Developing visualisations for spreadsheet formulae: towards increasing the accessibility of science, technology, engineering and maths subjects

Roxanne Leitão and Chris Roast

C3RI, Sheffield Hallam University
{r.leitao, c.r.roast}@shu.ac.uk

Abstract. Spreadsheets are widely used within Science, technology, engineering and maths education. Despite their widespread use, end-user spreadsheet errors are still extremely common and have been shown to have an adverse effect on learning. The textual representation of formulas can be particularly complex and error-prone, exacerbating barriers to dyslexic users. Our work focuses on the design and development of a visual language to graphically represent spreadsheet formulae, with the objective of making them easier to understand than their default textual form. This work contributes to a body of human factors research focused upon spreadsheets.

Introduction

It is estimated that 10% of the UK population is Dyslexic, with 4% being seriously affected (Association 2012). Regarding Dyscalculia, there are an estimated 3 million people in the UK who are affected, with 5% of school and higher education learners challenged by it. Dyslexic learners face a number of challenges related to reading, writing and numeracy skills (Association 2013), and may therefore find significant barriers when looking towards developing their education in numerate disciplines such as science, technology, engineering and mathematics (STEM).

One tool common to establishing confidence and educational progress in STEM subjects is the spreadsheet. Widely used in work and education (Chambers and Scaffidi 2010), at school level and in higher education, the spreadsheet is a core generic tool to understanding in numerate subjects. To help address the difficulties that dyslexics and dyscalculics face in STEM our work aims to aid the understanding of spreadsheets.

Despite the pervasive use of spreadsheets, studies have found that as many as 86% of user spreadsheets contain errors (Panko 2008). Although spreadsheets are commonly believed to be intuitive to use, research with actual users contradicts these claims (Hendry and Green 1994, Panko and Sprague Jr 1998, Panko 2008). Hendry and Greene (1994) found that users experience a wide-range of issues relating to the 1) mapping between spreadsheet layouts and functions and the task at hand, 2) the writing of textual formulae and their progressive debugging, and 3) the visibility of spreadsheets where the underlying formulae are hidden, or all shown, in which case the results are hidden. Panko (2008) classifies spreadsheet modelling errors according to 1) mechanical errors which include mistyping numbers, signs, pointing to the wrong cell, reading a number incorrectly, or selecting the wrong range, 2) logic errors which relate to faulty reasoning, and 3) omission errors where users leave something out of a model.

Although to the best of our knowledge, no work has been done to investigate dyslexia and the difficulties these users might specifically face with textual formulae, a wide pool of research exists regarding issues experienced while reading and writing letters, words and connected text. Some of the issues that dyslexics can experience are 1) difficulty in recognizing alphanumeric symbols and punctuation (Stein and Walsh 1997), 2) difficulties in scanning text without loosing their place (Rello, Kanvinde et al. 2012), 3) reversing letters and numbers, and 4) issues in perceiving individual letters separately from those around them (Gregor, Dickinson et al. 2003). Given the textual nature of the formulae in spreadsheets, we expect that the difficulties experienced with reading text will be similar to those with textual formulae. Additionally, the specification of formulae can be particularly complex as numbers, arithmetic operators, calls to built-in functions, and cell references are all textually represented in a single line of continuous characters. This example of the positive solution for quadratic equations helps illustrate the issue:

=(-B1+SQRT(B1*B1-4*A1*C1))/(2*A1)

Previous work has suggested that dyslexic people may be highly visual thinkers (Tafti, Hameedy et al. 2009, Davis 2010, Association 2012). Our work aims to make spreadsheet formulae easier to understand by transforming their textual representation into a graphical one. We have largely followed the conventions of dataflow diagrams to design a set of visual elements that come together to form diagrams that demonstrate the structure of these formulae. By providing a visual 'scaffold' of geometric forms, colours and connectors, we aim to make the relationship and sequence of formulae elements more evident and immediate, taking advantage of our target-audiences' strengths and skills.

By doing so, we expect to 1) enhance learners' comprehension of spreadsheet formulae, 2) better enable 'debugging' of formulae, and 3) aid the construction of syntactically correct formulae.

