Visualising Formula Structures to Support Exploratory Modelling

Chris Roast, Roxanne Leitão and Michael Gunning
Culture, Communication and Computing Research Institute, Sheffield Hallam University,
Sheffield, SW1 1WB, United Kingdom
{c.r.roast, r.leitao, m.gunning} @shu.ac.uk

Keywords: *Spreadsheets*, *Visualisation*, *Science Technology Engineering and Maths (STEM).*

Abstract: Visualisation is often presented as a means of simplifying information and helping people understand complex data. In this paper we describe a project designing interactive visualisations to support core learner competencies in the broad area of numeracy. The work builds upon: (i) the observation that while spreadsheets are traditional ICT tools, their widespread use means that they are often introduced as a means of exploring basic mathematical modelling; (ii) a research theme examining the human factors that influence the ease with which formal notations can be understood and applied appropriately. Our paper describes the iterative design and evaluation of a tool to visualise spreadsheets, with the aim of supporting mid-teen learners based on the premise that spreadsheets serve as a gateway tool for supporting learner experimentation and confidence within numerate subjects. This iterative process is informed by background research into notational design, graphic design as well as learner and tutor feedback.

1 INTRODUCTION

Visualisation is often presented as a means of simplifying information and helping people understand complex data. In this paper we describe a project designing interactive visualisations to support core learner competencies in the board area of numeracy. The work builds upon the premise that spreadsheets are a traditional, common and accessible ICT tool that supports learner experimentation of early mathematical modelling. This underpins confidence within numerate subjects. Within this context, we apply concepts from research examining the human factors that influence the ease with which formal notations and tools can be used effectively. This along with user feedback informs the design of a visualisation tool. Our paper describes the iterative design and evaluation of a tool to visualise spreadsheets, with an aim of supporting mid-teen learners in work-based education and/or prior to entering higher education. This iterative development process combines background research, graphic design and learner and tutor feedback to develop a valuable spreadsheet enhancement.

In this paper we describe the design development and evaluation of the tool, and some of the design decisions during development.

1 BACKGROUND

1.1 Why spreadsheets?

The relevance of numeracy as a foundation for educational, academic and professional skills is widely recognised. This is evidenced by the value placed on the development of numeracy skills within science, technology, engineering and maths (STEM) education. In the UK there are various programmes to develop maths skills and skills for employment in engineering and IT. One very common tool enabling the development of numeracy skills and enabling basic powerful numerical calculations 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 for understanding in many numerate subjects. However, at higher academic levels, subjects often focus upon more specialised tools. By contrast, from an employment and employability perspective, the spreadsheet is a widely used tool in most businesses. In order to help address the recognised need for numeracy skills, the research described in this paper has selected the spreadsheet as a key tool that could be enhanced to strengthen basic STEM related education for mid-teens prior to entering higher education or employment.

Although spreadsheets are a relatively familiar and mature tool for general purpose computation, there is evidence of them not being used to their full potential and often containing errors. A skills report identified, that in England, 95% of IT related skills gaps were in the area of spreadsheet skills (Technology Insights 2012, e-skills UK). In addition, research has identified that the likelihood of errors within spreadsheets occurring and going unnoticed is very high (Panko 2008). It has been found that as many as 44% of all end-user spreadsheets contain errors according to Hendry and Green (1993).

Research into addressing this issue has motivated many enhancements. Focusing supporting users in their use of spreadsheet and often encouraging a more disciplined or rigours approach to development and use (including automated assessment of quality features), (Burnett et al. Burnett 2002, Hendry and Green 1994, Hermans and Dig 2014, Panko and Sprague 1998, Sajaniemi 2000).

Despite these points, spreadsheets are widely available and there appears to be a consistent demand for, and interest, in them (Campbell-Kelly 2007). We proposed that this is, in part, due to their initial ease of use and responsiveness, paired with an information infra-structure that does not support self-documentation and modifications. Their responsiveness means their users quickly become embedded in 'solutions' that can become hard to manage. 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 use a limited number of functions with very few nested uses (Sajaniemi and Pekkanen 1988).

