Dependence Tracing Techniques for Spreadsheets: An Investigation

Sohon Roy
Delft University of Technology
S.Roy-1@tudelft.nl

Felienne Hermans
Delft University of Technology
F.F.J.Hermans@tudelft.nl

ABSTRACT

Spreadsheet cells contain data but also may contain formulas that refer to data from other cells, perform operations on them, and render the results directly to show it to the user. In order to understand the structure of spreadsheets, one needs to understand the formulas that control cell-to-cell dataflow. Understanding this cell-to-cell inter-relation or **dependence tracing** is easier done in visual manners and therefore quite a few techniques have been proposed over the years. This paper aims to report the results of an investigative study of such techniques. The study is a first step of an attempt to evaluate the relevance of these techniques from the point of view of their benefits and effectiveness in the context of real world spreadsheet users. Results obtained from such a study will have the potential for motivating the conception of newer and better techniques, in case it is found that the need for them is still not fully catered.

Categories and Subject Descriptors

H.4.1 [**Information Systems Applications**]: Office Automation – *Spreadsheets*

General Terms

Design, Experimentation, Human Factors

Keywords

End-user computing, Dependence tracing, Spreadsheet visualizations

1. INTRODUCTION

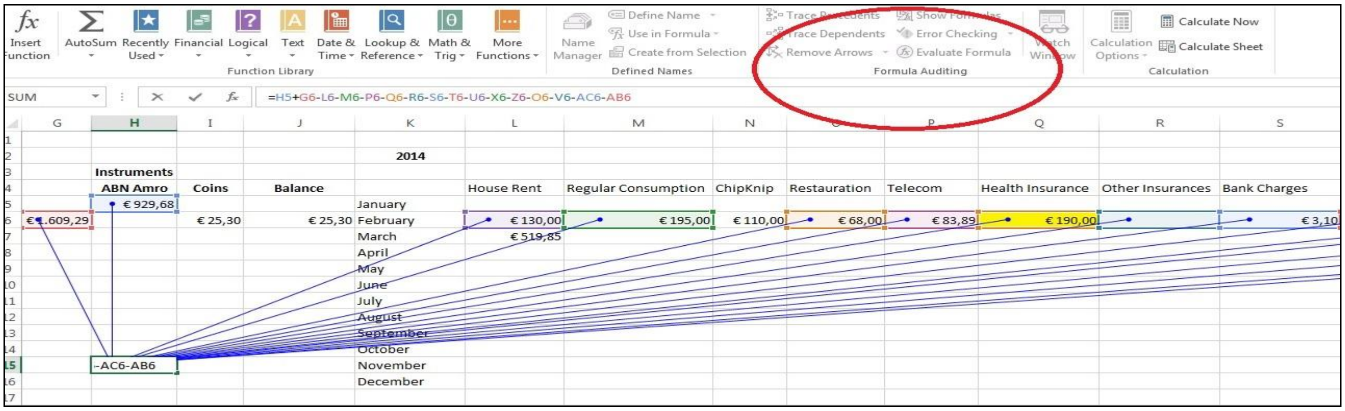
1.1 Background

Spreadsheets offer the end-users an interface that is incomparable in its simplicity and flexibility. However it is mostly beneficial for performing rapid calculations and quick simple analyses. This interface is not helpful at all in understanding the design logic behind a spreadsheet, especially the type of understanding that is necessary in order to make modifications to existing spreadsheets. Modification becomes harder in the case where it is done by a user different from the creator. This situation is fairly common in the industry as the average lifespan of spreadsheets have been found to be 5 years [3] which can often prove too long for the possibility that the original creator will be always available whenever some modifications are required. When understanding spreadsheets, the visual structure that is perceived from just looking at the cells is referred to as spreadsheet *surface structure* [2] comparable to the anatomical structure of the human body. However calculations are performed based on formulas and the formulas connect the cells to form another kind of structure called the *computational/deep structure* that is comparable to the nervous system of the human body. These two structures are often not similar and at times can be radically different. The deep structure reflects the data flow in the spreadsheet and is basically the cell-to-cell inter-dependence. In the understanding of a spreadsheet, this cell-to-cell inter-dependence plays a key role. Without having a clear idea of cell-to-cell inter-dependence, the modification of a fairly complex spreadsheet becomes impossible without ample risks of errors. It is considerably easier to understand for a user if the referred cell(s) in a formula are indicated in an enhanced manner with visualization techniques, instead of having to manually inspect each and every formula and trying to locate the exact cell(s) that it is referring to. Therefore a number of visualization techniques have been proposed in various research papers over the years. However there are some questions about these techniques that still need to be explored and they form the core of our investigation. They are listed in subsection 1.3.

