Introduction

* Prevalence of spreadsheets across business
  + Uses predicted data about future from past where future is now past
* Horror stories of getting them wrong
  + EuSpRIG reference and uncertain antecedent
  + “augur” is bad word, agreed
* The gathering of spreadsheet corpora and their uses so far
  + “Many” is contentious, “work” isn’t consistent
  + “Spreadsheet grammar” is uncertain
* The uses of a tool that used these corpora
  + Formula/formulae
  + “Understanding of this patterns”
  + Doesn’t completely necessitate tool – doesn’t distinguish enough from tools.
* Contributions – This paper presents…
  + Needs an overall more compelling tone

Related work

* Other spreadsheet visualization tools
  + Misguiding incrementality
  + “Tend to” is weasely
* Content of cells within spreadsheets
  + Nevertheless, this transitions poorly.
  + Need more comparison/contrast
* API usage
  + Incomplete sentence
  + More comparison/contrast

Own thoughts on intro:

* Recurring themes throughout the paper………
  + Exploration, Bad Smells, Education
  + How to introduce exploration in intro?
    - New paragraph on visualization? Would need more references…
  + How to incorporate bad smells?
    - Most clearly ties into the dangers of getting things wrong – in the horror stories, there were dire costs because the owners of spreadsheets did not have well enough an overview of the practices
  + How to incorporate education?
    - …Likewise, a bad education can cause more problems, take more time, just be really a drag on the entire process of working on spreadsheets.

Notes on spreadsheet motivation:

