wealth of CVs which use phrases like “proficient with Microsoft Excel”. All of these terms are undefined and their use is quite arbitrary – different people will have different ideas of what each means. Furthermore, those with lower ability may lack the metacognitive awareness to be able to accurately assess their own ability. This leads to several problems: Recruiters can’t rely on candidates’ professed abilities; job applicants can’t distinguish themselves as true experts; and trainers have a hard time accurately describing the level of user that their courses are aimed at.

While there are some systems out there for assessing spreadsheet knowledge, the ECAC felt that these were largely either too general – such as the European Computer Driving Licence [ECDL Foundation, 2019] – or which were only concerned with the very top percentile of users, such as Microsoft’s own MVP programme [Microsoft, 2019]. Syllabi for training courses often teach to an implicitly identified “appropriate” level, but divining this is not always straightforward.

Based on experience with crafting the 20P, the ECAC set up a working group, the members of which then each submitted their own ideas of what strata existing in spreadsheet ability, and what the key defining knowledge for each level was (the author transitioned to ICAEW staff during this process). After much iteration, the group eventually finalised its work under the title Spreadsheet Competency Framework [ICAEW, 2016], now in a second edition.

2.2 Overview

The Spreadsheet Competency Framework (SCF) lays out a set of four categories for spreadsheet users’ abilities: Basic User, General User, Creator, and Developer. These four levels are defined by which spreadsheet skills a user of that level would be expected to have, and also what role a person with those skills might play within an organisation.
The four levels are briefly explained as follows:

- **A Basic User** will mostly be carrying out data entry tasks. They will have a grounding in the essential skills needed to avoid major wasted effort or bad practice, but few technical skills beyond that. Anyone that uses spreadsheets should be at this level at a bare minimum.

- **General Users** make up the majority of spreadsheet users, essentially modifying and updating spreadsheets on a regular basis. They may have some formula and other more technical knowledge, but are rarely called upon to undertake highly complex tasks, or to make spreadsheets entirely from scratch.

- **Creators** use spreadsheets as a core element of their roles, and significantly use its functions and features. They often create spreadsheets from scratch and may create templates and workbooks for users at the first two levels.

- **Developers** are the true masters of spreadsheets, with a grasp on the majority of the features of the package and ability to handle many complex tasks. They may be specialists such as modellers, VBA programmers, or statisticians.

The SCF is designed to be of use to anyone needing to refer to spreadsheet ability in a structured and consistent way, although it is written particularly for an accounting and finance audience. Each level has a set of competencies which are ‘required’ for that level, and others classified as ‘beneficial’.

Since publication the SCF has been the subject of a EuSpRIG paper in its own right [Csernoch & Biró, 2017]. This paper consisted of a critique of the SCF and a reworking of the content in the educational sphere.

ICAEW has been exploring avenues to fund and develop an assessment tool that jobseekers, employers, and others could use to place themselves in the appropriate SCF level, allowing them to verify as well as communicate about spreadsheet ability.

### 2.3 Lessons learned

During the creation of the SCF, there was significant disagreement on several issues that identified just how complex of a question this is. For example, the number of levels that existed, and whether a single scale was appropriate for all user groups, was discussed at some length. There is a need to balance precision of measurement with avoiding over-complexity. The eventual compromise was reached of the four described levels, plus a *de facto* fifth level representing those with below-minimum spreadsheet package knowledge that the guide recommends are not permitted to work on spreadsheets until they are given a minimum level of training.

Even once the four-level model was generally accepted by the ECAC team, the mix of core skills at each level was considerably complex. The discussion highlighted just how varied the experience and skills of even a population of expert users could be. For example, some contributors were expert financial modellers with a decade or more of experience, but had never had cause to create a PivotTable. Others were experienced trainers, but did not work much in financial circles, and so might not know how to model a simple loan. VBA programming was also contentious, with some avoiding its use or even arguing that a little knowledge of the subject could be detrimental! The decision to mix between ‘required’
and ‘beneficial’ skills in the definition of each level is a nod to trying to acknowledge the possibility of specialisation, even within very expert users.

3 FINANCIAL MODELLING CODE

3.1 History

Shortly after releasing the 20P, ICAEW ran a roundtable at a ModelOff Global Training Camp event to discuss spreadsheet standards, principles, and best practice, comparing the approach of the 20P with some specific financial modelling standards. Resulting from this discussion, the idea of ICAEW creating a central, non-commercial guide to best practice was first raised. While this idea was of interest, it was not pursued at the time due to resource constraints.

Some years later, a group of expert financial modellers, working outside of ICAEW but including many active volunteers from the ECAC group, began to meet and discuss a joint project to create such a document – a shared vision of what good financial modelling practice looked like. This group began by lining up seven methodologies from various of its members, and looking for commonalities shared between all of them.

Eventually, this discussion generated a rough list of shared ideas. By this time, the external group had agreed that the final product would be most useful if it were finalised and owned by a neutral party external to the group, and identified ICAEW as the preferred destination. Not only is ICAEW seen as a neutral third party, but the work done in the first two publications had helped to establish ICAEW’s reputation and expertise in the field.

In early 2018, the early draft of the list was transferred to ICAEW. The ECAC formed a working group, comprised of financial modellers both who had worked on the project at the initiation stage and others, who then reviewed and revised the document as it was reshaped from a list of principles into a more fleshed-out document. The document was also circulated among a wide variety of external stakeholders via an open consultation period, with feedback being actively sought from as many financial modelling experts and organisations as could be identified.

The resulting document was released under the name Financial Modelling Code [ICAEW, 2018], and forms the final part of the trilogy of ‘core spreadsheet thought leadership’ made by ICAEW.

3.2 Overview

The Financial Modelling Code (FMC) is comprised of the essence of seven different modelling methodologies, with review from representatives of around twenty different modelling organisations of different sizes and specialisations. It has been endorsed by a wide range of players in the marketplace, including Microsoft itself.

The FMC consists of a series of recommendations on best practice for tackling common financial modelling problems, which are closely tied in to and cross-referenced with the 20P. Alongside the guiding principles in each case, a series ‘advocated’ and ‘discouraged’ approaches are listed as a suggestion for how one might go about putting the principle into practice. By intention, few absolute rules are permitted – allowing each modeller to make their own judgment about what’s right for them and their clients. Here’s an example of a principle from the document:
INCLUDE USER GUIDANCE

(cross-referenced to 20P #7, “Include an ‘About’ or ‘Welcome’ worksheet to document the spreadsheet”)

Although models should be built to require minimal external explanation, appropriate guidance to help facilitate understanding and proper use should be included. This should also include any key assumptions made in calculations (e.g., ‘all cash flows occur at the end of the appropriate period’).

Advocated approaches:
- Include in the model a worksheet dedicated to being a user guide.
- Embed a separate user guide document in the workbook.
- Add contextual user guidance throughout the model where appropriate.

Discouraged approach:
- Don’t store documentation in a separate file or email that may become separated from the model itself.

The FMC taken as a whole is a guidance document, and not a how-to guide – there are no Excel screenshots in the document and no formulas. There are other publications in the marketplace aimed at being modelling guides, many of which discuss best practice, but the FMC is aiming to be a higher-level guidance document than these, or even than modelling standards such as FAST.

An option for organisations to become official supporters of the FMC is available; at time of writing, eleven such organisations have put themselves forward, including several ‘Top 10’ accounting firms and Microsoft themselves.

3.3 Lessons learned

This was the first piece of work that ICAEW produced that originated outside of the organisation. While the final piece underwent considerable iteration and was reviewed by many sources, the original work was done by a separate group. However, the methodology used was similar to ICAEW’s previous two spreadsheet works – a group of domain experts was convened, and their knowledge was combined and contrasted to settle on a core of agreed practice.

Opinions about the best way to perform financial modelling are remarkably divergent. To take a clear example, opinions on the use of named ranges runs the gamut from “essentially forbidden” to “essentially mandatory”. To create a set of principles that would accept both extremes but not favour a particular solution required careful wording and consideration. In general, the final document comments on the aims of good practice more than how to achieve them in the main text, and several (sometimes mutually exclusive) options of how to achieve those aims are included.

While the SCF did include reference to the 20P, the creation of the FMC leaned more heavily on the 20P, and the final version includes both a list of the Principles and frequent cross-references to them. While the basic principles were developed by the external expert group, the resulting output closely aligned to the 20P and finding connections was straightforward. This shows that the 20P truly represent well-understood guiding principles of good practice.
4 CONCLUSION

Over the period since the ECAC was formed in 2013, ICAEW has worked on a variety of spreadsheet-related projects. The common methodology for the three major thought leadership pieces was, in brief:

1. Identify an area where a high-level guidance publication could be of use by consulting with experts and stakeholders
2. Convene a suitable and diverse group of experts and synthesise a consensus view from their disparate opinions
3. Invite commentary and critique via an exposure draft process
4. Complete the project based on the feedback received

This process has been successful in producing a trilogy of work which is of a high standard and which is well-regarded and accepted. Nevertheless, it is interesting that all three writing processes uncovered a wealth of individual preference, difference of opinion, and variety in what “good” really meant in the realm of spreadsheets. No doubt one of the difficulties that has plagued attempts to reduce spreadsheet risk in practice is this very heterogeneous field of viewpoints. But spreadsheet use is also very diverse, and perhaps it is only natural that agreement can be hard to find.

All three projects had to tackle similar issues with forging common ground out of many differing views. Spreadsheets are in use across a very broad cross-section of business and other life, and naturally users will have different priorities, experiences, and opinions. Our approach in each case was to identify the most broadly agreeable version of contentious statements, making things more generic or more fundamental where possible. Leaving room for individual users to interpret and enact our recommendations in their own way was an active decision.

Spreadsheet risk remains a significant issue in modern business, and stories of spreadsheet mistakes tripping up companies continue to be common. So it’s reasonable to ask how the cause of fighting spreadsheet risk has been affected by ICAEW’s work in this area, and what remains to be done. These three guides have all been well-received, but the key issue remains attracting attention to these issues and convincing the audience to seek out recommendations for improving their practice. Anecdotally, many users have been happy to take on recommendations for improvements when they have been presented with them, but do not actively seek them out. ICAEW’s strategy is to continue to promote both the case for improving practice, and its trio of publications aimed at meeting that end.

References


Practical aspects of applying an End User Computing policy

Roger Turner
Wesleyan Assurance Society, Colmore Circus, Birmingham, B4 6AR
roger.turner@wesleyan.co.uk

ABSTRACT

The Wesleyan Assurance Society revised its End User Computing (EUC) policy in 2017 and the plan was put in place to address the EUC risks (Turner, 2018). This paper describes the challenges which were faced and how these were overcome.

We find that EUC applications are clustered in certain business areas and this information supports the need for addressing these risks on a wider scale with a view to improving overall business efficiency.

A BAU process has been put in place to monitor activity and we are seeing an improvement in the quality of EUC in the Society.

1 INTRODUCTION

The End User Computing (EUC) policy at Wesleyan was revised in 2017 and this paper describes how it has been rolled out since, identifies some of the challenges and how these were overcome.

The Wesleyan Assurance Society is a financial service mutual founded in 1841 that provides specialist advice and solutions to doctors, dentists, teachers and lawyers. Wesleyan aims to build life-long relations with its customers, providing them with products and services at every stage of their life from graduation to retirement and beyond.

The Wesleyan group of companies employs approximately 1,500 staff divided between the Head Office in Birmingham, Oswestry, New Malden and Northwich as well as sales staff located throughout the UK.

There are £7.6 bn assets under management and Wesleyan is successful in passing on good performance to its policyholders through its financial strength and long-term investment policy.

The EUC policy covers any application not supported by IT and 90% are spreadsheets. In this paper we refer to them all as “applications” for complete coverage.

2 BACKGROUND

In 2017 we decided to keep pace with best practice by updating our EUC policy (Turner, 2018).

By the end of 2017 the End User Computing policy had been successful in two pilot runs and had been reviewed by the Chief Operating Officer (COO) as sponsor. He approved the policy being rolled out starting Q2 2018.
End User Computing control is part of the Data Governance function at Wesleyan.

3 CHALLENGES

The main challenges we faced in fully rolling out the revised End User Computing policy were:

- Identifying the business areas at greatest EUC risk
- Obtaining buy-in from each business area to adhere to and implement the EUC policy
- Establishing effective storage and retrieval of EUC application metadata
- Getting managers to record applications which fall short on Magique (the Society’s already existing risk management system) (Magique, 2018)

These challenges are not untypical across the industry. An example of this is the experiences faced by Chambers and Hamill in respect of understanding where the EUC risks are in a banking environment and assigning responsibility for them (Chambers & Hamill, 2008).

4 STORAGE OF EUC APPLICATION METADATA

Early in the project we decided to follow IT’s existing strategy roadmap by using Orbus’ iServer (Orbus, 2018) which is the recognised repository for all Wesleyan’s assets in terms of systems, interfaces, servers and processes. We defined the data dictionary for EUCAs and arranged for iServer to have corresponding attributes for this metadata. It made sense to make iServer the master repository for EUCA information.

![EUCA Metadata Flow](image_url)

The user records one or more EUCAs in the Risk Assessment Application which calculates the risk rating band and advises if any remedial action is needed and why. The user passes the completed Risk Assessment Application to Data Governance who uploads the metadata into iServer. iServer exports metadata about a selection of or all the EUCAs for the inventory, KPI reporting and re-assessment by the Risk Assessment Application. The user can be given a copy of the Risk Assessment Application containing his or her own data for updating and returning to Data Governance for upload again into iServer.
5 ROLLING OUT THE EUC POLICY

We rolled out the EUC Policy (Turner, 2018) in these stages:

5.1 Head Office Managers Meeting, April 2018

This monthly meeting at the Wesleyan consists of approximately 200 managers and the COO had given permission for us to deliver a short presentation on EUC at that meeting. The presentation focussed the audience on the risks posed by EUC (Financial Times, 2013) and by citing some of the public loss events (Chartis, 2016). Then the action was given to everyone to

- Read the EUC policy (which was already on the intranet)
- Complete a short assessment template by 30 June 2018 which was issued to everyone immediately after the meeting (See screenshot in Appendix B)
- Flag any high-risk applications to Data Governance as soon as possible.

The assessment template is an Excel spreadsheet which is applicable to the whole of the manager’s department. The main point is that it is easy to complete and it asks for the manager to identify the process which has the most complex calculations or which has the most material impact on the business.

The spreadsheet asked the managers to assess that process according to the metrics already established (Turner, page 3, 2018). Then they had to return the completed template to Data Governance by 30 June 2018. Very often the managers enlisted the help of their Subject Matter Experts and other staff to select and assess their most complex processes.

The spreadsheet provided the manager with one of three possible messages:

- You are Green. Please return this spreadsheet to Data Governance - no further action needed, however you are accountable for the results which you have returned. Any incidents as a direct result of spreadsheet errors that impact on a material process will need to be reported to Data Governance as a matter of urgency.
- You are Amber. Action is needed. Return this spreadsheet to Data Governance. Your spreadsheets and applications need to be assessed, errant ones recorded on Magique and there needs to be an action plan to fix.
- You are Red. Urgent action is needed. Return this spreadsheet to Data Governance. Your spreadsheets and applications need to be assessed, errant ones recorded on Magique and there needs to be an urgent action plan to fix.

Figure 2 (right) shows the number of assessments returned in each category. Forty-six of the replies were to the effect that the department had no EUCAs to assess. There were an additional 69 replies (not included in Figure 2) saying that their area was within someone else’s and to include it would be a duplication.

Figure 2: Assessment Returns by Department
5.2 Collating the returns and publicity, August 2018

By the end of July all the results were in (not without a certain amount of gentle persuasion including walking the building and meeting the stragglers face to face)! This was the time for publicising the importance of controls on EUC and having an agenda item on various Data Governance meetings to update people on the progress. Publicity was helped by the author of this paper being shortlisted for a CIR Risk Management award in the Newcomer of the Year category (CIR Magazine, 2018).

The next important deadline was to table whatever remedial actions to be taken on applications falling short of the controls at the Group Executive meeting in November 2018 so that remedial action could be put into each departments’ 2019 business plans.

5.3 Drilling down to expose further risks, September and October 2018

We identified those departments which had an assessment of red or amber and issued them with the full EUC Risk Assessment Application to identify any applications which fall into the amber or red category, therefore requiring remediation. (Turner, page 9, 2018).

Chambers & Hamill sets out certain minimum controls (Chambers & Hamill, 2008). We found that an important control is the ability to support an application once the author has ceased to be available to provide that support, especially without leaving technical documentation which the new incumbent would find virtually essential.

![Figure 3: Applications Recorded on iServer](image)

Figure 3 shows when EUC applications were created on iServer. All the assessment templates were completed by department heads or their representatives between May and July 2018 so those entered before this time had resulted from the pilots and other sources. The presentation to the Executive (next section) covered everything on iServer up to October apart from the pilot in Financial Accounting and Solvency Reporting in recognition of the fact that they have separate controls.
During this period, we gave priority to ensure that iServer was kept up to date with all changes to do with EUC applications and iServer became the master repository for this data. The schematic in Figure 1 shows the flow of data between the Risk Assessment Application, iServer and elsewhere.

5.4 Presentation to the Executive, November 2018

A short paper was presented at the Group Executive Meeting.

The paper included Figure 4, a table of risk rating v impact relating to the 158 EUC applications extant at the time.

![Figure 4: Distribution of EUCAs by Risk Rating and Impact](image)

There were 8 applications in the red quadrant, 14 in the amber, 116 in the green and 20 in the blue. The three highest risks in the paper were recommended to have the most urgent programme for remediation are

(a) An Access database dating back to 2003 with no existing support and carrying substantial value of business. This risk was immediately put on Magique and a replacement system is being sought.

(b) The Complaints Compliance Operations Database, again in Access and without supporting technical documentation. Again, this risk is on Magique and technical documentation is being prepared.
A cluster of spreadsheets used by HR for disciplinary, grievance and absence management exhibiting poor data management in spite of the privacy and access risks being well controlled. This issue, in addition to other business needs, has led to another system being sought to replace these spreadsheets.

The Chief Risk Officer (CRO) approved the paper and this action enabled

- The revised EUC Policy to be included in the Company Controls Documentation.
- Remedial action to be put in each department’s business plan for action during 2019 commensurate with the level of risk.

The Executive require an update of the EUC situation in November 2019.

5.5 Amendments to the EUC Risk Assessment application

Two amendments were required to the Risk Assessment application and these were discovered whilst preparing for the executive meeting and to streamline the process. They were:

1. Risk Rating Band

   The Risk Rating Band had been previously calculated from a numeric risk rating which in turn is a function of the materiality, complexity and other controls on the EUC application, but not the impact.

   The impact which is collected by the Risk Assessment Application takes one of six values, being

   1. Inconvenient
   2. Poor Customer Outcomes
   3. Reputational
   4. Loss of Business
   5. Financial
   6. Statutory / Legislative

   The user who is assessing the application in question provides the highest number out of these six governed by the outcome should the application fail to function correctly.

   We realised that if an application was given a red rating the impact upon a risk crystallising would be much less if the impact was classified as inconvenient than it would be if it was (say) financial.

   Therefore, we decided to amend the calculation of Risk Rating Band to say that if the impact is “inconvenient” the risk rating band can only be blue or green, and the risk rating band can be red only if the impact is “loss of business”, “financial” or “statutory / legislative”.

2. Streamlining the process

   The requirement exists for producing the inventory, KPI reporting and re-assessing each EUCA at least annually and iServer exports metadata about a selection of or all the EUCAs to satisfy this need (see section 4).
5.6 Position at the end of 2018

We provided an EUC inventory to all department heads and asked them to keep it up to date in line with the EUC policy.

Figure 5 shows the skew distribution of applications amongst departments, the point being that 85% of the applications are contained within only 7 departments.

Appendix A lists the 7 departments in the left-hand side of Figure 5 and summarises the use of each EUCA. This information is intended to help the reader to find similar EUCAs in his own organisation.

![Spread of EUCAs amongst Departments](image)

Those departments with a smaller number of EUCAs (right hand side of Figure 5) could assess these within the time requested. For the remainder we adopted an understanding approach whereby the assessment could be part of the 2019 business plan instead. The plan for these is as follows:

- **Accounting Operations**: We received a sample of 8 assessments in 2018 and this is the tip of the iceberg. They use thousands of spreadsheets as part of their transaction processing function and one of each kind is up for risk assessment. They are enthusiastic about implementing a phased approach during 2019 and by May the number assessed reached 60.
- **Financial Assumptions**: This department provided a green rating from the Head Office Managers Meeting the previous April (Section 5.1) so no action is being taken for the time being because we see this as a lower EUC risk. This will be reviewed at some stage in the future.
- **With Profits and Capital Management**: The manager has already provided information about 62 spreadsheets and knows to submit more assessments as the need arises. No further action needed for the time being.
• Financial Accounting and Solvency Reporting: Spreadsheets of high complexity and materiality were part of one of the pilots in Autumn 2017 and 20 of these revealed a red or amber rating when assessed. Solvency Reporting uses Finsbury Spreadsheet Workbench to provide audit and version control in this area (Finsbury, 2014). Solvency Reporting also successfully uses a Spreadsheet Controls Framework which ensures appropriate peer review whereby results are challenged and locked down with financial and actuarial analysis as necessary. This is achieved by having three tabs on every spreadsheet and these are recognised as positive indicators within the EUC policy. They are:
  - Control: Contains doer & checker evidence and sign-off of the spreadsheet and version history.
  - Validation: Describes the changes, what checks are done, who did them and when, the checker and date and if necessary the reviewer and date.
  - Documentation: Outlines the purpose of the workbook, details individual sheets, and gives instructions for how to use the spreadsheet.

The Head of Actuarial rigorously enforces the Spreadsheet Controls Framework on every spreadsheet in the valuation folder by means of a macro to ensure that elements have been completed.

These three indicators are in line with the first three items as recommended in the “initial remediation plan” (McGeady & McGouran, Page 3, 2009). Our control is retained within the business and not migrated to a controlled IT environment.

In addition, audit work is done to ensure that accounts are prepared in line with statutory rules and that regulatory responses are compliant.

In April 2019 we agreed a plan with the Head of Actuarial to identify any gaps in the Spreadsheet Controls Framework where the EUC Policy is not met and then plan how to amend both the framework and the EUC policy so that the objectives of the EUC policy are still met. To be complete by the end of 2019.

6 MITIGATION OF EUC RISK

2018 gave us the opportunity to determine where EUC risk exists within the Wesleyan and how it can be mitigated.

Firstly, why EUC? EUC applications (usually spreadsheets) are written to solve a problem — otherwise why write the spreadsheet? For example, a business need has been satisfied by a system run and supported by IT for years (sometimes decades) and the need is modified or it changes.

Historically a change request takes too long for IT to implement so the department using the system goes its own way and writes a spreadsheet or cluster of spreadsheets to fill the gap. Thereupon the EUC risk arises if the control is lacking and due to any errors in the transmission of data.

The options for areas where there are EUCAs are:

(a) To mitigate the risk — in other words to ensure that controls are in place to reduce the likelihood of the risk crystallising. This is a major part of our EUC policy.
(b) To remove the risk — this is to completely avoid or bypass the EUC applications which constitute the risk, sometimes by creating a new system.
To accept the risk. When the risk crystallises, the cost is a better alternative than either of the two options above.

Where EUC applications (especially if they have a red or amber risk rating) cluster in one place we find that putting controls round the EUC applications may not completely solve the problem and the EUC policy encourages and supports option (b) above, for example:

- In HR, errors due to manual transcription of data from one spreadsheet to another, in addition to other business needs, has resulted in a new system for HR and Payroll administration being sought.
- The risks indicated by EUC in Financial Accounting have supported the requirement for an existing project to replace the Finance system and this is now in progress.
- The Access database dating back to 2003 is an example where only option (b) above is appropriate. The fact that it is isolated from our point of sale system means that we are potentially missing out on cross-selling opportunities so a replacement which will integrate with this system adds to the business case. This is also in progress.

7 BAU ACTIVITY IN 2019

Activity is ongoing to continually improve the EUC control situation and ensure that a society-wide awareness of the EUC policy continues.

KPI reporting on End User Computing has been incorporated within Data Governance from March and is supported by monthly requests of each EUC application owning manager in two areas:

7.1 Annual review of applications

Each application according to the EUC policy needs to be reviewed annually so each month when one or more applications have come up for review, these are listed and the owning manager is simply asked to confirm that the application is still fit for purpose and in use. The return of this information causes the next review date to be stepped forward a year.

If there are substantial changes to the use, materiality or complexity of the application or if there are any new applications, the owner is asked to download the Risk Assessment Application from the intranet and complete it, returning it to Data Governance.

Particularly in Q1, the returns have indicated that certain applications are no longer in use and were retired and some others indicate a change in ownership. This all helps in keeping the inventory up to date.

7.2 Risks entered on the Magique system

iServer keeps a record of whether each red or amber rated application has its risk recorded on the Magique system. If there is no such record on Magique, the monthly communication to the owning manager lists these, asking for the Magique Risk ID and the opening and closing dates on Magique. When remedial action on the application has taken place, the application is re-assessed and the assessment returned to Data Governance.

Where appropriate, risks are recorded on Magique and are tracked by Corporate Audit. This focusses the user’s mind on remediating the EUC risk.
We use established methods for turning inherent risk into residual risk and expressing the risk rating as a product of likelihood and severity (Herrera, 2017), (Xenon, 2019).

7.3 Monthly Key Performance Indicator (KPI) Reporting

7.4 Reaction from the business to the introduction of the EUC Policy

The business has not been compelled to assess EUC risk for several years prior to the introduction of this policy. Starting the communication at executive level has meant that each department has known in advance that involvement and commitment was expected of them.

Knowledge that a risk based approach is being used provides an understanding that action needs to be taken to identify and mitigate against, remove or accept the most serious risks and this has certainly helped.

Reaction is mainly as follows:

- There is general acceptance of the policy because of the way in which it was introduced.
- 108 (81%) (see figure 2) indicated that there is negligible risk so no action is being taken for the time being (or unless they get caught out)! When this happens, they are only too keen to comply because they are made aware of the risks.
- The remainder had action to do, nearly everyone accepted and are using the policy (see section 5.6). Some departments are quite enthusiastic in wanting to comply.
- Where there are existing controls (Actuarial) there was quite understandably resistance from the point of view of having to change and because of the additional work. We are working together to achieve the objectives of the EUC policy.
Resistance is at its greatest where the policy demands that an EUC deficiency calls for an entry to be made on Magique (the risk register) because this exposes the risk to Internal Audit and eventually gets executive attention. Wherever possible we encourage the department to introduce appropriate EUC controls which will avoid an entry being made on Magique.

One of the most visible issues is the matter of communication. The policy is well documented on the intranet and offers to help by showing what to do in various situations is always accepted.

8 INTERESTING SURPRISES

It’s amazing what you find if you look (or what you miss if you don’t)! When we visited one department we were expecting to receive resistance to the governance which EUC controls would apply. Instead we found that the manager was only too keen to cooperate because this policy has allowed her to justify key changes to her department’s system. The advent of the EUC policy has brought the matter to executive attention and a plan is now in place to replace the system.

We adopted a top-down approach for identification which means that each manager recognises his or her areas at risk. We were ‘tipped off’ concerning the administration of one system. We discovered that it is administered using a cluster of spreadsheets which overcome shortcomings in some very old IT systems. We saw the manager the same day only to find that the sole person who could look after the spreadsheets was leaving within a month. Somebody else was quickly brought in to be trained up to sufficient level to mitigate the EUC risks.

9 CONCLUSION

We have worked successfully during the last year to increase the exposure and awareness of EUC to the Wesleyan. We have used a top-down approach to identify areas of greatest risk. Buy-in at Executive level from the start of the year was essential because only by this means can meaningful resource be devoted to areas of need.

A flexible and understanding approach has yielded dividends. EUC control generally takes a back seat in relation to business priorities and if an action to remediate an application cannot be completed immediately because of lack of resource then we have been able to agree a plan of action.

KPI reporting has started and the Executive are looking to see an improvement in the EUC estate by the end of 2019. At the time of writing this is already evident. Now that the momentum is established, it is important not to let it go to waste in subsequent months or years.

Any models or information contained in this paper are intended for educational purposes only. To the extent permitted by law, the author and Wesleyan Assurance Society shall not be held liable for any liability or loss suffered by a third party who uses the models or information within this document for purposes for which they were not intended.

© Wesleyan Assurance Society June 2019 All Rights Reserved
REFERENCES


Orbus Software, (2018), Capabilities, Online [available] https://www.orbussoftware.com/enterprise-architecture/capabilities/?selectRegion=1, accessed 03/06/2018


APPENDIX A – USE OF SPREADSHEETS IN BUSINESS AREAS

This appendix is a brief summary of the use to which spreadsheets are put, in the areas which use the most spreadsheets.