Research into complex interactive systems and user empowerment (Blackwell, et al. 2001) has provided a range of dimensions that capture some of those core characteristics. The notion of 'premature commitment' describes systems that, in effect, enforce unwanted constraints upon users before they know if they want such constraints. In terms of traditional spreadsheets, this corresponds to supporting users developing simple numeric models but in doing so committing them to a tool not well suited to model documentation. Hence, a spreadsheet 'solution' may be developed but the result solution in itself is not easily documented or explained within the sheet. Complementing this is the notion of 'viscosity' (a resistance to change). Once used substantively a sheet's information structure is one that allows input values to be changed, but more extensive changes are complex. The premature commitment and viscosity combined, result in users becoming locked-in to early solutions and the subsequent attempt to refine/improve them. These points reveal themselves in anecdotal evidence of businesses and professionals who have working spreadsheets, which are rarely refined or modified because of the risk of 'breaking' a working 'solution'. The work of Hermans and Dig (2014) in supporting re-factoring within spreadsheets, illustrates one way of reducing the emergent complexity of spreadsheet modelling.

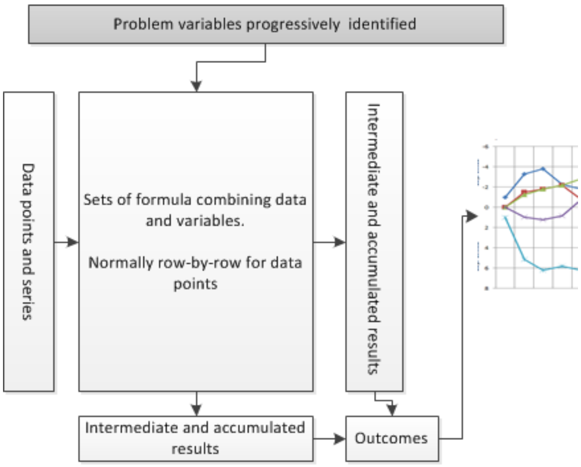
Problem variables progressively identified
Data points and series
Sets of formula combining data and variables.
Normally row-by-row for data points
Intermediate and accumulated results
Intermediate and accumulated results
Outcomes

Figure 1. Schematic of spreadsheet base modelling

1.2 Educational uses

The same characterisation is evidenced in the assessed educational use of spreadsheets. As part of our preliminary research, teaching and assessment materials related to spreadsheet skills were reviewed. This, in combination with conversations with tutors, revealed that spreadsheet skills were focused largely upon developing sheets to reflect a given model, structure and layout. For example, a weekly budgeting spreadsheet task will provide the appropriate structure in terms of the names of columns and rows. (For examples, see "City and Guilds" spreadsheet qualification, samples available at: www.cityandguilds.com). Within the same assessment of spreadsheet skills, non-numeric skills of layout and presentation are also assessed. This focus upon prescribed solutions left little opportunity for exploratory modelling or problem solving that could depart from given examples.

1.3 Exploration and development

When used in model development, the 'locking-in' effect of spreadsheets influences how they are used. In simple terms one can view a spreadsheet model as structured in figure 1. There are variables (at the top) used in a formula (central rectangle) applied to input data (on the left). Model outputs are accumulated in rows and/or columns feed into summary statistics, results, reports or graphs (on the right hand side). While this specific layout might be adopted, its structural features are very common in spreadsheets.

The iterative development of a model is in general a process of refinement, starting with a simple model and reifying details as the need for them becomes evident. In terms of our figure, reification tends to add more subtle variables (at the top) and associated computations (in the central rectangle). However, structurally, the model output is already present (on the right hand side). Hence, as opposed to re-designing a sheet's structure with each new variable introduced, the formulas used rise in complexity to accommodate new variables.

While not all examples of iterative modelling follow this account, we believe it is realistic and captures the inherent complexities involved. Note that this account is the antithesis of skills based training where the variables, layout and requirements are all prescribed as part of the problem.

1.4 Example

We illustrate our account of iterative modelling with a work based training example set in the domain of construction. It concerns the cost of tiling an irregular floor shape - in this case an "L" shaped room. The floor area can be treated as three adjoined rectangles (2m x 3m, 3m x 3m and 3m x 1m). So, assuming the price per 1m x 1m tile is given as, say, 4.99 euros the total cost of the tiles would be:

=4.99*(2*3+3*3+3*1)

Educationally, the progressive modelling could include recognising that the price of a tile is a variable that can be kept in cell for that purpose (say, A2). In which case the formula would become:

=A2*(2*3+3*3+3*1)

Similarly, these tiles are a specific size. Other tiles may be of a different size. In that case another variable, the area of a tile may be kept in, say, A3 and formula updated to:

=A2*(2*3+3*3+3*1)/A3

A delivery charge can be modelled too as a fixed amount added to the total:

=A4+A2*(2*3+3*3+3*1)/A3

While this is clearly a simple numeric problem it illustrates the rise in complexity. If the model goes on to account for, say, free delivery with orders over a certain amount, then the formula becomes more complex.

