A computational framework for the near-elimination of spreadsheet risk

Dr. Yusuf Jafry
Chief Technical Officer
Risk Integrated

This paper was first presented in July 2006 at EuSpRIG (European Spreadsheet Risk Interest Group) conference in Cambridge, UK

ABSTRACT

We present Risk Integrated's Enterprise Spreadsheet Platform (ESP), a technical approach to the near-elimination of spreadsheet risk in the enterprise computing environment, whilst maintaining the full flexibility of spreadsheets for modeling complex financial structures and processes. In its Basic Mode of use, the system comprises a secure and robust centralized spreadsheet management framework. In Advanced Mode, the system can be viewed as a robust computational framework whereby users can "submit jobs" to the spreadsheet, and retrieve the results from the computations, but with no direct access to the underlying spreadsheet. An example application, Monte Carlo simulation, is presented to highlight the benefits of this approach with regard to mitigating spreadsheet risk in complex, mission-critical, financial calculations.

1. INTRODUCTION

Spreadsheet risk is the danger that errors in a common business tool such as Microsoft Excel can cause material losses when used inappropriately by financial organizations. Today most banks already have established sets of rules, standards, and controls over their accounting systems and many of the databases they access. However controls have not yet been put into place for their smaller systems such as spreadsheets. This lack of internal control and audit reporting is something now being addressed in response to the Sarbanes-Oxley Act of 2002 banking regulation in the US, and the operational risk section of Basel II globally.

The problem with spreadsheets is that they are being built, in general, by non programmers. Errors creep into formulas, some from formatting, and some from links to other spreadsheets. Most are due to negligence, although a few are due to fraud. There are no thorough procedures in place to check the accuracy of the spreadsheets, or to test multiple runs of data through them.

The financial software industry has responded in a number of ways. For example, there are various spreadsheet auditing tools available that can assist in the process of building reliable spreadsheets and there are tools available to help end-users build new spreadsheets from scratch in the most robust manner. Also, more comprehensive solutions for managing the use of spreadsheets are beginning to emerge in the marketplace. These so-called business intelligence (BI) platforms are aimed at delivering insight to managers from masses of quantitative data centrally-held in the enterprise. For example, Microsoft is reportedly working on a BI solution around a future release of their Excel spreadsheet product, due for release in 2007 [The Banker, 2006].

In Section 2, we present Risk Integrated's technical approach to the near-elimination of spreadsheet risk in the enterprise computing environment whilst maintaining the full flexibility of spreadsheets for modeling complex financial structures and processes. The approach is particularly suited to those applications where a common set of complex spreadsheet-based calculations has to be applied across multiple instances of data inputs (e.g., individual deals within a portfolio, distributed across the enterprise) whilst retaining centrally-managed consistency and integrity throughout. By way of example, Section 3 discusses the application of the approach to the computationally-intensive process of Monte Carlo simulation for assessment of credit risk (another topical subject in the sphere of Basel II). Section 4 contains concluding remarks.

2.2 Advanced Mode: a secure & robust computational engine

The full power of ESP is realized in the Advanced Mode of ESP operation whereby normal end-users (in contrast to superusers) can utilize a given spreadsheet model as a computational engine, but without having direct access to the underlying spreadsheet itself. The rationale here is that those users whose primary goal is to input data and make use of results of computations do not need access to the underlying spreadsheets. In other words, they can only submit jobs, via their browser window, to a centralized cluster of computational servers, which then retrieve the spreadsheets via the management framework, automatically spawning service sessions of the underlying spreadsheet application, and run the computations against the inputs submitted by the users.