* UCheck - Studies have shown that each year, tens of millions of professionals and managers create hundreds of millions of spreadsheets [1], thereby making spreadsheets the most widely used programming environment [2]
  + Spreadsheet errors: what we know, what we think we can do, in: Symposium of the European Spreadsheet Risks Interest Group
  + C. Scaffidi, M. Shaw, B. Myers, Estimating the numbers of end users and end user programmers, in: IEEE Symposium on Visual Languages and Human-Centric Computing, 2005, pp. 207–214
* Grammar for Spreadsheets - Winston [1] estimates that 90% of all analysts in industry perform calculations in spreadsheets. Their use is diverse, ranging from inventory administration to educational applications and from scientific modeling to financial systems. It is estimated that 90% of desktops have Excel installed [2] and that the number of spreadsheet programmers is bigger than that of software programmers [3].
  + W. Winston, “Executive education opportunities,” OR/MS Today, vol. 28, no. 4, pp. 8–10, 2001.
  + [2] L. Bradley and K. McDaid, “Using bayesian statistical methods to determine the level of error in large spreadsheets,” in Proc. of ICSE ’09, Companion Volume, 2009, pp. 351–354.
  + [3] C. Scaffidi, M. Shaw, and B. A. Myers, “Estimating the num- bers of end users and end user programmers,” in Proc. of VL/HCC ’05, 2005, pp. 207–214.
* Detecting Bad Smells (A) - End users may include teachers, students, children, accountants, scientists, or anyone wishing to create spreadsheets or databases for their own use [1]. Spreadsheets are commonly used as end-user programming tools to create, for instance, simple programs using conditionals or formulas, to compute data by writing formulas, or to store data during personal tasks. Many studies of spreadsheets have shown that end users continually create more complex spreadsheets, with formula content and size doubling every three years [5] [7]. Field audits conducted by researchers found that 80-90% of these increasingly complex spreadsheets contain errors [19].
  + A. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrence, H. Lieberman, B. Myers, M. Rosson, G. Rothermel, M. Shaw, and S. Wiedenbeck, The state of the art in end-user software engineering, Journal ACM Computing Surveys, 2009
  + D. WHITTAKER, Spreadsheet errors and techniques for finding them. Management Accounting, 77(9), pages 50–51, 1999
  + [7] IMA, Avoiding Costly Errors in your Spreadsheets. Contractor's Business Management Report, Issue 12, pages 2-4, 3p, 2004.
  + Panko and Raymond R., What We Know About Spreadsheet Errors. Journal of End User Computing, 10(2), pages 15-21, 2009.
* Fuse - End-user programmers today constitute a broad class of users, including teachers, accountants, administrators, managers, research scientists, and even children [1]. Although these users are typically not professional software developers, their roles routinely involve computational tasks that, in many ways, are similar to those of developers — not just in activity, but also in their underlying cognitive demands on users [2]. Perhaps the most ubiquitous form [3] of end-user pro- gramming software are spreadsheets, a table-oriented visual interface that serves as the underlying model for the users’ applications [4]. Cells within these tables are augmented with computation, such as expressions, functions and macros [4]. This interplay between presentation and computation within the spreadsheet environment has garnered significant interest from the software engineering research community [5]. Researchers have adopted techniques and approaches to studying errors [6], code smells [7], and refactoring in spreadsheets [8], similar to traditional.
  + ] A. J. Ko, B. Myers, M. B. Rosson, G. Rothermel, M. Shaw, S. Wiedenbeck, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, and H. Lieberman, “The state of the art in end-user software engineering,” ACM Computing Surveys, vol. 43, no. 3, pp. 1–44, Apr. 2011.
  + [2] A. Blackwell, “First steps in programming: A rationale for attention investment models,” in IEEE 2002 Symposia on Human Centric Computing Languages and Environments, 2002, pp. 2–10.
  + [3] C. Scaffidi, M. Shaw, and B. Myers, “Estimating the numbers of end users and end user programmers,” in VL/HCC ’05, 2005, pp. 207–214.
  + [4] B. A. Nardi and J. R. Miller, “The spreadsheet interface: A basis for end user programming,” in Human-Computer Interaction: INTERACT ’90, 1990, pp. 977–983.
  + [5] M. Burnett, “What is end-user software engineering and why does it matter?” in End-User Development SE - 2, ser. Lecture Notes in Computer Science, 2009, vol. 5435, pp. 15–28.
  + [6] S. G. Powell, K. R. Baker, and B. Lawson, “A critical review of the literature on spreadsheet errors,” Decis. Support Syst., vol. 46, no. 1, pp. 128–138, Dec. 2008.
  + [7] M. Pinzger, F. Hermans, and A. van Deursen, “Detecting code smells in spreadsheet formulas,” in ICSM ’12, 2012, pp. 409–418.
  + [8] S. Badame and D. Dig, “Refactoring meets spreadsheet formulas,” in ICSM ’12, 2012, pp. 399–409
* Smellsheet - Spreadsheets play a crucial role in modern society. They are inherently multi-purpose and widely used both by individuals with simple needs as well as by large companies as integrators of complex systems and as support for business decisions. In fact, it is estimated that 95% of all U.S. firms use them for financial reporting, that 90% of all analysts in industry perform calculations in them and that 50% of all spreadsheets are the basis for decisions. Effective mechanisms for error prevention, however, did not grow proportionally: up to 94% of real-world spreadsheets contain errors, which each year cause losses worth around 10 billion dollars [1]!
  + R. Panko, “Facing the problem of spreadsheet errors,” Decision Line, 37(5), 2006.
* Analyzing and Vizzing - However big the damage of errors in spreadsheets is, we consider errors as merely symptoms of the real problems with spreadsheets. The underlying prob- lem is the fact that the design of a spreadsheet is hidden behind the formulas and worksheets of a spreadsheet. For instance, if a worksheet dependents on cal- culations in another worksheet, one cannot easily see this at the worksheet level unless one looks deep down in the formulas and cells.
* Enron - Spreadsheets are an important type of software. Scaffidi and colleagues estimate that there are more than 55 million end user programmers in the US alone [1]. In many ways, developing a spreadsheet is similar to writing code. Both entail analyzing data, manipulating operations on that data and understanding dependencies between different parts of the program. One of the main differences is that in software engineering many methods and techniques have been constructed that support developers in managing complexity and understanding existing artifacts. Although recent efforts to transfer software engineering methods to spreadsheets have been relatively successful [2], there is still a lot to be gained, as spreadsheet errors remain common.
  + [1] C. Scaffidi, M. Shaw, and B. Myers, “Estimating the numbers of end users and end user programmers,” in Visual Languages and Human-Centric Computing, 2005 IEEE Symposium on. IEEE, 2005, pp. 207–214.
  + [2] F. Hermans, “Analyzing and visualizing spreadsheets,” Ph.D. dissertation, Delft University of Technology, the Netherlands, 2013.
* Supporting - Panko [18] estimates that 95% of U.S. firms, and 80% in Europe, use spreadsheets in some form for financial re- porting
  + R. Panko. Facing the problem of spreadsheet errors. Decision Line, 37(5), 2006.
* Spreadsheets are used widely within companies. It was, for example, estimated that 95% of U.S. firms use spreadsheets for financial reporting [1]. The information in spreadsheets often forms the basis for significant business decisions [2]. However, from previous research we know that spreadsheets are error-prone [3].
  + [1] R. R. Panko and N. Ordway, “Sarbanes-oxley: What about all the spreadsheets?” CoRR, vol. abs/0804.0797, 2008. [Online]. Available: http://arxiv.org/abs/0804.0797
  + [2] F. Hermans, M. Pinzger, and A. van Deursen, “Supporting professional spreadsheet users by generating leveled dataflow diagrams,” in Proceed- ings of the 33rd International Conference on Software Engineering. ACM, 2011, pp. 451–460.
  + [3] R. R. Panko, “What we know about spreadsheet errors,” Journal of Organizational and End User Computing (JOEUC), vol. 10, no. 2, pp. 15–21, 1998