= (A2*(2*3+3*3+3*1)/A3) +
IF((A2*(2*3+3*3+3*1)/A3) > A5, 0, A4)

Other factors can be introduced, such as, the need to over order to account for tiles that may be broken when cut.

Different approaches to the examples just illustrated may help avoid complexity. We propose that exploratory model development in its nature will be incremental and will not necessarily adopt the best design for the needs that emerge during its development. Hence, although the illustrated complexity can be easily avoided, it is only done so with the benefit of hindsight. And therefore the complexity within the formula can be viewed as symptomatic of model development within spreadsheets.

1.5 Visualisation

Given our concern with the complexity of formula expressions being represented in a single line, we turned to research aimed at simplifying notations using visualisation. 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). However, it is of interest to note that in visual computational language the empirical evidence of their compelling and appealing character is limited, as is their educational utility (Sorva, et al. 2013).

Previous work has investigated the relationship between spreadsheet structures and proposed ways of presenting and visualizing them. Saariluoma and Sajaniemi (1991) showed that surface structures congruent with the resulting computation were learnt more easily. Igarashi, et al. (1998) propose 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, et al. (2003) present several spreadsheet visualisation techniques, exploring dataflow and cell dependencies. However, unlike Igarashi, 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, et al. (2001) propose the Forms/3 language exploring the spreadsheet paradigm as a way of 'programming' graphical outputs, including animations and GUI elements. It is of interest that these works only consider the wider structure of spreadsheets, and the dependencies between cells, but 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 individual formulae and the processes occurring within them, it relies heavily upon users' previous knowledge of the Prograph programming language.

More generally, the psychology of programming has studied both textual and visual notations extensively, with a view to examining how programming languages (and notations) are perceived and understood. One issue arising from this work is that notations and visualisations are rarely empirically assessed formally for their impact or value, and as such, while they are often compelling they do not necessarily yield the envisaged benefits.

1.6 The Idea

Our visualisation tool is motivated by the fact that the notation for exploring models and modelling in spreadsheets is less than ideal. Given that the spreadsheet is a widely adopted tool, an improvement to enhance modelling is to provide a visualisation of the expressions used within a model.

While there are various enhancements to support users, none appear to have addressed the fact that the formulas language is computationally powerful but contracted onto a single line. It is this complexity of language presentation that can complicate its effective use. Our enhanced spreadsheet tool employs a visual language that graphically represents spreadsheet formulae.

With many numeracy support visualisations the tools concentrate on presenting the mapping between input values (to a formula) and the resulting value (as shown schematically on the right hand side in figure 1). But in our case we are interested in visualising the formula that defines such a mapping. This related back to work in the psychology of programming in which the formula can in effect be viewed as notation representing computations that is expressed in the constrained manner of a single line of text. For example, in our modelling illustration the example goes on to include a conditional expression which is structurally complex but still contained in a single line when input to spreadsheet.

In developing such a visualisation, the objective is to reduce the complexity of understanding formulae and thus support user confidence with complex mathematical expressions and the activity of exploratory modelling.

2. DEVELOPING VISUALISATIONS

A visualisation offers a 'scaffold' of geometric forms, colours and connectors that take advantage of human perceptual ability to recognise patterns and associations - and support "visual thinking". We aim to make the relationship and sequence of formulae elements more evident and immediate using such techniques. Examples of how this might reveal itself include: learners recognising when a formula result is not fit for its intended purpose; identifying where an error is in a formula, or; identifying what modifications are necessary to ensure a formula does work. For example, if a cell is computing an unexpected result, the learner will need to closely inspect the formula and essentially 'debug' it. With good visual 'scaffolding', any problem in the formula should be more easily identified.

Structures for the visual language were initially inspired by graphical notation used in programming and taking account of the known difficulties learners face. Specifically with textual formulae, issues can arise with reading and discriminating terms and sub-expressions, their order and structure.