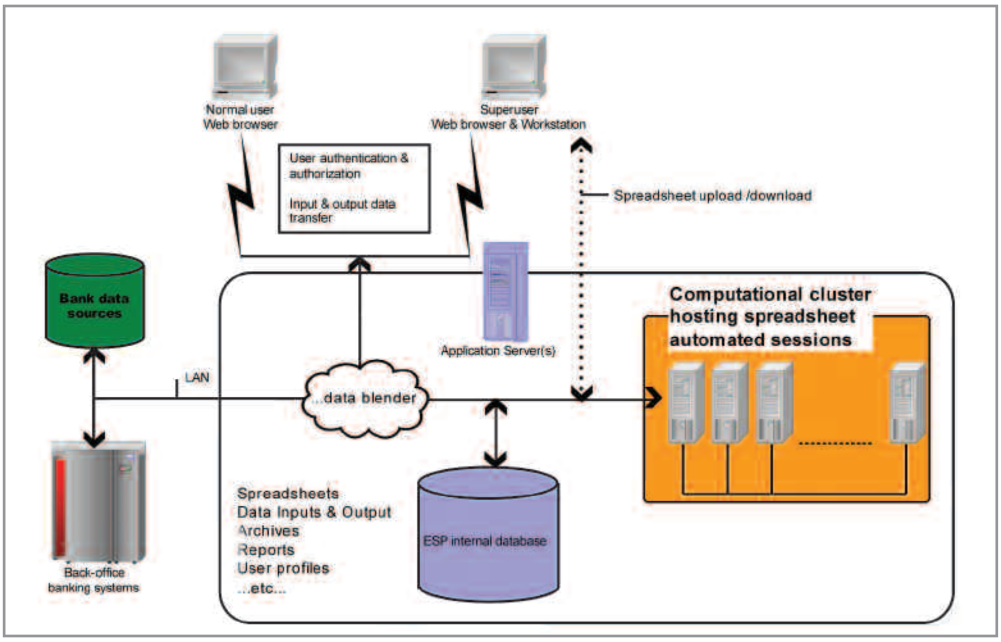


Figure 1. Risk Integrated's Enterprise Spreadsheet Platform (ESP) Architecture

2. RISK INTEGRATED'S ENTERPRISE SPREADSHEET PLATFORM

2.1 Basic Mode: a secure spreadsheet management system

As depicted in Figure 1, Risk Integrated's Enterprise Spreadsheet Platform represents a secure spreadsheet management framework whereby the spreadsheets themselves are exposed only to a few designated experts (superusers) in the organization. These users have the responsibility for maintaining the integrity of the spreadsheet models, and for uploading their tested and approved versions to a centralized server/database. ESP provides sophisticated tools that monitor, assign, and track changes to those spreadsheets. A full audit trail is available for tracking the versioning of the spreadsheet models back to those users submitting the changes, thereby eliminating the fundamental source of spreadsheet risk associated with the proliferation of unversioned models, scattered around the organization.

The employment of ESP in the manner described, i.e. as a secure and robust spreadsheet management system, is considered the Basic Mode of ESP operation, and may be considered comparable with the aims of the other emerging business intelligence platforms with regards to securing access to spreadsheets.

The spreadsheets never end up on the end-users' machines and are never opened interactively by the end-users, thereby eliminating the major source of spreadsheet risk caused by end-users, namely, the introduction (inadvertently or otherwise) and propagation of errors within the core logic of the spreadsheets themselves. Depending on the specificity of the spreadsheet in question, the data submitted by the users via their browser can be screened before being sent to the computational servers, using validation technology built into the web application interface, thereby minimizing another source of spreadsheet risk, namely nonsense data being fed into the computations.

The results of the computations are time-stamped and archived in the central database for auditability and reporting, before being sent back to the end-user's browser for display. This minimizes the third major source of spreadsheet risk, namely the manipulation of the results coming out of the spreadsheet calculations.

Risk Integrated

2.3 Security layer

The security of the ESP management framework leverages the authentication and authorization layers inherent to the operating system and/or relational database management system (RDBMS). ESP allows for completely configurable user-and group-level security and permissions.

2.4 User and Data Interfaces

A possible criticism of the approach pertains to the perceived limited flexibility of the user interface for communicating with the underlying spreadsheet. One of the major reasons why spreadsheets have become so prevalent is because of their extreme ease of use, particularly with regard to rapid prototyping of ideas. To diminish this ease and flexibility would undermine the use of the spreadsheet format.