1.2 Motivation

It is our opinion that visualization based dependence tracing techniques, as found in research literature, are not making across to the industry of spreadsheet users. In a study conducted by Hermans *et al*. [3] with spreadsheet users working in a large Dutch financial company, it was found that "*the most important information needs of professional spreadsheet users concern the structure of the formula dependencies*". This study also mentions the feeling of inadequacy felt by the users while using the only available dependence tracing tool within their reach the Excel Audit toolbar [Fig.1]; a feature of MS (Microsoft) Excel which is by far the most popular [1] spreadsheet application in the market. This feature demonstrates cell inter-dependencies with an overlaid dependency graph over a worksheet, with graph edges shown as blue arrows; the edges however are generated on a cell-by-cell basis which has to be interactively activated by the user. Findings of another informal survey conducted in October 2013 at the offices of the UK based financial modeling company F1F9[1] also point repeatedly at the direction of the sense of inadequacy the spreadsheet users are suffering from when depending heavily on this Excel Audit tracing feature. These findings lead us to the question why there are no better tools available to spreadsheet users? Nevertheless, as will be shown in this paper, there is considerable amount of research already done on this topic. This gives rise to the question why implementations of such research are not making it to the industry? Only a handful of highly

[1] F1F9: A financial modeling company http://www.f1f9.com/



=H5+G6-L6-M6-P6-Q6-R6-S6-T6-U6-X6-Z6-O6-V6-AC6-AB6

	G	H	I	J	K	L	M	N	O	P	Q	R	S
1													
2					2014								
3		Instruments											
4		ABN Amro	Coins	Balance		House Rent	Regular Consumption	ChipKnip	Restauration	Telecom	Health Insurance	Other Insurances	Bank Charges
5		€ 929,68			January								
6	€1.609,29		€ 25,30	€ 25,30	February	€ 130,00	€ 195,00	€ 110,00	€ 68,00	€ 83,89	€ 190,00		€ 3,10
7					March	€ 519,85							
8					April								
9					May								
10					June								
11					July								
12					August								
13					September								
14					October								
15		-AC6-AB6			November								
16					December								
17													

Figure 1: Tracing dependents with Excel Audit toolbar: blue connecting arrows and coloring of precedent cells

customized tools are existing today and that also are mostly used internally by organizations; they are not compared against each other based on any well accepted metrics framework. Their efficacy in actually helping in the end-user experience is not measured. Our investigation is therefore dedicated to evaluating the effectiveness of these proposed techniques in the context of real world spreadsheet users. Such an evaluation might also open up specific areas in which to improve upon or come up with newer techniques that are not only innovative but viable in terms of practically realizable implementations that can be adopted by spreadsheet users in the industry.

1.3 Hypothesis and Research Questions

Hypothesis: Proposals thus far described and demonstrated in research literature about visualization based techniques for spreadsheet dependence tracing have not adequately made it across to the industry in forms of reliable, user-friendly, wide-spread, multi-platform, and standardized software tools of both stand-alone and plug-in type.

On the basis of the premise established in Subsection 1.2 and the above mentioned hypothesis, we arrive at the following three research questions.

Research Questions:

R1. **Why the proposals thus far described and demonstrated in research literature have not reached the industry as implementations?**

An attempt to study what may be the key causes of the perceived bottleneck between research and industrial implementations.

R2. **Is there any well-accepted metrics framework with which such implementations as above (R1) can be compared to each other?**

If and when implementations are made available to the industry, it is necessary to measure their usefulness in actually helping the end-user computing experience. If such a framework is not there, then it can be devised and made into an industrial standard.

R3. **Is there any well-defined opportunity for improvement in the dependence tracing context**?

Improvement not just from the aspect of innovativeness of idea but also from the angle of how well the idea can be translated into a user-friendly and reliable implementation; the efficacy being measured against metrics as mentioned in R2.

