Evaluating the Effectiveness of Static Analysis Programs Versus Manual Inspection in the Detection of Natural Spreadsheet Errors

Salvatore Aurigemma, Collins College of Business, University of Tulsa, Tulsa, OK, USA
Ray Panko, Department of IT Management, University of Hawaii at Manoa, Honolulu, HI, USA

ABSTRACT

Spreadsheets are widely used in the business, public, and private sectors. However, research and practice has generally shown that spreadsheets frequently contain errors. Several researchers and vendors have proposed the use of spreadsheet static analysis programs (SAPs) as a means to augment or potentially replace the manual inspection of spreadsheets for errors. SAPs automatically search spreadsheets for indications of certain types of errors and present these indications to the inspector. Despite the potential importance of SAPs, their effectiveness has not been examined. This study explores the effectiveness of two widely fielded SAPs in comparison to manual human inspection on a set of naturally generated quantitative errors in a simple, yet realistic, spreadsheet model. The results showed that while manual human inspection results for this study were consistent with previous research in the field, the performance of the static analysis programs at detecting natural errors was very poor for every category of spreadsheet errors.

Keywords: Natural Errors, Seeded Errors, Spreadsheet Engineering, Spreadsheet Error Detection, Spreadsheet Error Taxonomy, Spreadsheet Quality Control, Static Analysis Programs, Verification and Validation

INTRODUCTION

Decades of research on spreadsheets have resulted in several commonly accepted tenets. Spreadsheets are widely used internationally in manufacturing (Schwartz, 2005) education (Wagner, 2003), business (Chan & Storey, 1996), and government (Butler, 2000). Spreadsheets are used for modeling problems ranging from trivial to epic, from simple calculations to incredibly complex amalgamations spanning multiple applications (Panko & Port, 2013; Warfield & Hihn, 2009) and tens of thousands

DOI: 10.4018/joeuc.2014010103

Given the widespread use of spreadsheets, errors are a potential concern. Since 1995, nine studies have inspected operational spreadsheets to look for errors (Panko, 2013). These studies found errors in 84% of the 163 spreadsheets they inspected. Among inspection studies that only reported errors if they were serious, the percentage of incorrect spreadsheets was actually higher, 91% (Panko, 2013). Although much research has been conducted on spreadsheet errors, there is no consensus on how to prevent, detect, and deal practically with these errors (Clermont & Mittermeir, 2002; Panko, 2007; Croll, 2003; Powell, Baker & Lawson, 2008; Butler, 2000; Rajalingham, Chadwick, Knight & Edwards, 2000).

One way to reduce errors is to manually inspect spreadsheets. Unfortunately, humans are only partially effective in detecting spreadsheet errors, even when using aggressive error detection techniques. In eight laboratory experiments collectively using 982 participants, the participants only discovered 63% of all seeded errors (Panko, 2010). In comparison, manual code inspection in software testing experiments has shown similar detection rates (Panko, 1999). Not only are spreadsheets consistently developed with errors and people are generally poor at detecting the errors, developers are overconfident in both their ability to build error-free models and detect errors (Panko, 1999; Howe & Simkin, 2006; Davis & Ikin, 1987; Reithel, Nichols & Robinson, 1996).

**Motivation**

Spreadsheet software add-ins have become an attractive option for end users to enhance the quality of end-user models (Grossman, 2002b). Several researchers (Butler, 2000, Nixon and O’Hara, 2001, Clermont, Hanin & Mittermeir, 2002; Powell et al., 2009) have included the use of spreadsheet software tools in their prescriptions towards better spreadsheet quality control. Static analysis programs (SAPs), in spreadsheet and software engineering, offer the potential to identify coding problems automatically without executing the program (or deploying the spreadsheet). SAPs represent an automated form of manual code inspection (also considered static analysis) with the benefit of significant time savings. Both SAPs and manual inspection require predetermined rules or standards to in order to identify potential flaws (Chess & McGraw, 2004).

Static analysis programs, for the purpose of this paper, are defined as automated software tools that conduct spreadsheet cell inspection for the purpose of improving the overall quality of spreadsheet models. SAP functionality can include automatically checking for spreadsheet errors, generating documentation, providing visual mappings of spreadsheet cell dependencies, and assisting in the use and building of new models. Figure 1 shows a sample error checking output of one of the SAPs examined in this study. Although recommended for use, there have been few studies conducted on the effectiveness of popular spreadsheet static analysis programs. This paper specifically compares the effectiveness of two popular static analysis programs that proffer spreadsheet error detection capabilities. SAP error detection performance is compared with manual spreadsheet inspection error detection, using the same natural spreadsheet error corpus.

Static analysis programs for spreadsheets are widely used, although they are far from universal. One reason to study them is this widespread use itself. Another is that vendors often make claims about the efficacy of static inspection versus manual inspection, arguing that SAPs avoid the human errors of inspectors. If SAP use really is substantially better at finding errors than tedious and expensive manual inspection, then SAPs should be used far more than they are.

However, in traditional software testing, SAPs play a minor role. They are certainly widely used. In a discussion with one SAP vendor in 2010, the vendor noted that 67% of all software development firms use static analysis programs and that this figure is close to 100% in the United States. However, he added that SAPs are used primarily to run preliminary sweeps for errors before more formal testing, much as

---

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Related Content

Mobile Accessibility in Touchscreen Devices: Implications from a Pilot Study with Blind Users on iOS Applications in iPhone and iPad
www.igi-global.com/chapter/mobile-accessibility-in-touchscreen-devices/80370?camid=4v1a

VDT Health Hazards: A Guide for End Users and Managers
www.igi-global.com/chapter/vdt-health-hazards/4426?camid=4v1a

Computer Security and Risky Computing Practices: A Rational Choice Perspective
www.igi-global.com/chapter/computer-security-risky-computing-practices/4482?camid=4v1a
A Comparison of Foreign Government Computing Policies
Rick Gibson and Mary Alice Mcdonough (1996). Journal of End User Computing (pp. 3-10).
www.igi-global.com/article/comparison-foreign-government-computing-policies/55729?camid=4v1a