For example, if a cell is presenting an unexpected result, the student will need to closely inspect the formula and essentially 'debug' it. Based upon existing knowledge regarding dyslexia, inspecting the formula in textual form might present difficulties arising from mis-reading tokens and reading tokens in the wrong order or sequence. Providing a graphic alternative has the potential to help address such problems and thus enhance users' ability to correctly read and interpret what is going on.

This paper describes an approach to making textual formulae easier to read and understand for dyslexic users. We are developing a visual language that will be integrated into a Google Sheets plugin. This plugin will allow users to call upon a visual representation of any formula within the spreadsheet, which will be displayed as a popout window. Our aim is to develop a prototype that will not only visualise formulae but also enable:

1. the dynamic manipulation of the geometric figures, so users can alter and 'write' formulas by manipulating visual blocks
2. visualising and navigating between referenced formulas by selecting Cell references within the visualised formulae.

The following sections describe related work in visualising spreadsheets, a description of the work we're currently developing, future work and the conclusions.

Related work

Graphical representations, such as flowcharts, and pictorial representations of data structures have long been known to be a significant aid in the understanding of programs and their underlying processes (Myers 1986).

Previous work has investigated the relationship between spreadsheet structures and proposed ways of visualizing them. Igarashi, Mackinlay, Chang et al. (1998) proposed a tool to visualise the dataflow structures associated with individual cells, which they call Local Transient Views, while the Static Global Views and Animated Global Explanations visually present the entire data structure at once. Ballinger, Biddle and Noble (2003) present several spreadsheet visualisation techniques, exploring dataflow and cell dependencies. However, unlike Igarashi, Mackinlay, Chang et al. (1998), their work does not explore visualisations within common spreadsheet tools, but instead creates a tool that is independent of the programs used to create the spreadsheets themselves. Burnett, Atwood, Djang et al. (2001) propose the Forms/3 language exploring the spreadsheet paradigm as a way of 'programming' graphical outputs, including animations and GUI elements (Burnett, Atwood et al. 2001).

Existing version of Microsoft Excel provide debugging tools that highlight a cell when users place their cursor on a cell reference in the formula bar, as well as the ability to trace and visualise precedent and dependent cells within a spreadsheet.

These works and tools consider the wider structure of spreadsheets, and the dependencies between cells. However, they do not explore dataflow and computations within each individual formula.

Cox and Smedley (1994) apply the principles of Prograph, a visual object-oriented programming language, to allow users to view and manipulate formulae within individual cells. Although their approach provides a visual display of these formulae and the processes occurring within them, it relies heavily upon users' previous knowledge of the Prograph programming language.

The following sections describe our approach to visualizing spreadsheet formulae. We have focused upon school age children (15+) who are considering their options for further and higher education. In particular it is this population, which may find that difficulties related to dyslexia and dyscalculia, impair confidence and achievement in STEM-related subjects. Despite this within the context of our empirical work is relatively rare for individuals at this level to be formally diagnosed with specific learning difficulties such as dyslexia.

Spreadsheets STEM education

Spreadsheets are widely used in educational settings, from mathematics, to engineering (Oke 2004), science and technology (Sjøberg 2002). Spreadsheets can provide learners with experience in mathematical modelling, a skill that is transferable to professional settings such as teaching, and engineering. In engineering education, spreadsheets have been found to facilitate the understanding of the essential features of a model instead of overly focussing on the mathematical aspects (Oke 2004), and have been widely used in mechanical, electrical, nuclear, and civil engineering, etc. Spreadsheets and other Information and Communication Technologies (ICTs) have also been established as fundamental tools in mathematics education (Stohl Drier, Harper et al. 2000).

Our work: developing visualisations for spreadsheet formulae

Objectives

Considering the issues that dyslexic users face in reading and correctly interpreting text, and therefore textual formulae, our approach aims to graphically make evident:

- The sequence of numbers/values and operators in a formula.
- The distinction between cell references and other numerical values.
- The distinction between values and operators or functions.
- The sequence of calculations.
- The flow of data throughout calculations.

We expect that this will allow users to interpret and debug a formula without depending on its textual representation to infer the formula's structure, underlying order of operations and cell references.

The structure of our visual languages

In spreadsheets, computations are defined by cells and their formulae. The visualisations being developed are a test bed that allows us to explore which visual characteristics are better or worse understood by our end-users, with the objective of developing a final single visual language that will enhance the construction, comprehension and 'debugging´ of spreadsheet formulae.