However, the ESP, by definition, retains that flexibility for the appropriate users in the organization (namely the designated experts) by providing them with complete access to the underlying spreadsheets (albeit through a secure content-management layer). They can prototype, manipulate, and test the spreadsheets in the normal manner. Normal end-users, who, in order to minimize operating risk, do not have access to the underlying spreadsheets, are provided with non-programmable graphical user interfaces (GUIs) which enable them to pass data into the spreadsheet, perform the desired calculations, and retrieve the results back. For maximum flexibility, these interfaces can be generic. They may comprise a simple suite of data entry grids which map on to the corresponding "inputs worksheet(s)" of the underlying spreadsheet, plus a corresponding suite of output data fields mapped on to the respective "outputs worksheet(s)" of the underlying spreadsheet. Alternatively, for well-established spreadsheet models (i.e., those which are used as deployed applications rather than prototyping scratchpads), the GUIs can be customized to reflect the model-specific input and output fields (thereby further facilitating the use of field-wise validation technology to minimize spreadsheet risk).

With ESP's implicit separation between the data and the spreadsheets, data can be fed to and from the spreadsheets in a variety of ways in addition to the generic (or customized) user GUIs discussed above. For example, it is straightforward to automatically populate the spreadsheets via links to the company's existing banking systems and data sources. This eliminates the spreadsheet risk associated with manual double-entry and/or the "copy and paste" of data from the data sources into the spreadsheets. The system's outputs are similarly flexible. They can be displayed to the user, exported to a variety of formats, or used to repopulate a banking system, database, or data warehouse.

2.5 Possible Uses

The ESP can accommodate any spreadsheet for any purpose. For example, in a typical basic mode usage scenario, it would simply be employed as a secure spreadsheet management system where users are all superusers. They use the spreadsheets by first downloading them from the central store, manipulate them for their present purpose, and then upload them again for safekeeping. Even this basic usage scenario dramatically reduces spreadsheet risk by eliminating the proliferation of models scattered across users' desktops. However, it does not take advantage of the full capabilities of ESP in Advanced Mode. By contrast, the example discussed below in Section 3, illustrates the maximal use of the ESP as a secure and robust computational engine, in addition to its spreadsheet management role.

2.6 Domain and Mechanical Errors

Ultimately the weak link in the chain is with the superusers. Domain (i.e. incorrect specification) and/or mechanical (i.e. incorrect implementation) errors introduced by them (unintentionally or otherwise) into the underlying spreadsheets can adversely affect the computations. However, ESP assists in these areas too.

Specialization

As discussed in the example presented in Section 3, ESP enables applications to be developed such that only the core business logic is programmed in the spreadsheet by the business analyst. All other aspects (e.g., data handling, numerical algorithms) can be housed within the ESP computational framework, which can be independently tested and qualified as fit-for-use. In this way, the business analysts only need to program or build spreadsheets in their areas of expertise, thereby minimizing the chance of introducing errors outside of their area of core specialization. In this way, ESP still allows business users to create their own applications (i.e., spreadsheets encompassing business logic) and, therefore, "avoid much of the IT bottleneck" [Gartner, 2006] that would be incurred with software development.

Testability

However, errors do occur, and can be reliably identified and removed only through a rigorous testing process. ESP provides for this in a systematic way. Since the data layer is separated from the spreadsheet core, it is straightforward to establish a set of standard tests comprising a collection of input/output data sets which have been agreed and signed-off as being valid and correct. Thereafter, whenever a spreadsheet is modified by a superuser, the policy can be imposed that the spreadsheet can only "go live" after it has successfully passed the battery of standard tests. Moreover, as discussed in Section 3, for any calculations involving pseudo-random numbers, the issue of testability becomes more severe. ESP enables full control of the pseudo-random "seeds" such that input/output data sets can be fully replicated (in contrast, for example, to Excel, which does not allow such control).