- With Profits and Capital Management, and Solvency Monitoring – Actuarial – keeps the Society’s financial position up to date and provides information to support Solvency II legislation.
- Financial Accounting and Accounting Operations – Accounting – Preparation of the Wesleyan’s accounts, receipts and payments.
- Field Support and Proposition and 1st Line Risk – Support for the Financial Consultants and logging of brokered business
- Human Resources – Joiners, Movers, Leavers, Benchmarking, employee relationship activity, workflow
- Risk & Regulatory – Work management, Regulatory changes
APPENDIX B – ASSESSMENT TEMPLATE

Screenshot of the Assessment Template provided to those attending the Head Office Managers Meeting, April 2018 (See section 5.1)

End User Computing Scoring template

Purpose: To assist line managers with assessing the level of risk associated with use of spreadsheets and other applications as applicable. This will take no more than 10 minutes to complete.

Use the Tab button to navigate from one field to the next.

Your name
Name of your Department

A Manager
Any Department

Have you any spreadsheets or applications which are not supported by any of the Wesleyan Group’s IT departments and which are in any way material to the Business? Enter 1 for Yes, 0 for No

If you have, put a 1 here and continue with the rest of this form.
If you haven’t, put a 0 here and go straight to the end to read the instructions.

1

Pick the process which has the most complex calculations, or which has the most material impact on the Business operation

Name of the process and in which business area

Materiality of the process (insert 1 to 3 as appropriate):
1 = Internal operations or day to day decisions only
2 = Management reporting or making key business decisions
3 = Financial or regulatory reporting or Private / Confidential information

Complexity of the spreadsheet or application in the process (insert 1 to 3 as appropriate):
1 = No calculations, just logging and information tracking
2 = Simple formulae or reformatting data
3 = Complex formulae, external references, any programming languages

Control Score [1 (bad) to 3 (good)] for each of these
(Decimals, for example 1.5, 2.5 are allowed)

How confident are you that there is the knowledge in the department able to fix an application if it breaks after the person who wrote it has left?

Does this area handle personal data or sensitive personal data (GDPR)? Put 0 for no, 1 for yes

How confident are you that the processes will continue to get good, reliable results even if you’re short staffed in key areas?

Is there a lack of service, equipment breaks or is there a power outage? How confident are you that you can get back to a working state quickly?

Have you confidence that controls are in place to ensure that the latest version is being used when needed

How confident are you that the applications don’t give misleading results resulting from misuse or unauthorised access (which might be due to malicious intent)?

1.5
0
2
3
2
3

Please review your responsibilities regarding End User Computing as described within the policy document, link given below

You are Amber. Action is needed. Return this spreadsheet to Data Governance. Your spreadsheets and applications need to be assessed, errant ones recorded on Magique and there needs to be an action plan to fix.

Figure 8: Departmental EUC Risk Assessment Template
Abstract

This paper presents the findings of a case study of spreadsheet use in a higher education institution in the UK. The paper considers the use of spreadsheets in two units of the organisation, academic registry and finance. Spreadsheet use is explored in terms of importance, training, experience, purpose, techniques deployed, size of spreadsheets created and sharing of spreadsheets. The implications of the results are then considered in terms of accurate reporting to external funding bodies such as the funding councils, internal data integrity and internal data efficiencies. The results show a large volume of spreadsheets being created and used, that the profile of spreadsheet developers is typical of other studies of spreadsheet use and the need for the organisation to have clear principles and guidelines for the development of spreadsheet models in the organisation to ensure data integrity, reduce duplication of effort and to optimise the use of spreadsheets to meet the institution's goals.

1.0 Introduction

This paper is generated from a case study of spreadsheet use in a higher education institution focusing on two departments that were thought to have a potentially high concentration of spreadsheets. The institution sanctioned the research in an effort to better understand and control spreadsheet use with the organisation. This report is comprised of data generated through a detailed questionnaire, informal unstructured interviews and some background investigation of spreadsheets stored on internal shared network drives.

1.1 Aim and research questions

The aim of this research is to appraise and quantify the veracity by which spreadsheets are used in the institution and identify the risks associated with such use.

The study was conducted in two units within a higher education institution, the academic registry and finance units since they were deemed to have the highest concentrations of spreadsheets.

1.2 Research questions

- How critical are spreadsheets to the units studied?
• What approaches to standards, validity and development are undertaken by institution staff using spreadsheets
• Is it possible to utilise existing corporate systems instead of using spreadsheets?
• Could the institution educate its staff to use spreadsheets more efficiently?

2.0 Method
The two units selected for the study were academic registry and the finance unit. These two units were selected because they were likely to have the highest concentrations of spreadsheet developers and users.

2.1 Academic Registry and Finance Units
Academic Registry (AR) is a sub unit of the Institution that is concerned with official record keeping and student data, communicating with external funding bodies and regulators and producing statistics used by external bodies. Registry employs 35 full time staff across various different roles all concerned with official record keeping of various types.

Recording keeping and student data is fundamental to how the institution keeps track of student progress, fees, degree awards and other relevant student related activities. The unit communicates with external funding bodies such as the Higher Education Funding Council (HEFCE) who ensure the institution compliance on fees and academic standards. The unit produces attainment statistics used for various purposes but are used in part for the universities league table calculations.

The Finance Unit (FU) is concerned with internal budgeting, student fee collection, forecasting and management of the universities finances. Internal budgeting processes manage the operational running costs of the institution, collecting student fees, managing debtors and forecasting the universities future developments. The finance department consists of 15 full time members of staff.

2.2 Research process
Two separate processes were undertaken as part of this study. Firstly a data capture phase sought out all spreadsheet files saved onto shared drives within the finance and academic registry units. Ideally, this would have been extended to personal hard disks on desktop machines but this was not possible through privacy concerns. The intention was not to audit the spreadsheets found rather to generate an overall impression of the scale of spreadsheet use in institution and these two units.

The second part of the study explores a questionnaire designed to probe the activities and attitudes of those working in the registry and finance units. This questionnaire is a modified version of Baker et al. (2008) and as such is designed to probe processes, approaches and standards of spreadsheet creation within the departments. The questionnaire was circulated to all 50 members of both units with 39 responses returned. Some questions were only partially responded to and some questions allow more than one response.

The questionnaire asked questions around the following themes:
• Levels of spreadsheet usage.
- Approaches to spreadsheet creation.
- Training acquired and available.
- Organisation standards and policies.
- Awareness of risk

3.0 Results

This section considers the data that has been produced from the questionnaires sent to the Finance and the Academic Registry Units. This chapter will interpret the data to show any trends and relationships which can be attributed to spreadsheet risk.

3.1 An overview of spreadsheets in the institution

As part of the study, a script was written to scan all shared drives across the entire institution to discover the sheer number of spreadsheets in use or stored on servers. The results showed a staggering 228,704 spreadsheets on shared drives for around 1000 staff in the institution, this number excludes spreadsheets sitting on personal hard disks. For the units of study, AR had a total 39,006 spreadsheets and FU had 39,311. Of these 78,317 spreadsheets in both AR and FU departments, it is likely that a large percentage of these are not actively used. However, due to scale it was not possible to determine with any accuracy what these figures are. What can be concluded though is that the institution is paying to store a large number of spreadsheets that probably have no current application, that could be deleted or archived and could possibly present a significant GDPR threat to the institution.

3.2 Questionnaire Responses

It could have been possible to have upwards of sixty responses to the questionnaires, but due to factors such as internal audits, sick leave and holidays the total number of responses was thirty nine. That number consisted of twenty-four from Academic Registry (from a potential of 35) and fifteen from Finance, which represents the whole department.

3.2.1 Experience and Training

The results of the first question in this section showed that participants from both units use Microsoft Excel. A very low percentage use similar programs in conjunction with Excel.

<table>
<thead>
<tr>
<th>Spreadsheet development experience levels</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Little or no experience</td>
<td>6</td>
<td>5</td>
<td>11</td>
<td>29%</td>
</tr>
<tr>
<td>Some experience</td>
<td>11</td>
<td>5</td>
<td>16</td>
<td>42%</td>
</tr>
<tr>
<td>Extensive experience</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>18%</td>
</tr>
<tr>
<td>Expert experience</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>11%</td>
</tr>
</tbody>
</table>

Table 1 SpreadSheet development experience levels

Table 1 shows that the large majority (71%) have either ‘little or no experience’ or ‘some’ experience. These findings are then reflected in table 2 which shows only 27% as having formal training and 52% as having either no or colleague demonstration training.
### 3.2.2 Purpose, Functions and Use

Table 3 shows the main purposes of spreadsheets created by the respondents. The highest proportion of the sample indicated they use spreadsheets for data analysis (finance and operational) with maintaining lists and tracking data being the second most popular choice.

Breaking the results down into the different departments, FU, as expected, predominantly use spreadsheets to analyse data, 80%, with tracking data, such as budgets, being second, 60%. Over half of staff in the FU use statistical analysis, 55, see table 4. Budgets are critical to the financial stability of institution, so this enforces the finding that Finance consider spreadsheets critical to their role.

AR use spreadsheets for three predominate reasons, maintaining lists 64%, tracking data 50% and analysing data 50%, see table 4.

<table>
<thead>
<tr>
<th>Spreadsheet modelling purposes</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintaining lists eg names and addresses</td>
<td>14</td>
<td>6</td>
<td>20</td>
<td>27%</td>
</tr>
<tr>
<td>Tracking data eg Budgets, Sales, Inventories</td>
<td>11</td>
<td>9</td>
<td>20</td>
<td>27%</td>
</tr>
<tr>
<td>Data analysis eg financial, operational</td>
<td>11</td>
<td>12</td>
<td>23</td>
<td>31%</td>
</tr>
<tr>
<td>Determining trends and projections</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>8%</td>
</tr>
<tr>
<td>Evaluating alternatives</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5%</td>
</tr>
<tr>
<td>Other</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3%</td>
</tr>
</tbody>
</table>

### Functions used in spreadsheets

In both the AR and FU responses (table 3), data analysis implies non-trivial spreadsheet programming whilst tracking data and maintaining lists is probably less computationally complex.

Table 4 shows the functions used in spreadsheets, as can be seen “none of the above” is the most popular choice. This is mostly down to the FU department who employ ‘financial instruments’ in their spreadsheets but don’t consider those instruments to fit within the specified categories. In the AR department, statistical analysis was the most popular technique, which ties in with the nature of the work of the AR department.

<table>
<thead>
<tr>
<th>Functions used in spreadsheets</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical analysis</td>
<td>6</td>
<td>6</td>
<td>12</td>
<td>39%</td>
</tr>
<tr>
<td>Optimisation</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3%</td>
</tr>
<tr>
<td>Simulation</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>10%</td>
</tr>
<tr>
<td>None of the above</td>
<td>10</td>
<td>5</td>
<td>15</td>
<td>48%</td>
</tr>
</tbody>
</table>
When asked how many spreadsheets are used per week, 64%, use between 2 and 5 different spreadsheets per week, see table 5

<table>
<thead>
<tr>
<th>Spreadsheets used per week</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>14%</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>14%</td>
</tr>
<tr>
<td>2 to 5</td>
<td>17</td>
<td>6</td>
<td>23</td>
<td>64%</td>
</tr>
<tr>
<td>Over 10</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>8%</td>
</tr>
</tbody>
</table>

Table 3 Spreadsheets used per week

When asked if the participants create spreadsheets, the majority of participants create spreadsheets as part of their job, FU 87% and AR 91%. When asked if the spreadsheets created are used to inform their line management, 40% of FU staff and only 9% of AR staff indicated so.

3.2.3 Spreadsheet Creation

When asked about the first steps in creating a spreadsheet, 60% of participants from the FU and 70% from AR say that they only sometimes create spreadsheets from scratch. This implies that templates or previous iterations of spreadsheets are used as a starting point for both AR and FU departments. If there are mistakes in these previous iterations, then these mistakes are compounded every time the spreadsheet is used as a starting point.

When asked about the size of the spreadsheets created the majority of participants from the FU create spreadsheets with more cells than AR, 43% create spreadsheet between 101 – 10,000 cells. Interestingly, Finance have two spreadsheets and AR have one spreadsheet that has between 10,001 – 100,000 cells.

When asked about the first step the participants take in creating a spreadsheet, 47% of staff input data directly into the spreadsheet without any planning, 42% indicated that they borrow a design from an existing spreadsheet and only 7% sketch the idea before engaging the software.

<table>
<thead>
<tr>
<th>First steps in creating a spreadsheet</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Borrow a design</td>
<td>9</td>
<td>7</td>
<td>16</td>
<td>42%</td>
</tr>
<tr>
<td>Sketch the spreadsheet on paper</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>7%</td>
</tr>
<tr>
<td>Enter data directly</td>
<td>12</td>
<td>6</td>
<td>18</td>
<td>47%</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2%</td>
</tr>
</tbody>
</table>

Table 6 First steps in creating a spreadsheet

Table 7 shows further evidence of inadequate preparation. No one in the FU and very few staff in AR use any form of design methodology. This is obviously a risky practice, especially considering that some spreadsheets are very large with more than 10,000 cells.

<table>
<thead>
<tr>
<th>Do you use a software development methodology?</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3%</td>
</tr>
<tr>
<td>Sometimes</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3%</td>
</tr>
<tr>
<td>Never</td>
<td>14</td>
<td>21</td>
<td>35</td>
<td>95%</td>
</tr>
</tbody>
</table>

Table 7 Do you use a software development methodology for creating spreadsheets?
3.2.4 Testing and documentation

When asked about testing spreadsheet models, 59% said they never test and only 11% said they always test, see table 8.

<table>
<thead>
<tr>
<th></th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td>15</td>
<td>7</td>
<td>22</td>
<td>59%</td>
</tr>
<tr>
<td>Sometimes</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>16%</td>
</tr>
<tr>
<td>Usually</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>14%</td>
</tr>
<tr>
<td>Always</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>11%</td>
</tr>
</tbody>
</table>

Table 8 Do you test the spreadsheets you create?

Both Units have a high percentage of staff that do not test the spreadsheets they create (table 8), those that do test their spreadsheets, most use their own “common sense”, see table 9.

<table>
<thead>
<tr>
<th>Approaches to testing spreadsheets</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test extreme cases</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>7%</td>
</tr>
<tr>
<td>Manually calculate answers</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>16%</td>
</tr>
<tr>
<td>Display formulae</td>
<td>2</td>
<td>5</td>
<td>7</td>
<td>16%</td>
</tr>
<tr>
<td>Code inspect formulae</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>18%</td>
</tr>
<tr>
<td>Test performance for plausibility</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>11%</td>
</tr>
<tr>
<td>Error checking in Excel</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4%</td>
</tr>
<tr>
<td>Use Common sense</td>
<td>6</td>
<td>7</td>
<td>13</td>
<td>29%</td>
</tr>
</tbody>
</table>

Table 9 Approaches to testing spreadsheets (Multiple answer question)

Table 10 shows the majority of staff (58%) do not document any of their spreadsheets. Of those who said they do document, they most commonly spend only 1-10% of the development time on documenting the spreadsheets they create.

<table>
<thead>
<tr>
<th>Do you document your spreadsheets?</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Never</td>
<td>11</td>
<td>10</td>
<td>21</td>
<td>58%</td>
</tr>
<tr>
<td>Sometimes</td>
<td>10</td>
<td>3</td>
<td>13</td>
<td>36%</td>
</tr>
<tr>
<td>Usually</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>6%</td>
</tr>
</tbody>
</table>

Table 10 Do you document your spreadsheets?

3.2.5 Dissemination and sharing

64% of FU and AR staff share their spreadsheets with between 2 and 5 people, as shown in table 10.

<table>
<thead>
<tr>
<th>Typical number of users per spreadsheet</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>14%</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>14%</td>
</tr>
<tr>
<td>2 to 5</td>
<td>12</td>
<td>11</td>
<td>23</td>
<td>62%</td>
</tr>
<tr>
<td>6-10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2%</td>
</tr>
<tr>
<td>Over 10</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>8%</td>
</tr>
</tbody>
</table>

Table 11 Typical number of users per spreadsheet
As shown in table 12 the majority of staff from both Units share their entire spreadsheet model via email or sharepoint.

<table>
<thead>
<tr>
<th>Approaches to sharing spreadsheets</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>I rarely share any part of my spreadsheets</td>
<td>5</td>
<td>4</td>
<td>9</td>
<td>19%</td>
</tr>
<tr>
<td>I provide a summary of results</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>15%</td>
</tr>
<tr>
<td>I provide parts of my spreadsheets</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>11%</td>
</tr>
<tr>
<td>I share the entire model</td>
<td>16</td>
<td>10</td>
<td>26</td>
<td>55%</td>
</tr>
</tbody>
</table>

Table 12 Approaches to sharing spreadsheets

3.2.6 Modification and backup

With more than 39,000 spreadsheets stored on the network by both units, it was surprising to find that participants from the Finance Unit commented that only 29% of major spreadsheets have a lifetime of more than 2 years and in the AR only 9%. Both AR and FU responses are shown in table 13. This suggests that the vast majority of spreadsheets stored on the institutions servers are redundant. The disk space these files are using is significant and could be utilised more efficiently.

<table>
<thead>
<tr>
<th>My organisations uses standards for spreadsheet development</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>One week or less</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>11%</td>
</tr>
<tr>
<td>A few weeks or months</td>
<td>8</td>
<td>6</td>
<td>14</td>
<td>38%</td>
</tr>
<tr>
<td>One to two years</td>
<td>10</td>
<td>3</td>
<td>13</td>
<td>35%</td>
</tr>
<tr>
<td>More than two years</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>16%</td>
</tr>
</tbody>
</table>

Table 13 what is the typical lifecycle of the spreadsheets you create?

When asked about backing up of spreadsheet models, about an equal amount use sharepoint or personal storage space provided by the institution.

<table>
<thead>
<tr>
<th>Backup destinations</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>No backup, save to C:</td>
<td>4</td>
<td>7</td>
<td>11</td>
<td>19%</td>
</tr>
<tr>
<td>USB Flash drive</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>9%</td>
</tr>
<tr>
<td>Sharepoint</td>
<td>14</td>
<td>8</td>
<td>22</td>
<td>38%</td>
</tr>
<tr>
<td>Networked personal space</td>
<td>18</td>
<td>2</td>
<td>20</td>
<td>34%</td>
</tr>
</tbody>
</table>

Table 14 Backup destinations

When asked how frequently participants archive their spreadsheets, the response showed that very few archive regularly.

<table>
<thead>
<tr>
<th>How frequently do you archive your spreadsheets</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rarely if ever</td>
<td>7</td>
<td>10</td>
<td>17</td>
<td>47%</td>
</tr>
<tr>
<td>Occasionally</td>
<td>11</td>
<td>3</td>
<td>14</td>
<td>39%</td>
</tr>
<tr>
<td>Frequently</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>14%</td>
</tr>
</tbody>
</table>

Table 15 How frequently do you archive your spreadsheets?

Considering the data in table 15, it is possible to understand why there are so many spreadsheets (39,000) in comparison to the amount of staff members. The results of the question show that only 14% frequently archive their work with 47% saying they rarely if
ever archive. This is part of the reason why so many ‘active’ spreadsheets exist on shared drives within these departments.

3.2.7 Standards, Policies and Risk Perception

The responses from the participants in both Units suggest they know very little, if anything regarding spreadsheet standards and policies employed by the institution. The question asked about standards the institution sets, as shown in Table 16, 76% answered ‘I don’t know’ and 13% declaring there are no standards.

<table>
<thead>
<tr>
<th>My organisations uses standards for spreadsheet development?</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Standards</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>13%</td>
</tr>
<tr>
<td>Informal guidelines</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>8%</td>
</tr>
<tr>
<td>Formal written guidelines</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3%</td>
</tr>
<tr>
<td>I don't know</td>
<td>18</td>
<td>11</td>
<td>29</td>
<td>76%</td>
</tr>
</tbody>
</table>

Table 16: My organisations uses standards for spreadsheet development

When asked about how important spreadsheets are to the institution as a whole, 60% said very important or critical which shows that the participants perceive the use of spreadsheets to be central to the success of the institution, see table 17.

<table>
<thead>
<tr>
<th>How important are spreadsheets to the institution?</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderately important</td>
<td>8</td>
<td>6</td>
<td>14</td>
<td>40%</td>
</tr>
<tr>
<td>Very important</td>
<td>9</td>
<td>1</td>
<td>10</td>
<td>28%</td>
</tr>
<tr>
<td>Critical</td>
<td>5</td>
<td>6</td>
<td>11</td>
<td>32%</td>
</tr>
</tbody>
</table>

Table 17: How important are spreadsheets to the institution?

When asked how critical spreadsheets were to the specific unit the participants worked in, similar answers to table 17 were gathered.

<table>
<thead>
<tr>
<th>How important are spreadsheets to your individual unit?</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderately important</td>
<td>6</td>
<td>4</td>
<td>10</td>
<td>26%</td>
</tr>
<tr>
<td>Very important</td>
<td>12</td>
<td>3</td>
<td>15</td>
<td>39%</td>
</tr>
<tr>
<td>Critical</td>
<td>5</td>
<td>8</td>
<td>13</td>
<td>34%</td>
</tr>
</tbody>
</table>

Table 18: How important are spreadsheets to your individual unit?

From tables 17 and 18 it is clear that the participants understand the criticality of spreadsheets to the institution with 60% indicating they thought spreadsheets were either critical or very important to the institution, see table 17, and 73% indicating they were critical or very important the unit the participants worked in, see table 18. If we consider table 1, which was concerned with training and experience, we can conclude that even though 73% consider spreadsheets as very important or critical to their unit, 30% consider themselves to have little or no experience and only 27% have had any formal training.

When asked about the risks that spreadsheets pose to the institution, 74% of participants indicated that they thought spreadsheets posed a medium or high risk, see table 19. In combination with tables 17 and 18, it would seem that the participants have a good understanding of the level of risks posed by spreadsheets given their criticality to the institution.
Table 19 Perceived level of risk spreadsheets pose to the institution

<table>
<thead>
<tr>
<th>Perceived level of risk spreadsheets pose</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>High risk</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>11%</td>
</tr>
<tr>
<td>Medium risk</td>
<td>15</td>
<td>7</td>
<td>22</td>
<td>63%</td>
</tr>
<tr>
<td>Low risk</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>14%</td>
</tr>
<tr>
<td>No risk</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>11%</td>
</tr>
</tbody>
</table>

When asked about who should manage the risks spreadsheets pose, 86% indicated they didn’t know who was responsible and only 5 and 8% thought it was the managers or the developers responsibility respectively, see table 20.

Table 20 Who is responsible for managing the risks spreadsheets pose?

<table>
<thead>
<tr>
<th>Who is responsible for managing the risks spreadsheets pose?</th>
<th>AR</th>
<th>FU</th>
<th>Total</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>The developer</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>8%</td>
</tr>
<tr>
<td>The manager</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5%</td>
</tr>
<tr>
<td>Don’t know</td>
<td>20</td>
<td>12</td>
<td>32</td>
<td>86%</td>
</tr>
</tbody>
</table>

This confusion is likely down to a failure to develop and communicate a clear policy around spreadsheet across the institution, the answers here are mirrored in table 16 which shows that participants overwhelmingly “didn’t know” if there were policies and standards for spreadsheet use at the institution.

These results show that whilst participants are aware of some of the risks spreadsheets pose to the institution, they have no idea how to mitigate these risks, what the standards are they should adhere to and who bears responsibility for these risks. A large part of this failing comes directly from the centre of the institution who must be unaware of the risks posed by spreadsheets to institution.

3.3 Summary of risks to the institution

Given the data collected, the institution is placing themselves at risk in a number of veins. There is the risk of making errors in spreadsheets that go unnoticed and propagate to strategic decision making. There are risks in the institution communicating erroneous data to third parties and regulators via data entry or erroneous analysis in spreadsheets. There is a risk of fraudulent activities going unnoticed through lack of oversight, ownership and standards governing spreadsheets. From the FU, there is the risk financial planning, forecasting and tracking could be compromised resulting in internal failure to plan and budget, there could be dire financial consequences as a direct result. For AR which is more student focussed, risks exist in the mishandling of data, errors in student data and errors in reporting to external bodies which could put the institution in legal and financial jeopardy.

3.4 Criticality

From the evidence available, spreadsheets play a critical role in the institution on a number of levels. All financial planning and reporting is conducted using spreadsheets as evidenced by the FU responses to the survey. Information is collected, analysed and communicated to external bodies such as the Higher Education Statistics Authority (HESA) and other regulatory bodies that can affect levels of funding to the institution, strategic choices for the institution’s direction and how the external credibility of the institution is viewed.
If we apply criticality tests developed by Hamill and Chambers (2008), McGeady and McGouran (2008) or Thorne and Shubbak (2015), a large number of spreadsheet applications would be deemed critical and in need of management. If we apply other criticality tests such as those used to determine Sarbanes Oxley (SOX) compliance, communicating with external bodies and internal financial planning would place a number of spreadsheet applications in the critical category, which would need formal management.

Hence, the institution has a significant number of critical spreadsheet applications that need proper mitigation. The data obtained in this survey suggests there are a number of areas that need immediate and focussed attention to reduce the likelihood of a major problem arising.

3.5 Specific Spreadsheet Risks

As with any organisation using spreadsheets, there is a risk of making errors in critical spreadsheet applications. These errors risk the integrity of the internal decision making process but the institution also faces risks arising from other practices too such as communicating with third parties, fraud and falsification and GDPR.

The validity of the models built could be strengthened by addressing the weaknesses identified in documentation, complexity, training, organisational standards, development methodologies and testing.

3.5.1 Documentation

Table 10 shows the majority of staff from both Units do not document any of their spreadsheets. Of those who said they do document, they most commonly spend only 1-10% of the development time on documenting the spreadsheets they create. Documentation is vital for the future maintenance of spreadsheets (Pryor, 2006), since most indicate they do not document, there is a significant risk that an undocumented spreadsheet may become a major problem for the organisation if a key employee leaves without imparting crucial information.

Cleary et al. (2003) discuss one such example where a spreadsheet was used to manage contracts at a large NHS health trust. The employee responsible for the spreadsheet left the organisation without providing any documentation on the function and purpose of the spreadsheet. This spreadsheet managed clinical contracts, it took data from corporate databases, manipulated the data inside the spreadsheet and then inserted data back into the corporate database bypassing all validation and integrity controls on the database. Since its function was critical to the health trust and it was a ‘blackbox’, it could not be simply removed but since there was no documentation, it could not be maintained or checked for validity. Eventually the trust engaged a software house to forensically dismantle the spreadsheet application and engineer software to replace it.

3.5.2 Complexity of spreadsheet tasks

In terms of the complexity of spreadsheet modelling activities, figure 3 implies that some of the activities carried out are non-trivial spreadsheet programming tasks that include writing formulae which are considered more complex than simple data entry and tracking tasks.

‘Complex’ spreadsheet modelling activities generally attract higher error rates of around 5% Base Error Rate (BER). Maintaining lists and tracking data attracts a smaller BER of around 2.5% in data entry (Panko, 2008).
Hence, there is a significant chance that around 5% of the spreadsheets used for data analysis contain material errors and 2.5% of the data used for maintaining lists and tracking is erroneous. In addition, if the data used in maintaining lists and tracking is of a personal nature, there is a risk that the institution is exposing themselves to GDPR risks especially if the spreadsheets in question are not active and dormant. Indeed this must be a risk for many organisations who possess dormant spreadsheets containing personal information.

In terms of the size of spreadsheet models in the institution, the FU have two spreadsheets and AR have one spreadsheet that has between 10,001 – 100,000 cells. These large spreadsheets are probably complex, mission critical and candidates for migration to the IT function. A lack of development methods, testing, documentation and validation is a real problem when considering the scale and likely complexity and importance of 10-100K cell spreadsheets.

For the more complex applications, some mitigating retroactive action should be taken, such as code inspection. It may also be prudent to consider migration of some of these applications to formally designed software led by the information systems department inside the organisation.

3.5.3 Formal training

Tables 1 and 2 show that the majority of staff have no formal training and cite their own experience levels as either beginners with some experience or no real experience. These findings are typical and reflected in other studies, which show that spreadsheet developers usually have no formal training and little experience (Taylor et al. 1998, Grossman and Ozluk 2004 Panko 2008, Baker et al. 2006, Thorne and Altarawneh 2017, Thorne and Shubbak 2015, Mireault and Gresham 2015). Hence, the profile of the participants is broadly typical of other spreadsheet users worldwide.

The institution should aim to provide some basic how to type training and provide some training on the context of risk in spreadsheet development as advocated by Hamill and Chambers (2008).

3.5.4 Organisational standards

Table 16 shows the majority of respondents ‘don’t know’ if there are standards surrounding the development and deployment of spreadsheets.

The institution does not have any such policy on the development, management and maintenance of spreadsheet applications, which explains the responses given by participants. One participant indicated that there were formal written standards but this participant must have been mistaken since no guidelines exist of this nature. The institution should develop guidelines based around an assessment of criticality of each individual spreadsheet application to ensure that sufficiently critical applications are subject to some minimum standards.