2.1 The design of the proposed visual languages

The visualisations were developed on paper to allow the authors, tutors and learners to explore and provide rapid feedback on which visual characteristics are appropriate and of value.

Initially visual languages were developed based upon a rationale drawn from visual design practice, as well as learning scenarios and educational employing spreadsheets (for example see: Gretton and Challis, 2008). The influencing principles that drove this initial design phase included:

- 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*X1*X1+B1*X1+C1. The repeated use of X1 is important for understanding what is expressed. A more complex abstraction is the repeated use of the expression (2*3+3*3+3*1), in the simple example above.
- Evidencing computation. In contrast to abstractions, there is the value of evidencing the specific values used in determining the resulting value of a formula. Hence, when a formula such as, =2+3*4 produces the result 14, it is important to understand that arises form 2+12 and the 12 arises from 3*4.
- 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.

Given these principles, two visualisation approaches were identified and developed, termed ‘Explicit Visualisation’ (EV) and ‘Dataflow Visualisation’ (DV). Both were largely based on a data flow metaphor, which presents a set of interconnected components, or nodes with directed dependencies between each other. The interconnectedness was used to represent the flow of results between operations within a formula (normally defined by its syntax). The nodes represented inputs to the formula and points of computation. Both also presumed a top-down reading with the starting expression at the top and the outcome at the bottom. Categories of node included: numeric values, cell references, strings, operators and built-in spreadsheet functions. All such token types are given a distinct visual identity.

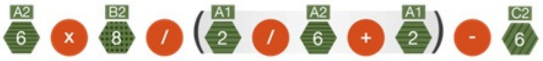

Figure 2. An initial tokenised visualisation of a formula =A2xB2/(A1/A2+A1)-C2 as graphical tokens.

2.2 Dataflow Visualisations

The Dataflow Visualisation (DV) focuses on the order of operations and flow within a formula. It embodies an abstract view and was developed according to the following rules:

- Inputs to the visualisation are all original values, whether they are, or not, they are specific values or cell references.
- Cell references are not replaced by their numeric values, since it is presence of a value (and not the literal amount) which is important to the model.
- The visualized formula has values flowing down and into functions.
- The outputs from functions and operations consistently flow down any functions that use them as inputs.
- Brackets are eliminated, as scoping can be inferred by the order of operations represented by the visualisations. This allows us to significantly reduce the number of visual elements and thus support visual simplicity and scalability.

DV emphasises formula structure, and minimises numeric details. The rationale behind this is that a 'wrong' formula is because of it not linking its components correctly. Hence, displaying the structure in this way will help identify important errors or slips. The merit of this approach is that it is able to demonstrate the order of operations in a formula as an entity independent of specific cell values. It captures what is being proposed as a solution and not the details of any specific instance of the solution. For an example of DV see figure 3.

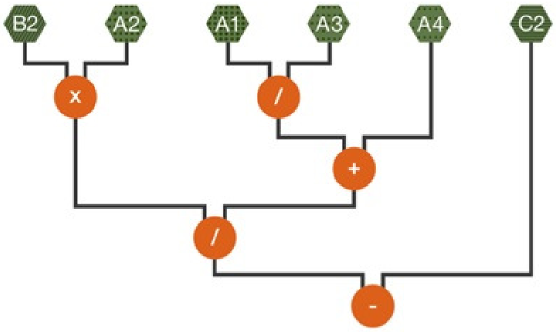

Figure 3. A DV visualisation of the expression in figure 2..

In terms of our initial principles, evidencing visual mapping is weak since the formula as typed in the spreadsheet cannot be immediately obvious in the visualisation. In addition, the evidencing of computation is relatively weak since the flow is shown but not the effect of individual operations or functions.

2.3 Explicit Visualisation

Unlike DV, the Explicit Visualisation (EV) approach graphically represented each computation step in processing a formula, and was developed according to the rules:

– The visualised formula is a direct match to the original spreadsheet formula. Thus supporting the concept of visual mapping.
– Cell references include the numeric values in those cells. While this detracts from the visual mapping it does support evidencing of basic computations.
– Values, functions and operators flow down into additional nodes ("monitors") which themselves show the result of the associated operator or function applied to its arguments. This further supports evidencing computation.