We have developed two languages that explore different features to address the same issues. The first explicitly shows the original formula, the order of operations and computations, and the results of each computation. The second provides a higher-level more abstract view, where the order of operations and computations are represented, but the results of computations are not. We will call the first the 'Explicit Visualisation' (EV) and the second 'Dataflow Visualisation' (DV). Our aim is to evaluate both languages with learners, in order to understand which better enhances the comprehension of spreadsheet formulae and engage in a continual development of the languages until we achieve a final optimal result. This result will be a single visual language that has been constructed from the best features of the test bed languages, based upon the results of our evaluations with learners.

Our visualisations are largely based on a data flow metaphor, which presents a set of interconnected components, or nodes with dependencies between each other.

Both EV and DV share some basic graphic characteristics:

- They are read from left-to-right and top-to-bottom.
- They follow the order of operations within a formula.
- Numeric values (Fig.1), cell references (Fig.2), strings (Fig.3), operators (Fig.4) and built-in spreadsheet functions (Fig.5) all have different combinations of shape and colour, making them visually distinct from each other.

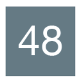


Fig. 1. Numeric values.

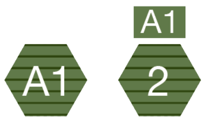


Fig. 2. Cell references in DV (right) and EV (left).

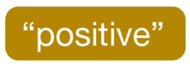


Fig. 3. Strings.

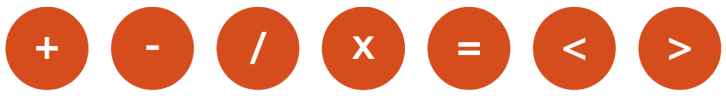


Fig. 4. Operators

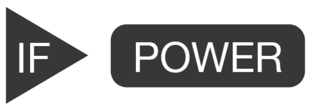


Fig. 5. IF and POWER built-in functions

- To indicate repeated cell references, within the same formula, the same cell reference will always have the same background pattern.
- ‘Wires’ show the flow of data between computations.

Both languages are also based on a set of shared definitions:

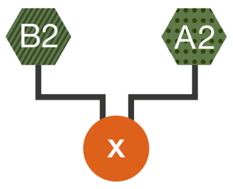
- Calculations are executed when all sources of data are available. For example, given =3+2*4 the addition will only be computed when the result data is available from the multiplication.
- Sources can be constants, variables, or strings.
- Operators are mathematical functions (e.g., add, subtract, multiply and divide).
- Functions are built-in spreadsheet functions, such as IF and POWER.
- Monitors visually show the results of computations.

The following sections describe the specific underlying structures of the EV and DV languages.

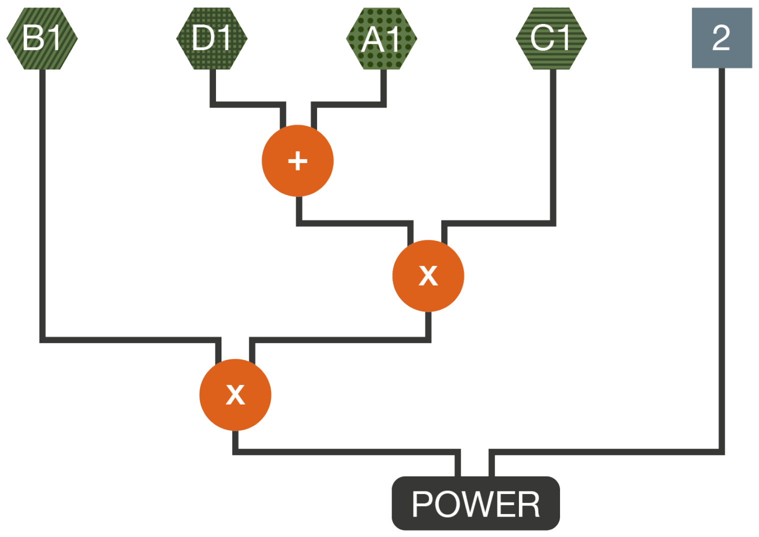
Dataflow Visualisation (DV)

DV focuses on the order of operations and dataflow within a formula. It embodies a higher-level, more abstract view than EV, and has been developed according to the construction rules presented below.