3.5.5 Development methodologies

Tables 6 and 7 show that approaches to development are ad-hoc with most participants entering data directly into the spreadsheet or borrowing a pre-existing design. Participants also indicated that they don’t use any sort of planning methodology for developing their spreadsheets.
The institution should impose methodology standards on spreadsheet applications that are considered critical; many papers discuss how the use of a methodology can help improve the quality control and the accuracy of the spreadsheet. (Butler 2000, Diemer 2002, Thorne 2009, Thorne and Altarawneh 2017, Thorne and Shubbak 2015, Mireault and Gresham 2015)

3.5.7 Lack of testing

Tables 8 and 9 show participant approaches to testing with 59% saying they do not test their applications. For those that say they do test, most cite ‘common sense’ as the means for testing.

It is likely that the common sense test is the same as the “sniff test” discussed by Grossman et al. (2010) and Caulkins et al. (2007) which is essentially a visual check to see if the numerical values are approximately correct. In other words, these respondents do not test their spreadsheet models with any rigour or reliability. A few do employ some techniques such as code inspection, manually calculating the outcomes, plausibility testing and error checking functions in excel. However, we can infer the large majority of these models are not tested rigorously or reliably. Even those who do audit their own work are still bound by the probability that alone they will find around 60% of their own mistakes (Panko, 2008).

3.5.8 Communicating with third parties

The institution places themselves at particular risk in communicating data or analysis conducted in spreadsheets with external third parties. This is especially so if those third parties are regulatory, since making erroneous statements or having to retract statements would cause the institution embarrassment, damage its reputation and could have financial or legal implications. The institution engages with a number of regulatory bodies, both the FU and AR units provide statistics, forecasts and accounting for the Higher Education Statistics Authority and the Funding Council. Retracting financial or statistical statements could have knock on effect on the institution in terms of funding, reputation and legal consequences. In the case of AR, mistakes in student records or statistics surrounding students could be pursued legally by those affected (Nightingale, 2017).

3.5.9 Fraud and falsification

The institution also risks fraud or data falsification incidents through a basic lack of oversight and standards. Spreadsheets are often the vehicle used to commit fraud since they can be used to hide fraudulent transactions and create a clean falsified version of events. Thorne (2013) discusses examples of spreadsheet fraud and falsification involving spreadsheets in finance. Spreadsheets can be used to create false paper trails, making fraudulent transactions seem legitimate. The mechanism of falsification can be difficult to spot inside spreadsheets too, for the same reason genuine errors are hard to find. The mix of data and programming structures and the ability to hide parts of the calculation within the software or away from the visible portion of the spreadsheet makes spotting fraudulent calculations difficult. One approach to securing artefacts critical to the organisation would be to adopt a similar approach to the pharmaceutical industry who have electronic vaults that contain spreadsheets that present drug control trials.

Title 21 CFR Part 11 (FDA, 2011) protects against security violations and ensures the reliability of electronic records via the use of digital vaults to store official record evidence base for the effectiveness of drugs. 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. This should discourage fraud since the process of auditing, validating and documentation would most likely uncover any mistakes or fraud in spreadsheets. Such a system would prevent fraudulent alteration of spreadsheets, since access is limited by ‘lock and key’. Authorisation would be needed to implement any changes with subsequent auditing, validation and documentation. Such a system would make the prospect of committing fraud fraught with difficulties, higher risk and therefore a less attractive option. In addition if all organisations relying on spreadsheets were to use a system that complies with Title 21 CFR Part 11, regulators could easily check compliance.

3.5.10 General Data Protection Regulations (GDPR)

The institution also faces risks arising from GDPR introduced in the EU in 2018. This legislation dictates how information can be collected, stored, pseudo-anonymised processed and disposed of for all EU citizens. Any data that can personally identify an individual is a potential GDPR breach if such information were accidently imparted to an unintended party. GDPR breaches are usually dealt with via fines that can amount to €10 Million or 2% of the total turnover. Breaches can arise in any of the following areas: Implementation of privacy by design, data processing activities, recording of data processing activities, the data processor’s main obligations, notifications of breaches and appointments of data protection officers.

A fine of up to €20 million or 4% of total turnover for breaches in basic data processing including privacy consent, individuals’ rights of access and to be forgotten and transfer of data outside of the European Economic Area. The AR and FU units have some 39,000 active spreadsheets, some of which must contain personal identifiable data. Particularly in the AR unit, there are most likely a substantial number of spreadsheets that contain information that can personally identify students inside the institution, these spreadsheets if accidently imparted could cost the institution a large amount of money. The Italian Data Protection Authority recently issued fines amounting to €11 Million against five separate companies for breaches in transferring information outside of the EU (Corragio, 2019). With most higher education institutions operating on a truly international scale, it is not infeasible a similar situation could arise through the transfer of information internationally at the institution.

Beyond the AR and FU units, the institution stores approximately another 190,000 spreadsheets, some of which are very likely to contain personal identifiable data. It would be reasonable to assume that since the FU and AR units generally do not archive their work, this is reflected across the institution and that a good number of these spreadsheets are no longer needed for an active purpose. Hence, the institutions risk of GDPR breaches is significant and immediate action to pseudo-anonymise or delete the information should be taken.

3.6 Mitigation of risks at the institution

The following practical steps should be taken over the short and long term to address the risks the institution places themselves at

1. **Generate an inventory of active critical spreadsheet applications and take appropriate mitigating actions with the riskiest spreadsheets.**

This step is critical to controlling the institutions spreadsheet risk and should identify the highest risk applications and take immediate action to secure them. This should at the very
least allow the institution oversight on the critical applications currently in use and provide a Key Risk Indicator (KRI) that the institution can manage and track their spreadsheet risk by. One tool for assessing risk is discussed in Thorne and Shubbak (2015) called ‘Risk Calculator’ which can take a large volume of spreadsheets and give a relative and absolute risk score for each spreadsheet. This should identify the priority targets for mitigation activities. Mitigation activities should be matched to the risk each spreadsheet poses and should be considered a longer-term goal of the process. Hamil and Chambers (2008) discuss how mitigation activities should be applied to spreadsheets, they assert that for critical applications a number of minimum controls should be implemented: version control, access control, change control, business continuity measures, documentation and testing. These measures should be implemented by the institution as soon as possible. In addition, large complex spreadsheet applications should be considered for redevelopment by the IT function of the institution – there is at least one very large complex spreadsheet in the FU unit that should be considered for redevelopment.

2. Development of a robust spreadsheet development policy governing approaches to development, deployment, documentation, testing and accountability

The institution should immediately construct an organisation wide policy, which covers the development, deployment, documentation, testing and accountability of individual spreadsheet applications based on the criticality of individual applications. This policy should aim to make individuals and managers accountable for the quality, truthfulness and accuracy of spreadsheet artefacts in use in the organisation. This should prevent the risk the institution faces increasing in an uncontrolled manner and provides an avenue of accountability for these actions. This is essential in controlling and protecting the institution.

3. As part of the organisational policy, decide on development methodologies, documentation standards and testing policies for spreadsheets

The institution should adopt a structured approach to spreadsheet development that is sympathetic to demands of spreadsheet development. Grossman and Ozluk (2004) present a structured framework for spreadsheet development that takes into account the typical skills reported by spreadsheet developers. There are other approaches that would be suitable also, Mireault 2015 describes a development technique called Structured Spreadsheet Modelling and Implementation (SSMI) methodology which is designed for spreadsheet modellers. This technique is also designed with non-computing professionals in mind and could be used to model and implement complex spreadsheet models.

In a similar vein, documentation standards should also be considered and adopted. Standards presented by O’Beirne (2005) and Pryor (2006) both advocate using documentation within the spreadsheet itself in some cases using cell comments to give more detail on data structures and formulae and in other cases using a welcome sheet that covers the basic principles of what the spreadsheet does and how it calculates.

For testing strategies, Panko (2006) advocates a variety of testing approaches, some from traditional software development such as unit testing and others that are more tailored to spreadsheets such as the Fagan method. Panko also discusses code inspection and auditing as a means to find and correct errors, which remains the most effective approach.

4. Identify and deliver appropriate training in planning, development and testing of spreadsheet models
From the results of this survey, there is a clear need to train staff in AR and FU units in the basics of spreadsheet development, use of methodologies, production of documentation and testing.

5. Peer working, user groups and risk awareness sessions

The institution should establish a spreadsheet user group and encourage members to work collaboratively in development and testing of spreadsheet models. This group should have its own internal space on the network and should be allowed to share suitable spreadsheet templates, tips on writing spreadsheets and a mechanism for peer review of spreadsheet models. Chambers and Hamill (2008) and McGeady and McGouran (2008) both advocate spreadsheet user groups and noted the impact of presenting spreadsheet modellers with facts about spreadsheet risk.

3.7 Conclusions

In summary, the institutions approach to using spreadsheets is typical of most organisations. There is a lack of awareness of how critical spreadsheets are to the running of the institution, a lack of awareness over the pervasiveness of errors and the risks spreadsheets pose. Staff developing spreadsheets are not trained and generally have ‘self taught’ experience, they do not plan, develop with a methodology or test their models. The organisation has no standards or policies whatsoever to govern the use of such computing artefacts. As a direct result, the institution places itself at significant risk of a decision being made on bad numbers, assumptions or models for which the consequences could be severe. This report has highlighted those points and others in detail and provided a framework for mitigating these risks for the institution.
References


mba.tuck.dartmouth.edu/spreadsheet/product_pubs_files/SurveyPaper.doc


Mireault. P. and Gresham. L, (2015), Governance and structured spreadsheets: Your spreadsheets don't need to be black boxes, Proceedings of the 5th annual International Conference on Accounting and Finance, Singapore


Abstracting spreadsheet data flow through hypergraph redrawing

David Birch*1, Nicolai Stawinoga1, Jack Binks2, Bruno Nicoletti2, Paul Kelly1
1 Imperial College London
2 Filigree Technologies, London
*David.Birch@Imperial.ac.uk

ABSTRACT

We believe the error prone nature of traditional spreadsheets is due to their low level of abstraction. End user programmers are forced to construct their data models from low level cells which we define as “a data container or manipulator linked by user-intent to model their world and positioned to reflect its structure”. Spreadsheet cells are limited in what they may contain (scalar values) and the links between them are inherently hidden. This paper proposes a method of raising the level of abstraction of spreadsheets by “redrawing the boundary” of the cell.

To expose the hidden linkage structure we transform spreadsheets into fine-grained graphs with operators and values as nodes. “cells” are then represented as hypergraph edges by drawing a boundary “wall” around a set of operator/data nodes. To extend what cells may contain and to create a higher level model of the spreadsheet we propose that researchers should seek techniques to redraw these boundaries to create higher level “cells” which will more faithfully represent the end-user’s real world/mental model. We illustrate this approach via common sub-expression identification and the application of sub-tree isomorphisms for the detection of vector (array) operations.

1 INTRODUCTION

Spreadsheets are known to be the most common form of end user programming. This likely makes the spreadsheets end users create the most common software representation of end users’ mental model of their modelling and data manipulation problems. For example a user may model the growth of capital in a savings account using a simple compound interest spreadsheet model.

As discussed by (Hutchins, et al., 1985) users iteratively build improved mental models of how a system works in order to bridge both the “gulf” of evaluation and of understanding. Users first encounter a “gulf evaluation” in understanding the current state of the a system such as a spreadsheet which may be held in hidden formulas. Users then face a “gulf of execution” in understanding how to impact the system as they desire. However even if the user build a mental model that mirrors exactly how the spreadsheet functions we must be aware that many spreadsheets do not quite function as their creators intended (not everyone understands compound interest for example!). Indeed spreadsheets are well known to be error prone (Panko, 1998) and regularly underly large financial mishaps. Aside from human nature many reasons are posited for this in relation to the underlying spreadsheet model and its implementation in popular software packages such as Microsoft Excel or Google Sheets.

In this paper we consider one aspect of these challenges namely the low level of abstraction within a spreadsheet. In creating spreadsheet models users are forced to map their data and it’s manipulation into a rectangular cell grid. This means that lists, tables, hierarchies and other multidimensional data structures are split apart into cells only related by their locality and possibly by visual formatting cues. Such entities may no longer be manipulated conceptually as a high level entity, as a user would naturally think of doing (“Sum sales of Widgets for 2016”). Instead such manipulations are expressed as multiple operations on collections of atomised cells. This is known to be an error prone process.
Coupled with this low level of abstraction is the absence of any higher level of modelling abstraction which would be present in most other programming environments (structures, enumerations, objects, class hierarchies, UML diagrams ...). The absence of a formal higher level description of the user’s model precludes model verification – that is ensuring that the model implementation is correct “are we building the right model”. More importantly it makes the validation of models difficult – that is checking if “we are building the right model” (Easterbrook, 2010). This is particularly challenging when users are presented with a spreadsheet they did not create as they must build a mental model of the purpose of the spreadsheet from its atomised low-level primitives.

Unfortunately few tools exist (see Section 10) to express a higher level of abstraction in spreadsheet models. Further we find that few tools or methods exist to discover a higher level model description from the spreadsheet.

In this paper we seek to introduce and illustrate a method of creating higher level abstractions for spreadsheet models.

2 PROBLEM

We believe that one of the key reasons for the challenges encountered with spreadsheets is their low level of abstraction. This abstraction level rests firmly at the level of cells which for the purpose of this paper we define as:

“a data container or data manipulator
linked by user-intent to model their world
and positioned to reflect its structure”

Cells contain single data values such as numbers, text or dates or the formula to calculate such a value. These calculations may reference other cell values or groups of cell values. These links form the a calculation structure which builds a model of the user’s world. For example as in Figure 1 summing the sales of all sales agents and calculating their sales bonuses. In addition we see that the positioning of cells within the grid structure is generally done to reflect higher level structures in the real world – e.g. a list sales.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Agent</td>
<td>Sales (sqft)</td>
<td>Sales(m2)</td>
<td>Bonus</td>
<td>1</td>
<td>Agent</td>
<td>Sales (sqft)</td>
</tr>
<tr>
<td>2</td>
<td>Fred</td>
<td>5221</td>
<td>485.2952</td>
<td>100</td>
<td>2</td>
<td>Fred</td>
<td>5221</td>
</tr>
<tr>
<td>3</td>
<td>Dave</td>
<td>3872</td>
<td>359.9048</td>
<td>0</td>
<td>3</td>
<td>Dave</td>
<td>3872</td>
</tr>
<tr>
<td>4</td>
<td>Bob</td>
<td>3651</td>
<td>339.3627</td>
<td>0</td>
<td>4</td>
<td>Bob</td>
<td>3651</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Total Sales</td>
<td>1184.563</td>
<td></td>
<td>5</td>
<td></td>
<td>Total Sales</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Total Bonus</td>
<td>100</td>
<td></td>
<td>6</td>
<td></td>
<td>Total Bonus</td>
</tr>
</tbody>
</table>

Figure 1 Simple spreadsheet to calculate total sales and bonuses for agents. Note the repetitive formulas.

This definition leads us to highlight to problems with current implementations of cells which have bearing directly upon the low level of abstraction of the spreadsheet.

Firstly we see that spreadsheet cells contain a single value rather than a conceptual entity such as a list of products, this creates fragmentation of concepts into a collection of cells which are grouped only by their collocation in a grid, their adjacent labelling or formatting or by being commonly referenced as a group.
Consequently we see that all data manipulations (excluding array or table formulae) must again produce a single value such that conceptual operations must also be atomised rather than expressed concisely (“let the sales target for next year be 10% higher than this year’s sales”)

Secondly we see that the links between cells are normally hidden and difficult to comprehend. End users must interrogate individual cells to identify what references they make and how they link with other cells to contribute their calculation to the users model. While some tools exist to expose this structure (Hermans, et al., 2011), it remains inherently hidden. This makes understanding the model structure difficult.

Thirdly we see that the higher level structure of a model is only expressed informally through the positional structure of the spreadsheet. Generally we find that adjacent cells are conceptually related with some forming table dimensions (e.g. time expressed as Q1,Q2,Q3..) some forming the names of cells (e.g. “Profit:”). By this we find that cells may form lists, tables and dimensions to represent higher level concepts which model the structure of the user’s world.

Finally we note that these encumbrances mean that the manipulation of the abstract concepts within a spreadsheet is challenging as not one cell but many together represent a higher level concept. Unfortunately few tools are available to manipulate groups of cells effectively without large amounts of manual work, for a good example see (Sarkar, et al., 2018).

As such we believe that while the concept of a cell remains effective there is a pressing need to augment the current implementation of a cell which restricts the level of modelling abstraction to be too low and difficult to manipulate.

3 THE IDEA

In this paper we propose that one way to achieve a higher level of modelling abstraction is to expand what constitutes a cell. We believe the best method to do this is to start from existing spreadsheet cells and manipulate them in such a way as to both raise the abstraction level and provide users with a more faithful representation of their model.

Conceptually we undertake this by “redrawing the boundaries” of what constitutes a “cell”. We propose this by the following general steps:

1) First we seek to make the link structure explicit by transforming the spreadsheet into a graph structure. The reference structure now exposed should reflect the model of the users intent. In contrast to previous approaches (discussed in related work) we generate a fine-grained graph with functions, operators and values as nodes.

2) We then consider this graph as a hyper-graph, that is a graph where edges may connect not two but a set of any number of nodes. We observe that under this conceptualisation Cells may be represented precisely as hypergraph edges - that is a linked set of function, operator, value nodes. Visually one may express a hyper edge for a cell by drawing a boundary line or “cell wall” around a group of function/operator/value nodes. An example may be seen in Figure 7.

3) In order to raise the level of abstraction in spreadsheets we propose that researchers should seek techniques to redraw these boundaries of what constitutes a cell. Such “cells” would then provide end users with a better building block to more faithfully represent their world.

To illustrate this concept this paper seeks to use this approach to detect vector (array) operations using isomorphic subtree detection. Conceptually this algorithm seeks to identify “identical” cells and group them into higher level vector operations.
We demonstrate the generality of this approach via the detection of common sub tree expressions across the whole spreadsheet graph. These may represent unit conversions or other repeatedly used calculation.

We report the application of this method to the Enron spreadsheet corpus and are able to report a reduction in model complexity (as measured by cell count) of 27% via the identification of input vectors and vector operations.

We believe this graph-driven approach provides a mechanism for a wide range of spreadsheet analysis and refactoring tools to be constructed and many recently proposed tools could be re-implemented using this approach which we commend to the community via our open source toolkit.

We illustrate the effectiveness of raising the abstraction level by translating a series of spreadsheets to a high level data-flow language called ModL which has much bigger “cell walls”. Such models have a significantly reduced number of “cells” which are of a higher level of abstraction.

4 CONTRIBUTIONS:

We believe that the contributions of our work are as follows:

1. In section 5 we demonstrate how two force directed layout algorithms reveal hidden spreadsheet structure when applied to spreadsheet derived cell level graphs.
2. In section 6 we introduce a fine-grained operator-level graph representation of spreadsheets and demonstrate its versatility by constructing such a graph for each of 99+% of the Enron spreadsheet corpus.
3. We propose a general method for abstracting spreadsheet models by “redrawing the cell walls” when cells are represented as hyper-graph edges in a fine-grained graph representation of the spreadsheet, this is shown in section 7.
4. In section 8 we demonstrate this by introducing vector operation detection via sub-tree isomorphism and find a compression rate of 27% on a sample of 2124 Enron spreadsheets.
5. We further show the potential opportunity for common sub-expressions to be factored out of the spreadsheet hypergraph by highlight how prevalent structurally isomorphic expressions are within the Enron corpus.
6. Finally in section 9, we introduce a graph based data flow modelling in an application called Wire with “larger cell walls” which can raise the abstraction level by enabling “cells” to contain and operate on multi-dimensional vectors. Manual translation results in a 95-98% reduction in model complexity as measured by formula counts.

5 EXTRACTING GRAPH STRUCTURE

“linked by user-intent to model their world”

Inspired by a long history of graph based representation of spreadsheets (Pertti & Sajaniemi, 1991) (Igarashi, et al., 1998) we seek to expose the hidden linking structure of the spreadsheet. These links are formed from cells with formulas which reference other cells, or ranges. This creates a directed acyclic graph. We believe this structure is formed by endusers as a reflection of their model of their world. That is, the way cells or groups of cells are linked together are a reflection of the entities and their relationship in the users world. For example profit may be calculated by subtracting sales from the cost of doing business.
5.1. Cell level Graph

Formally we form the cell graph by the following worklist algorithm starting with a user selected final result cell:

1. Add first cell to the worklist
2. For each cell in the worklist:
3. Parse the cell formula
4. Identify references within the formula
5. Remove this cell from the worklist
6. Expand all range references and named ranges to their constituent cells, add to the worklist

This algorithm may be run in parallel. We then form the graph with nodes for cells, ranges and named ranges; the latter with edges to their constituent cells. We then add edges to the graph between cells and the cells and ranges they have been determined to reference. Graphs are written to the standard GraphML file format (Brandes, et al., 2001) which can encode properties of nodes and edges. We encode the cells location (sheet, row and column) as well as their values and formula. This graph is then used as the basis for visual analysis.

5.2 Graph layout for Visual Analytics

To make sense of a graph it is necessary to map the nodes into two dimensional (XY) space. The objective of this mapping is to permit the structure of the graph formed by its edges to become apparent. A common family of algorithms for such graph layout are force direct algorithms. These algorithms treat the layout of a visualisation as a particle physics problem. A common example of such a problem would be to predict the motion of one or more planets as they orbits a sun which can be complex as each particle (sun, planet) exerts a gravitational force upon every other body. These algorithms treat the graph nodes as a particles with forces acting on each node. These forces are allowed to interact for a period of time as the graph nodes move through 2d space. These forces are defined in a variety of ways which reflect the edge structure of the graph and permit it to influence the layout of the final graph. The intention being to find “structure from chaos”.

Force Atlas 2

In principle these algorithms are highly computationally expensive as the forces acting on one node may in general act upon all others, however a number of optimised algorithms have been developed which approximate the solution of these physics problems. One efficient such algorithm is the Force Atlas 2 algorithm (Jacomy, et al., 2014).

Under this algorithm the following forces are defined:

- A “spring” force is defined for each edge which seeks to draw its two nodes closer together.
- A “repulsion” force acts between every pair of nodes which seeks to spread the nodes apart.
- A “gravity” force is placed in the centre of the 2d space which seeks to draw all nodes toward it. This acts as a gravity well and stops nodes from flying off into infinite space.

Solving the particle physical problem for node locations influenced by these forces enables, with suitable configuration of their relative strengths, an informative trade off of clarity and complexity in the layout. Under these circumstances structures should become apparent.

As an example we consider a small 2700 cell spreadsheet model developed in the construction industry to answer a commercial question and explore the scenarios and circumstances around it. We create the cell level graph as described above and show it in Figure 2. Within this figure we see large islands of structure represent calculation models, generally, but not universally, there is a correspondence between these islands and different worksheets since they tend to be broadly
independent from one another. These can be highlighted by a change of colour mapping. The orphaned cells are unreferenced and usually blank cells or textual labels which surround the extracted models. Further structures apparent in Figure 2:

A. A dragged out chain of cells of the form A2=A1*1.10 used to roll a percentage increase over time.
B. A shared and commonly used numeric assumption in the model.
C. A triangular structure of cells calculating net present value of future profit. Each year along the arrow is further into the future and must be projected back one year further, giving the growing complexity of the calculation.
D. A comparison between two business cases. Note the two structurally identical calculation chains being compared differ only by different numeric business assumptions.
E. Examples of the business assumptions being frequently referenced in the calculation model.

Figure 2 Cell level graph of a typical model from the construction industry to answer a commercial question and exploring the effect of different build completion dates.

**Multi-source of Gravity Force Atlas 2**

“(and positioned to reflect its structure)”

Developed by Suryansh Rastogi and David Birch the multi-source of gravity algorithm extends the Force Atlas 2 algorithm by enabling an infinite number of sources of gravity. Each node may be given
its own source of gravity. This is done by specifying a gravity_x and gravity_y coordinate for each node. This enables a number of new possibilities for graph layout. Specifically for spreadsheets we may assign the gravity x and y to be the (integer) columns and row of each cell. For range and named range nodes we assign the top left value.

This permits the original spreadsheet grid placement structure to be reflected in the graph layout. Since the graph layout is solving a particle physics problem we may trade off the relative strengths of the various forces acting upon the nodes. Specifically we may trade the relative strengths of:

- The “spring” forces acting where edges are present to reflect the reference and dataflow structure of the spreadsheet
- The gravity forces which seek to tether nodes to their original grid locations.
Figure 3 shows the effect of changing the relative strength and interaction of these forces. The top left figure has gravity forces high enough to force all nodes to reside at their grid location. The triangular in the references which is given by the user is clearly apparent under high gravity conditions. This represents the triangular nature of the Net Present Value structure identified in Figure 2 C. As gravity decreases relative to the edge forces we see clusters of cells being to form each of which clearly depends upon the subsequent one – a structure not entirely clear in the triangular layout. In this way
the increasing size and complexity of the group of cells calculating each year’s net present value structure is clear from the growing clusters and the dependency year to year is also shown. Array (range) operations are also shown in the top half of each year with the dependency between them being clearest at intermediate levels of gravity.

As the strength of gravity is reduced through 5 orders of magnitude we see the grid structure weaken and enable the reference structure to determine, via its forces, the layout of the graph. This transition in force strengths enables a compromise to be sought which enables both the grid structure imposed by the user and the reference structure to be seen and learned from. This we believe will create the most informative graph layout in terms of identifying structure from both sources visually. In this example we believe the first graph of the second row has the best compromise of user dictated

Unfortunately it is currently unknown how to automatically balance these forces to find the most visually insightful graph and this must be done by human eye. Inspired by (Igarashi, et al., 1998) we would encourage the reader to experiment with the interactive variation of the relative balance of these forces particularly with sufficient computing power to enable interactive animation of the spreadsheet graph as it transitions between grid structure and reference structure. The graph layout is available as an open source plugin to the Gephi graph visualisation tool.

6 FINE LEVEL GRAPH

In contrast to the preceding graphs and the literature on graphs created from spreadsheet we now introduce a more fine-grained graph. Cell level graphs while they show structure still hide the calculation formulas, much as they are hidden within Microsoft Excel, requiring user interrogation to view.

To provide a graph view of the calculation structure inside cells we start with the parse tree graphs generated by the excellent XLParser (Aivaloglou, et al., 2015). This robust parser produces an Abstract Syntax Tree reflecting the formula structure with nodes representing values, functions and operators. Also included within these per-cell graphs are various annotation nodes which hint to an evaluator or printer how to parse or print the formula effectively.

Since this graph is complex and difficult to read we introduce 8 refactoring steps to reduce the verbosity of these syntax trees. These are shown in the table below and have the effect of turning the Parse tree into a clearer tree structure which operates from the root downward with the arguments for operators and functions forming their direct children.

<table>
<thead>
<tr>
<th>Refactoring</th>
<th>Purpose / Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveReferenceNodes</td>
<td>Ensure that reference nodes appear only before cell references</td>
</tr>
<tr>
<td>BeforeFunctionCalls</td>
<td></td>
</tr>
<tr>
<td>RemoveFormulaEqNode</td>
<td>Remove the root node for formula trees so that the cell node may be used instead</td>
</tr>
<tr>
<td>InlineFunctionNames</td>
<td>Make the function name the root of the subtree</td>
</tr>
<tr>
<td>RemoveConstantNodes</td>
<td>Remove type indicator for constant nodes.</td>
</tr>
<tr>
<td>RemoveFormulaNodes</td>
<td>Make the function name the root of the subtree</td>
</tr>
<tr>
<td>RemoveNumberNodes</td>
<td>Remove type indicator for constant nodes.</td>
</tr>
<tr>
<td>RemoveArgumentNodes</td>
<td>Make the arguments of a node its direct children.</td>
</tr>
<tr>
<td>TruncateReferences</td>
<td>Truncate reference signatures for later processing.</td>
</tr>
</tbody>
</table>
Figure 4 Parse tree graph of the formula “=IF(C4>AVERAGE(C$2:C$4),100,0)” in cell D4 as generated by XLParser, prior to our refactoring. Compare with Figure 5 for the whole spreadsheet in Figure 5.
Figure 5 Fine-grained graph of the spreadsheet shown in Figure 1
Thus far we have created a single cell’s formula tree. While these are interesting we seek to form a whole spreadsheet fine-level parse tree led structure. This is done by expanding the Cell level graph discussed in section 5 by linking each cell node with its parse tree graph, such that each cell node links to the root of its parse tree. The edges originating from the cell node are now transferred to the relevant parse tree reference node which expresses the generated cell reference. Such references are identified by seeking Reference nodes in the formulas abstract syntax tree and converting their subtree into standard spreadsheet references to cells, ranges or named ranges. For the avoidance of doubt we continue to include explicit range and named range nodes which link to the cells they reference.

We further note that such graphs can be formed by extracting every cell within a spreadsheet rather than by recursively following references from one formula. The latter graphs form a sub-graph “slice” of a whole spreadsheet. For the remainder of the paper we now consider whole spreadsheet graphs.

