Interfaces
Vol. 38, No. 4, July–August 2008, pp. 281–288
ISSN 0092-2102 | EISSN 1526-551X | 08 | 3804 | 0281

informs®
DOI 10.1287/inte.1080.0372

Hypo International Strengthens Risk Management with a Large-Scale, Secure Spreadsheet-Management Framework

Yusuf Jafry
Risk Integrated, Douglas, Isle of Man, IM1 2LH, British Isles, yusuf.jafry@riskintegrated.com

Christopher Marrison
Risk Integrated, Garrison, New York 10524, chris.marrison@riskintegrated.com

Ulrike Umkehrer-Neudeck
Hypo Real Estate Systems GmbH, Münich, Germany, ulrike.umkehrer@hyposystems.de

We present the solution that Hypo Real Estate Bank International (Hypo) implemented to perform Monte Carlo simulations of its commercial real estate credit risk. The solution uses Risk Integrated's proprietary software, the Specialized Finance System, which is supported by another Risk Integrated technology, the Enterprise Spreadsheet Platform. The platform embeds individual master spreadsheets within a high-performance, server-based, computational-engine architecture; thus, it enables them to be accessed enterprise-wide. The major benefit of this approach is the near-elimination of spreadsheet risk in Hypo's banking system. The solution allows domain experts to have the flexible programming power that spreadsheets provide; however, it does not sacrifice the reliability and auditability expected from traditional business applications.

Key words: software; risk management; systems; models; portfolio; banks; insurance; investment; commercial real estate; simulation; Basel II.
History: This paper was refereed.

Based in Stuttgart, Germany, Hypo Real Estate Bank International (Hypo), with a portfolio of over €40 billion in commercial real estate lending outstanding, undertakes some of the world's largest real estate transactions. Hypo operates in 20 countries; it has branches in Amsterdam, London, Hong Kong, Milan, Madrid, Munich, New York, Paris, Stockholm, Stuttgart, and Tokyo, thus maintaining a presence in all major markets.

Previously, Hypo stored its loan data in several legacy banking systems. As is common in banks, it used Microsoft Excel for more complex analytics. These analytics included calculations for determining the structure and setting the price of new loans, generating reports about the potential loan pipeline, and stress-testing the portfolio of existing transactions. Stress-testing requires collecting information on all loans in the portfolio, and estimating how many loans would be seriously affected if, for example, interest rates rose by a percentage point. Collecting all the necessary data and estimating the effect of the stresses using a complex set of spreadsheets and VBA macros required a multiweek manual effort.

The most profitable commercial real estate loans are often highly customized. Hypo used Excel because (1) Hypo has a requirement for flexibility (for example, the ability to incorporate extra structural loan features on an ad hoc basis), and (2) spreadsheets facilitate the creation of financial models. This latter point is especially true because bank analysts typically have a financial or business background rather than a programming background. Despite their flexibility and popularity, the use of spreadsheets meant that it was very difficult to enforce standard models; management always had to request the assumptions made in analyzing prospective loans. In addition, some of the organization's most important data, the loan and underlying property details, were on disparate spreadsheets on Hypo employees' hard drives; this made it cumbersome to develop detailed overall reports on the portfolio without having to reenter data. In addition to the competitive need to continually make operations more efficient, Hypo was faced with the challenge of complying with Basel II banking regulations in Europe (Bank for International Settlements 2005).

The Requirements of the Basel II Capital Regulations

Basel II is the new regulation for setting the minimum capital to be held in reserve by internationally active banks. The capital is required to keep the banks safe in a time of crisis; however, it is costly to banks because raising capital is generally more expensive than issuing normal debt. Therefore, banks tend to hold the minimum capital required, given the risks that they face. Basel II has several levels of increasingly stringent standards, ranging from the standardized to the advanced approach. If a bank is able to comply with the more demanding requirements of the advanced approach, it will generally be required to hold less capital than it would under the standardized approach. By complying with the advanced approach, banks potentially save €20–60 million per year in capital costs. To qualify under the advanced approach, the bank must satisfy three conditions: (1) it must have credible models that are able to estimate the probability-of-default (PD) and loss-given-default (LGD) for each of its outstanding loans, (2) it must use the PD and LGD in its day-to-day business to demonstrate that it has faith in the results of the models, and (3) it must be able to publish detailed information about its portfolio so that the market can judge the bank's risk profile in comparison to the capital that it holds. To attain advanced compliance, Hypo needed new risk models and reporting systems. The company also wished to upgrade its internal reporting and management framework to provide better analytical tools to its lending officers who were responsible for structuring new loans, and to provide its managers with better insights into the risks of the overall portfolio.