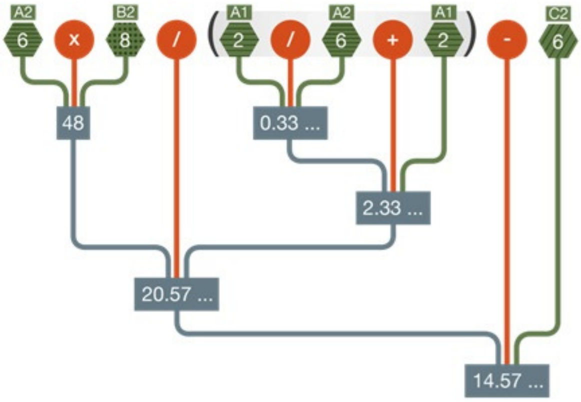

Figure 4. An EV visualisation of the expression in figure 2.

2.4 Preliminary Evaluation

Examples of these two concepts for visualisation were initially illustrated for a range of formula based on teaching materials and examples identified by the authors. Informal feedback was sought from tutors, which at this stage did not reveal any specific preference between the two.

2.5 Conditionals, abstraction and concrete

One of the issues with complexity, illustrated with our simple example, is the use of conditional functions (such as "IF"). Interestingly, conditionals represent visualisation challenges for a variety of reasons that highlight tensions between the proposed principles.

The most common conditional is the "IF" function. Briefly: "IF" takes three arguments, and behaves as follows: if the first argument (the CONDITION) is evaluated to TRUE, then the second argument (the THEN-PART) is evaluated and the result is returned as the value of the "IF" expression. Otherwise, the third argument (the ELSE-PART) is evaluated and that value is returned.

Unlike other functions, conditionals have the characteristic that they embody more than one computational behaviour. In short, for any instance, they will only compute the THEN-PART or the ELSE-PART (and never both). This characteristic is exaggerated with nested IF's – were there to be a conditional with a conditional in both its sub-expressions. Then for any single use of that formula, there would be three of the four sub-formulae expressed but never computed.

This exposes the difficulties of visualising conditionals in EV and DV. Treating a conditional as though any other function, means that both visualisations would be misleading. In the case of EV, a non-computed ELSE-PART would need to be shown and it would be necessary to indicate that its value is not computed. However, the same formula in the contrary case would show the THEN-PART not computed (see figure 5). So in simple terms, there would not be single visualisation for a formula but one that would change depending upon the input. This dynamism is at odds with the idea of representing a formula that itself does not change.

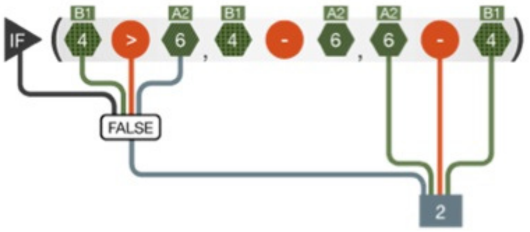

Figure 5. A simplistic EV visualisation of a conditional expression, in which the THEN-PART is present but not used.

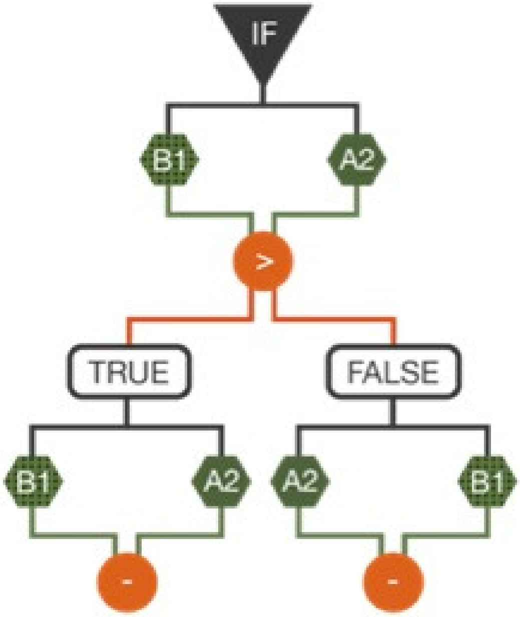

Figure 6. A DV visualisation of a conditional expression, illustrating the separation of flow into two alternate computations.

By contrast using the DV visualisation approach, conditionals appear complex since the flow for each part is drawn. Alternatively, the visualisation requires an additional mechanism to indicate that the flow from both the THEN-PART and ELSE-PART has the same output, although they are never both used together. See figure 6 for an illustration of a DV style conditional.