To consider the generality of forming such complex graphs we sought to extract a fine-grained graph for each of the spreadsheets within the Enron spreadsheet corpus (Hermans & Murphy-Hill, 2015), of 15,399 spreadsheets considered 94.49% parsed, 4.72% failed due to external libraries providing a success rate of 99.2%. The remaining cases are caused by unusual reference structure. These graphs form the basis of our further analysis as we seek to raise the level of abstraction.

7 HYPERGRAPH VIEW

One challenge with the fine level graph is its complexity which hides the cell structure of the spreadsheet. To address this we can consider the graph as a hyper-graph, that is a graph where edges can connect not just two nodes but a set containing multiple nodes. Represented visually one can consider a hypergraph edge as drawing a boundary around a group of nodes which are contained within the hyperedge set.

An example of this hyper graph view of the graph may be found in Figure 7 which shows a hypergraph view of the fine grain graph shown in Figure 5.

Such a view provides both the high level cell-level graph discussed previously as well as the detailed structure of each cells data manipulation. Such graphs may be laid out recursively using either of the algorithms mentioned above at the cell level followed by a second application within each hypergraph edge. Alternatively the Sugiyama (Eiglsperger, et al., 2004) family of layered graph layout algorithms provide the additional benefit of providing a flow-diagram structure with links all focused in the same direction (e.g. top down in Figure 7). There also exist implementations capable of laying out nested (hyper)graphs (see for example Microsoft Automatic Graph Layout https://github.com/Microsoft/automatic-graph-layout).

The premise of this paper is that the key to reducing the risk of using spreadsheets is increasing their level of abstraction. We contend that this is precisely analogous to redrawing the boundaries of the hypergraph edges representing the cells of the spreadsheet.

Results of 15399 tests.  
Total Results:  
PASSED:14551 (94.49%)  
EXTERN_ClosedXML_Pivot:51 (0.33%)  
EXTERN_ClosedXML_Parse_Failed:440 (2.86%)  
EXTERN_XlParser_Failed:54 (0.35%)  
EXTERN_XlParser_ArrayFormula:75 (0.49%)  
EXTERN_OutOfMemory:106 (0.69%)  
BUG_StrangeReference:28 (0.18%)  
BUG_ComplexReference:61 (0.40%)  
BUG_WeirdNamedRange:14 (0.09%)  
BUG_NoKnownEdge:15 (0.10%)  
Other:4 (0.03%)  

Final score = 99.20%
Figure 7 Hypergraph view of the spreadsheet shown in Figure 1. Note the repeated structure at each level of the graph

8 STRUCTURE DETECTION

By analysing this hypergraph view of the spreadsheet we seek mechanisms for abstracting the spreadsheet by redrawing the boundaries of cells. To identify opportunities for this process we seek to find repeated structure within the graph which may be replaced by a higher level more abstract “cell”. In order to do this we explore three abstracting operations:

1) We seek table structures of input values.
2) We seek vector (array) operations which apply the same operations to related data.
3) We seek common sub-expressions within cells which may be factored out to simplify the graph.
8.1 Input vector detection

“{and positioned to reflect its structure}”

Our objective here is to identify the inputs to a spreadsheet graph which have been entered by the user. Such inputs fall into two general groups, those that are set and used alone such as model assumptions like exchange rates and those that are used in lists or tables. We term such larger sets of inputs measures which are often related by an index or dimension such as time, sales office or product name. Such dimensions may be shared across multiple measures for example sales in different offices over time. Such cross products are common and lead to multidimensional models which can be difficult to model in spreadsheets.

These larger structures are closer to the users mental model and are generally what the users seek, in objective, to manipulate (as appose to a group of loosely connected cells). We seek to identify them so as to abstract them as single logic entities.

To undertake this we consider the subgraphs defined by each cell hyper-edge (these sub-graphs form a tree structure with the cell node at the root) and identify those which a) have no external references and b) who’s root node is referenced by at least one (standard) edge. This definition permits us to include simple calculations (=24*60) as inputs in contrast to other approaches. One could further extend this definition to permit references to assumptions as defined above.

Once such candidate input cells are detected we seek to learn from the intuition the user has positioned them with. We do this by using the common Greedy Rectangles algorithms as in (Sarkar, et al., 2018) where a cell is chosen from the list and a vector is grown rightward if the cells exist in the list, the vector is then grown downward if all the cells in that new row exist in the list. Once a vector may grow no more the cells are removed from the list and the process is repeated with a new seed.

We suggest one improvement in this approach which is to permit the vector to grow over blank cells (which are not in the list) provided conditions are met which avoid the elision of adjacent but spatially disjoint vectors. For example should the first row of a table contain a blank cell then the table may truncated. Alternatively one might consider merged cells above or right which likely represent table legend to permit to enable vectors to bridge blank spaces and grow beyond them. Continuous columns or rows of textual legend may also serve the same purpose. Such an extension enables sparse tables to be identified. Further we note that the choice of initial search direction (leftward or downward) does place limitations on the size of vectors which are identified in some cases. Computing in both directions and removing the largest discovered vector may be preferable.

8.2 Isomorphic Subtree Vector detection

Using the premise that cells that manipulate data approximately the same way are likely related we seek to identify isomorphically identical cell subgraphs. Two sets are said to be isomorphic if there is a “one to one” mapping function between them which preserves relations between their elements. In our hypergraph we are seeking to identify group of hypergraph edges which demark sufficiently identical sub-graphs that we may consider replacing them with a single “master” version which operates on more data. This abstracting may be considered redrawing the boundary of a cell to include all similar copies. The new “cell” with larger cells walls would, we hope be easier to interact with and manipulate than a large group of cells.

To detect vector operations we first start with the subgraphs formed from the nodes of each of the cell’s hyper-edge. These form a strict tree structure. We identify these graphs by walking the graph from the root cell node until other cell nodes are identified (we include cell nodes but walk no
further). This permits us to walk over and include Range and Named ranges such that a reference to A1:A2 and A1:A3 may be distinguished by the number of children of the Range node.

The nodes across all of these trees are split into equivalence classes based on their type which is assigned a prime number as an identifier. The rules for defining whether two nodes are within an equivalence classes permit interesting variation in the applicability and success of what this approach will find. For the objective of identifying structurally isomorphic “cells” in terms of the formula they represent we assign all function and operator nodes to their own classes (+, -, sum, if, …), all textual inputs are grouped into one class, numeric inputs also form their own single class. Finally all reference nodes (to a single cell) are placed in the same class.

Having defined such classes each “cell” tree is then walked in Post order (visiting children prior to the node) so that an equivalence class value may be set for each node. This is done by multiplying together the equivalence classes of each of a node’s children with its own equivalence class identifier. Since all identifiers are prime numbers we are guaranteed that, up to the permutation of its children, the identifier generated for a node will be unique. Of course this permutation may affect the outcome of a formula (e.g. 0/1 vs 1/0) and should be checked as part of result validation.

Once classes have been calculated for all nodes in each sub-tree, vector detection may proceed by grouping cells by the equivalence class of their root node. This signifies they root an isomorphic subtree. Since we wish to learn from the intuition the user imparted to the spreadsheet by positioning the cells to reflect the structure of their world we seek to create spatially adjacent vectors within each isomorphic group of cells. This is achieved by running the modified greedy rectangles algorithm discussed in the previous section. We further suggest that one might consider accepting vectors with constant gap or offset (e.g. every other column) as this is a common pattern.

Unfortunately it is not sufficient to identify simply Structurally Isomorphic cells, further tests must be made:

i. **Constant Isomorphism** – isomorphic formulas must contain identical constants (textual and numeric). This is achieved by walking each tree in the same order (e.g. post order) and extracting a list of constants. The list of cells may then be partitioned by testing pairwise equality on their lists of constants.

ii. **Reference Isomorphism** – Isomorphic cell graphs should make reference in a isomorphic manner, that is there should be an equivalent reference in B for each of the references in A. The obviously isomorphic case to identify is if each graph references the same cell (or same set of cells), such as a shared assumption. However if all cell graphs reference a cell to their left we would still seek to identify an isomorphism by identifying this mapping. This is done by extracting the relative offset of the cell reference. Thus in a similar way to i, one may walk each cell graph in post-order forming a reference list and then partition a group of cell graphs into reference isomorphic subgraphs by considering a pairwise comparison of their reference lists which successes if either A) the cell referenced has the same identifier or B) the same relative offset from the origin cell at the root of the tree. We note that since we walked over the Range and Named range nodes so as to include them in the “cell” subtree they are dealt with by this algorithm without special case.

iii. One might also consider isomorphism by ensuring that cells are referenced in an identical way within the graph, and this may be done via the same method but considering the incoming edges to the root cell node. However experience shows us that this is not an effective means of recovering user intent as vectors are often broken by users “special” cases – such as adding a reference to the 3rd column of a vector to highlight “Sales in March”.

Having now grouped cell subgraphs into vectors that meet all of the following isomorphic criteria:

1. **Structural isomorphism** – the same operators in the same order
2) **Colocation** – located in a contiguous cell space
3) **Constant isomorphism** – using the same textual and numeric constants in its calculation
4) **Reference isomorphism** – using isomorphic references within its calculation.

We may redraw the boundaries of these cells to coalesce them into a single larger more abstract “cell” which operates upon vector(s) of data. Such a single data manipulation should prove easier to work with than individually modifying the original group of cells.

One can consider this abstraction process as analogous to extracting and using a function method in a programming language. As in that case our goal is to provide the users with a more abstract and reusable representation of a piece of logic which makes it easier to “link cells by user intent”.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Agent</td>
<td>Sales (sqft)</td>
<td>Sales(m2)</td>
<td>Bonus</td>
</tr>
<tr>
<td>3</td>
<td>Fred</td>
<td>5221</td>
<td>485.29521</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>Dave</td>
<td>3872</td>
<td>359.90482</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Bob</td>
<td>3651</td>
<td>339.36273</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Total Sales</td>
<td>1184.5628</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Total Bonus</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 8 Detected input vector B3:B5 and two sets of isomorphic cell graphs C3:C5 and D3:D5*

To demonstrate this, we have conducted experiments on a sample of 2124 Enron spreadsheets. For these, we report an average of 11% input vector cells out of all non-blank cells. Likewise, we identified an average of 16% vector cells, adding to a total compression rate of 27% on average. Figure 9 plots these three components, showing spreadsheet size measured in non-blank cells on the y axes, as well as the input, vector, and total percentages on the x axes. Naturally, the plots do not include experiments with non-zero vector components.

*Figure 9 Plotting the number of vector, input, and total percentages for a sample of the ENRON spreadsheet corpus*

### 8.3 Common Sub-expression Detection

Another redrawing operation for our hypergraphs is to consider the centralisation of duplicated parts of the graph. For example perhaps a calculation is commonly performed – such as conversion from
square feet to square meters in the spreadsheet in Figure 1. Alternatively perhaps a common series of formulas are frequently combined to perform a common task. We know this frequently the case from (Middleton & Murphy-Hill, 2016) who studied common function pairings. Extracting these common sub-expressions would further abstract the spreadsheet and reduce the burden placed upon the end user.

To explore the potential for this approach on a finer grained graph we analysed 694 spreadsheets taken the Enron corpus and sought to identify how many structurally isomorphic subtrees they contained. On average each spreadsheet had 43.4 different subtrees of at least 3 nodes. The average number of nodes within a subtree was 15 and the average number of instances of the subtree per spreadsheet was 90. Figure 10 shows the distribution of the size of each subtree and the number of instances for subtrees of at least 3 nodes.

While these instances would need to be tested for other forms of isomorphism we believe this demonstrates the potential for a common sub-expression analysis to reduce the scale of spreadsheets by factoring out commonality and permitting end user programmers to follow the DRY (don’t repeat yourself) principle common in programming.

9 A WORLD WITH BIGGER CELLS

Our grant partner Filigree Technologies is building a commercial data modelling platform 'Wire' based on the idea of working at a higher abstraction level than available in spreadsheets. The model logic is presented, and interacted with, in a graph based dataflow environment, which processes the multi-dimensional measure structures discussed earlier. The data flow environment may be seen in Figure 11.

By allowing the user to operate at a higher abstraction level, where the formulae expressions ('modl') embody the conceptual meaning of the processing applied, models are significantly reduced in perceived complexity, resulting in fewer errors and easier maintenance plus collaboration. In internal
tests models can be expressed in around 2-5% of the formulae utilised for equivalent spreadsheet models, with a corresponding reducing in boilerplate errors.

Separating data structure, view structure and logical data flow additionally offers considerable benefits in:
- reusing logic as templates which can be simply reapplied to new situations, without the need for repetitive manual restructuring seen in spreadsheets
- handling live data sources, where incoming data structure potentially varies over time, to allow model results to stay up to the minute
- exploring the resultant model to ascertain response to volatility in input assumptions, so as to make the most informed decisions

Utilising the research presented has allowed Excel spreadsheets to be imported into the Wire environment via an intermediate ModL representation, enabling spreadsheet models to be translated to a higher abstraction level, for subsequent maintenance, modification and exploration. As an example in Figures 11 and 12, we introduce a ModL implementation of the spreadsheet in Figure 1. The number of cells in the Excel model is 20 for 3 sales staff (8+N*4) the corresponding ModL graph contains 12 nodes. Each of which has a bigger boundary than a spreadsheet cell. Moreover consider the relationship between the number of cells and the numbers of sales staff in the spreadsheet it is 8+N*4. In ModL the number of graph nodes remains constant at 12 and requires no manual intervention to scale – whereas the sales and bonus totals would both need manual, error prone, adjustment.

Figure 11 Translation of the spreadsheet from Figure 1 in Filigree technologis ModL language (right) and shown as a dataflow graph (bottom centre).
10 RELATED WORK:

There is a rich history of academic research into spreadsheets and this work draws heavily from it. Most notably this builds our Extraction and Analysis methodology for spreadsheets introduced in (Birch, et al., 2013). There is a wide literature exploring the prevalence and nature of errors in spreadsheets. A good survey is provided in (Panko, 1998). The spreadsheet grammar used as the basis of our work is introduced by (Aivaloglou, et al., 2015). Other areas of related literature are as follows:

10.1 Spreadsheet Corpuses

A number of spreadsheet corpuses are available to researchers. FUSE (Barik, et al., 2015), EUSES (Fisher & Rothermel, 2005) and ENRON (Hermans & Murphy-Hill, 2015) being the most notable. We would commend the comparison in (Jansen, 2015) which inspired our choice of the ENRON corpus as test matter for this paper.

10.2 Extraction of Structure

A number of works have sought to extract structure from spreadsheets through 2d search algorithms exploiting the “structured to model their world” motif of cells

- (Pertti & Sajaniemi, 1991) is an example of early work on extraction of the tree structure formed by spreadsheet cells and their references a simple categorisation of such references is then proposed.
- (Mittermeir & Clermont, 2002) seek to identify high level structure and introduce a concept of copy equivalence and structural equivalence which is equivalent to our isomorphic AST graph. The author would commend study of (Clermont, 2003) for many detailed insights into structure extraction.
- (Hermans, et al., 2010) go one step further and seek to extract class diagrams which resemble those familiar to software engineers
- The visualization of these structures is generally represented within the spreadsheet such as in (Hodnigg & Pinzger, 2015)
- On a more fine-grained basis (Middleton & Murphy-Hill, 2016) seek to understand frequently combined functions in a visual way.
- (Eberius, et al., 2013) takes a higher level view and seeks to extract relational data from spreadsheet tables through a process of normalisation more commonly found in the database community.
More recently machine learning has been used for the detection of spreadsheet tables (Koci, et al., 2016) this method has been shown to be effective (Koci, et al., 2017) since it is able to deal with the noise generated by inconsistent labels and formulas introduced by spreadsheet users.

In addition there are several spreadsheet audit tools which include the ability to colour tools that visualise the blocks of identical formulas, these include FastExcel (http://www.decisionmodels.com/fastexcel.htm) and XLTest (http://www.sysmod.com/xltest/index.htm).

10.3 Spreadsheet Refactoring

In recent years a few works have applied the software engineering process of code refactoring to spreadsheets. This is where code is transformed so that the same functionality is expressed in a “better” way, perhaps using a different more concise code concept – such as a for loop or ternary expression. This refactoring process is in essence a transformation of a spreadsheets underlying AST graph. Indeed the redrawing of cell boundaries is precisely a form of refactoring and a quite general one. We believe that several recent works could be implemented through the process we describe.

- (Badame & Dig, 2012) introduce 7 refactoring tools for end users such as the introduction of cell names and the extraction of constants as well as checking whether a range of cells have constant formulas which is precisely analogous to our isomorphism check.
- (Zhang, et al., 2018) use an AST driven approach to specifically consider the simplification of the very common nested IF statement.

Most of these refactoring techniques work within the existing cell walls in that they always refactor to a cell compatible format, we believe that further value could be given by permitting refactoring “outside” the cell walls.

10.4 Visualisation

The transformation of spreadsheets into a different visual formats for easier comprehension has a rich and inventive history:

- (Igarashi, et al., 1998) introduce the now common Dataflow graph and permit it to be edited directly by the end users. Unusually the animation of these graphs and spreadsheet annotation (colouring and arrows) is discussed, primarily through the step by step reveal of the “march of computation” to provide a “narrative expression”. It would be interesting to evaluate the efficacy of animation in aiding comprehension as to the authors knowledge this has not been considered before. The force directed algorithms discussed in our paper provide an interesting opportunity for animation to transition between the grid structure and a data flow driven graph layout.
- (Hermans, et al., 2011) uses a concept of a levelled data flow diagram to represent different conceptual levels of a spreadsheet reflecting the workbooks of the model. This is extended in (Hermans, et al., 2011) to provide a visualisation tool for aiding spreadsheet comprehension.
- (Shiozawa, et al., 1999) provides an introduction to the use of 3d to visualise different layers of spreadsheets which are taken as the data, formula and reference abstractions.
- (Kankuzi & Ayalew, 2008) visualise a cell level reference graph using tree view and compound fisheye graph layouts generated using a Markov Clustering Layout algorithm to group cells. It would be interesting to compare this algorithm to its force-directed cousins outlined in the paper.
- The efficacy of spreadsheet visualisation is an important question and the authors would commend the reader to consider (Goswami, et al., 2008)
10.5 New Spreadsheet tools

Over the years many tools have been presented with a view to replacing the spreadsheet. Some such as (Litt, 2018) seek to augment implementations such as Excel with new functionality such as the ability to manage multi-dimensional data. It is speculation to consider the future of the spreadsheet. It is surprising the technology is now approaching its 40th anniversary a collection of the challenges facing spreadsheets may be found in (Birch, et al., 2017)

10 CONCLUSIONS AND FURTHER WORK:

In this paper we have argued that one of the central challenges of the spreadsheet and the cause of much of its error-proneness is its low level of abstraction centred around the concept of a cell. We believe there is nothing inherently wrong with the concept of a cell which define as

“a data container or data manipulator
linked by user-intent to model their world
and positioned to reflect its structure”

However we believe current implementations are constrained by their low level nature in that they only hold simple scalar values or the means to generate them. The lack within modern spreadsheets of higher level abstractions which can be manipulated as easily as cells leads to large volumes of cells which in aggregate represent a high level concept. Such groups of cells are generally only associated by their spatial positioning to reflect a higher level structure through use of a grid.

What constrains spreadsheets to their grid? We believe the grid is now primarily used to enable end users to reflect the structure of their world and its data by positioning cells adjacently. If the concept of a cell is expanded to permit it to contain higher level structures such as lists and data tables then the need for this constraining structure is reduced. Instead the natural dataflow with which users link the cell to model their world can be brought to the fore through graph-based dataflow. We believe this is the future for spreadsheets.

We have shown that the transformation of a spreadsheet into a fine-grained operator graph permits us to consider cells as hypergraph edges grouping a set of operators. We have discussed how visualisation of this graph structure may be achieved to highlight the hidden calculation structure of the spreadsheet and introduced a graph layout algorithm which permits the interactive blending of the layout between the calculation structure and the user imposed grid structure.

Within this hypergraph many spreadsheet refactoring’s can then be seen as the redrawing of the boundaries of this group. For example the introduction of a named cell or the replacement of a common subexpression.

Further we have explored spreadsheet vectorisation through this motif and shown that through the use of subtree isomorphisms we can identify numerous opportunities in real-world spreadsheets for vectorisation to reduce the size and complexity of spreadsheets and their volume of formulas.

The utility of such an approach however depends upon the adoption of a modelling framework with “larger cell walls”, that is where cells may contain more than scalar values and where data flow is a primary user facing concern.

We believe that there remains much further work to done to help end user programmers and the spreadsheets they create enter the age of big data successfully. We see three streams of future work.
Firstly there is much scope for the introduction of new spreadsheet refactoring’s using hypergraph redrawing and subtree isomorphisms and the reconstruing of existing refactoring’s into this formalism.

Secondly there is a need to develop new spreadsheet inspired formalisms with larger cell walls which support a higher level of abstraction without alienating accomplished end user spreadsheet programmers.

Finally we have shown the potential for the vectorisation of spreadsheets through the identification of isomorphic operations, this paves the way for rapid performance increase and for spreadsheets to process dramatically larger volumes of data in the age of big data.

Acknowledgements:

The work has been supported by InnovateUK under grant reference 104141 and through EPSRC Platform Grant EP/P010040/1. This work would not have been possible without the contribution of Filigree technologies who are actively “enlarging the cell walls”.

10 BIBLIOGRAPHY


ABSTRACT

Seasoned Excel developers were invited to participate in a challenge to implement a spreadsheet with multi-dimensional variables. We analyzed their spreadsheet to see the different implement strategies employed. We identified two strategies: most participants used a projection of three or four-dimensional variables on the two-dimensional plane used by Excel. A few participants used a database approach where the multi-dimensional variables are presented in the form of a dataset table with the appropriate primary key. This approach leads to simpler formulas.

1 Introduction

Expert guidelines for developing spreadsheets, such as (Raffensperger, 2003) and (FAST Standard Organisation, 2016), recommend shorter formulas with fewer references.

Researchers have studied the complexity of spreadsheets because of its impact on the risk of having errors. (Bregar, 2004) developed a complexity metric that takes numerous factors into account, such as the number of operators and operands, the nesting, and the dispersion of references. Combining approaches from Software Engineering and Linguistics, (Reschenhofer, Waltl, & Matthes, 2016) have also developed a metric based the analysis of a spreadsheet’s formulas. Using the concept of smell in Software Engineering, (Hermans, Pinzger, & van Deursen, 2014) measured complexity with the number of operators and of references, the length of calculation chains and the presence of duplicated formulas.

What all those research projects and expert recommendations fail to do is take into account the complexity of the problem itself, and that cannot be done with only an analysis of the formulas in a spreadsheet file.

Even though a spreadsheet has two dimensions, rows and columns, one of the spreadsheet's dimension is usually used to represent variables. This leaves the other spreadsheet’s dimension to represent one of the model’s dimension, such as time. But real-world problems can be more complex. For example, we may want to model unit sales by month, by region and by product, thus needing a three-dimensional variable. The challenge's problem had variables in up to four dimensions: month, sector, product and region.

In this paper, we will describe how some Excel users implement a multidimensional structure. The structure is the one described in (Mireault, 2018).

We recruited the participants in the study through an Internet challenge. The challenge kit consisted of the problem description and its solution, presented in the form of a Formula Diagram and a Formula List. The kit also contained an Excel file with the data and the desired Interface sheet and it also contained two screen captures showing the desired results given two different inputs. The participants could then validate themselves to make sure that their calculations were correct.

This research is an explorative study to see what techniques Excel developers use to implement multidimensional variables. Since participation was voluntary, we cannot do any statistical inference with the results.
2 Recruiting the participants

An invitation to participate in the challenge was sent to the EuSpRIG mailing list (EuSpRIG, 2019) and on the LinkedIn Excel Developers group (LinkedIn, 2019), with encouragements to redistribute the invitation in other circles. The challenge lasted one month from November 1 to November 31, 2018. The participants had to request the kit by email, which allowed us to measure the interest level. There were 109 kit requests and 17 of them submitted a spreadsheet. We did not investigate why potential participants did not complete the challenge, but some did communicate that they were too busy or that they found the problem too complicated and did not know how to implement it.

The participants were given a problem along with its solution: all the variables and the formulas were presented in the Formula List. Their task was not to solve the problem, but to implement its given solution. The participants were free to use any feature that did not require non-standard features: it had to work in a simple Excel installation.

The Formula List contains 5 variables with more than two dimensions. MSP Unit Sales and MSP Sales Amount are in the (Month, Sector, Product) dimension set, MSPR Unit Sales and MSPR Variable Cost are in the (Month, Sector, Product, Region) dimension set, and MPR Unit Sales is in the (Month, Product, Region) dimension set. Those are the variables that interested us in this analysis.

3 Initial analysis

Given the voluntary lack of directives, some participants did not implement all the variables of the Formula List. They avoided implementing some multidimensional variables by building more complex formulas. For example, the two four-dimensional variables, MSPR Unit Sales and MSPR Variable Cost can be avoided by changing the given formulas for the variables that depend on them: MPR Unit Sales, Monthly Unit Sales and Monthly Variable Cost. So, the formulas

\[
\text{MPR Unit Sales} = \text{SUM( MSP Unit Sales } \times \text{ Region Sales Distribution per Sector })
\]

and

\[
\text{MPR Unit Sales} = \text{SUM( MSP Unit Sales } \times \text{ Region Sales Distribution per Sector })
\]

were combined by some participants to calculate

\[
\text{MPR Unit Sales} = \text{SUM( MSP Unit Sales } \times \text{ Region Sales Distribution per Sector })
\]

This was unfortunate for this study, as we were interested in seeing how the variable MSPR Unit Sales was implemented in their worksheet. But since the four-dimensional variables were not presented as output variables in the Interface sheet their implementation was mathematically correct.

4 Implementation strategies

The implementation strategies used by the participants can be assigned to two broad categories that we'll call the Database approach (DB) and the Variables and Formulas approach (VF). The VF approach is well used in one or two-dimensional spreadsheets in business and finance applications.

In a one-dimensional spreadsheet, the variables, like Unit Sales, Cost and Revenues, are usually represented in rows, with columns on the left-side documenting them with their name and, sometimes, their units. One dimension, like Month or Region, is presented in a row at the top and columns represent the instances of the dimension, such as Jan, Feb, Mar or South, North. The formulas appear in a rectangular shaped area below the dimension row and to the right of the variables. Normally, the formulas linking the variables are written for one instance of the dimension and then copied under the other instances to the right.

Implementing two dimensions starts to be a bit more involved. One method consists of presenting the second dimension along with the first one. All the instances of the first dimension are then repeated for each instance of the second dimension. Thus, if we have twelve months and three regions, the set of twelve months (Jan, Feb… Dec) will be repeated for each of the three regions, giving 36 columns.
Another method consists of subsuming a dimension with variables. Thus, instead of having one variable Sales with 36 instances—Sales(North, Jan), to Sales(East, Dec)—we create three variables named Sales North, Sales South and Sales East. And there remains only one dimension, Month, with 12 instances. Usually, calculations with variables with the same suffix are grouped together. Thus the calculations of Sales North, Cost North and Profit North would appear in the same area and then would be reproduced and adjusted for Sales East, Cost East and Profit East.

### 4.1 Three-dimensional variables

We had two 3-dimensional variables, MSP Unit Sales and MSP Sales Amount in the (Month, Sector, Product) dimension set, and MPR Unit Sales is in the (Month, Product, Region) dimension set. This variable was also shown as an output variable in the Interface sheet given to the participants, as shown in Figure 1.

<table>
<thead>
<tr>
<th></th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unit Sales</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Product</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Regions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>181</td>
<td>182</td>
<td>194</td>
<td>199</td>
<td>211</td>
<td>189</td>
<td>171</td>
<td>148</td>
<td>131</td>
<td>122</td>
<td>128</td>
<td>143</td>
</tr>
<tr>
<td>S</td>
<td>118</td>
<td>125</td>
<td>128</td>
<td>128</td>
<td>132</td>
<td>118</td>
<td>109</td>
<td>97</td>
<td>96</td>
<td>87</td>
<td>83</td>
<td>103</td>
</tr>
<tr>
<td>W</td>
<td>145</td>
<td>169</td>
<td>182</td>
<td>190</td>
<td>208</td>
<td>194</td>
<td>177</td>
<td>151</td>
<td>130</td>
<td>119</td>
<td>118</td>
<td>128</td>
</tr>
<tr>
<td>E</td>
<td>128</td>
<td>141</td>
<td>145</td>
<td>143</td>
<td>147</td>
<td>130</td>
<td>120</td>
<td>108</td>
<td>101</td>
<td>95</td>
<td>96</td>
<td>118</td>
</tr>
<tr>
<td><strong>Product and Region</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Product</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Regions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>184</td>
<td>171</td>
<td>173</td>
<td>177</td>
<td>180</td>
<td>149</td>
<td>135</td>
<td>120</td>
<td>118</td>
<td>108</td>
<td>127</td>
<td>145</td>
</tr>
<tr>
<td>S</td>
<td>107</td>
<td>108</td>
<td>108</td>
<td>96</td>
<td>98</td>
<td>72</td>
<td>72</td>
<td>70</td>
<td>75</td>
<td>76</td>
<td>90</td>
<td>101</td>
</tr>
<tr>
<td>W</td>
<td>109</td>
<td>115</td>
<td>114</td>
<td>112</td>
<td>114</td>
<td>97</td>
<td>90</td>
<td>82</td>
<td>82</td>
<td>78</td>
<td>89</td>
<td>100</td>
</tr>
<tr>
<td>E</td>
<td>111</td>
<td>110</td>
<td>102</td>
<td>89</td>
<td>81</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>77</td>
<td>82</td>
<td>97</td>
<td>108</td>
</tr>
</tbody>
</table>

*Figure 1 MPR Unit Sales in the Interface sheet*

Not surprisingly, all but one participant adopted that representation in their calculation sheet. Since participants had different ways of showing a particular structure (with different formats, spacing, etc.), we will represent this structure with a diagram as shown in Figure 2. The diagram shows that the Month dimension is in columns and the Product and Region dimensions are in row. Furthermore, the Products are in sequence and the Regions are repeated for each instance of the Products. The body of the structure is the calculation of the MPR Unit Sales variable.