Alternative Approaches for Measuring Risk

There are many types of risk models; each has advantages depending on the data available and the nature of the loans. The most common type is a regression model based on historical loan-default data. However, in commercial real estate, there is very little applicable historical data on loan defaults because, particularly in Europe, historical data have not been collected systematically for very long; in addition, loan structures are constantly evolving in the competitive market. Today's commercial real estate loans differ greatly from the loans of five years ago. This makes it difficult to use regression models. However, there is ample data available on external market conditions, such as historical property prices and interest rates. Moreover, there is a high degree of structure, including customized payment conditions, and leases, which is specific to commercial real estate loans. Monte Carlo simulation is a good way of bringing together the market information and transaction structure to assess the risk. In many ways, it can be considered as an extension of options pricing.

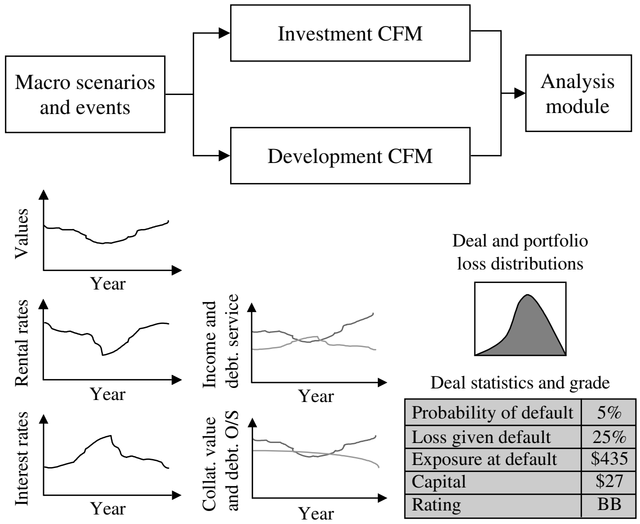
The Simulation Approach to Risk Measurement

Simulation brings together all known information on market conditions, tenant credit quality, loan-structure terms, and historical defaults.

Monte Carlo simulation is a well-established technique for analyzing credit risk. As Figure 1 illustrates, the concept is to construct a cash flow model that captures the logic of the given loan, drive this model "into the future" with a set of randomized macroeconomic variables, then perform the risk analyses on the outputs generated from all the iterations.

The set of randomized macroeconomic variables can be extensive, depending on the market sector being modelled. For example, in the SFS commercial real estate application, 200 different macroeconomic and market variables are typically simulated over 20 years. The random scenarios generated for each iteration of the simulation are shaped by regressors; these are calculated from historical market data observed over the previous 20 years, and result in a complex macroeconomic model and simulator.

The cash flow modelling process can be even more complex, particularly if the effects of all the intricate details of the transaction must be quantified. For example, if the customer can be persuaded to take steps to make the loan safer, such as adding a reserve



Probability of default	5%
Loss given default	25%
Exposure at default	$435
Capital	$27
Rating	BB

Figure 1: This chart illustrates the Monte Carlo simulation approach as used in Risk Integrated's specialized finance system (SFS) for analyzing credit risk. The simulation framework (macroeconomic scenario generation and data analysis) is common to all asset classes, and the cash flow model (CFM) is tailored to a specific asset class or to an individual loan.

account, the cash flow model must be sufficiently detailed to provide the lending officer with a quantification of the reduction in price that the bank might offer the customer while still maintaining profitability. As an example of the complexity, the cash flow model for commercial real estate has approximately 10,000 rows of formula-intensive Excel cells. Typically, each row contains 20 different formulas, ranging from simple additions to complex logic such as the following:

```
=Tenants!$AV36∗IF(COUNTIF($AD36:$AX36,
  “>”&TEXT(DJ$29,“0.00000”))-
  COUNTIF($AD36:$AX36,“>”&TEXT(DK$29,
  “0.00000”))=0,0,(SUMIF($AD36:$AX36,
  “>”&TEXT(DJ$29,“0.00000”),$AZ36:$BT36)-
  SUMIF($AD36:$AX36,“>”&TEXT(DK$29,
  “0.00000”), $AZ36:$BT36))/
  (COUNTIF($AD36:$AX36,“>”&TEXT(DJ$29,
  “0.00000”))-COUNTIF($AD36:$AX36,
  “>”&TEXT(DK$29,“0.00000”)))).”
```

Translating such a model into another language would be a significant task.