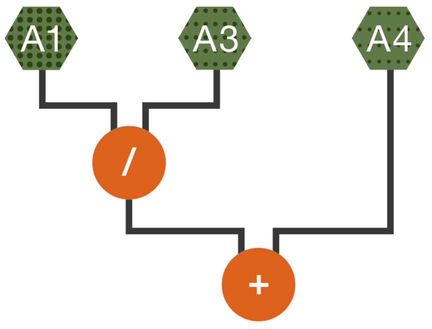
1. The visualized formula is not an exact visual match of the original Excel formula, where Sources constitute the first line of the visualization and lead into operators. Sources have no inlets, only outlets. For example, considering =B2*A2:



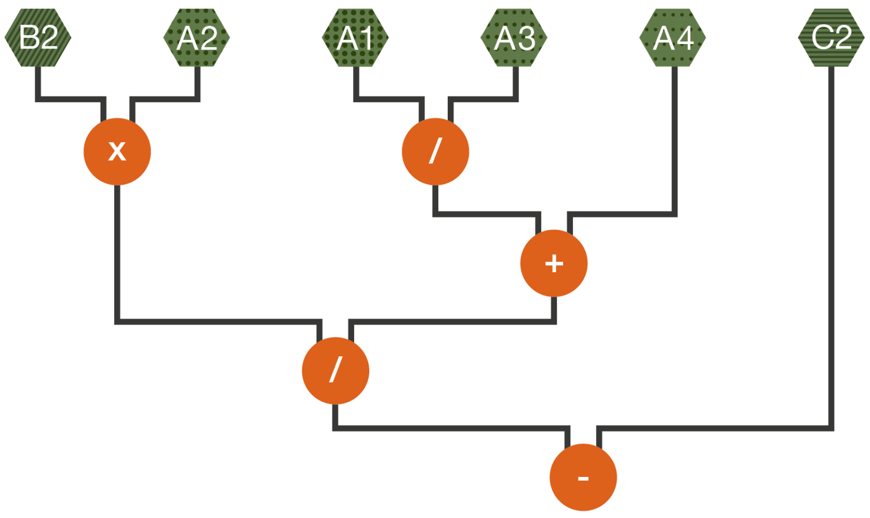
2. Sources are all values, whether they are, or not, contained within a Cell.
3. Cell references are not replaced by their numeric values. Our approach in DV is that the visualisation should demonstrate the underlying processes within a function, independently of any numeric values placed within Cells.
4. Brackets are eliminated, as they can be inferred by the order of operations represented by the visualisations. This allows us to significantly reduce the number of visual elements and simplify the overall graphics.
   For example, for =POWER(B1*((D1+A1)*C1), 2):



5. Sources feed into Operators and Functions, where computation happens ‘behind the scenes’. Operators and Functions have inlets where data comes in, and outlets where the result of a computation flows out of one operator/function, and into the next. For example:



6. Given that DV visualises formulae independently of the numeric values placed within Cells, there are no Monitors to visualise the result of a computation. For example, considering =B2*A2/(A1/A3+A4)-C2:

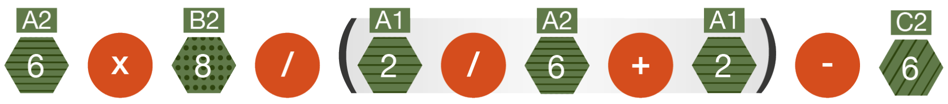


This approach's strength is that it is able to demonstrate the workings of a formula without requiring Cells to be already filled with values. DV makes evident the underlying order of operations within a given function and therefore, allows users to better understand how a particular formula is working and how it might need to be modified to achieve the desired result.