*Figure 2 Diagram of Structure MPR1*

One participant chose to calculate the variable with a different structure. In this case, all the dimensions were represented in rows, as shown in Figure 3. Our representation of that structure is shown in Figure 4.
With the MSP structure, the participants had no suggested representation and they had a variety of structures. We identified six different structures, shown in the following figures.
Structures MSP1 and MSP3 are similar, and so are MSP2 and MSP4. We can also note the similarity of structures MPR2 and MSP6: they both perform all the calculations in a single column.

### 4.2 Four-dimensional variables

Seven participants avoided calculating the two four-dimensional variables, MSPR Unit Sales and MSPR Variable Cost. The 10 other participants had 6 different structures, shown in the following figures. Structures MSPR1 and MSPR4 were each used by 3 participants, all the others were used by a single participant.
4.3 Database approach

Some participants used a strategy we call the Database approach. This approach consists of using structure MSPR4 and calculating all the problem variables in each row of the table. The columns M, S, R and P collectively form the primary key of the table and, separately, serve as foreign keys to form relations with the other dimension sets. The resulting table was then used with Excel database tools like Power Query, Pivot Tables or SUMIF functions to calculate all the aggregate variables required in the Interface sheet.

Using Power Pivot and Pivot Tables requires a manual recalculation after changing the Base Price input variable, and can be inconvenient for users who want to do scenario analysis with Excel’s What-If Tables.

5 Discussion

As noted above, some participants avoided creating 3 or 4 dimensional variables that were not required in the Interface sheet, even though they were in the supplied Formula List. As a result, their formulas tended to be much more complex than those using the multi-dimensional variables. For example, the formula to calculate the MPR Unit Sales using MSPR Unit Sales is described in the Formula List as:

\[
\text{MPR Unit Sales} = \text{SUM(MSPR Unit Sales)}
\]

It is implemented as

- \(=\text{SUM(J51:J54)}\)

and

- \(=\text{SUMIFS(tblProcess[MSPR Unit Sales], tblProcess[Product Name], $E$4, tblProcess[Region Name], $G3, tblProcess[Month Name], $H$1)}\)

and

- \(=\text{SUMIFS(Sales[UnitSales], Sales[Region], $F3, Sales[Month], G$1, Sales[TypeAbbrev], $D$4)}\)

by participants who used a MSPR structure,

and as

- \(=\text{SUMPRODUCT}((G$31:G$34)*(W$31:W$34)*(Y$31:AC$34)*($Y$30:AC$30=IF3))\)

and

\(=\text{SUMPRODUCT(Data!F$29:F$32*Data!B$11:B$14*TRANSPOSE(Data!J$5:M5) * TRANSPOSE(OFFSET(Data!J$14:M$14,COLUMNS($G$1:G$1),)))}\)
and

\[
\text{SUMPRODUCT(OFFSET(Data!$J$5,MATCH($F3,Data!$I$5:$I$9,0)-1,0,1,4),}
\text{OFFSET(Data!$T$15,MATCH(G$1,Data!$I$15:$I$26,0)-1,0,1,4))}
\]

by participants who did not use a MSPR structure.

Some participants also mixed variables of different dimension sets in the same calculation areas. This is often done to calculate aggregate values and presents a visually pleasing table view, as shown in Figure 17 where the body of the table calculates variable MSP Unit Sales and the column labelled Total calculates MS Unit Sales. This organization makes it hard to copy formulas from left to right, and is a potential error risk during future maintenance.

<table>
<thead>
<tr>
<th>Month</th>
<th>Government</th>
<th>Military</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
<td>Deluxe</td>
</tr>
<tr>
<td>Jan</td>
<td>133</td>
<td>72</td>
</tr>
<tr>
<td>Feb</td>
<td>148</td>
<td>80</td>
</tr>
<tr>
<td>Mar</td>
<td>177</td>
<td>96</td>
</tr>
</tbody>
</table>

*Figure 17 Mixing variables of different dimension sets*

Two participants used worksheets to represent one or two dimensions. One used *bookends* sheets to simplify the use of the SUM function (see Figure 18 and structure MSPR6).

*Figure 18 Modeling dimensions with sheets*

The formula to calculate MPR Unit Sales is simply \(\text{SUM(Ga:Gz!G3)}\). The bookend sheets (Ga, Gz, Ma, Mz, etc.) are empty, and one would only need to add a sheet between them to have it used in the SUM. Thus, adding a Sector-Region sheet in the proper place would not require any change in the aggregate formulas.

Participants using the Database approach usually have the simplest formulas for aggregated variables. We expect that they will be the easiest to maintain.

6 Conclusion

The analysis of the Multi-dimensional Spreadsheet Challenge submissions has given us insight on the different implementation strategies used by seasoned spreadsheet developers. Given the same problem and its solution, developers had very different results. This lack of standardization may explain why it is hard to understand and maintain somebody else’s spreadsheet.

This study should open the door for more research on complexity and on error reduction. All previous research on complexity has been done on the analysis of existing spreadsheets without examining their original requirements. By ignoring the nature of the problem, they implicitly assume that either the nature of problem itself is not important, or that all problems are equivalent.

An interesting experiment would be to submit the spreadsheets to different developers, along with some requirement changes, and examine the process they follow to understand it and modify it to satisfy the new requirements. We could then infer if a multi-dimensional structure leads to a better understanding and reduces maintenance errors.

Since the solution was supplied in this research, another future research project could just provide a problem statement to participants and examine their resulting spreadsheet with its documentation. This would more closely represent a real-world situation. We expect this to be hard to set up such an experiment outside of an academic setting.
Appendix A – Multidimensional Spreadsheet Challenge Kit

This section presents the material that was sent to the participants of the Multidimensional Spreadsheet Challenge in November 2018. It consists of the problem statement, the Formula Diagram, the Formula List and two screen captures showing the result for two different inputs.

Acme Techno Widgets Company

The Acme Techno Widgets Company (ATW) produces and sells widgets. Its salesforce is assigned to four major sectors: Government, military, education and private. It produces two products, the Standard widget and the Deluxe widget.

Market research has established that the annual demand for widgets depends on each sector’s Standard widget price. The Pricing Director explains:

We start by setting a global base price. Then, for each sector, we tell our salesforce that they can offer a rebate. For instance, we offer a 70% rebate to the education sector and it’s 10% for the private sector because purchases are usually made by researchers with limited funds. The military sector gets a 20% rebate and the government 40%. This is not made public: all our price lists show the base price, but our clients in each sector are aware of the rebate they can get.

Each sector reacts differently to a change of price. We consulted with a market research expert and she came up with multiple demand functions, one for each sector. The demand function estimates a sector’s annual demand for a given base price. The demand function has the form \( B/\text{Price}^{A} \). The parameters \( A \) and \( B \) are different for each sector, and \( \text{Price} \) is the sector’s price, after the rebate. This table shows the values the expert gave us:

<table>
<thead>
<tr>
<th>Sector</th>
<th>Government</th>
<th>Military</th>
<th>Private Sector</th>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebate Percentage</td>
<td>40%</td>
<td>20%</td>
<td>10%</td>
<td>70%</td>
</tr>
<tr>
<td>DemParA</td>
<td>3.59</td>
<td>3.46</td>
<td>3.18</td>
<td>4.11</td>
</tr>
<tr>
<td>DemParB</td>
<td>22000000000</td>
<td>22000000000</td>
<td>22000000000</td>
<td>22000000000</td>
</tr>
</tbody>
</table>

The price of the Deluxe widget is 45% higher than the Standard widget.

The Sales Manager explains the sales pattern:

The annual demand of each Sector is split between the Standard and Deluxe products, but the distribution is very different in each sector. For instance, in the education sector, with its limited funds, the split is 80%-20% and it is 25%-75% in the military sector. I guess these guys always go for the best, and they have higher budgets. The distribution is 65%-35% for the government sector and 40%-60% for the private sector. The ratios are then applied to the sector’s annual demand to get the annual demand by product.

Another interesting pattern is the distribution of sales during the year. We noticed that our clients buy more just before the end of their fiscal year, when some want to spend their budget surpluses, and the beginning, when others have new funds allotted. Each sector has a different pattern, and we noticed that it is pretty stable year after year.

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>9%</td>
<td>10%</td>
<td>12%</td>
<td>12%</td>
<td>11%</td>
<td>9%</td>
<td>7%</td>
<td>6%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>6%</td>
</tr>
<tr>
<td>8%</td>
<td>9%</td>
<td>10%</td>
<td>12%</td>
<td>13%</td>
<td>11%</td>
<td>9%</td>
<td>7%</td>
<td>6%</td>
<td>5%</td>
<td>5%</td>
<td>6%</td>
</tr>
</tbody>
</table>

The Acme Techno Widgets Company (ATW) produces and sells widgets. Its salesforce is assigned to four major sectors: Government, military, education and private. It produces two products, the Standard widget and the Deluxe widget.

Market research has established that the annual demand for widgets depends on each sector’s Standard widget price. The Pricing Director explains:

We start by setting a global base price. Then, for each sector, we tell our salesforce that they can offer a rebate. For instance, we offer a 70% rebate to the education sector and it’s 10% for the private sector because purchases are usually made by researchers with limited funds. The military sector gets a 20% rebate and the government 40%. This is not made public: all our price lists show the base price, but our clients in each sector are aware of the rebate they can get.

Each sector reacts differently to a change of price. We consulted with a market research expert and she came up with multiple demand functions, one for each sector. The demand function estimates a sector’s annual demand for a given base price. The demand function has the form \( B/\text{Price}^{A} \). The parameters \( A \) and \( B \) are different for each sector, and \( \text{Price} \) is the sector’s price, after the rebate. This table shows the values the expert gave us:

<table>
<thead>
<tr>
<th>Sector</th>
<th>Government</th>
<th>Military</th>
<th>Private Sector</th>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rebate Percentage</td>
<td>40%</td>
<td>20%</td>
<td>10%</td>
<td>70%</td>
</tr>
<tr>
<td>DemParA</td>
<td>3.59</td>
<td>3.46</td>
<td>3.18</td>
<td>4.11</td>
</tr>
<tr>
<td>DemParB</td>
<td>22000000000</td>
<td>22000000000</td>
<td>22000000000</td>
<td>22000000000</td>
</tr>
</tbody>
</table>

The price of the Deluxe widget is 45% higher than the Standard widget.

The Sales Manager explains the sales pattern:

The annual demand of each Sector is split between the Standard and Deluxe products, but the distribution is very different in each sector. For instance, in the education sector, with its limited funds, the split is 80%-20% and it is 25%-75% in the military sector. I guess these guys always go for the best, and they have higher budgets. The distribution is 65%-35% for the government sector and 40%-60% for the private sector. The ratios are then applied to the sector’s annual demand to get the annual demand by product.

Another interesting pattern is the distribution of sales during the year. We noticed that our clients buy more just before the end of their fiscal year, when some want to spend their budget surpluses, and the beginning, when others have new funds allotted. Each sector has a different pattern, and we noticed that it is pretty stable year after year.

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>9%</td>
<td>10%</td>
<td>12%</td>
<td>12%</td>
<td>11%</td>
<td>9%</td>
<td>7%</td>
<td>6%</td>
<td>5%</td>
<td>4%</td>
<td>5%</td>
<td>6%</td>
</tr>
<tr>
<td>8%</td>
<td>9%</td>
<td>10%</td>
<td>12%</td>
<td>13%</td>
<td>11%</td>
<td>9%</td>
<td>7%</td>
<td>6%</td>
<td>5%</td>
<td>5%</td>
<td>6%</td>
</tr>
</tbody>
</table>
Sales to a sector are not uniformly distributed by region. For example, there are more universities in the South-West than in the West. The following table shows the distribution of a sector’s sales by region. With it, we can calculate the expected monthly sales per product per region, which helps our Logistics Department do its planning.

<table>
<thead>
<tr>
<th>Region</th>
<th>Government</th>
<th>Military</th>
<th>Private Sector</th>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>25%</td>
<td>52%</td>
<td>22%</td>
<td>24%</td>
</tr>
<tr>
<td>SE</td>
<td>18%</td>
<td>13%</td>
<td>21%</td>
<td>15%</td>
</tr>
<tr>
<td>SW</td>
<td>18%</td>
<td>18%</td>
<td>17%</td>
<td>32%</td>
</tr>
<tr>
<td>E</td>
<td>22%</td>
<td>0%</td>
<td>25%</td>
<td>17%</td>
</tr>
<tr>
<td>W</td>
<td>17%</td>
<td>17%</td>
<td>15%</td>
<td>12%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

The costs of producing a widget are $48 and $72 for the Standard and the Deluxe widget respectively. The monthly fixed costs for this year are $20000. Delivery costs depend solely on the region and are shown in this table:

<table>
<thead>
<tr>
<th>Region</th>
<th>North</th>
<th>South-East</th>
<th>South-West</th>
<th>East</th>
<th>West</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Delivery Cost</td>
<td>$10.25</td>
<td>$9.73</td>
<td>$9.58</td>
<td>$8.26</td>
<td>$11.02</td>
</tr>
</tbody>
</table>

The company CEO wants to see the following results:

- The monthly unit sales per product per region.
- The monthly sales amount and unit sales per product.
- The monthly unit sales and profit.
- The total profit.

Acme Techno Widget Company Formula Diagram
<table>
<thead>
<tr>
<th>Var No</th>
<th>Variable</th>
<th>Type</th>
<th>Dimension Set</th>
<th>Value / Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Base Price</td>
<td>Input</td>
<td></td>
<td>$100</td>
</tr>
<tr>
<td>2</td>
<td>Base Price Multiplier</td>
<td>Data</td>
<td>Product</td>
<td>(1, 1.45)</td>
</tr>
<tr>
<td>3</td>
<td>Unit Production Cost</td>
<td>Data</td>
<td>Product</td>
<td>list of values</td>
</tr>
<tr>
<td>4</td>
<td>Rebate Percentage</td>
<td>Data</td>
<td>Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>5</td>
<td>Sector Price Factor</td>
<td>Calculated</td>
<td>Sector</td>
<td>1-Rebate Percentage</td>
</tr>
<tr>
<td>6</td>
<td>Sector Base Price</td>
<td>Calculated</td>
<td>Sector</td>
<td>Base Price * Sector Price Factor</td>
</tr>
<tr>
<td>7</td>
<td>DemParA</td>
<td>Data</td>
<td>Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>8</td>
<td>DemParB</td>
<td>Data</td>
<td>Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>9</td>
<td>Sector Annual Demand Units</td>
<td>Calculated</td>
<td>Sector</td>
<td>DemParA*DemParB^-Sector Base Price</td>
</tr>
<tr>
<td>10</td>
<td>Unit Delivery Cost</td>
<td>Data</td>
<td>Region</td>
<td>list of values</td>
</tr>
<tr>
<td>11</td>
<td>PR Unit Cost</td>
<td>Calculated</td>
<td>Product-Region</td>
<td>Unit Production Cost + Unit Delivery Cost</td>
</tr>
<tr>
<td>12</td>
<td>Product Distribution per Sector</td>
<td>Data</td>
<td>Sector-Product</td>
<td>list of values</td>
</tr>
<tr>
<td>13</td>
<td>Annual Sector-Product Unit Sales</td>
<td>Calculated</td>
<td>Sector-Product</td>
<td>Sector Annual Demand Units * Product Distribution per Sector</td>
</tr>
<tr>
<td>14</td>
<td>Price</td>
<td>Calculated</td>
<td>Sector-Product</td>
<td>Sector Base Price * Base Price Multiplier</td>
</tr>
<tr>
<td>15</td>
<td>Annual Sector-Product Sales Amount</td>
<td>Calculated</td>
<td>Sector-Product</td>
<td>Annual Sector-Product Unit Sales * Price</td>
</tr>
<tr>
<td>16</td>
<td>Region Sales Distribution per Sector</td>
<td>Data</td>
<td>Sector-Region</td>
<td>list of values</td>
</tr>
<tr>
<td>17</td>
<td>Monthly Sales Distribution per Sector</td>
<td>Data</td>
<td>Month-Sector</td>
<td>list of values</td>
</tr>
<tr>
<td>18</td>
<td>MSP Unit Sales</td>
<td>Calculated</td>
<td>Month-Sector-Product</td>
<td>Annual Sector-Product Unit Sales * Monthly Sales Distribution per Sector</td>
</tr>
<tr>
<td>19</td>
<td>MSP Sales Amount</td>
<td>Calculated</td>
<td>Month-Sector-Product</td>
<td>Annual Sector-Product Sales Amount * Monthly Sales Distribution per Sector</td>
</tr>
<tr>
<td>20</td>
<td>MSPR Unit Sales</td>
<td>Calculated</td>
<td>Month-Sector-Product-Region</td>
<td>MSP Unit Sales * Region Sales Distribution per Sector</td>
</tr>
<tr>
<td>21</td>
<td>MSPR Variable Cost</td>
<td>Calculated</td>
<td>Month-Sector-Product-Region</td>
<td>MSPR Unit Sales * PR Unit Cost</td>
</tr>
<tr>
<td>22</td>
<td>Monthly Variable Cost</td>
<td>Calculated</td>
<td>Month</td>
<td>SUM(MSPR Variable Cost)</td>
</tr>
<tr>
<td>23</td>
<td>Monthly Unit Sales</td>
<td>Output</td>
<td>Month</td>
<td>SUM(MSPR Unit Sales)</td>
</tr>
<tr>
<td>24</td>
<td>Monthly Sales Amount</td>
<td>Calculated</td>
<td>Month</td>
<td>SUM(MSP Sales Amount)</td>
</tr>
<tr>
<td>25</td>
<td>Monthly Fixed Cost</td>
<td>Data</td>
<td></td>
<td>$20000</td>
</tr>
<tr>
<td>26</td>
<td>Monthly Costs</td>
<td>Calculated</td>
<td>Month</td>
<td>Monthly Fixed Cost + Monthly Variable Cost</td>
</tr>
<tr>
<td>27</td>
<td>Monthly Profit</td>
<td>Calculated</td>
<td>Month</td>
<td>Monthly Sales Amount - Monthly Costs</td>
</tr>
<tr>
<td>28</td>
<td>MPR Unit Sales</td>
<td>Output</td>
<td>Month-Product-Region</td>
<td>SUM(MSPR Unit Sales)</td>
</tr>
<tr>
<td>29</td>
<td>MP Unit Sales</td>
<td>Output</td>
<td>Month-Product</td>
<td>SUM(MSP Unit Sales)</td>
</tr>
<tr>
<td>30</td>
<td>MP Sales Amount</td>
<td>Output</td>
<td>Month-Product</td>
<td>SUM(MSP Sales Amount)</td>
</tr>
<tr>
<td>31</td>
<td>Total Profit</td>
<td>Output</td>
<td></td>
<td>SUM(Monthly Profit)</td>
</tr>
</tbody>
</table>
References


LinkedIn. (2019, 04 01). Excel Developers. Retrieved from LinkedIn: https://www.linkedin.com/groups/58704/


ABSTRACT

Visualisation is often presented as a means of simplifying information and helping people understand complex data. In this paper we describe the design development and evaluation of an interactive visualisation for spreadsheet formula (EQUS). The work is justified on the grounds that these are widely used tools for significant numerical processing and modelling, yet the formula developed can be easily misunderstood. The development process was one of iterative refinement engaging an initial target audience of mid-teen learners, involving re-design and formative evaluation. The resulting visualisation techniques have been found to be broadly relevant to spreadsheet users beyond the initial target audience. EQUS has since been developed as fully integrated plug-in for MS Excel.

1 INTRODUCTION

Visualisation is often presented as a means of simplifying information and helping people understand complex data. In this paper we describe the design and development of interactive visualisation designed to help understand spreadsheets. Our premise is that, spreadsheets are a traditional, common and accessible ICT tool that is often used to perform any number of numeric activities. Widely used in work and education (Chambers et al. (2012)), at school level and in higher education, the spreadsheet is a core generic tool for understanding in many numerate subjects. Their responsiveness means users quickly become embedded in ‘solutions’ in an information infrastructure that is intrinsically easy to mismanage.

Despite the difficulties that arise from embracing a spreadsheet infrastructure, the scale of their user population and widespread familiarity make them a legitimate area of study. Despite the spreadsheet being a familiar tool for general purpose computation, with significant longevity, it is widely established that they are difficult to manage (Panko and Sprague (1998); Hendry and Green (1994)). Disappointingly, the research behind the work reported occasionally encountered educational contexts in which tutors themselves did little to encourage effective spreadsheet use. It is interesting to note, that skills based training with spreadsheets involves very little modelling and numerical work, but focuses more on duplicating prescribed models and ensuring that presentation and formatting is focused upon.

Research into addressing issues of spreadsheet quality has motivated very many enhancements. This includes additional features to ensure they are more transparent as well as to encourage more discipline in their use. For some examples, see: Burnett et al. (2001, 2002); Hendry and Green (1994); Hermans and Dig (2014); Panko and Sprague (1998); Sajaniemi (2000); Ayalew (2009). Specifically, with regard to the complexity of inter-cell referencing, the understanding of formulae has been found to be particularly demanding, with evidence that business and governmental spreadsheets tend to avoid the use of many functions and function nesting (as with Sajaniemi (2000)).

1.1 Motivation

The starting point of the work was a recognition that initial numerical modelling with spreadsheets benefits from working with easily manipulated and refined formulae. Once a model formula is stable and not being experimented with, it is possible that the formula will be embodied in surrogate
structures, such as sub-computations in separate cells. However, to maintain flexibility when exploring solutions there is a benefit to not having to commit to surrogate structures.

Previous work has proposed ways of presenting and visualising spreadsheets, see: Saariluoma and Sajaniemi (1994); Igarashi et al. (1998); Ballinger et al. (2003); Burnett et al. (2001). However, these works on the whole consider the wider structure of spreadsheets, and the dependencies between cells. None appear to have addressed the fact that the formula’s language is computationally powerful but contracted onto a single line. It is this complexity of language presentation that can impair its effective use, especially when experimenting with formulae.

This observation motivated consideration of spreadsheet formulae as, in effect, small programs expressed in an inappropriate environment. One potential improvement would be to have a clearer visual language that graphically represents spreadsheet formulae. To this end we took a lead from work on program comprehension and visual programming. We chose to develop a visualisation to take advantage of human perceptual ability to recognise patterns and associations - and support “visual thinking”. Graphical representations, such as flowcharts and pictorial representations of data structures have long been used to support the understanding of programs and their underlying processes (Myers 1986). However, it is of interest to note that in visual computational language, empirical evidence of their compelling and appealing character is limited (Sorva et al. (2013).

Motivating Potential Users Initially, tutors and learners were targeted as end users of our tool, with a focus upon numerate subjects and disciplines. With numeracy at the core of most effective spreadsheet use, we motivated end user engagement initially with the question:

What does 2 + 3 × 4 equal?

This was effective as a means of engaging the user community, since it was sufficiently simple to allow an open discussion of why it is useful to understand formula structure. If the user has no expectation of the result, they’ll accept 14 (or 20) without worrying about what they meant by the formula. By contrast, if the user has an expectation of the result and output does not match it, they’ll need support in understanding whether their formula or their expectation is wrong.

Although this case is trivial the same arguments are just as applicable for more complex formulae. Reflecting this point, our focus was to treat the visualisation as a re-presentation of a formula that would at least help disabuse users of misconceptions about the formula.

2 DESIGN AND DEVELOPMENT

In this section we outline our approach to design and development of EQUS. This was an iterative process informed by various factors: knowledge of visual programming research, professional graphic design and user feedback. The overall structure and phases of development were largely governed by features of technical development.

2.1 Designing a visual language

The visualisations were initially developed on paper to allow tutors and learners to explore and provide rapid feedback on which visual characteristics are appropriate and of value. Initially good visual design practice was followed, informed by learning scenarios and educational uses of spreadsheets (e.g. see: Gretton and Challis (2008)). The principles for the initial design where:

- Evidencing structure. Within a given formula, the syntactic structure is core to comprehending meaning.
• Visual mapping. The ease of mapping between the formula and visualisation. Clearly, if this mapping is complex for a learner, the visualisation may be of little value.
• Evidencing categories. Within a given formula, being able to recognise the different categories of tokens and structures.
• Evidencing abstractions. There are various abstractions apparent in the way formulae are used. For example, the same sub-expression appearing in a number of places in a single formula. A simple example would be the formula for a quadratic, such as, \( =A1^2 + B1^2 + C1 \). The repeated use of \( X1 \) is important for understanding what is expressed.
• Evidencing computation. In contrast to abstractions, there is the value of evidencing the specific values used in determining the resulting value of a formula.
• Visual simplicity and scalability. Although not easily defined, this principle discourages apparently empty space, redundant arcs or overlapping lines or structures. In view of our motivation, this point is most relevant for complex formulae.

Clearly these principles can guide design decisions, though they can conflict with each other. As such they provide a basis for interpreting design changes and user feedback.

2.2 Interaction Design

In addition to providing visualisations, the manner in which users should interact and work with them was explored. In the context of this work, the objective was to enhance the use of spreadsheets and not undermine existing users skills. Within the applied educational context, departing from established materials and tutor knowledge was considered inappropriate. Hence, the interaction design was developed with aim of adding value without undermining existing skills. This perspective informed how the interaction between the tool and the spreadsheet should operate. The design thus focused upon the tool to re-present formulae responsively with the following rationale:

• The visualisation for the formula matches the formula in the current cell. If no cell is selected or it is empty, the visualisation is blank.
• The visualisation should be interactively integrated, in the sense of following the conventions of other supplementary interactive tools in spreadsheets.

These rationale in principle were to minimised the operational cognitive for spreadsheets users when working with the new visualisation.

2.3 Development

As described above, the design and development followed a user centred design approach, with frequent formative evaluation by end users. This progressed with both a professional visual designer engaged as well as technical developers for prototype ideas. Throughout the process each engagement with end users generated issues and questions about EQUS. These were assessed and potential tool enhancements were identified. Enhancements were judged based on relevance, taking into account prior research, and also in terms of technical feasibility.

Paper prototyping

To help explore the wide range of potential visualisations and help ensure users engaged in early design decisions, paper prototyping of visualisations was initially employed. This was useful in also establishing existing skill and knowledge levels of potential users. All the designs tended to use a flow visualisation aiming to convey data, dependencies and computational information. Key design decisions focused around whether to visualise structure purely in terms of operators alone (taking up less visual space) or to visualise the value computed by each intermediate operator. Despite the latter being more visually complex, it was preferred by users and could also be mapped to the principle of evidencing computation (Leitao and Roast, 2014).
Operational prototypes

Once key visual design features had been agree upon, prototypes for dynamically generating visualisation were built. The first of these were developed in python and loosely integrated with MS Excel. This initial proof of concept helped tackle some of the initial technical questions of how the visualisation should operate. The second prototype required stronger integration with the spreadsheet and thus was developed in Javascript and to inter-operate with GoogleSheets. This provided a sufficiently robust prototype for real-world trailing and feedback.

Feedback as a whole was positive and compelling (see table 1), and used to justify moving towards a full product. Although GoogleSheets is growing in popularity, MS Excel was judged to be the primary target for a full product version. The tool was developed as a plug-in, integrated using Visual Studio Tools for Office (VSTO).

Product development

During product development, user feedback gathered which informed more design details (summarised in table 2). Engaging users highlighted spreadsheet usage and uses that had not initially been identified. These provoked technical and design revisions. During this phase of development additional qualitative feedback reaffirmed the potential value of elements of this design (see table 3). Despite the lack of familiarity with the visualisations, their presence and use did not impair learner performance. In follow-on interviews all agreed that the visualisation approach had merit. Overall feedback was positive, with those interviewed seeing the potential to help “de-mystify” spreadsheets for learner populations we are targeting.

The outcome of this process resulted in the EQUS tool which is described and illustrated in the following section.

