EuSpRiG
European Spreadsheet
Risks Interest Group

EuSpRIG 2007 conference report

Dean Buckner of the UK Financial Services Authority gave the *Regulator's View* on the progress in the control of End User Computing (EUC) in the financial markets. His themes were echoed by many subsequent speakers

1. Change of mindset. He referred to the acceptance that spreadsheets are not going to be replaced by bigger systems, but rather that they are here to stay.

2. User training. This is still shockingly neglected; he still finds *dumb solutions* that could be replaced by cleaner methods.

3. There is no accepted base of 'good practice'.

4. Because of (3) there is therefore no accreditation of spreadsheet skills.

5. He sees increasing mention of spreadsheet controls in audit reports.

6. Data standards – including data quality in Access databases, created to get over the 65535 row limit in Excel prior to the 2007 version.

7. Software support – more tools and technologies are becoming available to manage spreadsheets.

Steve Powell of the Tuck School of Business at Dartmouth College revealed the results of their audits into quantitative errors that showed impacts of up to $100 million in real world spreadsheets. These are reasonably large spreadsheets, having on average 15 sheets and 1.5MB in size. They discovered how difficult auditing really is, even with tools like Spreadsheet Professional. The first fully documented study of its kind, this paper received the highest rating by the attendees of all the papers at the conference. Here is one quote describing one end of the scale:

> *"Organisation 5 is a small consulting company with highly educated employees and a culture that demands excellence. The spreadsheets we audited from this firm were works of art: thoughtfully designed, well documented, easy to understand, and error free."*

Ray Panko of the University of Hawaii provocatively titled his paper *Thinking is bad*. He quoted human error research indicating that logical thought is the most difficult thing we creatures can do. In his categorisation of Blameful vs Blameless errors, he urged people to be frank about unavoidable error rates. He ended with a challenge to those prescribing *good* or even *best* practices to produce the evidence that their recommendations are safe and effective.

Paul Bach of Compassoft, the main conference sponsor, reported that 30 to 40% of corporate data is in uncontrolled end-user applications. He described the processes necessary to get them under control: discovery, change management, and validation. He mentioned one user who guessed that they had 300 critical spreadsheets; they discovered millions (not all critical of course).

Ralph Baxter of ClusterSeven, another conference sponsor, outlined *Enterprise Spreadsheet Management* (ESM) and described the efficiencies that come from saving wasted time. Taking Dean Buckner's 6th point for example, the 'spot the difference' task of data integrity checking is easier with ESM.

Simon Murphy of Codematic sympathised with the tortured souls in *Spreadsheet Hell*. Maintenance represents 90% of spreadsheet use, yet the initial development phase gets most of the attention. Most users have only had at most two days formal training in Excel, so they build unstable and fragile *frankensheets* which are terminally undebuggable and break if anyone changes anything. 'Keeping it working' therefore becomes the desperate state of these users like the little Dutch boy keeping his finger in the sea wall.

Soheil Saadat of Prodiance highlighted the larger-scale problems in multiple linked workbooks. Links become broken because paths are changed when these files are moved into a document management system. He described their WebDAV system for discovery, risk analysis and automatic updating such links.

Andy Kumiega of the Stuart Graduate School of Business described hybrid applications using Matlab for serious matrix manipulation and Excel for the user interface and data integrity control. For example, Excel cannot handle matrices bigger than 52x52 but Matlab's interface is text-based.

Patrick O'Beirne's paper entitled *Facing the Facts* argued for making visible the cost of rework. Track the time (and therefore the cost) spent on each phase of spreadsheet development, distinguishing when you are creating new work, or correcting existing work. You may be surprised to discover that more than half your time is spent on rework, on correcting defects. It would then be good engineering practice to trace the causes of these defects back to their root and put in earlier checks to prevent and detect errors. For example, it is much less costly to catch a specification or design mistake by early proofing and review, rather than having to unwind it after implementation, with all the knock-on effects on dependencies of such late changes.

David Chadwick of Greenwich University presented a questionnaire on a *Minimum Generic Skill Set* for spreadsheet development. The first such list of testable skills was created by the European Computer Driving Licence Foundation in their syllabus *Spreadsheet Check and Control*. David Chadwick expands on this to include teaching practices such as auditing models seeded with errors.

Kath McGuire of SmallSpark asserted that training based on the tasks that people need to accomplish in their context is more effective than generic classroom-type training. People learn more effectively when they are told what they need to know, at the time when they need to know it, in the situation where the need arises. In such cases, the productivity benefits are immediate and obvious.

Simon Thorne of the University of Wales Institute Cardiff (UWIC) had several distinctive points in his presentation. Firstly, the topic Example Driven Modelling was novel enough, but secondly he provided statistical analysis of the significance of the results found, which is welcome whenever we get it. For classification problems (i.e. grading) people find it easier to provide examples than to create spreadsheet formulas to do the classification. Such examples can be fed into neural network software so it can learn how to classify attributes where the answer is not yet known.

As an example of a classification spreadsheet, he cited the Cardiac Anaesthesia Risk Evaluation (CARE) model from the Medical Algorithms website medal.org. His analysis revealed some clearly incorrect results from its formulas, which may have been automatically generated.

Mukul Madahar also from UWIC described various measures of risk and impact, including urgency.

Philip Howard of Bloor Research presented an overview of the types of software tools in the marketplace: auditing, automation, control and compliance. Along with the presence of the four other vendors (Compassoft, SecureXLS, ClusterSeven, and Prodiance) that is an indicator of the maturing of the market.

Derek Flood of Dundalk Institute of Technology (DKIT) reported on the limitations of voice control technology that experimental subjects encountered when using it to debug spreadsheets. When entering a formula the Dragon software does not adapt the vocabulary in order to offer relevant keywords first. It uses the same vocabulary in all situations so it persistently offered "sun" as opposed to "sum" during the auditing process.

Brian Bishop also of DKIT investigated end-user behaviour while correcting a debugged spreadsheet. His novel approach was to record the timing of their navigation and editing. This gave some insight into patterns of scanning and fixing, which would be worth more research.

David Colver let us in on one of the secret weapons in the Operis armoury: inclusion/exclusion analysis. For efficiency in his final review of the work of the analysts, he has client spreadsheets transformed into a standardised form. As part of that institutionalised process, the analysts rearrange the data to make explicit which items are included and which excluded from key bottom line figures such as IRR and financial ratios. This is a deceptively simple but powerful method to raise questions about the appropriateness of the choices.

Tom Grossman of the University of San Francisco in his paper "*Source Code Protection for Applications Written in Microsoft Excel and Google Spreadsheet*" discussed the different possibilities of spreadsheet protection available in Excel and in Software as a Service (SaaS).

Patrick Kemmis and Giles Thomas of Resolver Systems described their Python-based spreadsheet that unusually has both a grid and a code tab in view at the same time.

Jocelyn Paine of Spreadsheet Factory pushed the boundaries of spreadsheet deconstruction with his tutorial on modularity, and a paper on Semantic Wikis and Literate Programming. The latter featured an economical and elegant way of writing documentation with embedded code, contrasted to the usual approach of code with embedded comments.

The organisation of the conference received favourable comments from the attendees. All the speakers kept within their time slot and had time for questions.