1.4 Approach

To ascertain answers to the research questions, as a first step, we did a critical review of the existing research literature on this specific topic of visualization based dependence tracing techniques for spreadsheets. This paper summarizes in brief the findings of the review and the conclusions drawn from it. It essentially presents preliminary results and indicators related to the research questions. In order to illustrate our findings for this paper, we chose a number of research papers relevant on this topic and revisited their contents from the following aspects:

I. The basic technique/principle/strategy

II. Characteristic features related to dependents tracing

III. Tools or prototypes developed if any

IV. Comments or details available on testing, performance, and limitations

V. Current status of the research and its implementation, and its perceived relevance or influence in the industrial scene

2. THE SELECTED RESEARCH PAPERS

2.1 Fluid Visualization of Spreadsheet Structures [4]

In this paper Igarashi *et al*. provide the description of a spreadsheet visualization technique mainly based on superimposition of visual enhancement and animations on top of the regular tabular structure of spreadsheets. The strategy is primarily the use of graphical variation (color, shading, outlining, etc.), animation, and lightweight interaction that allows the user to directly perceive the spreadsheet dataflow structure, keeping the tabular spreadsheet view unchanged. The *transient local view* feature is a visual enhancement based on outlining and shading that allows a user to view the dataflow associated with a particular cell. There is a *static global view* that visually enhances the entire spreadsheet by overlaying the complete dataflow graph of all the cells. *Animated global explanation* plays an animation to illustrate the dataflow of the entire spreadsheet. *Visual editing techniques* is a graphical manipulation technique that allows the user to directly edit the generated dataflow graph in *global static view* by dragging and its effect is then reflected in the spreadsheet structure as the textual formulas are updated automatically. A prototype for UNIX was developed using Pad++ and Python. Pad++ was a visualization platform developed and maintained by University of Maryland. A video demonstration of the tool in

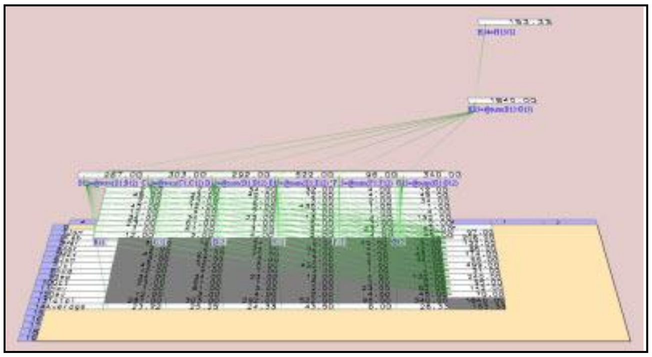


Figure 2: Recursive lifting-up operation

action is available. It is mentioned that the smoothness of animation is limited to spreadsheets of 400 cells[2] or lesser. Performance of the tool radically degrades with increase in size of the spreadsheets. There is no information if the efficacy of the prototype was tested with real spreadsheet users. No future plan is provided on how this tool can be implemented or scaled up for use in the industry of spreadsheet users. Pad++ and its support has been long discontinued and the project is closed by UMD. However, an extension of the idea of "transient local view" as proposed in this paper can be observed in MS Excel version 2007 onwards. In Excel 2007 the precedent cells of a cell are outlined in different colors. In Excel 2013 the precedent cells are actually shaded fully in different colors [Fig.1].

2.2 3D Interactive Visualization for Inter-cell Dependencies of Spreadsheets [5]

In this paper Shiozawa *et al*. propose a technique of cell dependence visualization in 3D based on an interactive lifting up operation. The technique utilizes the fact that spreadsheets are two dimensional tabular structures and therefore the third dimension can be used to depict complementary information like cell inter-dependencies. A spreadsheet is first graphically re-rendered in a 3D space. Next, users are allowed to select a cell and drag it upwards level-wise along the z-axis. The selected cell's dependent cells are pointed with arrows [Fig.2] and they themselves are also lifted up but kept one level below the selected cell. However in this case the advantage is in the fact that unlike in Excel, arrows connecting dependent cells lying on the same row would never overlap with each other to generate visual ambiguity. The lifting up operation is recursively repeated on the dependent cells as well to generate a leveled tree structure in 3D. This provides the user a clear idea of which cells in the sheet are more important by looking at the levels of dependents lying below them. A prototype for UNIX was developed by modifying the spreadsheet program SLSC. The 3D graphics were implemented with OpenGL APIs. No information regarding the performance of the prototype is provided. For an application such as this, making heavy use of computer graphics, it is presumable that performance and scaling could be a concern. Unfortunately the paper does not throw any light on this matter. Neither was given any detail about how beneficial or acceptable the tool proved for spreadsheet users.