![Figure 1: The formula: =2+3*4 visualised by EQUS](image)

Table 1: Summary of initial evaluation studies and outcomes

<table>
<thead>
<tr>
<th>Study</th>
<th>Population and context</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper based study with spreadsheet comprehension questions</td>
<td>44 work-based learners studying Electrical engineering, Engineering and Maths at NVQ level 3.</td>
<td>Visualisation showed a positive effect. Average score was 55% with a visualisation, and 51% without. (Not significant)</td>
</tr>
<tr>
<td>Interactive prototype based study and qualitative interviews</td>
<td>14 full-time learners were given a spreadsheet “refresher” and then completed 37 spreadsheet formula questions</td>
<td>Visualisation conditions showed a positive effect over the no visualisation condition. Average score 73% with a visualisation, and 67% without. (Not significant)</td>
</tr>
<tr>
<td>Interviews and demonstration with experts.</td>
<td>Three STEM educators, three STEM education researchers and five support staff</td>
<td>Positive feedback with specific formative feedback.</td>
</tr>
</tbody>
</table>
Table 2: Summary of follow-on evaluation studies and outcomes

<table>
<thead>
<tr>
<th>Study</th>
<th>Population and context</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tutor lead usage by learners in Science education classes: GCSE and A level. Linked with the analysis of data gathered in earlier lesson</td>
<td>Number of learners est. 75.</td>
<td>Positive tutor feedback on the value of the tool for enhancing learner familiarity with BODMAS, and the physics equations that they have been learning.</td>
</tr>
<tr>
<td>Controlled study with participants attempting a range of given spreadsheets tasks and formulae based questions, working both with and without EQUS.</td>
<td>42 higher education students (about 50 minutes of spreadsheet tasks, with and without EQUS)</td>
<td>Highly positive feedback. 70% would re-use EQUS, 91% would recommend it to others, 64% felt their understanding had improved.</td>
</tr>
</tbody>
</table>

Table 3: Selective evaluation quotes

<table>
<thead>
<tr>
<th>Source</th>
<th>Quotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>School child</td>
<td>“It would help me a lot with other formulas”</td>
</tr>
<tr>
<td>School child</td>
<td>“You can see the values and how they are worked out, that’s great.”</td>
</tr>
<tr>
<td>Secondary level Physics tutor</td>
<td>“I’d draw a diagram like this on the whiteboard, but it would take a while and I might get it wrong.”</td>
</tr>
<tr>
<td>Functional skill tutor</td>
<td>“Absolutely brilliant when it comes to more complicated formulas for our learners. With regards to the IF statement, I particularly like the way it checks the condition and identifies whether it is TRUE or FALSE. Additionally really good for formulas of non-adjacent cells.”</td>
</tr>
<tr>
<td>Tutor</td>
<td>“I am sure that it could add value to the teaching of mathematics.”</td>
</tr>
<tr>
<td>Tutor</td>
<td>“It would help anyone willing to learn about spreadsheets”</td>
</tr>
<tr>
<td>Maths education researcher</td>
<td>“It will be very useful to many students to have a product that enables a better conceptual understanding of the equation format. There is a clear need for such a tool to be suitable for the many students who do not have high levels of mathematical skills and yet use mathematical symbolism every day in their studies. This will include students from Chemistry, Business, Economics, Psychology, Geography and many more.”</td>
</tr>
<tr>
<td>Trainee STEM tutor</td>
<td>“I struggle a lot with spreadsheets and find it hard to understand them. Seeing the spreadsheet visualisation prototype made it clearer to understand the formulas and feel that if I had chance to use a programme of that kind I would have a greater understanding and be able to pick up the skills I require much quicker. I feel that this product could help people like myself that struggle with spreadsheets.”</td>
</tr>
</tbody>
</table>
Figure 2: Various examples of formulae involving cell references visualised by EQUS

3 EQUS

Here we describe the EQUS tool introducing features that proved to be useful for a range of end users, illustrating various cases.

3.1 Basic formula

As mentioned in the introduction, some of the most basic formulae have presented problems for some of our initial end users. In keeping with a data flow model and the desire to see intermediate results, the visualisation for \(2 + 3 \times 4\) is shown in figure 1. This has been found to be of most use in dynamic settings, where simply being able to quickly see what is happening, and try alternatives, has been useful. In some cases the fact that a spreadsheet is the basis for the visualisation has been of secondary interest.

Figure 3: Examples of prefix operators being visualised in EQUS

Cell References Spreadsheet utility primarily comes from the use of cell references for arbitrary values. Within the EQUS visualisation this is shown by differentiating the shape cell values used in a formula and an additional label for the cell reference. Cell referencing examples are shown in figure 2. The compositional character of mathematical expressions is intrinsic to their power. Interesting cases can easily be explored with the EQUS visualisations responding to any numerical expression entered into the spreadsheet. Some compelling illustrations of this include unary pre-fix operators - and +. While learning, such unary operators are, not surprising, confused with the same infix binary
operators, subtract (-) and add (+). Figure 3 shows some of the examples of repeated uses of unary operators that help illustrate their structural character and their function.

Building upon this, examples of negations interacting are useful, especially when working with non-associate operators. The importance of non-associative operators is that the order of operands makes a difference to the result, despite there being little to distinguish associative and non-associative operators by appearance. Figure 4 shows a variety of EQUS visualisations for formulae involving negation and non-associative binary operators.

\[
\begin{align*}
-2 & + -5 & -6 & / & 3 \\
& -7 & -5 \\
= & -2 & -5 & -6 / & 3
\end{align*}
\]

\[
\begin{align*}
-9 & / -4 & -2 & -6 \\
& 2.25 & 0.25 & 0.25 \\
= & -9 & -4 & -2 - 6
\end{align*}
\]

\[
\begin{align*}
-(-8 & -6) & / -(2 & -7) & / 3 & / 6 \\
& 14 & 14 & 1 & 0.33333 & 0.05556 \\
= & -(-8 - 6) & / -(2 & -7) & / 3 & / 6
\end{align*}
\]

Figure 4: A number of operator based formulae visualised in EQUS

### 3.2 Functions

With spreadsheet functions EQUS visually encodes them keeping to the principles of ensuring that the formula maps to the visualisation. Figure 5 illustrates a number of standard functions combined and nested. The benefit of the EQUS visualisation is that the intermediate results are evident and therefore helping users comprehend why a result is what was intended. For real number results and intermediate values, users’ expectations are likely to be estimates. However, with more discrete domains specific values are core to verifying how data is being processed. One good example of this can be found with the Boolean functions that spreadsheets support. In figure 6, the EQUS visualisation shows various Boolean expressions being computed.

### 3.3 Errors

Operator, functions and cell linking in combination give spreadsheets their power. One consequence of this is also the possibility of formulae and results being invalid and/or
generating errors. During EQUS development, erroneous formulae were largely avoided. However, it became obvious as the work progressed that errors were part and parcel of spreadsheet usage - both a cause of annoyance for users but also a formative mechanism when developing numeric skills.

\[ \text{Figure 5: Example formulae with common spreadsheet functions shown in EQUS} \]

\[ \text{Figure 6: Example Boolean formulae visualised in EQUS} \]

For standard spreadsheets two types of error should be distinguished. First, an invalided expression error arises from an input that simply is not a formula. In MS Excel invalided expressions are prevented from being entered, and a valid alternative expression is proposed. Second, an erroneous result arises from a syntactically valid formula that cannot be evaluated to produce a normal result. In terms of EQUS’s technical design as a plug-in, there is no means of accessing invalid expressions with MS Excel, and thus, in keeping with our interaction design principles, there is no visualisation treatment of them. However, although erroneous results are not “normal” results for a formula, within MS Excel such errors are treated as to the normal domain of values that functions and operations work over. To
illustrate this point, a trigonometric function, such as, \( \tan \) would be mathematically characterised as operating over the domain of real numbers (\( \mathbb{R} \)), as:

\[
\tan : \mathbb{R} \to \mathbb{R}
\]

The same function when digitally computed would be best characterised as:

\[
\tan : \mathbb{R}_D \to \mathbb{R}_D
\]

This characterisation, reflect the imperfect representation of the real numbers in digital technology using \( \mathbb{R}_D \), as opposed to \( \mathbb{R} \). However, the same function \( \tan \) in MS Excel would be characterised as operating over the same domain extended and some additional distinct error values:

\[
\tan : (\mathbb{R}_D \cup \mathbb{E}) \to (\mathbb{R}_D \cup \mathbb{E})
\]

Here, \( \mathbb{E} \) represents error-type values that extend the conventional domain of \( \tan \) — in MS Excel there are eight such defined error-type values. The same domain extension is applied to all functions and operators in MS Excel. So for functions, such as \( \sqrt{\text{t}} \) and \( \text{trunc} \) their characterisation would be:

\[
\sqrt{\text{t}} : (\mathbb{R}_D \cup \mathbb{E}) \to (\mathbb{R}_D \cup \mathbb{E})
\]

\[
\text{trunc} : (\mathbb{R}_D \cup \mathbb{E}) \times (\mathbb{R}_D \cup \mathbb{E}) \to (\mathbb{R}_D \cup \mathbb{E})
\]

And so on, for all functions and operations.

\[
\begin{align*}
1/0 & \rightarrow \text{#DIV/0!} \\
\tan(1/0) & \rightarrow \text{#DIV/0!} \\
\tan(1/0)+\sin(40/3) & \rightarrow 13.3333 \\
\end{align*}
\]

Figure 7: Three progressively complex examples of erroneous outputs visualised in EQUS

The simplest error to demonstrate this extended domain is that which arises from a division by zero. (Formally, dividing a number by zero has no sensible numeric value and thus it constitutes an erroneous result — looking like a valid expression, but with no proper result.) In the case of MS Excel, dividing by zero \( =1/0 \) results in an error “\#DIV/0!”, and thus \( =\tan(1/0) \) reflects this by also giving same error, as does \( =\tan(1/0)+\sin(40/3) \), and so on. Any expression will compute a result when an error is present, and on most occasions the result is also the same error. Feedback from potential users about erroneous results and MS Excel’s treatment of them led to the refinement of EQUS, so as to
operationally treat error-type values as any other legitimate value. However, their status as “abnormal” results was made distinctive in the resulting visualisation. The examples from above are illustrated in figure 7.

A valuable result of the visualisation is that it shows how an error propagates through a computation. Thus it is possible to see where it originates and what parts of the expression are in effect error free.

4 CONCLUSIONS

We have reported the iterative development of the EQUS visualisation tool in terms of the design and development process, and details of its operation and resulting design. Further development of EQUS related products is envisaged in specific domains, including early education when BODMAS is being introduced, and more specialist spreadsheet integration with business needs. Additional functionality in many areas can be easily envisaged, although the merit and impact of specific suggestions requires careful analysis, so as to not over complicate a tool that benefits from simplicity.

In addition, there is the opportunity to develop a more sophisticated visual language, as recommended by Moody (2009), and discussed in Roast and Uruchurtu (2016). However, using a more subtle visual language makes significant demands upon the consistency and characterisation of functions and types. If the visualisation were to discriminate between types of function, the rationale for the types would have to be carefully established and validated. One basis is to help users recognise specific function types easily, however classifying functions appropriately is more complex — by popularity, by likely domain, by type of output are all viable alternatives.

This point reflects one of a number of design opportunities that EQUS has not incorporated despite compelling contexts of use. Examples of these include: presenting formulae in a manner visually consistent with their mathematical origins, for example, showing $\sqrt{A1}$ as opposed to $\text{sqrt}(A1)$ and allowing users to edit a formula by manipulating the visualisation. As a spreadsheet augmentation such opportunities are hard to justify in terms of the design rationale since they risk specialising the tool and thus work against the objective of supporting spreadsheet users in general.

At a more academic level, experience with EQUS’s development, refinement and evaluation helps inform research into visualisation and the exploration of how best to present what are, in effect, computations to users who are not thinking computationally. Work on visualising SQL provides an early illustration of this (see Roast et al., 2018).

ACKNOWLEDGEMENTS

We are indebted to tutors, students and educational experts who were willing to support and participate in the evaluation and development of the tool. The technical development of the tool benefited significantly from support in part by: UK Higher Education Academy, The Small Business Research Initiative (part of Innovate UK) and Department for Business, Innovation & Skills (UK). VeryViz Ltd. is also acknowledged for supporting recent development activity.
References


From webtables to datatables

Mária Csernoch
University of Debrecen, Hungary, 4028 Debrecen Kassai út 26.
csernoch.maria@inf.unideb.hu

ABSTRACT

Webtables – tables and table-like structures on webpages – are excellent sources for teaching spreadsheeting, in commercial and professional organisations by utilizing and developing knowledge-transfer items, presenting and handling various real-world problems and solutions, discussing and debugging, and in general, developing and utilizing computational thinking skills. In the present paper the conversion process of one of the LOL Boards (League of Legends, Riot Games Inc. 2019) is detailed. After presenting the algorithm of the conversion, two solutions are offered – one in a word processor, the other purely in a spreadsheet application –, leaving space for discussions, inventing other solutions and combining them.

1. INTRODUCTION

Teaching spreadsheeting from the programming point of view (Sestoft, 2011) is an approach which evokes various knowledge-transfer items which have traditionally not been focused on in end-user computing. The programming aspect allows us to pay attention to data analysis, data types, n-dimensional vectors, n×m dimensional matrixes – without the burden of explicit declaration –, file and data conversions, handling n-ary and multilevel functions, algorithms, and coding in general concept-based (Pólya, 1957; Csernoch, 2017) high-mathability (Baranyi & Gilányi, 2013; Biró & Csernoch, 2015a, 2015b) computer problem-solving approaches.

To make problem- and programming-oriented spreadsheeting classes efficient and effective, teachers must present data which students are interested in, which suit their level of background knowledge and their age, and which they are eager to use for information retrieval. In short, teachers must make students motivated. However, most of the tutorials available are not like this. They are long (hundreds of pages), cover non-spreadsheet knowledge items (how to handle ribbons and toolbars, how to setup the spreadsheet environment, how to format and colour data cells and tables, how to handle operating system features, how to type data, etc.), lack any real world problem-solving, and are claimed to be suitable for everyone (Katz, 2010; Walkenbach, 2010; ECDL, 2016; ICAEW, 2016; JobTestPrep, 2016; Microsoft, 2016; Test ECDL, 2019).

In teaching-learning practice, instead of relying on ready-made tutorials, teachers must select contents which best suit their students’ interests. One option is to work with the students’ data collected in other classes in school or at work. However, these data are not necessarily available and/or do not match the students’ background knowledge. One solution is to turn to the internet and download and modify contents which match all the requirements.

The present paper details how webtables are selected and converted into datatables and how the conversion process can be used for developing students’ and participants’ computational thinking skills by applying knowledge-transfer items. In this context webtables are structures in which data are presented in tables or table-like forms on...
webpages, while datatables are $n \times m$ matrixes with $n$ data records and $m$ data-fields, and usually with an extra row for fieldnames.

2. WEBTABLE→DATATABLE CONVERSION: THE BACKGROUND

Webtable→datatable conversion requires various ICT skills which in traditional teaching approaches are presented as software-specific elements of knowledge (discussed in Csernoch, 2017; Csernoch & Biró, 2018). This approach is clearly detectable in most teaching materials (Katz, 2010; Walkenbach, 2010; ECDL, 2016; JobTestPrep, 2016; Microsoft, 2016; Test ECDL, 2019), in widely accepted exams (ECDL/ICDL, SAM), and even in the Spreadsheet Competency Framework (ICAEW, 2016).

The aim of the present paper is to demonstrate how different applications and background knowledge deriving from various sources would lead to (1) long lasting knowledge, (2) deep approach problem-solving with slow thinking, (3) reliable fast thinking problem-solving based on schemata, and (4) a realisation of algorithms and their importance in computer problem-solving, regardless of interfaces.

2.1. Selecting webtables

In the teaching-learning process, especially at the beginning, the selection of webtables and datatables and how they are presented is primarily the teacher’s responsibility (Kirschner et al., 2006). The content and the form in which tables are introduced in class greatly affects the outcome. In no particular order of importance, the following points of view must be taken into consideration when authentic data sources are presented in class.

- Using authentic tables in class would save students/participants (students for short) from the labour involved in typing (typing is not a spreadsheet competence; furthermore, it is boring, time consuming, unreliable, small-sized tables discredit the use of spreadsheets, etc.). In our approach, typing is not banished, but mostly applied in creating formulas (coding in spreadsheet interfaces), instead of typing data.

- Content plays a crucial role in motivating students. Tables which students are interested in and which contain data from which they are willing to gain information would best serve our purposes. Another option would be to handle various data collected in non-ICT classes. However, this second option, at the time of writing of the present paper, is hardly used, due to the low level of spreadsheet knowledge, both on the part of students and non-computer teachers. Consequently, computer teachers are responsible for providing data sources which are compatible with other subjects and sciences and match the students’ interest, age, occupation, etc.

- A third aspect of selecting tables is how these data sources are presented in class. The form in which the data is available for students is primarily affected by the aims of the class. From pure webpages to normalized tables, and any state in between – all are acceptable.

- The fourth aspect is the issue of timing, i.e. when these tables should be introduced in spreadsheet classes. The answer is simple: they should be introduced right at the very beginning, during the very first class. It has been found that the first class plays an important role in persuading students that they really need spreadsheeting. If we are able to make them believe that they need this knowledge, that they can handle large worksheets from the very beginning, and that they can even form questions and provide answers with formulas, we can win.
2.2. The challenges of conversions

In our experience, the webtable \(\rightarrow\) datatable conversion (WeT \(\rightarrow\) DaT) started as a ‘must’ for presenting authentic data in classes. However, in the course of events, we have found that the conversion processes might serve as a webpage semantic validator (Csénoch & Dani, 2017). Building the algorithms for the conversion and using traditional methods for handling different file formats (4.1.1 and 4.1.2 in Table 1, 4.3.2 in Table 2, 5.1.1, 5.1.2, 5.1.4, 5.1.5 in Table 5) might reveal the discrepancies of webtables: redundancy, the amount of data stored in one data-field, the structures used to build the tables or table-like formations, how consequently data are recorded and stored, the characters used in the tables (especially spaces, paragraph and line breaks, decimal and thousand separator characters), character coding, etc. We must note here that importing data from webpages with the user-friendly Data \(\rightarrow\) From Web command works properly only on well-structured and well-designed webtables, which is not often the case. The method presented in this paper would serve as an alternative solution for handling poorly designed webtables also. It might look “old-fashioned”, but its advantage to the ready-made solution is that it is program, version, and data-structure independent, furthermore highly supports knowledge-transfer activities, built upon the content of long-term memory (Sweller et al., 2011).

This deeply felt need to present interesting and processable tables in spreadsheet classes became a passion. Encountering a new webtable is extremely exciting. In hyper attention mode (Csénoch & Dani, 2017) it is impossible to tell what challenges the selected webtable holds. Is it convertible or not? If it is convertible, would we be able to do it or not? What new tricks and algorithms are required to solve problems specific to the actual table? The webtable \(\rightarrow\) datatable conversion process has turned out to be a great challenge, and we are addicted to it.

2.3. Selection of the application

As mentioned above, it is always the teacher’s responsibility to decide in which form tables are presented in class. This statement also implies that all the tables must be tried and converted beforehand. Furthermore, all conversions must be remade shortly before the class. This extra caution must be taken due to the ever-changing structure and content of webpages. Without checking the webtables in detail before the class takes place, they might surprise us and lower the efficacy of the teaching-learning process (e.g. the IMDb Top 250 table has gone through several changes over the years and various versions are available: IMDb, 2019a, 2019b).

One of the most fundamental knowledge-transfer items in end-user computing is that there is life beyond opening a datafile by double clicking, based on the datafile-application assignment. Relying on this knowledge, webpages can be opened both in spreadsheet and word processor applications. However, opening a webpage in a spreadsheet application requires more attention due to the automated, and frequently undesired, datatype recognition, which in European languages (i.e. not English) can cause serious conversion problems. In Figure 1–Figure 3 (Social Blade LLC, 2019) the comma as the thousand separator character is the source of the conversion problem (Figure 3). A test was carried out by adding 1 to the converted values (Figure 3). In the example, the Excel-converted values are stored in Columns C–E, while the results of the addition are shown in Columns F–H (\(=\text{C2:}\text{C251+1}\), \(=\text{D2:}\text{D251+1}\), and \(=\text{E2:}\text{E251+1}\), respectively) (Figure 2–Figure 3).
In Figure 4–Figure 6 (Fatty Weight Loss, 2019) the decimal-character-problem is presented. The original webpage uses a dot as the decimal character, which is a reasonable decision, given the language of the page. However, the webpage, with the left alignment format, presents these data as string, so the conversion process would tell how these data are stored.

Opening the webpage in an English version of Excel is a convenient start in the conversion process and provides correct data in a fast way (Figure 5). All the text contents are recognized with a string, while all the numbers with number data types. However, opening the webpage in a non-English Excel converts decimal real numbers to dates (Figure 6; Cells D2, F2, F3, and F314) or strings (Figure 6, Columns D and E, except Cell D2 and Cells F4, F312, F313, and F315).
When the real numbers are converted into dates, there is no way back, the original data is not available anymore, due to the conversion process operating in the background, in which dates are converted into whole numbers. Data being converted into a string datatype is a more favourable solution.

3. WEBTABLE → DATATABLE CONVERSION: LOL BOARD
In the following, the major issues involved in a conversion process are presented. For this demonstration, we selected a webtable which middle- and high-school students are extremely interested in, namely the LOL boards (League of Legends, Riot Games Inc., 2019) (a 14-old student suggested using this webtable) (Figure 7). The table can also serve didactic purposes, since it (1) contains several data fields with different datatypes, (2) redundancy can be detected, (3) pictures in different positions and with various meanings are displayed, and (4) minor lemmatization problems must also be handled.

3.1. At first sight – hyper attention mode

At this stage of the conversion a data-scanning is processed, activating hyper attention (Csernoch & Dani, 2017). The webpage opened in a browser would reveal whether the data are organized in a table-like formation or not (Figure 7). In this sense, the LOL board is quite promising. At first sight, the only distracting factor is the presence or absence of pictures in the second column, and the small icons in the third column. The table can also be suspected of redundancy, since all the records contains the words ‘votes’, ‘Comments’, and ‘Views’.

At this scanning stage, activating mixed attention (Csernoch & Dani, 2017), we can begin to consider the number of data and their data types stored in the records. The second column is not clear, since it holds several pieces of text-data: title, username, server (in parenthesis), source, and time (Figure 8). The question is how these data are stored and separated. The first column is also tricky, because, by pointing on the votes, the numbers of positive and negative votes are presented instead of the total votes (Figure 8). Apart from these distracting factors and concerns, the webtable seems promising, and convertible.

![Figure 8. The change in the way in which the number of votes is presented in the LOL Board.](image)

One further concern of the conversion process is the selection of applications. The LOL board webtable selected for the present paper, can be converted both in word processors (Section 4) and spreadsheet applications (Section 5), using the same data analysing techniques and algorithms, but requiring different background knowledge, tools, and skills to carry out the conversion.

4. KNOWLEDGE-TRANSFER ITEMS – WORD PROCESSOR (WP)

4.1. Saving and opening the webpage (WP)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1. opening and saving webpage in a browser</td>
<td>Save Page As</td>
</tr>
<tr>
<td>4.1.2. converting the webpage to a word document</td>
<td>starting MS Word→File→Open→File→Save as→File type: Word Document</td>
</tr>
<tr>
<td>4.1.3. deleting the contents outside of a table</td>
<td>selection of contents above and below table→deleting content</td>
</tr>
</tbody>
</table>

After opening the webpage in a word processor and deleting the contents above and below the table, a relatively clean text table is obtained (Figure 9). This form of the table reveals a couple of unexpected features. The votes – positive, negative, and total – are stored in
different paragraphs; however, this can easily be handled by conversion. The column with the icons, the comment category, reveals no problem in this form of presentation, since all the data-cells are filled with text, substituting the icons or their absence, and most importantly, not one cell is left empty. Furthermore, as was suspected, we are faced with redundancy; however, this can be handled rapidly and easily. Finally, disturbing Space characters – primarily data closing spaces – must be handled in order to obtain pure data.

File types (extensions): Web Page complete (html), Word Document (docx)

Knowledge-transfer items: opening a datafile in a non-assigned application (starting an application, opening a datafile, Ctrl+O, optional but fast), selecting and deleting contents, key combinations for selection (Ctrl+Shift+Home and Ctrl+Shift+End), file conversion: webpage→word document (Save As).

![Figure 9. The LOL board webtable is opened in a word processor.](image)

However, the second (with picture, the third) column is not arranged according to the data. The title is in a separate paragraph, in accordance with our expectations. The username and the server share another paragraph, but they can be separated along the parentheses. The source and the time share one further paragraph. However, this case is more complicated than it seems, since in this table there are four different ways to indicate the time: “about an hour ago”, “a day ago”, “about … hours ago”, “… days ago” (not presented in Figure 9 but handled in Formula 27 – Formula 34). Finally, for unknown reasons, without checking the page source and/or info – i.e. by applying deep attention (Csernoch & Dani, 2017) –, we are surprised by an extra paragraph for the word “by”.

4.2. Arranging/re-arranging columns (WP) – mixed attention mode

The second (with picture, the third) column seem frightening due to the presence or lack of a picture. In the case an uncertain number of columns it is worth moving these columns to the far right (Figure 10). The new order of columns is the following: (A–1) category, (B–2) new comments, (C–3) views, (D–4) votes, and (E–5 / 5½) title, for short.

Knowledge-transfer items: recognizing the number of columns, recognizing a column as an object, selecting columns, moving columns
In the current phase of the conversion it is worth knowing the number of records. To count them, we can insert an extra column and set up automated numbering. After registering the record-number we can undo these last steps. The downloaded LOL Boards consist of 966 records (Figure 11).

**Knowledge-transfer items:** records, number of records, inserting a new column, changing column-width, selecting and deleting a column, automated numbering, undo, and stack.

The final action in this phase does not change the number of columns, but it is better to handle this problem here, specifically by removing the automated bullets (automated numbering with symbols).
**Knowledge-transfer items:** selecting a column, turning off automated numbering/bullets (one of the most important knowledge-transfer items is the similarities between automated numbering and bullets)

### 4.3. Separating data (WP) – deep attention mode

In the following, without giving long explanations, we provide the algorithm, the tools, and the file types to carry out the real data conversion. Separating the algorithm from the tools brings various advantages, as is obvious in programming. The advantages of building algorithms are the following. Algorithms are interface independent, can be constructed and written in any natural language, and inscribed on any surface; tools, on the other hand, are application specific. Since one of our purposes is to support knowledge-transfer, separating the algorithms from the tools plays a crucial role. With this approach various coding solutions can be carried out at various stages of development, and in various applications and versions, without rebuilding the algorithm.

Table 2. The algorithm and the tools to build up the records of the datatable by removing varying layout solutions of the webtable, presented in Figure 12 (the conversion to a text file and back to a word document is not detailed). Results presented in Figure 13.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1. changing paragraphs to data fields</td>
<td>replace: paragraph→tabulator</td>
</tr>
<tr>
<td>4.3.2. deleting all formats and pictures</td>
<td>converting the word document to a text file and then back to a word document</td>
</tr>
<tr>
<td>changing column separators</td>
<td>Table Tools→Layout→Convert to Text→Separator: Tabs</td>
</tr>
<tr>
<td>4.3.3. deleting duplicate tabulator characters</td>
<td>replace: tabulator→tabulator</td>
</tr>
<tr>
<td>4.3.4. deleting data-closing space characters</td>
<td>replace: space→tabulator</td>
</tr>
<tr>
<td></td>
<td>replace: space→paragraph</td>
</tr>
<tr>
<td>4.3.5. changing column separators (back to table) (Figure 14)</td>
<td>Select all → Insert → Table → Convert Text to Table → 10 columns and 966 rows (recognized by Word and correct)</td>
</tr>
</tbody>
</table>

**File types (extensions):** Word Document (docx), text file (txt)

**Knowledge-transfer items:** replace, copy, escape sequences for end of paragraph and tabulator characters, saving the file with a new name, deleting formats and pictures by saving as a text file, different separator characters of tables, refreshing files (MS Word and Excel do not do this automatically), Ctrl+F4, Ctrl+O (both are optional but fast), file conversion with Save as, a range of commands

Figure 13. Data-fields are separated with the Tabulator character where data-closing Space characters are deleted.
Figure 14. 10 columns and 966 records are recognized by the word processor.

Table 3. Separating positive and negative votes.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.6. separating positive and negative votes</td>
<td>replace: space</td>
</tr>
</tbody>
</table>

The separation of positive and negative votes evokes several knowledge-transfer items. This is the first action where the number of records plays a crucial role. If the replacement is carried out in the text form, the number of replacements does not match the number of records (Figure 15), indicating that there are other instances of the searched string, apart from the votes. Consequently, this command can only be carried out when the column is selected (Figure 16).

Figure 15. The number of replacements does not equal the number of records when the replacement is carried out in the whole document.

One further concern is whether or not to continue the search in the rest of the document. By default, the “Yes” option is offered (Figure 16). In most cases, students do not read the message, they do not pay attention to it, and accept what is suggested by the “almighty” Microsoft. This is one of the knowledge-transfer items which must be taken seriously throughout computer problem-solving: no one else but the user, who is working on the problem, can decide how to continue. Regardless of the interface, students must be trained that all messages must be read, and action must be carried out according to the algorithm previously set up.
Figure 16. The number of replacements is equal to the number of records. The replacement is restricted to the selection, so further searching is not allowed.