Spreadsheet Audit Trail

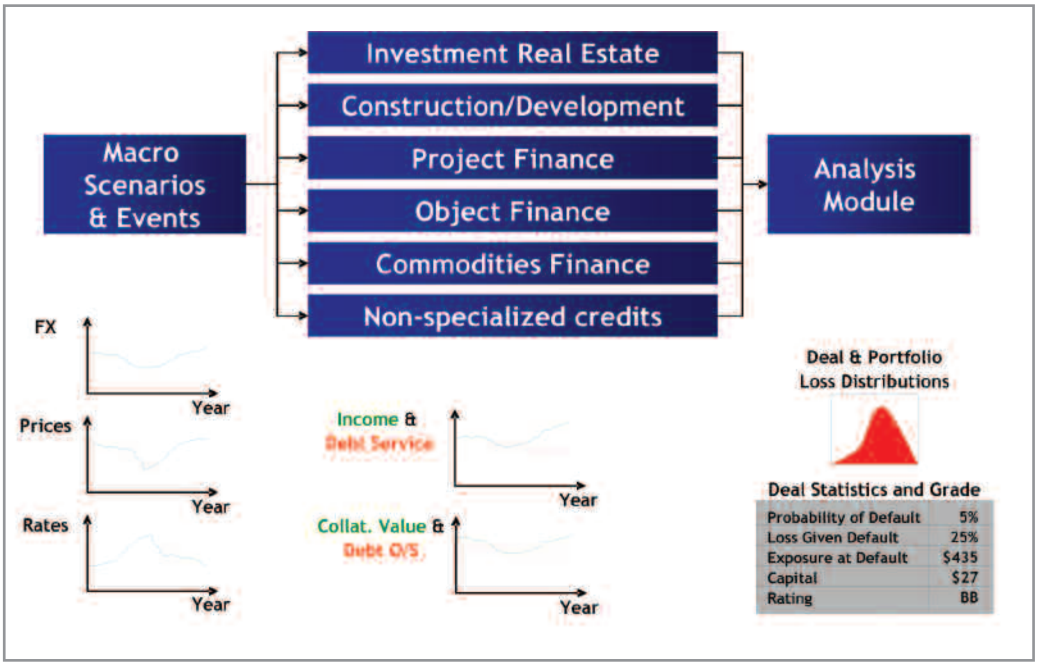
In the event that the effects of domain or mechanical errors do creep in to the core spreadsheets, the auditability inherent to ESP can resolve the issue. All modifications to the spreadsheet models are logged (with user IDs of who is making the modifications) by the ESP content-management system. Moreover, any time a spreadsheet model is called by the system at runtime, a copy is automatically made and archived on the server against the given usage instance. This provides a secure electronic audit trail back to the specific model used, and the ID of the superuser who last modified it. It would require collusion between the superuser, database administrator, system administrator, security officer, and network administrator to circumvent this audit trail.

3.2 Spreadsheet or Black Box?

A central issue faced by banks when constructing such Monte Carlo simulations is that the cashflow models are invariably constructed in a spreadsheet format (typically in MS Excel) either by their own business analysts or by the banks' clients or other external stakeholders in the deal. They present the Excel cashflow model as part of the deal documentation. Under these highly typical circumstances, there are essentially two options for performing a Monte Carlo simulation: 1) the cashflow model is retained in Excel, and the rest of the simulation framework is built in Excel and/or Visual-Basic-for-Applications (VBA) add-ins, or 2) the cashflow model is coded by a software developer in a different language and then integrated into a non spreadsheet-based simulation framework.

Figure 2.

The basic principle of Monte Carlo simulation for analyzing credit risk is illustrated. The simulation framework (macroeconomic scenario generation and data analysis) is common to all asset classes, and the cashflow model is tailored to a specific asset class (e.g., investment real estate, project finance) or to an individual deal.



3. EXAMPLE APPLICATION: MONTE CARLO SIMULATION

3.1 Introduction

Monte Carlo simulation is a well-established technique for analyzing credit risk. As illustrated in Figure 2 above, the basic idea is to construct a (typically time-domain) cashflow model to capture the logic of the given deal, drive this model into the future with a set of randomized macroeconomic scenarios (albeit with historically-imposed correlations between the variables), then perform the risk analyzes on the outputs generated from all of the scenarios. The modeling process can be complex such as for analyzing a complicated power generation deal or a large commercial real estate deal containing many properties and related securities. The computational process can be numerically intensive too, especially if globally-linked economies have to be included with hundreds of interacting macroeconomic variables from many different geographies over long lifecycles.