2.3 Visual Checking of Spreadsheets [2]

In this paper Chen *et al*. propose a set of strategies aimed at checking and debugging of spreadsheets using visual methods to reveal the *deep structure* of spreadsheets to the users. A set of visual methods is described followed by strategies on how to best use those visual tools for different purposes of checking. The *functional identification* feature demarcates cells with different colors according to whether they behave as input, output, processing or standalone and this classification is based on whether a cell is having dependents, precedents, both or none. *Multi-precedents and dependents tool*, *block-precedents tool*, and the *in-block-precedents-dependents tool* are all tools that illustrate various types of inter-cell dependencies with pointed arrow-heads similar to the Excel feature. The difference here being that arrows not only connect individual cells but also have the capability of offering the visual perception that they are connecting a set of related cells that are visually grouped together by shading or coloring; such group of cells are termed in the paper as *cell block*. Three debugging strategies each for global and local context were described to illustrate the use of these tools. The tools were implemented using VBA (Visual Basic for Applications) and authors claimed that they can be plugged in to any Excel installation. In spite of claims that the tools increase usability of spreadsheets, no details were given about user acceptance or any measurement of by how much they increased usability.

2.4 Spreadsheet Visualisation[3] to Improve End User Understanding [1]

In this paper Ballinger *et al*. provide description of a visualization toolkit that could ease understanding of spreadsheets by introducing visual abstraction with types of images that emphasize on layout and dependency rather than values of cells. In order to achieve this, their idea was to extract all the information contained in a spreadsheet and utilize that in a more versatile programming environment to quickly generate visualizations. They chose Java for this purpose and since Excel is the most popular spreadsheet application, their toolkit was designed to operate on Excel spreadsheets. The toolkit is capable of extracting low level structural information and data from spreadsheet files, analyze that information, and produce visualization. The *data dependency flow* feature is capable of generating 2D and 3D maps that illustrate the general drift of dataflow in a spreadsheet with arrows of unit magnitude [Fig.3 (a)]. This helps reduce the visual clutter which normally occurs with arrows of different lengths due to different distances between cells. The *graph structure* feature provides the *spring view* [Fig.3 (b)] which is a generated graph of cells stripped of their values. The *detailed inspection of formula* feature provides visualizations that are similar to Excel Audit and

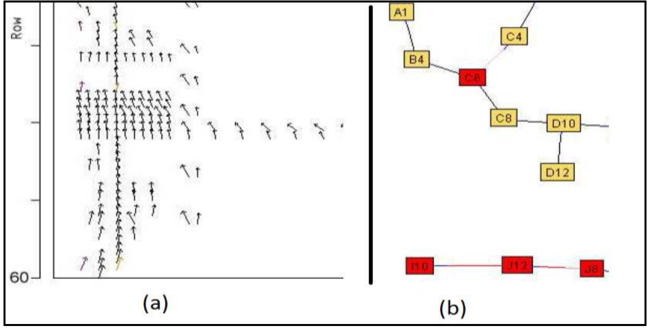


Figure 3: (a) Data dependency unit vector map (b) Spring view graph structure

[2] This is a much smaller number of cells than what is observed in typical real life spreadsheets

[3] Paper is in New Zealand English

block precedents tool (subsection 2.3) but they are not overlaid on spreadsheets; the images are generated on spreadsheet-like matrix structures and the cells are reduced to row-column intersection points, their values wiped out to reduce visual overhead on the user's understanding. The toolkit was run successfully on a corpus of 259 workbooks. User-studies were not conducted and no details were given on whether real users found it convenient enough to understand the various types of images.

2.5 Supporting Professional Spreadsheet Users by Generating Leveled Dataflow Diagrams [3]

In this paper Hermans *et al.* propose a spreadsheet visualization technique and the description of an implementation along with the findings of a user study. The work in this paper extends that of previous work by the authors about extraction of class diagrams from spreadsheets. The basic principle depends upon classifying all cells in a spreadsheet as either of type *data*, *formula*, *label*, or *empty*. Diagrams similar to ER (Entity-Relationship) diagrams are next created by representing data cells as entities and formula cells as method (operation) + entity (result). The interconnections are illustrated as relationships. Next these elements are grouped together based on the presence of *label* type cells to form larger entities that represent cell blocks. These are then assembled inside entities that represent their respective worksheets. In this manner the hierarchical leveled dataflow diagrams are generated. The *global view* [Fig.4] feature offers the users a high level interactive visualization of the whole workbook showing the dependencies between worksheets. The *worksheet view* shows the dependencies between blocks in the same sheet and the low level *formula view* shows in details how individual cells are inter-connected via formulas. A tool was developed called GyroSAT (Gyro Spreadsheet Analysis Toolkit) in C# 4.0. The output dataflow diagram is produced in DGML (Directed Graph Markup Language) which can be viewed and navigated in Microsoft Visual Studio 2010 Ultimate's built-in DGML browser. This tool was extensively evaluated with a user group consisting of 27 professional spreadsheet users working in a large Dutch financial management company. A set of 9 spreadsheets that were used for testing in 9 case studies had number of worksheets ranging from 4 to 42, and number of cells ranging from 1048 to 503050. Subsequently this tool and its features have been integrated into the set of services offered by the spreadsheet solutions company Infotron.[4]