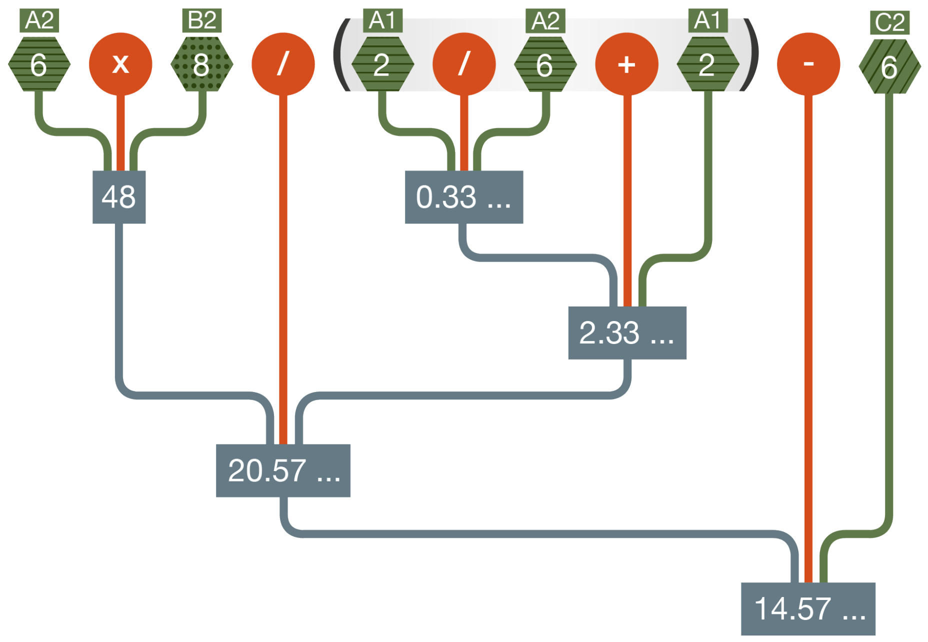
Explicit Visualisation (EV)

Unlike DV, this approach graphically represents each step in the process of executing a formula. EV has been designed and developed according to the rules of construction presented below.

1. The visualised formula is a visual match of the original spreadsheet formula, where it is integrally represented on the first line of the visualisation. This approach provides a more immediate mapping between the textual (default) and visualised versions of the same formula.
   For example, considering: =A2*B2/(A1/A2+A1)-C2:



2. Cell references are replaced by their numeric values. The visualisation maintains the original Cell reference, but places emphasis on its numeric value. This allows us to compute results at every step of the visualisation and provides a more explicit view of the process and results within a formula:



3. Sources and Operators feed into Monitors, where the result of a calculation is shown. Unlike in DV, users can see the results of a computation and view how data is being manipulated throughout the calculation of a formula.

In our understanding, the strength of EV lies in the fact that it visualises every step of the process, and the results of every computation. It is our expectation that such an explicit visualisation will enhance learners understanding of the processes unrolling within a particular formula.

On-going and future work

These visualisations are currently being evaluated with users in a paper-based format. We are evaluating them with learners (15+ years of age) from a range of Colleges within Yorkshire, UK. In each session, we have 1) a control group that performs the tasks without any of the visualisations, 2) a group solving tasks with the aid of the Dataflow Visualisations, and 3) another group with the Explicit Visualisations. Before solving the tasks, users fill in a questionnaire that assesses dyslexic tendencies, allowing us to compare the results of students who show dyslexic traits and those who do not (Vinegrad 1994). Although these evaluations fail to consider the environment (spreadsheets) in which these visualisations will eventually be integrated, they are rather looking at comprehension and adequacy of the visual languages before moving on to implement them in the functioning plugin.

We are also working on the extensibility of the languages, and their capability to deal with more complex built-in spreadsheet functions, such as conditional IF statements. We are particularly addressing issues related to the visual complexity of nested conditional statements, exploring concerns related to 'wires' crossing over each other, and the elevated number of visual elements in a graphic for a single formula.

In parallel, the visualizations are being implemented as a Google Sheets plugin. This plug-in analyses the formula within a Cell and automatically generates the visualisations, allowing users to easily access them while remaining in the spreadsheet environment. Future integration will look at issues such as 1) interactive manipulation of the graphic blocks to write and modify formulae, 2) the navigation between referenced formulae and cells, and 3) matching the colour coding between Google Sheets and the visualisations. Formal end-user evaluations of the prototype will be carried out in June 2014.

Conclusion

The widespread use of spreadsheets in work and education may pose significant barriers to learners with Dyslexia and/or Dyscalculia. The textual form in which spreadsheet formulae are presented may exacerbate Dyslexic symptoms, such as confusing similar letters and symbols, and issues in perceiving the correct order of sequences of characters.

Our approach to graphically visualising formulae and their computations aims to take full advantage of "visual thinking" which has been found to be a strength that is commonly associated with Dyslexia.