The first option—to build the entire simulation framework in Excel around the cashflow model—has the disadvantage of potentially incurring severe spreadsheet risk. Not least of these is because the programming challenge to construct a mathematically-consistent and numerically-robust Monte Carlo simulation framework is not trivial and beyond the skills of typical business analysts. Moreover, once such a framework has been built in spreadsheet form, it is highly-susceptible to the familiar forms of spreadsheet risk since the spreadsheet will, by its very nature, be large and complex, and thus vulnerable to the introduction of errors (inadvertent or otherwise).

The latter option—to translate the Excel cashflow model into another programming language—has the severe disadvantage that the underlying model can no longer be manipulated by the average business analyst. The model has become the proverbial black box, and all changes have to be implemented by a programmer.

Risk∫ntegrated

3.3 The ESP Solution

Using ESP in Advanced Mode, the advantages of both options can be realized, and disadvantages of each can be eliminated. Specifically, as illustrated in Figure 3, ESP incorporates a code layer which enables the spreadsheet to be embedded within a robust computational engine (written in C++) to communicate with an autospawned session of the spreadsheet application via a shared-memory interface.

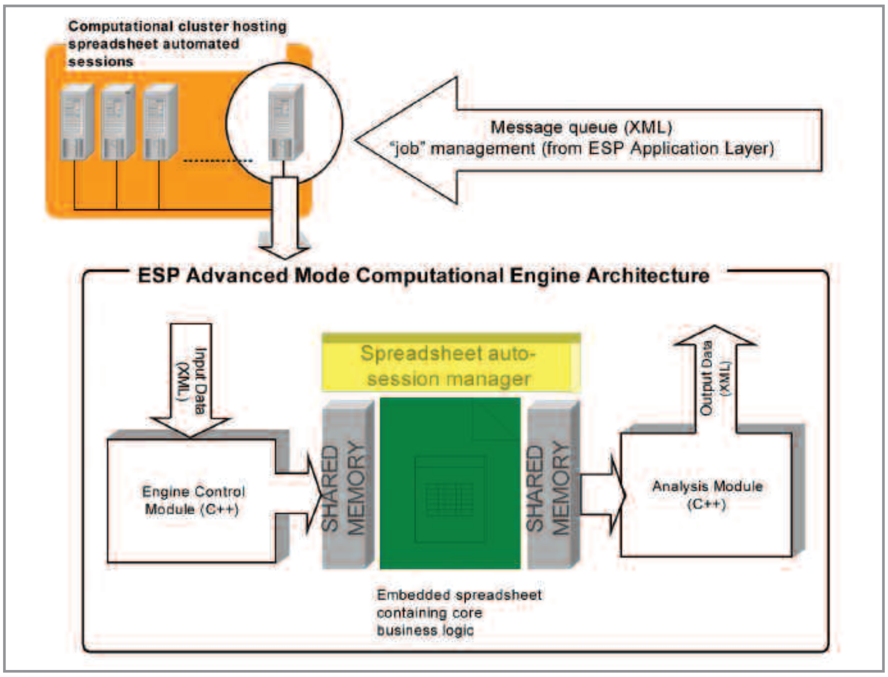


Figure 3.

ESP Advanced Mode. The autospawned spreadsheet session is embedded within a (C++ coded) computational framework via shared-memory, with XML-based input/output data feeds to the application layer via queued messaging.

Under this arrangement, various aspects of spreadsheet risk are eliminated. For example, with specific regard to the sample application presented:

1. Complex, numerically-intensive Monte Carlo calculations are not subjected to the risk of being programmed into the spreadsheet by business analysts. Only the core deal logic is programmed in Excel by the business analyst.

2. Complex numerical method algorithms such as matrix computations are central to many financial applications and are not subjected to the risk of being programmed in Excel or VBA. Rather, best-of-breed compiled libraries such as LAPACK for matrix computations can be linked in via the ESP Advanced Mode interface.