3. EVALUATION

In order to understand whether our initial designs served as effective visualisations for the intended tutors and learners we conducted a variety of user studies. In keeping with iterative design principles each study considered both assessing the appropriateness of the visualisations and also gathering formative feedback. The primary target users were college based learners developing skill for higher education entry and improved employability.

3.1 Initial evaluations

Initial evaluations were directed at assessing the comprehension of the visualised formula with the aim of comparing textual formula, DV and EV (Leitão and Roast 2014). This revealed one of the methodological challenges: for our target users despite using and learning about spreadsheets. Some of the complexities of formulae of interest in our research did not match those learners and tutors naturally encounter. In fact, the issues with modelling that we discussed earlier in this paper were rarely apparent since teaching materials did not encourage exploratory modelling per-se. In addition, approaching differing cohorts of learners at similar stages in their skills development was impossible and therefore overall familiarity with using spreadsheets varied greatly.

Initially the two styles of visualisation DV and EV were assessed with between groups with task completion being observed and along with some post task interviews. For individual classes, the approach to user engagement varied in response to the readiness of technology, users' academic levels, and support of their tutor. This included:

- Paper based materials with multiple choice formulae comprehensions tasks. This included: (i) a formula and a number of possible results; (ii) a mini-problem statement and a number of formulae (one of which is correct for the problem).
- Prototype implementations of DV and EV were developed as extensions to an existing spreadsheet package. With the prototype the tasks were to construct or modify a formula in the spreadsheet to solve a set of mini-problems. An example mini-problem is: *"Does =A1*(A1*A1) calculate cell A1 to the power of 4? If not, correct the formula."*

Performance measures assessed were the correctness of the choices made by the participants and where possible time to complete task. Subjects varied in age profile, familiarity with spreadsheets and readiness to engage with the tasks. Quantitative results from these studies (summarised in table 1) suffered from the wide variety of abilities, levels and topics that subject were studying. However, the fact that learners engaged with the tasks set and worked effectively with the visualisations did show that the visualisation were of some positive value and in no cases was there evidence of them impairing or disrupting the tasks set.

During the same period educational experts were consulted regarding the tool and the visualisations and encouraged to critic the approach takes. Feedback from this process and interviews with subjects were valuable in helping distinguish between DV and EV.

Table 1: Summary of initial evaluation studies and outcomes

Study	Population and context	Outcome
Paper based study with 8 spreadsheet comprehension questions.	44 work-based learners studying Electrical engineering, Engineering and Maths at NVQ level 3.	Visualisation showed a positive effect. Average score was 55\% with a visualisation, and 51\% without. (Not significant)
Interactive prototype based study and qualitative inter-views	14 full-time learners were given a spreadsheet "refresher" and then completed 37 spreadsheet formula questions	Visualisation conditions showed a positive effect over the no-visualisation condition. Average score 73\% with a visualisation, and 67\% without
Interviews and demonstration with experts.	Three STEM educators, three STEM education researchers and five support staff	

Outcomes from this initial evaluation stage were that the explicit EV style was of more value. The support for evidence of computation and mapping back to the spreadsheet formula counted highly for educational experts, tutors and learners. The fact that EV was less visually compact was not raised as a significant concern.

Feedback on visualisation conditionals was not so straight forward partly due to learners and tutors being less familiar with using conditionals. Hence the outcome was to review the visualisation taking into account the general points arising from the evaluation of DV and EV.

3.2 Design, development and evaluation

The outcome from the initial development and evaluation resulted in a need to develop a more robust prototype tool suitable for broader scale and assessment. This development process proceeded hand-in-hand with the commitment to the EV visualisation style. The specific technical development focused upon was to move to a platform that interacted with the most widely used spreadsheet, specifically MS Excel. This naturally opened further evaluation opportunities because of its widespread availability and use (Campbell-Kelly, 2007).

The visualisation developments focused upon developing an EV-based visualisation of conditionals that aimed to ensure a good mapping with the formula while indicating the dynamic character of conditional behaviour. The resulting visualisation is illustrated in figure 7. In this design the THEN-PART and the ELSE-PART are shown, but in addition, the un-used part is faded to indicate it is not in use and the conditional expression is shown to be "selecting" the relevant part.