Knowledge-transfer items: replace, escape sequence of the Tabulator character, number of replacements, changing separator tools in a table, selecting a column, deciding to search the rest of the document

4.4. Handling redundancy (WP) – deep attention mode

Redundancy can also be handled with replacements in both word processors and spreadsheets. In the present section the word-version is described.

Table 4. The algorithm and the tools to reduce redundancy in the table. The numbers in parentheses indicate the numbers of replacements carried out.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.1. deleting columns votes and by</td>
<td>selecting columns→deleting columns</td>
</tr>
<tr>
<td>4.4.2. deleting inspace</td>
<td>replace: inspace→nothing (966)</td>
</tr>
<tr>
<td>4.4.3. deleting spaceViews (Column C)</td>
<td>replace: spaceViews→nothing (966)</td>
</tr>
<tr>
<td>4.4.4. deleting spaceNew Comments (Column D)</td>
<td>replace: spaceNew Comments→nothing (802)</td>
</tr>
<tr>
<td>4.4.5. deleting spaceNew Comment (Column D)</td>
<td>replace: spaceNew Comment→nothing (164) (802 + 164 = 966)</td>
</tr>
<tr>
<td>4.4.6. deleting spaceComments (Column A)</td>
<td>replace: spaceComments→nothing (965)</td>
</tr>
<tr>
<td>4.4.7. deleting spaceComment (Column A)</td>
<td>replace: spaceComment→nothing (1) (965 + 1 = 966)</td>
</tr>
</tbody>
</table>

One further aspect of reducing redundancy and separating data with replacements is to build up schemata of Find and/or Replace by applying various contexts. With this method drilling is carried out, with a high number of repeating opportunities. Beyond offering practising environments for Find/Replace, these activities require creativity and, not infrequently, a strict execution order of commands, skills which are extremely important in computer problem-solving.

Knowledge-transfer items: replace, selecting and deleting a column, range of commands, order of commands
5. KNOWLEDGE-TRANSFER ITEMS – SPREADSHEET APPLICATION (SP)

5.1. Saving and opening the webpage (SP) – mixed attention mode

As mentioned above, building up the datatable requires the same algorithm in both applications, described above, with some minor changes.

Table 5. The algorithm for cleaning the webtable in Excel (conversion to a text file and back to a word document is not detailed). Results presented in Figure 17.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1. opening and saving the webpage in a browser</td>
<td>Save Page As</td>
</tr>
<tr>
<td>5.1.2. converting the webpage to a spreadsheet document</td>
<td>starting MS Excel→File→Open→File→Save as→File type: Excel Workbook</td>
</tr>
<tr>
<td>5.1.3. deleting contents outside of the table</td>
<td>selection of rows above and below the table, deleting rows (right click→local menu→Delete or Home→Delete→Delete Sheet Rows)</td>
</tr>
<tr>
<td>5.1.4. deleting all formats and pictures</td>
<td>converting the excel document to a text file</td>
</tr>
<tr>
<td>5.1.5. converting the text file to an excel document</td>
<td>Save as: File type: →Excel Workbook</td>
</tr>
</tbody>
</table>

The algorithm and the tools for cleaning the webtable do not differ at all in the two applications (Table 1 vs. Table 5). The only tool which should be taken into consideration is that in spreadsheets above and below the table it is not only contents, but objects (rows) which must be deleted in the first step of the cleaning process (Table 5).

File types (extensions): Web Page, complete (html), Excel Workbook (xlsx), text file (txt)

Knowledge-transfer items: saving and converting files with Save As, opening a file manually in a non-assigned application, selecting rows, deleting rows, refreshing a text file manually

5.2. Creating records (SP) – deep attention mode

After the webtable is cleaned of formats and pictures, the rearranging of data columns and creating of records must be dealt with. The table consists of Column A for the votes (positive, negative, and total), Columns B and C for the title, user, server, source and time (Columns B and C: originally without and with picture, respectively) (Figure 7), Column D for category, Column E for the number of new Comments, and Column F for the number of Views (Figure 17).

Figure 17. The cleaned webtable opened in Excel.
In the cleaned table, all the records occupy four rows (Figure 17, Worksheet LOL). To create one single record from the four rows is a modulo problem. Knowing the record number (3864/4=966; 3864=number of data-rows in Worksheet LOL) allows us to calculate the original position of the data by multiplying the new record number by 4. Within the range of a foursome the specified rows can be reached by a subtraction (Figure 18: Worksheet LOL_rec, Table 6: 5.2.2).

Similar to the solution in word processors (Section 4.3, Step 4.3.1), the next step in the conversion process is building up the records (Table 6). The output is presented in Figure 18. After generating the record numbers, a search in a vector must be completed to write out the data in the corresponding fields. Furthermore, to create the title, user_server, and the source_time columns a condition must be set up to handle the two title columns, due to the optional pictures.

Figure 18. Data arranged into records with Excel functions.

The comparison of the two solutions reveals that while the tools applied in word processors are based on general operating system knowledge-transfer items and some specifications frequently used in programming, the spreadsheet-solution requires items brought from mathematics and the ability to handle Lookup & Reference functions.

Table 6. The algorithm for creating the records in Excel. The fieldnames starting with LOL or R are in sheets LOL or LOL_rec, respectively (Figure 17 and Figure 18).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1.</td>
<td>numbering records (Rrecord)</td>
</tr>
<tr>
<td>5.2.2.</td>
<td>positive and negative votes (RVotes pn)</td>
</tr>
<tr>
<td>5.2.3.</td>
<td>total votes (RVotes)</td>
</tr>
<tr>
<td>5.2.4.</td>
<td>title (RTitle)</td>
</tr>
<tr>
<td>5.2.5.</td>
<td>user and server (Ruserserver)</td>
</tr>
<tr>
<td>5.2.6.</td>
<td>source and time (RSourcetime)</td>
</tr>
<tr>
<td>5.2.7.</td>
<td>category (RCategory)</td>
</tr>
<tr>
<td>5.2.8.</td>
<td>#new comments (Rnewcomments)</td>
</tr>
<tr>
<td>5.2.9.</td>
<td>#views (Rviews)</td>
</tr>
</tbody>
</table>

Knowledge-transfer items: counting rows and records, modulo, searching in a vector, writing out vector items, setting up a condition, output values depending on the condition, composite functions, array formulas (solutions presented in Table 6)

`=ROW(Rrecord)-1`
Formula 1: Section 5.2.1

`=INDEX(LOLvote,Rrecord*4-3)`
Formula 2: Section 5.2.2
5.3. Separating data (SP) – deep attention mode

One further aspect of building up the records is the separation of data. In the spreadsheet, the negative and positive votes are separated with functions (Table 7, Steps 5.3.1 and 5.3.2), using the same separator character as the word processors (Table 3). In a similar way, based on the same algorithm, the separation of the users and servers are carried out with composite functions (Table 7, Steps 5.3.3 and 5.3.4). The replacement process is not detailed in the word processor for the reason that the two problems require similar, but not identical, algorithms. Applying these algorithms both supports building schemata (called forth later on with fast thinking) and concept-based problem solving.

Table 7. The algorithm for creating records and data fields using Excel functions (Figure 18 and Figure 19).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.1. separating the positive votes – left to</td>
<td>Formula 4</td>
</tr>
<tr>
<td>5.3.2. separating the negative votes – right to</td>
<td>Formula 5</td>
</tr>
<tr>
<td>5.3.3. separating the user – left to ( character</td>
<td>Formula 6</td>
</tr>
<tr>
<td>5.3.4. separating the server – left to ) character</td>
<td>Formula 7</td>
</tr>
</tbody>
</table>

Figure 19. Data arranged into records and data fields with Excel functions.

5.4. Handling redundancy (SP) – deep attention mode

As presented in Section 4.4, handling redundancy is conveniently achieved with a sequence of replacements. In spreadsheets, the replacement of characters can be carried out either using the Replacement command (detailed in Section 4.4) or functions. In this section, to
avoid repetition, the latter version is presented applying the SUBSTITUTE() function with the empty string as the replacement string. We must note here that there are cases when the TRIM() and/or the CLEAN() functions also work in the cleaning process, but they are not as reliable as the SUBSTITUTE() function. Building functions for reducing redundancy is not complicated. However, the numbers of replacements are not displayed; consequently, checking the number of replacements requires further solutions and skills.

Table 8. Algorithm for reducing redundancy in categories, using Excel functions (Figure 18 and Figure 19).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.1. deleting space Comments (Column Ncategory)</td>
<td>Formula 8</td>
</tr>
<tr>
<td>5.4.2. checking the number of replacements</td>
<td>Formula 9 or Formula 10 or Formula 12 (Figure 20, Cells C2, C3)</td>
</tr>
<tr>
<td>5.4.3. colouring the cells which do not match the requirements (optional)</td>
<td>Formula 13 (Figure 20, Column B)</td>
</tr>
<tr>
<td>5.4.4. deleting space Comment (Column Ncategory)</td>
<td>Formula 11</td>
</tr>
<tr>
<td>5.4.5. checking the number of replacements</td>
<td>Formula 12 (Figure 20, Cells 4)</td>
</tr>
</tbody>
</table>

To check the number of replacements, we can calculate the number of those records which do not match our requirements. To highlight the non-matching cells conditional formatting can be applied. In the example presented a separate sheet (LOL_test) is set up for the discussion (Figure 20).

Ncategory: {=SUBSTITUTE(Rcategory, " Comments","")}
Formula 8: Section 5.4.1

{=SUM(IF(ISERROR(RcategoryT),1))} (Figure 20, C2)
Formula 9: Section 5.4.2

{=SUM(IF(ISERROR(SEARCH(" Comments",Rcategory)),1))} (Figure 20, C3)
Formula 10: Section 5.4.2

Ncategory: {=SUBSTITUTE(SUBSTITUTE(Rcategory, " Comments",""), " Comment","")}
Formula 11: Section 5.4.4

{=SUM(IF(ISERROR(SEARCH(" Comment",Ncategory)),1))} (Figure 20, C4)
Formula 12: Section 5.4.4

RcategoryT: ={=SEARCH(" Comments",Rcategory)} and conditional formatting: =ISERROR(B2:B976) (Figure 20, Column B)
Formula 13: Section 5.4.3

Figure 20. Testing the number of substitutions with Excel functions and conditional formatting.
Calculating and colouring in the number of failed replacements reveals the instances which do not match the string of the SUBSTITUTE() / SEARCH() functions, and the mismatching cell(s) can be easily recognized. In this case, similar to the procedure in Word, one failed replacement was found (Comments vs. Comment) (Figure 20, Cells C2 and C3; Sections 4.4.6 and 4.4.7; Table 4).

Table 9. Algorithm for reducing redundancy in new comments, using Excel functions (Figure 18 and Figure 19).

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.6.</td>
<td>deleting space new Comments and converting it to a number (Column Nnewcomments)</td>
<td></td>
</tr>
<tr>
<td>5.4.7.</td>
<td>checking the number of replacements</td>
<td>Formula 15 or Formula 16</td>
</tr>
<tr>
<td>5.4.8.</td>
<td>colouring the cells which do not match the requirements (optional)</td>
<td>Formula 13</td>
</tr>
<tr>
<td>5.4.9.</td>
<td>deleting space new Comment and converting it to a number (Column Nnewcomments2)</td>
<td>Formula 17</td>
</tr>
</tbody>
</table>

Formula 14: Section 5.4.6

\{=\text{SUBSTITUTE}(Rnewcomments,"," new Comments","")*1}\}

Checking the number of failed replacements in new comments is somewhat simpler than in categories, because the conversion from a string datatype to number fails when a substitution is not carried out (Figure 19, Sheet LOL_norm, Column Nnewcomments). The shorter formulas can be applied while using additional cells and vectors (Formula 15), without them longer formulas are needed (Formula 16) (Figure 20, Cells E2 and E3).

The test-algorithm of the number of replacements in the case of " new Comments" is faster than in the previous problem. The SUBSTITUTE() function is the same; however, one further step in the conversion from text to number is carried out, by a multiplication. The mathematical operator returns an error massage if non-convertible text is found (Figure 20).

\{=\text{SUBSTITUTE}(Rnewcomments,"," new Comments","")*1}\}

Formula 14: Section 5.4.6

\{=\text{SUM}(IF(ISERROR(NnewcommentsT),1))}\}

Formula 15: Section 5.4.7

\{=\text{SUM}(IF(ISERROR(SEARCH(" new Comments",Rnewcomments),1))}\}

Formula 16: Section 5.4.7

\{=\text{IF}(ISERROR(SEARCH(" new Comments",Rnewcomments)),\text{SUBSTITUTE}(Rnewcomments," new Comment","")*1,\text{SUBSTITUTE}(Rnewcomments," new Comments","")*1)\}\}

Formula 17: Section 5.4.9

Table 10. Algorithm for reducing redundancy in views, using Excel functions (Figure 18 and Figure 19).

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.10.</td>
<td>deleting space Views (Column Nviews)</td>
<td>Formula 18</td>
</tr>
<tr>
<td>5.4.11.</td>
<td>checking the number of replacements</td>
<td>Formula 19 or Formula 20 or Formula 21</td>
</tr>
<tr>
<td>5.4.12.</td>
<td>colouring the cells which do not match the requirements (optional)</td>
<td>Formula 13</td>
</tr>
<tr>
<td>5.4.13.</td>
<td>converting text to number</td>
<td>Table 11</td>
</tr>
</tbody>
</table>
Table 11. Algorithm for converting Views to number datatype, using Excel.

<table>
<thead>
<tr>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>=SUBSTITUTE(Rviews,&quot; Views&quot;,&quot;&quot;&quot;)</td>
</tr>
<tr>
<td>19</td>
<td>=SUM(IF(ISERROR(RviewsT),1))</td>
</tr>
<tr>
<td>20</td>
<td>=SUM(IF(ISERROR(SEARCH(&quot; views&quot;,Rviews)),1))</td>
</tr>
<tr>
<td>21</td>
<td>=SUM(IF(ISERROR(SEARCH(&quot; views&quot;,Nviews)),1))</td>
</tr>
</tbody>
</table>

5.4.14. searching for \(k\) string

5.4.15. checking the absence of \(k\) string

5.4.16. separating views based on the absence or presence of \(k\) string and removing \(k\) string

5.4.17. converting string data type to number datatype

Views are presented both as whole numbers (Figure 11, Records 1 and 2; Figure 19) and strings – a real number followed by \(k\) (kilo), marking a thousand (Figure 11, Record 966; Figure 19). To convert all the numbers of views into a number datatype, we must search for the \(k\) string, check its presence, remove it, convert the string datatype to a number datatype, and multiply the small numbers by 1 and the \(k\) numbers by 1000.

\(=\text{SEARCH}("k",\text{Nviews})\)

5.4.18. checking the absence of \(k\) string

5.4.19. separating views based on the absence or presence of \(k\) string and removing \(k\) string

5.4.20. converting string data type to number datatype

5.5. Handling leading space characters – deep attention mode

After deleting the redundant expressions, the leading Space characters must be deleted. The leading Space characters are recognizable in Excel and Word (Figure 18 and Figure 19, Column I; Figure 21 Column A; Formula 26). The WP-conversion (Figure 21) clearly reveals that one of the Space characters is a non-breaking Space, which must be taken into consideration while calling the SUBSTITUTE() function. Copying the original string into argument position is one solution, while another is the calling of the CHAR() function with 160 as the argument. However, this latter solution requires further knowledge-transfer items.

Table 12. Deleting leading Space characters in categories, using Excel.

<table>
<thead>
<tr>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>=SUBSTITUTE(Nviews,&quot;k&quot;,&quot;&quot;&quot;)</td>
</tr>
</tbody>
</table>

5.5.1. deleting leading Space characters

5.5.2. copying both non-breaking and normal Space characters into argument position
Figure 21. Leading space characters in categories (Column A).

Ncategory: {=SUBSTITUTE(SUBSTITUTE(SUBSTITUTE(Rcategory, "Comments", ""), Comment", ""), " ", "")}

Formula 26: Section 5.5.1

Knowledge-transfer items: recognizing leading Space characters, differentiating normal and non-breaking Space characters, copying string of Space characters, replacement command or SUBSTITUTE() function

5.6. Separating source and time – deep attention mode

There are four different forms of presenting time in the table: (1) about an hour ago, (2) a day ago, (3), about * or ** hours ago (one or two digit-long), and (4)* days ago (only one-digit-long) (Section 4.1).

All these forms must be checked and handled separately to separate the source and the time and to normalize the time, in which process all the presented time-forms are converted to numbers, using the hour as the general unit.

Without going into detail, the formulas of the source and time conversions are presented in Formula 27–Formula 34. To shorten the formulas named vectors were introduced, referred to as Nsource1, Ntime1, Nsource2, Ntime2, Nsource3, Ntime3, Nsource4, and Ntime4.

Nsource1:
{=IF(ISERROR(SEARCH(" about an hour ago",Nsourcetime)),
Nsource1,
LEFT(Nsourcetime,SEARCH(" about an hour ago",Nsourcetime)-1))}

Formula 27

Ntime1:
{=IF(ISERROR(SEARCH(" about an hour ago",Nsourcetime)),
Ntime1,
1)}

Formula 28

Nsource2:
{=IF(ISERROR(SEARCH(" a day ago",Nsource1)),
Nsource2,
LEFT(Nsource1,SEARCH(" a day ago",Nsource1)-1))}

Formula 29
6. CONCLUSION

Present paper describes how word processors and/or spreadsheet applications and/or any combination of these applications can be utilized to convert a webtable to a normalized datatable in educational environment. However, we must emphasise that the subject is highly relevant in commercial and professional organisations too.

In the conversion process, first surface/hyper then mixed or deep attention data analysis is carried out. Following the data analysis, the algorithms of the conversion process are set up. Based on the algorithm, we can decide which tools need to be applied to solve the problems and to carry out the steps of the algorithm. One further consideration of the conversion process is the discussion and debugging phase, when the outputs must be checked and corrected if discrepancies are revealed. All these steps are detailed and discussed in the paper, presenting solutions for the conversion of one of the LOL Boards both in Word and Excel.

In the teaching-learning process presenting authentic tables has a strong motivating effect. However, we must keep in mind that it is the teacher’s responsibility which form is introduced in class. The form and content of the table must match the students’ background knowledge, stored in long-term memory – originated in informatics and/or other subjects, sciences –, the goals of the classes, and how knowledge gained through these tasks would help them in their further studies. The webtable→datatable conversion processes open up
novel methods for developing various computer problem-solving and computational thinking skills by utilizing knowledge-transfer items.

7. ACKNOWLEDGEMENT

The project was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

8. REFERENCES


9. SOURCES


Will Dynamic Arrays finally change the way Models are built?

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

ABSTRACT

Spreadsheets offer a supremely successful and intuitive means of processing and exchanging numerical content. Its intuitive ad-hoc nature makes it hugely popular for use in diverse areas including business and engineering, yet these very same characteristics make it extraordinarily error-prone; many would question whether it is suitable for serious analysis or modelling tasks.

A previous EuSpRIG paper examined the role of Names in increasing solution transparency and providing a readable notation to forge links with the problem domain. Extensive use was made of CSE array formulas, but it is acknowledged that their use makes spreadsheet development a distinctly cumbersome task. Since that time, the new dynamic arrays have been introduced and array calculation is now the default mode of operation for Excel. This paper examines the thesis that their adoption within a more professional development environment could replace traditional techniques where solution integrity is important.

A major advantage of fully dynamic models is that they require less manual intervention to keep them updated and so have the potential to reduce the attendant errors and risk.

1 INTRODUCTION

This paper starts by reviewing the decisions made at the time electronic spreadsheet was invented and looks at the impact of those original decisions. Dan Bricklin required a means for recording the parameters used in a formula which is both usable for the calculation and intelligible for the user. Dan was well aware of the “programmer’s way” of achieving this through the use of named variables but instead plumped for a strategy that was more action-led and intuitive. Thus, the co-ordinate approach of identifying data by location on a grid was created. This required no effort on behalf of the user to define either the terms they wished to use or their relationship to business objects; to do so was deemed ‘tedious’.

These early decisions still characterise spreadsheet use today. Catering specifically for ‘end user computing’ has allowed the spreadsheet to become the most widely used and understood program throughout the world. It is the most widely used medium for the exchange of numerical data and across various business areas domain-specific software is likely to feature an ‘Export to Excel’ button.

Although professionally developed spreadsheets tend to be less haphazard in their construction, it is something of a surprise that the same primitive spreadsheet methods still dominate most spreadsheet development. The consistent use of defined names, array formula, structured references, pivot tables, Power Query and Power Pivot is the exception rather than the rule.

In place of adopting strategies that impose greater structure through the coding techniques employed, some of the financial modelling standards approach the problem by defining rules and standard practices to constrain the user to somewhat more ordered practices. This may start to address the ‘Wild West’ symptoms of endemic spreadsheet risk but does little to address the underlying disease.
2 THE SPREADSHEET CHALLENGE

2.1 Applicability

Standard Excel practice merely captures what the user does but provides no real indication of their intent. Worse, it does it in the most obscure manner imaginable e.g. ‘take the number one cells above wherever it is that your formula happens to reside and add to it the number you find 6 cells to the right’. What on earth does the position of two numbers on a worksheet tell you about the business problem? Absolutely nothing; such an approach utterly fails to capture any business logic or the user-intent.

This bizarre process is then usually followed by ‘and then copy the formula down’: why, one might ask, how far? The action implies the existence of an unacknowledged list or array. Any such structure that may exist with the problem domain has to be inferred from the pattern of formula references that results from user actions (bearing in mind that each action carries with it something like a 2% chance of being an error, Panko (2015).

Almost always, the business objects of interest turn out not to be single cells; they may be arrays representing a quantity varying over time or lists of related objects such as that contained within a database table. This is evident in the case of financial models where one of the few things that is universally accepted is that all formulas should be uniform across the time period of the model. Such uniformity is the defining property of an array formula, yet the standards tend to either reject the use of array formulas or at least regard them as methods of last resort. Only if the syntax captures these objects and relates them to domain knowledge does one have anything of value. Without the ability to name objects at an appropriate level of abstraction, one must resort to annotation as a means of communicating the significance of content. One might hope to see the annotation in reasonable proximity to the range it describes but there is no real link between such labels and the data.

It should be noted, that although such criticism of the concept of direct cell referencing, as well as the merging of presentation, formulas and values as single cell objects is unusual in the world of spreadsheets, others have raised similar objections over the years, Hellman (2001), and the companies behind Analytica, Quantrix and related software have also had harsh criticisms, even ridicule, of spreadsheet technology. Finally, a quote from the Quantrix website:

“Traditional spreadsheets use cell addresses – such as B2 for column B of row 2 – rather than meaningful names – such as Revenues or Expenses. Anyone who has tried to read such formulas, whether written by someone else or oneself, knows how hard they can be to understand or verify”.

Not many such critics still use Excel though.

3 THIS IS THE WAY IT HAS BEEN SINCE THE BEGINNING; WHY CHANGE NOW?

Last September fundamental changes to the way in which Excel both stores data and performs calculation were announced at the Ignite Conference held in Florida. Prior to that, the value property of a single cell was limited to a single value, be it a String, a Double, a Boolean or an Error. At the Florida meeting Joe McDaid of Microsoft described how the newest releases of Excel could associate array object with a single cell.

Before, the elements of an array would have to be individually stored in a contiguous range of cells, and formulas would be implemented by the user on a cell-by-cell basis. Clearly, this would be horribly inconvenient if the references within each formula had to be built individually but, unless one of the referenced arrays happens to be transposed, the relative
position of the formula cell and the cells it references will be uniform throughout the
calculation. It is this property that was exploited with the introduction of relative
referencing as a spreadsheet default; a relative reference has no intrinsic meaning, but it
does serve to associate the correct values across the remaining cells of a referenced array.
References in the form RC[-2] make the relationship explicit, whilst B7 conceals the
relative nature of the reference behind a notation that at first sight appears to be absolute.
It is the use of this ‘hybrid’ notation that gives rise to the characteristic pattern of references
seen on spreadsheets.

In the new Excel all that changes. Any formula that references one or more arrays will
return an array; it is no longer necessary to reference cells one by one using relative
references to build an array result. The formula is contained within a single cell, though
adjacent empty cells are needed in order to display the result, a process described as
‘spilling’. The notation used for the resulting array is the cell reference followed by the ‘#’
character. The reference to the anchor cell may be a direct reference or a defined name,
though in published work to date the former is more commonly seen

4 ARRAY FORMULAS IN MODERN EXCEL

Array formulas create amazingly simple solutions. Instead of a multitude of formulas that
should be the same but are in no way constrained to be so, the formula resides in a single
cell. The thing that has changed is that a cell can now accept a complete array as its ‘value’
property whereas previously it was limited to taking a scalar value, be it a string, a number,
a Boolean or an error.

Clearly, it is not possible to display an array within a single cell, so what is done instead is
to use adjacent blank cells to show individual values, a process referred to as ‘spilling’.
This will be demonstrated in the example section below.

If one wishes to link the formulas back to the business problem, it is only these single cells
that need to be named. If a name such as ‘revenue’ is applied to the anchor cell containing
the formula then the entire array can be referenced as ‘revenue#’ and, as such the reference
will adjust dynamically to match the content.

Whilst the headline functionality of the new modern Array Formulas is to make array
calculation the default mode of operation, the new functions that have accompanied the
release have a strong focus upon data analysis and list operations. Here, the distinction I
am making is that an array is ordered, so allowing elements to be returned by index,
whereas the significance of a list does not depend upon order. An array often represents a
function of some independent variable, time being the most important example, but many
other possibilities exist such as random variables within a probability calculation. The key
difference is that operations such as filtering and sorting are not valid operations applied to
an array whereas they are a mainstay of managing lists; a calendar sorted alphabetically is
not much use to anyone!

Despite the distinction made here, there is a huge overlap that permits lists to be treated as
arrays and conversely. In the following section, I will demonstrate the dynamic array
formulas used to create a crosstab report. Of course, for large datasets, one would use
PowerPivot but for small to medium datasets the new array functions offer a viable and
very responsive alternative.
5 EXAMPLE – DATA ANALYSIS

The following example is based upon a hypothetical computer business with nation-wide sales. The data table has 5000 rows and the objective is to create a crosstab summary using formulas.

![Data Table] (full database)

- record 25/12/2013 east laptops 1 $712.00 $712.00
- record 01/01/2014 south desktops 3 $471.00 $471.00
- record 02/01/2014 west tablets 3 $570.00 $570.00
- record 02/01/2014 midwest software 4 $349.00 $1,396.00
- record 02/01/2014 west laptops 3 $584.00 $584.00
- record 02/01/2014 midwest servers 1 $1,697.00 $1,697.00
- record 03/01/2014 south software 1 $482.00 $482.00
- record 03/01/2014 south tablets 2 $34.00 $68.00
- record 03/01/2014 mountain tablets 1 $118.00 $118.00
- record 03/01/2014 west tablets 2 $995.00 $995.00
- record 03/01/2014 midwest tablets 2 $622.00 $622.00
- record 03/01/2014 midwest laptops 1 $806.00 $806.00
- record 03/01/2014 laptops 10 $474.00 $4,740.00

Figure 1 Extract taken from source data table

The table itself is conventional with values providing information relating to sales volume and unit price, combined with other fields that provide further information related to the transaction. Excel tables support calculated fields and the ‘amount’ could well have been included as part of the table. Since the aim here is to discuss the potential impact of modern dynamic arrays an alternative route is taken in this case.

The formula

\[
\text{= Sales[units]} \times \text{Sales[price]}
\]

is inserted into the first cell of ‘amount’ (it happens to be J8 but that is immaterial).

![Array Formula] (2a) Inserting the formula (2b) Spill array (2c) Spill error

Figure 2 Building an Array Formula

Whole column references to the table are chosen so that as soon as the formula is committed using the ‘Enter’ key the entire array of 5000 values is calculated as shown in Figure 2b.
The process of displaying the resulting array over adjacent cells is known as ‘spilling’. It is not unusual for concern to be expressed at this point: ‘What happens if I have important data in the way of the spilt array; will I lose it?’ The data will not be lost; instead the dynamic array will simply give a #SPILL! error, as shown in Figure 2c. As soon as the offending content is deleted or moved, the array appears. Something else that is clear from the figure showing the #SPILL! error is that what appears as a full and busy worksheet can be almost entirely blank space with only a few cells populated by formulas. Next we turn to the data analysis.

![Cross tab using SUMIFS](image)

**Figure 3: Extract from Pivot Data by aggregation**

The key to producing a simple crosstab solution that updates dynamically as the data table is extended or modified is to use the newly introduced array functions to create the row and column headers.