3. In the specific case of Monte Carlo simulation, the use of pseudo-random numbers is central to the technique. In Excel, the programmer has no control over the seed of the pseudo-random number generator. Hence, it is impossible to replicate the outputs for a given set of inputs (e.g., during the testing phase of the simulator). By contrast, under ESP, non-Excel-based pseudo-random number generators can be employed. These have the benefit of providing full control over the seed, so that input/output replication can be realized for test cases. This eliminates the significant risk associated with not being able to properly test the simulator before deployment.

4. Since the ESP computational and data management framework is separate from the Excel application, proper runtime monitoring can be invoked. For example, if the spawned Excel session hangs (unfortunately, an all-to common occurrence, especially for large computations), the ESP framework can detect this, and, if necessary, shut down the spreadsheet session, and inform the user that the computation has not proceeded successfully. This eliminates the operating risk associated with accepting the results from a possibly incomplete spreadsheet computation.

5. Spreadsheet applications are notoriously slow at computation compared with compiled code. This is particularly evident when attempting to construct complex Monte Carlo simulations in Excel. By contrast, under the ESP computational framework, where the bulk of the numerically-intensive computations are performed in the compiled C++ code and only the deal logic remains in Excel, the speed of computation is optimized. For example, on a Monte Carlo simulation of a large commercial real estate deal on high-end PC hardware, the ESP simulation framework is typically hundreds of times faster than the same simulation programmed in Excel, operating on the same Excel deal logic. By providing such performance advantages, the risks associated with the user's temptation to run just a few Monte Carlo iterations is mitigated. This can be significant, especially when calculating probability of default and loss given default for Basel II, where statistically significant results can only be achieved with large numbers of Monte Carlo iterations. When running fewer iterations, the results can be wholly misleading.

6. With the separation between data and the spreadsheet, ESP enables any important parameters (e.g., number of Monte Carlo iterations, discussed in the previous item, macroeconomic assumptions, centrally-set deal parameters such as haircuts) to be locked down such that the computations are performed consistently across the portfolio.

Although we have intentionally presented a complex application example (involving Monte Carlo iterations) to highlight the key benefits of the approach, it should be noted that any Excel spreadsheet can be configured for use under ESP Advanced mode. This is facilitated in a straightforward manner by the inclusion of an add-in (which contains the code hooks to the shared-memory interface) plus two specific worksheets, in an agreed format, which comprise the lists of input and output variables.

4. CONCLUDING REMARKS

We have presented Risk Integrated's Enterprise Spreadsheet Platform. It is our approach to the near-elimination of spreadsheet risk in the enterprise computing environment. With a complex example application, Monte Carlo simulation, we have demonstrated how ESP provides a secure, robust, computational framework, yet with spreadsheets remaining at the core, thereby preserving the flexibility demanded by business analysts.

5. REFERENCES

SOX was introduced by the SEC to protect the public from accounting error and fraud in the wake of the Enron, WorldCom, and Arthur Andersen scandals. For one, it demands companies to archive for five years all business records including electronic records and emails. Additionally, each company's external auditors are required to audit and report on the internal control reports of management, submitting an annual report of the effectiveness of their internal accounting controls to the SEC.

Basel II defines operational risk as the risk of loss resulting from inadequate or failed internal processes, people or systems, or from external events. In the U.S., advanced measurement approaches are being approved by the government which factor in operational risk into the calculation of total capital requirements. The more effective a bank’s internal operational risk management system is, the less money it needs to set aside in reserve.

A bank needs to able to measure the risk and manage it with tools sensitive enough to analyze all internal and external data and perform scenario analysis. They also need to take into account any mitigating factors, such as insurance coverage. Central to Basel II compliance, and to a bank’s ability to ultimately drive down its capital requirements, is the need to create a so-called risk culture, one that recognizes there are many consequences for failing to handle information correctly.

The Banker, (2006), “Spreadsheet Risk”, May 2006, pages 130-131.

Gartner, Inc., (2006), “Magic Quadrant for Business Intelligence Platforms, 1Q06.”