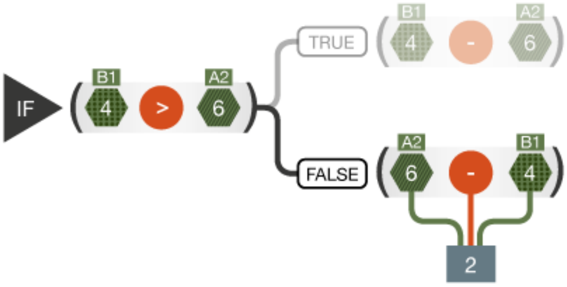

Figure 6. The revised EV style visualisation of conditionals.

The value of having the prototype tool working with the most common spreadsheet benefited the next evaluation activity. In this case cohorts of learners studying functional skills in various areas were approached and introduced to the tool. Where possible, this introduction mapped to their existing use of spreadsheets, such as their current topics of study or tutorial work.

The initial evaluation was taken to have demonstrated that our general approach visualisation was valid. However, evidencing performance improvements attributable to the visualisation was judged to be too methodologically complex, for the reasons described earlier. As a consequence the second phase of evaluation focused upon whether the prototype technology was recognised by users to be of potential value. It was assumed that this judgement could be made by users, even if the tool was not used comprehensively in the sessions when it was introduced to them. For this reason the Technology Adoption Model (Davis, et al. 1989) was used to develop a series of questions for both learners and tutors.

Over 15 learners were introduced to the tool during a taught element of work related courses. They subsequently attempted specified spreadsheet tasks at a level matching their normal class. The tasks lasted for between 30 and 60 minutes, during which they worked with the tool running with MS Excel. Responses were on the whole positively in terms concurring with positive statements about the experience of using the tool (on a Likert scale: 1= 'Strongly Disagree' through to 7='Strongly Agree'). The most positive responses were with respect to the visualisation (6.17) and responsiveness of the system (6.00). The least positive response (3.92) was just below the median of 4.00, and concerned whether learners viewed the tool as helping them work more efficiently.

As with the initial evaluation, the results are on the whole positive for a small number of subjects. Qualitative feedback supports this view, learners' quotes include: "It would help me a lot with other formulas", "You can see the values and how they are worked out, that's great." and "It would help anyone willing to learn about spreadsheets".

In addition supportive qualitative evidence came from tutors engaged during the sessions: *"I am sure that it could add value to the teaching of mathematics."*; *"I think it would be very helpful"*; *"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."*

Both tutor and learner feedback also supported identifying additional visualisation details. One example of this was the need in complex cases to indicate the flow of data more explicitly, as well as the final result node. Features such as these were introduced to the next iteration of the tools which currently under going evaluation. It is useful that both these points can be attributed back to the visualisation design concept of Evidencing computation.

4. DISCUSSION

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 population we are targeting. For example, trainee tutor commented:

"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."

An expert in maths education research commented:

"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."

5. FUTURE WORK AND CONCLUSIONS

We have reported the iterative development of the visualisation tool in terms of: preliminary design, initial development and evaluation and then the evaluation of prototype inter-operating with MS Excel. This is part of an on going process of evaluation and refinement, with learner and tutor feedback informing future enhancements.

The widespread use of spreadsheets in work and education (Chambers and Scaffidi, 2010) may pose significant barriers to learners. Hence their potential benefit as an ease to use tool for exploring STEM related topics is limited. Our approach to visualising formulae that can quickly become complex, offers a means of helping learners work more effectively with spreadsheets. Our evidence of the benefit of this approach is positive but it requires further investigation. Similarity the principles that underpin the effectiveness of the visualisation language require further development, specifically to address some of the complex structures found in spreadsheets.

The long-term benefit of making spreadsheets more usable is one that could impact upon academic progress for individuals as well as general numeracy skills. The value of the resulting improved ability aligns to national and international educational objectives regarding skills and employability.

ACKNOWLEDGEMENTS

We are indebted to local colleges' staff and students that were willing to support and participate in the studies reported. This work is supported in part by: UK Higher Education Academy and The Small Business Research Initiative (part of Innovate UK).

REFERENCES

Ballinger, D., Biddle, R. and Noble, J., 2003. Spreadsheet structure inspection using low level access and visualisation. In Proc. Fourth Australasian User Interface Conference (AUIC2003). Biddle, R. and Thomas, B., Eds. ACS. 91-94.