The first two functions turn manual data management processes into simple formulas that update automatically, thus eliminating one source of user error. The following formula yields the distinct row headers and the sorted list resizes dynamically without user intervention.

\[
= \text{SORT}\left( \text{UNIQUE(Sales[region])} \right)
\]

The next formula produces the column headers as a list of distinct values but then nests this within the classic array function \( \text{TRANSPOSE} \) to give a row of headers. Now, however, \( \text{Ctrl+Shift+Enter} \) is not required

\[
= \text{TRANSPOSE}\left( \text{UNIQUE(Sales[goods])} \right)
\]

Now it only remains to fill in the table

\[
= \text{SUMIFS( amount#, Sales[region], region#, Sales[goods], goods# )}
\]

Each value is calculated by referencing an element of these two vectors as criteria and a column of the table or a range reference for the values to be summed. The Names are single cell references but the addition of the “#” symbol changes the reference from a single cell into a dynamic range reference. The formula is placed in a single cell and the 25 values are an array property of that cell. The other 24 cells of the Range are empty but they serve to display the result to the user.

Used for small to moderate sized problems the new Dynamic Array functionality works seamlessly as a means of data aggregation.
6 EXAMPLE – MODELLING WITH SIMILAR LINE ITEMS

This problem was developed by Levi Bailey and Vishal Rander within the context of a LinkedIn Discussion Group to test and compare the various approaches to building financial models as advocated by several leading finance modelling groups. It was presented by the present author in the 2016 EuSpRIG conference, along with a variety of non-financial models, to demonstrate the feasibility of using defined names and CSE array formulas to build models. It was concluded that such methods can provide a coherent solution strategy in which the problem is solved by a sequence of formulas resembling the steps of a programmed language. Despite that it is acknowledged that the use of CSE array formulas can be inflexible and labour-intensive.

Since that time, the Excel calculation engine has been redeveloped and Array formulas are now the default mode of calculation. Only the anchor cell contains the formula and the entire array is a property of that cell. The size of the array as displayed no longer depends upon the actions of the user; it is determined by the sizes of the referenced arrays.

In the main, this presents massive advantages since an entire array is generated by committing formula in a single cell. As will be explained, though, this problem exhibits several features that makes the application of dynamic array formulas less than routine.

One such calculation that cannot be achieved with the array formula is an accumulation in which each element of an array refers to its immediate predecessor. In mathematics this would be known as a recurrence relation and in finance it might be a 'corkscrew' or an escalation. In a reply to an email Joe McDaid confirmed that Dynamic Arrays are not suited to corkscrew style calculations because they do not support breakup – which CSE arrays do. The solution is to revert to CSE arrays for the calculation, which will mean that they are no longer dynamic, or, as is done in the example, to reformulate the problem in a manner that allows each term to be calculated without reference to the recurrence relation. There are examples of this in the spreadsheet that accompanies this paper.

The first is the time ruler. A standard way of creating the start and end of each time period is to set each start date by incrementing the end date from the previous period by one day. The end of period is then calculated as the end of month, quarter or year as required. This will generate an error as a dynamic array formula. If the time interval is constant along the ruler the simple workaround is to calculate the dates directly from the period index 'p'.

The number of periods depends both on the time duration of the model and the number of months per period. The new function SEQUENCE can then be used to generate 'p'

=SEQUENCE(1, COLUMNS(monthlyDemand) / monthsPerPeriod, 0, 1)

starting at zero.

The start of each period then follows immediately using

= 1 + EOMONTH( input.startDate, p * monthsPerPeriod )

This time ruler automatically resizes to match the problem description. A similar problem occurs with the product price escalation formula. Again, step by step multiplication by the
An escalation factor would be a normal approach. Provided the escalation rate is assumed constant, however, a simple formula will give the price at any time period:

\[ \text{price.at.time } = \text{price.initial} \times \left(1 + \text{escalation.rate} \right)^{\text{number.of.periods}}. \]

In these situations, the accumulations have been replaced by the terms of an arithmetic sequence and a geometric sequence respectively. In this case the array ‘p’ causes the formula to spill across the sheet with minimal user input. In the latter case the initial prices and escalation factors are also arrays which specify values for each product. Therefore, the formula also spills down the sheet to give a 2D array of values.

\[ \text{price.at.time} = \text{price.initial} \times \left(1 + \text{escalation.factor} \right)^{\text{number.of.periods}}. \]

A more severe accumulation problem arises when one tries to accumulate the demand over the diverse products to compare the result with the total production capacity for each period. Previously a CSE array formula was used with the referenced array being offset one row from the result. Because this required array breakup, an alternative solution was sought using matrix multiplication. This used a lower triangular matrix in which the values below the leading diagonal were all ones. Multiplied into the demand array, this first gives the top row, then the sum of the top and second rows and so on. The matrix itself is build using the \text{SEQUENCE} array function within the named formula ‘accumulate’ that refers to:

\[ \text{MMULT}( \text{accumulate}, \text{demand#}) \]

where ‘\text{demand#}’ is the notation for a split array anchored at a cell named ‘\text{demand}’. \text{MMULT} implements the mathematical operation of matrix multiplication.
From there the residual capacity is

\[= \text{IF}(\text{maximum.production#} > \text{cumulativeDemand#}, \text{maximum.production#} - \text{cumulativeDemand#}, 0)\]

and the volume produced for each product is

\[= \text{IF}(\text{demand} < \text{initialCapacity#}, \text{demand#}, \text{initialCapacity#})\]

A striking feature of the extract from the workbook shown in Figure 4 are the magenta cells. These are produced by a conditional format based upon the formula

\[=\text{ISFORMULA(RC)}\]

The surprise for anyone versed in traditional methods is how few cells contain formulas. The second problem to be overcome when building solutions using array formulas is to aggregate a 2D array either row-by-row or column-by-column. Pretty much all the aggregation operators such as SUM, PRODUCT, SMALL, MIN, LARGE, MAX, AND, OR produce a single result from a 2D array yet it is perfectly reasonable to want to apply such formulas to each row or each column. The expectation is that the user will use relative referencing to select rows or columns that they wish to aggregate manually. This, however, means that the column of row sums, for example, does not expand dynamically as additional rows are added to the data. The one exception to this limitation is MMULT which will perform batches of weighted sums either row-wise or column-wise. The result obtained from MMULT will expand dynamically, provided care is taken to avoid non-numeric fields and to ensure the internal dimensions always match exactly.

In the present case, the total product revenues are calculation using

\[= \text{MMULT}(\text{TRANSPOSE(active)}, \text{product.revenue#})\]

where ‘active’ is a Boolean array that is 1 where data has been provided for the specific product and 0 if it has been omitted.

This solution is effective but is unnecessarily complicated in terms of its mathematics. In my mind it would be better to have a function that indicates that the aggregation should be carried out column by column but returned as a row array of results. Suggestions might include

\[= \text{SUM(BYCOLUMN(product.revenue#))}\]

or, if a more compact notation were required

\[= \text{SUM(|product.revenue#|)}\]

where the ‘pipe’ character here indicates a column.
7 ELEMENTS OF RISK

As yet there is virtually no experience of applying the methods outlined within the Excel community, so no evidence exists to back up statements concerning risk. However, using the Panko (2010) taxonomy of spreadsheet errors, Figure 5, the following might be considered to provide reasonable expectations.

Firstly, considering execution errors, the fact that dynamic arrays require an order of magnitude fewer formulas must reduce the opportunity for error. Typing errors and pointing errors, in particular, may be expected to reduce.

Some typing errors manifest themselves as consistency errors in which a region of formulae that may be intended to be the same have cells that differ from the surrounding region. Such inconsistencies can also arise from ill-conceived attempts to correct a specific instance of the formula or the data it references. Dynamic arrays eliminate such errors; consistency errors are simply not possible when only one instance of a formula is present.

Sizing errors, whereby only part of a range is generated, or subsequently referenced within an aggregation formula, may also be expected to reduce since these steps no longer require user interaction.

The consistent use of Names may also be expected to have a role in reducing the effect of typing errors in that wrongly typed names are more likely to generate #NAME! errors than erroneous references. In particular, erroneous references to ‘white space’ that has not been assigned to the model are eliminated.

Turning now to planning errors: it is still quite possible to introduce errors in the application of Domain knowledge. That said, the consistent use of meaningful names creates formulae that are semantically meaningful. This, in turn, may be expected to reduce the likelihood of such errors passing unnoticed.

A potential disadvantage of programming Excel through the use of named arrays is that the more abstract level of thinking required is alien to most practitioners. It is thought likely that planning errors associated with inadequate Excel knowledge will actually increase.
DISCUSSION AND CONCLUSIONS

The introduction of modern dynamic arrays could change the nature of spreadsheet development utterly. No longer need it be a process of manual manipulation of large blocks of data by reference to the rows and columns they occupy. As has been shown, dynamic formulas typically reside in a single cell and only appear to fill large regions of the worksheet for presentational purposes.

There remains scant need for the concept of relative referencing or for the default A1 notation introduced in the original spreadsheet concept. Some additional work is still required on behalf of the developer to name each formula, now occupying a single cell, but it requires little more effort than simply labelling the cell, as would be normal good practice. Referencing raw data, whether input directly to Excel or read in via Power Query, does require the use of multi-cell ranges but there, assuming Tables are used extensively, naming is largely taken care of by structured referencing.

The advantage of fully dynamic models is that they require less manual intervention to keep them updated or, alternatively, to modify them to accept new data sets. This reduces risk and the attendant errors. The advantage of consistent naming conventions is that the formulas are semantically meaningful and can be readily checked for correctness in terms of the both the Excel logic and, more importantly, the domain knowledge that links the proposed solution to the business problem. Consistency errors disappear because the formula is only written within a single cell. Array sizing errors in which the final terms of an array are accidentally omitted are also largely eliminated because it is the code that sizes the array and not the user. With DA, array formulas become the path of least resistance rather than some obscure method of last resort.

What is lost is the ability to identify relationships between individual cells using precedent trees and to check their correctness using a pocket calculator. I believe that is small loss, to check a representative sample sufficiently to estimate the probability of the workbook being correct would be a mind-numbingly tedious and repetitive task. Moreover, it is addressing the wrong problem. It is rare that Excel produces an incorrect result for a correctly formulated formula; it is far more likely that the formula is incorrectly implemented or does not capture the business logic.

So, having argued that modern Excel is capable of revolutionising spreadsheet practice, yielding something far more akin to a rigorous programming approach that the ad-hoc style of end user programming, the question remains “Will such a revolution actually come to pass?”

By providing backward compatibility one ensures that it is still possible to develop workbooks in the old way, that of replicating single-cell formulas. Users can just continue the way they always have, oblivious to all new functionality as they have, by and large, done in the case of Tables, Pivot tables, Power Query etc. If we assume that most users, who are not aware that Excel involves programming and do not see its relevance, will not move beyond the turn of the century, how should developers respond? I would argue that developers should no longer accept the limitations of their client’s knowledge as constraints on the techniques they employ. If a client wants an ad-hoc programming style they are at liberty to do it themselves. My experience is that, once a structured Excel application has been delivered to the client, it will over time acquire unstructured additions as a result of client interaction. Sooner or later it will go wrong and at that point there exists an opportunity for the developer to refactor these changes to provide new functionality.

Ultimately a well-constructed ‘app’ built on Excel as a platform provides a far more controlled risk environment than a loosely structured worksheet that everyone feels free to modify.
9 REFERENCES


McDaid J, “Dynamic Arrays and corkscrew calculations” (private communication October 2018)


Cubes an alternative to spreadsheets to mitigate risk and improve performance?

www.xcubes.net
Marc BRAUN
marcbraun@gmail.com

ABSTRACT

Multidimensional or even simple business modelling with traditional spreadsheets often leads to complex models which are difficult to understand and to maintain, is prone to errors and generally time consuming.

A cube-based system offers a structured and scalable approach to modelling, forecasting, reporting and data analysis with little manual intervention once the model is set up.

The practical demonstration will be made with XCubes a Multidimensional Spreadsheet System software package.

1. INTRODUCTION

For the purpose of this paper we started from the Paul Mireault, Honorary Professor at HEC Montréal, Multi-Dimensional challenge Kit on Excel (“Acme TechnoWidgets”) to design a XCubes solution.

(Mireault, Structured Spreadsheet Modelling and Implementation: A Methodology for Creating Effective Spreadsheets, 2017) has developed a Structured Spreadsheet Modelling and Implementation methodology (SSMI) which is a strongly recommended starting point whatever software solution is being chosen.

We will first describe the key concepts of Cubes, present the result of the XCubes solution and summarize the benefits of using cubes instead of spreadsheets.

2. CUBES VERSUS SPREADSHEETS

The purpose here is not to stress out limitations of traditional spreadsheets for modelling purposes whether flat or multidimensional. There have been numerous articles on this subject.

(Panko, 2015) has written several articles about Spreadsheet errors.

(Read & Batson, 1999) in Spreadsheet Modelling Best Practice have concluded that spreadsheets are week at handling multi-dimensional data.

We will rather focus on the benefits of a full multidimensional approach.

(Litt, 2017) wrote an article about a Spreadsheet risk mitigation in Complex Multi-Dimensional Models in Excel where the solution appears to be a “PivotModel” inside Excel, similar to a MOLAP solution.

(Braun, XCubes Download for PC, 2019) has developed XCubes a software solution which does not rely on Excel and uses natively cubes to perform complex multi-dimensional modelling.
(Murphy) has listed in his comparison of Spreadsheets with other development tools several areas where other tools perform better in terms of Data separation, Scalability, Type Safe, Links which are addressed in XCubes.

The main differentiators between Cubes and Spreadsheets are as follows.

2.1 Data separation (n-tier architecture)

The multidimensional nature of the cubes cannot accommodate the Rows and Columns flat reference system used by traditional spreadsheets.

Therefore, the formulas are removed entirely from the spreadsheet surface which is only meant to receive data.

The removal of formulas at this level eliminates the risk of damaging the model during routine operations like model maintenance, data entry or zeroing a model to bring it to a clean state.

The n-tier architecture effectively separates the data from the business logic and also from the presentation level.

2.2 Structure with Dimensions

A dimension is list of two kinds of items:
- Detail items which receive data.
- Calculated items (in bold by convention) which store user defined formulas: calculation logic or business rules for the model.

The items are strongly typed for safety: number, text, date, … to prevent data entry mismatches. In spreadsheets date types are inferred during data entry.

The formulas are written in plain text using self-explanatory labels defined by the user in the ‘Item Code’ field. The ‘Item Description’ is used for displaying the meaningful label.

<table>
<thead>
<tr>
<th>Item Code</th>
<th>Item Description</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>Sales</td>
<td></td>
</tr>
<tr>
<td>COGS</td>
<td>Cost of goods</td>
<td></td>
</tr>
<tr>
<td>Margin</td>
<td>Gross margin</td>
<td>SA - COGS</td>
</tr>
</tbody>
</table>

Storing formulas in dimensions reduces drastically the number of formulas required in a model. The system automatically propagates the calculations across rows and columns as required.

Dimensions can be combined freely to build any kind of cubes desired by the user. There is no need for preexisting data to create dimensions or cubes.

Dimension are strictly independent from each other (self-contained) so that can be used in different cubes.
As a result, changes made to a dimension will automatically be reflected in other cubes using it.

2.3 Cubes as efficient data containers
Cubes are built with at least two dimensions representing Rows and Columns.

Dimensions can be further added, removed or exchanged as needed. This will automatically trigger the reorganization of cubes. Existing data is preserved during the process.

The same is true when items are added or removed from individual dimensions, giving its dynamic nature to the system.

Cubes can be sliced and diced using Page dimensions. This will break a large area of data into smaller parts. It allows easier data analyzes and facilitates data entry by showing a specific portion of a cube like a Region (see below).

Data can be quickly filtered on empty rows/columns, on detail/formula items or by turning on or off items in the dimensions.

The views are created by dragging and dropping dimensions in various areas (Row, Column, Page).

Cubes are compact as they often replace multiple tabs in a spreadsheet used to perform intermediary data display or calculation.

2.4 Centralized view of all Links, References and variables

The interrelations between cubes using data Links, References, Variables are maintained in a single place as part of user projects.
Objects which cannot be treated as cubes because they do not have the two required dimensions can be defined as global variables for a project.

2.5 Data entry

As cubes are efficiently structured it easy to create robust and standardized data loading mechanisms. The most obvious is to load structured data directly from SQL Databases tables. This is achieved via data links using industry standard data sources: ODBC, OleDB or DataLink.

Data loading prevents common data entry mistakes like omitted or incorrect entries due to typo errors, entries made in wrong cells leading to possibly overwriting formulas in spreadsheets.

2.6 Project oriented approach

Storing and managing dimensions, cubes, tables, links, references, variables, data sources and other objects in a single place (Project) managed entirely by the application eliminates the risk of having broken External Links which can render spreadsheets potentially unusable. This will guaranty the integrity of the project and all of its components.

2.7 Pivot Tables in comparison

This feature has been built into Excel to allow data visualization and can be somewhat compared to a cube.

However, a Pivot Table cannot be created without and existing dataset. Calculation capabilities are limited although some level of customization is possible. PivotTable generally work from a single data source and it is not convenient to load data from multiple sources.

On the other hand, cubes can be freely built based on ad hoc created dimensions and populated either by manual data entry or by multiple data links from various.

2.8 When to use cubes

Most of the time business models have several dimensions and thinking multidimensional right from the start will bring immediate benefits liked structured and scalable models.

Dimensions can be easily construed from various lists: time periods, products, employees, geographies, versions, financial statement items, and so on.
3. THE ACME TECHNOWIDGETS CASE STUDY

3.1 Model layout

The challenge from (Mireault, Multi-Dimensional Spreadsheet Challenge, 2018) is summarized below. The list of formulas used in this model can be found in Appendix 1.

Based on an annual production capacity and using input parameters from:

<table>
<thead>
<tr>
<th>Sector data</th>
<th>Month split by sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monthly fixed costs</td>
<td>Regions-sector split</td>
</tr>
<tr>
<td>Sector-product split</td>
<td>Product data</td>
</tr>
<tr>
<td>Delivery cost</td>
<td></td>
</tr>
</tbody>
</table>

The result is a monthly summary showing the following calculated values:

<table>
<thead>
<tr>
<th>Unit sales by product and region</th>
<th>Unit sales by product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit sales by month</td>
<td>Sales amount by product and month</td>
</tr>
<tr>
<td>Profit by month</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4 - ACME TechnoWidgets set of cubes.
The Unit Fixed Costs in “Monthly Fixed costs” have to be calculated separately so that they can be properly allocated to the products in the “Months result” cube.

The resulting cube (‘Months result”) is built with 5 dimensions (4,680 cells):

- Months: Jan to Dec, Total
- Regions: North, South East, South West, East, West, Total
- Products: Standard, Deluxe, Total
- Sector: Government, Military, Private, Education, Total
- Measures: Quantities, Sales, Costs, Profit
The formula for the Quantity measure would be:

```plaintext
REFERENCE("Sector Annual Demand Units") * REFERENCE("Month and Sector Split") * REFERENCE("Region and Sector Split") * REFERENCE("Sector and Product Split")
```

REFERENCE () can be compared to the LOOKUP () function in Excel.
It automatically walks thru the items of any common dimensions of the source and destination cubes.

For example, REFERENCE("Month and Sector Split") will walk thru the ‘Month Split by Sector’ (Figure 7) which is the source cube and pick the value (split percentage) at the intersection of Month and Sector during each iteration since the Month and Sector dimensions are shared in both cubes.

As the dimensions are strictly independent from each other the REFERENCE (“Label”) function must be defined on a cube by cube basis.
The Label is just a description to identify the reference to define and has no further meaning.
The reference manager is used to connect each reference to a specific input cube (cf. Figure 3).The references are resolved i.e. the values are fetched during cube recalculation.
3.2 Size of cubes

![Figure 8 - Cube sizes](image)

The size of cubes is the product of all item count in each dimension for each dimension.

For Monthly the result is:

Months (13) X Regions (6) X Products (3) X Sectors (5) X Measures (4) = 4,680 cells

Although there are actually 12 months, 5 regions, 2 products and so on, the Total item is accounted for as an additional item.

3.3 Number of formulas

The challenge model contains 20 variables and is summarized below:

<table>
<thead>
<tr>
<th>Calculated &amp; Output variables</th>
<th>Items</th>
<th>12</th>
<th>4</th>
<th>2</th>
<th>5</th>
<th>Dimension Size</th>
<th>Number of Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Month</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>60</td>
</tr>
<tr>
<td>Sector</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Product</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Region</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Product-Region</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Sector-Product</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>Sector-Region</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Month-Sector</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>48</td>
<td>0</td>
</tr>
<tr>
<td>Month-Sector-Product</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>96</td>
<td>192</td>
</tr>
<tr>
<td>Month-Sector-Product-Region</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>480</td>
<td>960</td>
</tr>
<tr>
<td>Month-Product-Region</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>Month-Product</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>48</td>
</tr>
<tr>
<td>Total Profit</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1427</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 - Formulas required in Excel
### Dimensions, Number of Formulas, and Kind of Calculation

<table>
<thead>
<tr>
<th>Dimensions</th>
<th>Number of Formulas</th>
<th>Kind of calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures</td>
<td>4</td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sales</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Costs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Profit</td>
</tr>
<tr>
<td>Month</td>
<td>1</td>
<td>Dimension total</td>
</tr>
<tr>
<td>Products</td>
<td>1</td>
<td>Dimension total</td>
</tr>
<tr>
<td>Regions</td>
<td>1</td>
<td>Dimension total</td>
</tr>
<tr>
<td>Sector</td>
<td>1</td>
<td>Dimension total</td>
</tr>
<tr>
<td>Sector data</td>
<td>4</td>
<td>Sector price Factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Base price (reference to a global variable)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sector base price</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sector Annual Demand Units</td>
</tr>
<tr>
<td>Fixed Costs</td>
<td>1</td>
<td>Unit fixed cost</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>13</strong></td>
<td></td>
</tr>
</tbody>
</table>

*Table 2 - Formulas required in XCubes*

In the challenge the model contains **1,427** formulas whereas XCubes requires only **13** formulas to achieve the same results.

### 3.4 Scalability

The scalability of a model determines what efforts are required to grow or shrink it. Some models might be of fixed or static others need to be adjustable.

One of the metrics that can be used to assess the complexity and the risks of making changes is the number of formulas needed to be added/altered in order to accommodate the change of structure.

Based on the current Acme model the impacts are as follows:

<table>
<thead>
<tr>
<th>Adding one item in any dimension</th>
<th>In Excel</th>
<th>In XCubes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sector</td>
<td>297</td>
<td>0</td>
</tr>
<tr>
<td>Product</td>
<td>677</td>
<td>0</td>
</tr>
<tr>
<td>Region</td>
<td>218</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 3 - Adding an item in one dimension*

The number of additional formulas required in Excel is computed by adding one item in the dimensions in the Table 1 - Formulas required in Excel, and calculating the difference with the initial 1,427 formulas.

<table>
<thead>
<tr>
<th>Adding a new dimension like Years</th>
<th>In Excel</th>
<th>In XCubes</th>
</tr>
</thead>
<tbody>
<tr>
<td>With N items</td>
<td>(N-1) \times 1,427</td>
<td>1 (Total item – if wanted)</td>
</tr>
</tbody>
</table>

*Table 4 - Adding a dimension in a model*

1. N-1 indicates that there is already an implicit year in the model
Due to its extensible nature, XCubes requires no additional formulas when adding any number of items in any of the considered dimensions.

Furthermore, adding a new dimension with any number of items will generally add only a formula which is usually the Total for that dimension.

4. CONCLUSION

Cubes are best fitted when the model can be broken down in blocks to store well-structured data. They make analysis, calculation and visualization even on large data sets much easier.

They are most beneficial when scalability is an issue with models expected to grow over time.

Overall, they contribute to limit the risks of complex spreadsheet models by drastically reducing the number of required formulas and make even complex models easier to understand and maintain.

Finally, the user will be able to spend more time on analysis rather than worrying about reliability and performance.

The XCubes ACME TechnoWidgets solution is available for download (Braun, Windows Tutorials, 2019).
### Appendix 1: List of the 20 formulas used in ACME TechnoWidgets

<table>
<thead>
<tr>
<th>Variable</th>
<th>T</th>
<th>Dimension Set</th>
<th>Value / Formula</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sector Price Factor</td>
<td>C</td>
<td>Sector</td>
<td>1-Rebate Percentage</td>
<td></td>
</tr>
<tr>
<td>Sector Base Price</td>
<td>C</td>
<td>Sector</td>
<td>Base Price * Sector Price Factor</td>
<td>Base price of the Standard widget sold in the sector</td>
</tr>
<tr>
<td>Sector Annual Demand Units</td>
<td>C</td>
<td>Sector</td>
<td>DemParB/Sector Base Price *DemParA</td>
<td>Estimated demand for the sector, in units</td>
</tr>
<tr>
<td>PR Unit Cost</td>
<td>C</td>
<td>ProductRegion</td>
<td>Unit Production Cost + Unit Delivery Cost</td>
<td>Cost of producing and delivering one unit</td>
</tr>
<tr>
<td>Annual Sector-Product Unit Sales</td>
<td>C</td>
<td>Sector-Product</td>
<td>Sector Annual Demand Units * Product Distribution per Sector</td>
<td>Annual sales</td>
</tr>
<tr>
<td>Price</td>
<td>C</td>
<td>Sector-Product</td>
<td>Sector Base Price * Base Price Multiplier</td>
<td>Price of each product in each sector</td>
</tr>
<tr>
<td>Annual Sector-Product Sales Amount</td>
<td>C</td>
<td>Sector-Product</td>
<td>Annual Sector-Product Unit Sales * Price</td>
<td>Annual sales amount of each product in each sector</td>
</tr>
<tr>
<td>MSP Unit Sales</td>
<td>C</td>
<td>Month-Sector-Product</td>
<td>Annual Sector-Product Unit Sales * Monthly Sales Distribution per Sector</td>
<td>Unit Sales per month, sector and product</td>
</tr>
<tr>
<td>MSP Sales Amount</td>
<td>C</td>
<td>Month-Sector-Product</td>
<td>Annual Sector-Product Sales Amount * Monthly Sales Distribution per Sector</td>
<td>Sales Amount per month, sector and product</td>
</tr>
<tr>
<td>MSPR Unit Sales</td>
<td>C</td>
<td>Month-Sector-Product-Region</td>
<td>MSP Unit Sales * Region Sales Distribution per Sector</td>
<td>Unit Sales per month, sector, product and region. (This is the finest granularity of the Unit Sales)</td>
</tr>
<tr>
<td>MSPR Variable Cost</td>
<td>C</td>
<td>Month-Sector-Product-Region</td>
<td>MSPR Unit Sales * PR Unit Cost</td>
<td>The variable cost of producing and selling the widgets per month, sector, product and region.</td>
</tr>
<tr>
<td>Monthly Variable Cost</td>
<td>C</td>
<td>Month</td>
<td>SUM(MSPR Variable Cost)</td>
<td>The monthly variable cost of producing and selling all the products in all the regions and for all the sectors.</td>
</tr>
<tr>
<td>Monthly Unit Sales</td>
<td>O</td>
<td>Month</td>
<td>SUM(MSPR Unit Sales)</td>
<td>The monthly number of units of all the products in all the regions and for all the sectors.</td>
</tr>
<tr>
<td>Monthly Sales Amount</td>
<td>C</td>
<td>Month</td>
<td>SUM(MSP Sales Amount)</td>
<td>The monthly sales amounts of all the products in all the regions and for all the sectors.</td>
</tr>
<tr>
<td>Monthly Costs</td>
<td>C</td>
<td>Month</td>
<td>Monthly Fixed Cost + Monthly Variable Cost</td>
<td>The monthly costs of all the products in all the regions for all the sectors.</td>
</tr>
<tr>
<td>Monthly Profit</td>
<td>C</td>
<td>Month</td>
<td>Monthly Sales Amount - Monthly Costs</td>
<td>The monthly profit of all the products in all the regions for all the sectors.</td>
</tr>
<tr>
<td>----------------</td>
<td>---</td>
<td>-------</td>
<td>-------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MPR Unit Sales</td>
<td>O</td>
<td>Month-Product-Region</td>
<td>SUM(MSPR Unit Sales)</td>
<td>The number of units sold for all the sectors per month, per product and per region.</td>
</tr>
<tr>
<td>MP Unit Sales</td>
<td>O</td>
<td>Month-Product</td>
<td>SUM(MSP Unit Sales)</td>
<td>The number of units sold for all the sectors and in all the regions per month and per product.</td>
</tr>
<tr>
<td>MP Sales Amount</td>
<td>O</td>
<td>Month-Product</td>
<td>SUM(MSP Sales Amount)</td>
<td>The sales amount for all the sectors and in all the regions per month and per product.</td>
</tr>
<tr>
<td>Total Profit</td>
<td>O</td>
<td></td>
<td>SUM(Monthly Profit)</td>
<td>The total profit.</td>
</tr>
</tbody>
</table>

**T (Type): C = Calculated, O = Output**

SUM in a formula, needs to be implemented with the Excel function


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 image courtesy of EuSpRIG Authors.

ISBN          978-1-905404-56-8