We have developed two prototype visual languages, the first – DV – visualises formulae according to a dataflow model and abstracts the formula from numeric values placed within Cells. The second – EV – is more detailed in its approach and visualises each computation within a formula, along with the results of these computations at every step.

Both languages are still a work in progress and are currently being evaluated (in paper form) with students from a range of Colleges within the South Yorkshire, and will be evaluated as a functional Google Sheets plugin in June 2014. Based on the findings from these
evaluations we will further refine our approach and work toward a single visual language that takes advantage of the best characteristics of both DV and EV.

Our aim is to identify the approach that best 1) enhances learners' comprehension of spreadsheet formulae, 2) better enables 'debugging' of formulae, and 3) is more successful in aiding the construction of syntactically correct formulae.

Acknowledgements

The authors would like to acknowledge the Small Business Research Initiative's "Ready steady STEM" programme for supporting this work. The programme is managed by JISC Techdisc and funded by the Department for Business Innovation and Skills and the UK's innovation agency, the Technology Strategy Board. The work is also reliant upon the co-operation of colleges and learners' willingness to participate in studies. The authors would also like to thank reviewers for informative feedback on the first version of this paper and identifying relevant additional research.

References

1. Association, B. D. (2012). Adults and Dyslexia, 40 years on . . . , British Dyslexia Association.
2. Association, T. B. D. (2013). "Handy Hints for Secondary School Teachers." Retrieved 12 May, 2014, from http://www.bdadyslexia.org.uk/about-dyslexia/schools-colleges-and-universities/secondary-hints-and-tips.html.
3. Burnett, M. M., et al. (2001). "Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm." Journal of functional programming **11**(2): 155-206.
4. Chambers, C. and C. Scaffidi (2010). Struggling to excel: A field study of challenges faced by spreadsheet users. Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium on, IEEE.
5. Davis, R. (2010). The gift of dyslexia: why some of the brightest people can't read and how they can learn, Souvenir Press Ltd.
6. Gregor, P., et al. (2003). "SeeWord—a personal word processing environment for dyslexic computer users." British Journal of Educational Technology **34**(3): 341-355.
7. Hendry, D. G. and T. R. G. Green (1994). "Creating, comprehending and explaining spreadsheets: a cognitive interpretation of what discretionary users think of the spreadsheet model." International Journal of Human-Computer Studies **40**(6): 1033-1065.
8. Myers, B. A. (1986). "Visual programming, programming by example, and program visualization: a taxonomy." ACM SIGCHI Bulletin **17**(4): 59-66.
9. Oke, S. A. (2004). "Spreadsheet applications in engineering education: A review." International Journal of Engineering Education **20**(6): 893-901.
10. Panko, R. R. (2008). "Spreadsheet errors: What we know. what we think we can do." arXiv preprint arXiv:0802.3457.
11. Panko, R. R. (2008) What We Know About Spreadsheet Errors. Journal of End User Computing's Special issue on Scaling Up End User Development **10**, 15-21
12. Panko, R. R. and R. H. Sprague Jr (1998). "Hitting the wall: errors in developing and code inspecting asimple'spreadsheet model." Decision Support Systems **22**(4): 337-353.
13. Rello, L., et al. (2012). Layout guidelines for web text and a web service to improve accessibility for dyslexics. Proceedings of the International Cross-Disciplinary Conference on Web Accessibility, ACM.
14. Sjøberg, S. (2002). "Science and technology education: Current challenges and possible solutions." Innovations in science and technology education **8**.
15. Stein, J. and V. Walsh (1997). "To see but not to read; the magnocellular theory of dyslexia." Trends in neurosciences **20**(4): 147-152.
16. Stohl Drier, H., et al. (2000). "Promoting Appropriate Uses of Technology in Mathematics Teacher Preparation." Contemporary Issues in Technology and Teacher Education **1**(1): 66-88.

17. Tafti, M. A., et al. (2009). "Dyslexia, a deficit or a difference: Comparing the creativity and memory skills of dyslexic and nondyslexic students in Iran." Social Behavior and Personality: an international journal **37**(8): 1009-1016.
18. Vinegrad, M. (1994). "A revised adult dyslexia checklist." Educare **48**(1): 21-23.