Blackwell, A.F., Britton, C., Cox, A. Green, T.R.G., Gurr, C.A., Kadoda, G.F., Kutar, M., Loomes, M., Nehaniv, C.L., Petre, M., Roast, C., Roes, C., Wong, A. and Young, R.M., 2001. Cognitive Dimensions of Notations: Design tools for cognitive technology. In M. Beynon, C.L. Nehaniv, and K. Dautenhahn (Eds.) Cognitive Technology 2001 (LNAI 2117). Springer-Verlag, pp. 325-341.

Burnett, M., Atwood, J., Djang, R., Gottfried, H., Reichwein, J. and Yang, S., 2001. Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm. Journal of functional programming 11(2): 155-206.

Burnett, M., Sheretov, A., Ren, A. and Rothermel, G., 2002. Testing homogeneous spreadsheet grids with the "what you see is what you test" methodology," IEEE Trans. Softw. Eng., 28(6) pp. 576-594. http://dx.doi.org/10.1109/TSE.2002.1010060

Campbell-Kelly, M.,2007. Number crunching without programming: The evolution of spreadsheet usability. IEEE Annals Of The History Of Computing, 29(3), pp.6-19, ISSN: 1058-6180

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.

Cox, P. T. and T. J. Smedley (1994). Using visual programming to extend the power of spreadsheet. Proceedings of the workshop on Advanced visual interfaces, ACM.

Davis, F.D., Bagozzi, R.P. and Warshaw P.R., 1989. User acceptance of computer technology: A comparison of two theoretical models. Management Science, 35 (8) pp. 982–1003

Gretton, H. and Challis, N., 2008. "Fundamental Engineering Mathematics; A Student Friendly Workbook" Woodhead Publishing, 2008, ISBN-13: 978-1898563655.

Hendry, D.G. and Green, T.R.G., 1994. "Creating, comprehending and explaining spreadsheets: a cognitive interpretation of what discretionary users think of the spreadsheet model." in International Journal of Human-Computer Studies, 40(6), 1033-1065, DOI=10.1006/ijhc.1994.1047

Hermans, F. and Dig, D., 2014. BumbleBee: a refactoring environment for spreadsheet formulas. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014). ACM, New York, NY, USA, 747-750. DOI=http://dx.doi.org/10.1145/2635868.2661673

Igarashi, T., Mackinlay, J.D., Chang, B-W and Zellweger, P.T., 1998. Fluid Visualization of Spreadsheet Structures, Proceedings of the IEEE Symposium on Visual Languages, p.118.

Leitão,R. and Roast, C., 2014. Developing visualisations for spreadsheet formulae: towards increasing the accessibility of science, technology, engineering and maths subjects. In: 9th Workshop on Mathematical User Interfaces, Coimbra, Portugal, 10 July 2014.

Myers, B. A. (1986). Visual programming, programming by example, and program visualization: a taxonomy. ACM SIGCHI Bulletin 17(4): 59-66.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

Panko, R. R. and Sprague Jr., R. H., 1998. "Hitting the wall: errors in developing and code inspecting a simple spreadsheet model" Decision Support Systems 22(4): 337-353.

Peyton Jones, S., Blackwell, A. and Burnett, M., 2003. A user-centred approach to functions in Excel. In Proceedings of the eighth ACM SIGPLAN international conference on Functional programming (ICFP '03). ACM, New York, NY, USA, 165-176. DOI=10.1145/944705.944721

Rello, L., Kanvinde, G. and Baeza-Yates, R., 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.

Saariluoma, P., Sajaniemi, J.,1991. Extracting Implicit Tree Structures in Spreadsheet Calculation. Ergonomics 34(8), 1027-1046.

Sajaniemi, J., 2000. Modeling Spreadsheet Audit: A Rigorous Approach to Automatic Visualization, Journal of Visual Languages and Computing. 11, 49-82. doi:10.1006/jvlc.1999.0142

Sorva, J., Karavirta, V. and Malmi, L., 2013. A Review of Generic Program Visualization Systems for Introductory Programming Education. *Trans. Comput. Educ.* 13, 4. DOI=http://dx.doi.org/10.1145/2490822

Stein, J. and V. Walsh, 1997. "To see but not to read; the magnocellular theory of dyslexia." Trends in neurosciences 20(4): 147-152.

Technology Insights 2012, e-skills UK (Available at http://www.e-skills.com).