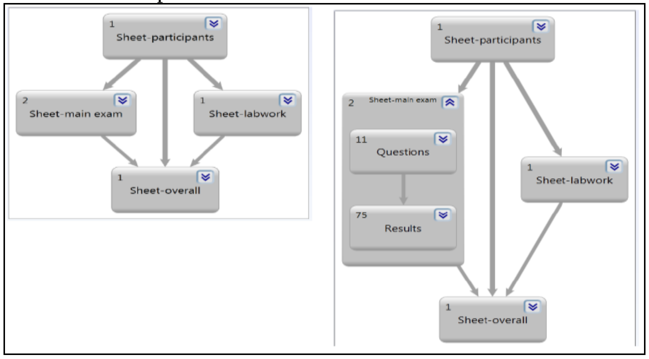


Figure 4: Global view (L) and Worksheet view (R)

3. CONCLUSIONS

Our study indicates that each of the five research papers proposes unique and innovative visualization techniques based on different strategies. All of them offer rich set of features intended to help spreadsheet users from different angles. Only two of them have prototypes running on UNIX, both of which, to the best of our beliefs will prove incompatible for current use on any popular platform. One has Excel based VBA implementation which supposedly should work as plug-in to any Excel version but is subject to be tested against version incompatibility. Two of them have full-fledged standalone implementations based on Java and C#, both accepting Excel spreadsheets as inputs, but only one of them has found practical exposure in the industry. This reinforces the need to explore our research question "**R1**. **Why the proposals thus far described and demonstrated in research literature have not reached the industry as implementations?"**

Only one of the research ideas has been properly validated against a set of real world professional spreadsheet users. The efficacies of the rest of the research ideas have only been claimed in writing but not demonstrated by user studies. This further reinforces the need to explore our second research question "**R2**. **Is there any well-accepted metrics framework with which such implementations as above (R1) can be compared to each other?"**

The above findings also lead us towards the general conclusion that our third research question "**R3. Is there any well-defined opportunity for improvement in the dependence tracing context**?" is an open question indeed. In that light we therefore judge that a suitable next step would be to do a more exhaustive search of available spreadsheet visualization tools and 1) actually test them on industrially used spreadsheets such as those available in the EUSES corpus and if the tools are found to be performing in a reliable manner then 2) test them on an adequately large and well represented spreadsheet users group to measure usability.

4. REFERENCES

[1] Ballinger, D., Biddle, R., Noble, J. 2003. Spreadsheet Visualisation to Improve End-user Understanding. In proceedings of the Asia-Pacific Symposium on Information Visualisation - Volume 24 (APVIS 2003), Adelaide, Australia, pp. 99–109.

[2] Chen, Y., Chan, H. C. 2000. Visual Checking of Spreadsheets. In proceedings of the European Spreadsheet Risks Interest Group 1st Annual Conference (EuSpRIG 2000), London, United Kingdom.

[3] Hermans, F., Pinzger, M., Deursen, A. van. 2011. Supporting Professional Spreadsheet Users by Generating Leveled Dataflow Diagrams. In proceedings of the 33rd International Conference on Software Engineering (ICSE 2011), Waikiki, Honolulu, HI, USA, pp. 451–460.

[4] Igarashi, T., Mackinlay, J., Chang, B.-W., Zellweger, P. 1998. Fluid Visualization of Spreadsheet Structures. In proceedings of the IEEE Symposium on Visual Languages (VL 1998), Halifax, NS, Canada, pp. 118–125.

[5] Shiozawa, H., Okada, K., Matsushita, Y. 1999. 3D Interactive Visualization for Inter-Cell Dependencies of Spreadsheets. In proceedings of the IEEE Symposium on Information Visualization (Info Vis 1999), San Francisco, CA, USA, pp. 79–82, 148.

[4] Infotron is a spreadsheet solution company offering web based spreadsheet analysis services http://www.infotron.nl/