By using Monte Carlo simulation, the effect of all the intricate details of the transaction can be quantified using stochastic analysis. This is the only practical approach for analyzing models of such complexity (closed-form solutions are unobtainable, and closed-form approximations forego too much of the complexity to be useful in practice). However, the computational process of Monte Carlo simulation is numerically intensive because the entire spreadsheet must be recalculated both for each iteration of the simulation and each individual asset (or transaction) within the portfolio. Typically 1,000 to 10,000 iterations are used; however, for some requirements, such as economic-capital calculations, up to 100,000 iterations are required per asset. This pushes the limits of stand-alone Excel, even for a single asset. Moreover, because the bank is usually interested in analyzing its entire portfolio of thousands of assets, in practice, it becomes impossible to do so using stand-alone Excel.

Implementation Choices

Implementation Objectives

Given the need for new risk systems and the applicability of simulation to this asset class, Hypo needed a way to implement the complex analytics of simulation in a way that its global offices could use on all their thousands of loans. The approach had to work well for multicurrency loans, be flexible enough to use in new markets, and be able to handle the complexity of loans applicable to both completed buildings and buildings to be developed. With the financial complexity of Hypo's loans, it was important to have the underlying cash flow models in a form that could be understood easily and modified, if necessary.

Spreadsheet or “Black Box”?

A central issue that Hypo faced when considering the use of Monte Carlo simulation is that most bankers construct cash flow models using a spreadsheet format, typically Excel. Generally, two options are available: (1) the cash flow model is left in Excel, and the rest of the simulation framework is built using Excel with or without add-ins, or (2) the cash flow model is translated by a programmer into a standard programming language, such as C++, and integrated into a nonspreadsheet-based simulation framework.

In addition to the unreliability and slow speed for the computationally intensive simulation analytics, the option to build the entire simulation framework in Excel has significant exposure to human error. These errors, when discussed in relation to using spreadsheets, are commonly referred to as *spreadsheet risk*. Spreadsheet risks that Hypo wished to minimize included the following:

- Proliferation of spreadsheet models that are stored on individual users' desktop computers throughout the organization, are untested, and lack version data, and the unsanctioned manipulation of the results of spreadsheet calculations.
- Potential for serious mistakes resulting from typographical and "cut and copy-and-paste" errors when entering data from other applications or spreadsheets.
- Accidental acceptance of results from incomplete calculations.
- Errors associated with running an insufficient number of Monte Carlo iterations because of data or time constraints.

Given these potential problems, Hypo deemed a pure Excel solution as impractical. However, the alternative approach of translating the Excel cash flow model into another programming language would have eliminated the key advantage of spreadsheets because the underlying model(s) would no longer be transparent and could not be modified by the business analysts ("super users" as defined below in the *Accessing Data from Hypo's IT Architecture* section). The models would have become the proverbial "black boxes" because implementing any change would have required a software developer. Therefore, both of these traditional options posed significant problems for Hypo.

Spreadsheet and "Transparent Box"

To address the challenge of building a reliable, auditable system that still allowed the domain experts to build their models using Excel, Hypo adopted the SFS, which is built on Risk Integrated's Enterprise Spreadsheet Platform (ESP) technology. ESP enables applications such as the SFS to be developed such that only the core business logic is programmed in the spreadsheet. All other aspects (e.g., data handling and numerical algorithms) can be embodied within the ESP computational framework, which can be independently tested and qualified as usable. Thus, the designated experts only need to "program" (i.e., build spreadsheets) in their areas of expertise (e.g., business logic), minimizing the chance of introducing errors outside their core specializations. Thus, ESP allows business users to create their own business applications (i.e., spreadsheets encompassing business logic). Therefore, they avoid much of the IT bottleneck that is incurred by the software development process, which is traditionally required to translate a spreadsheet prototype into an enterprise system.

How the Solution Is Used

Normal Users

The SFS allows Hypo to quickly embed a spreadsheet model into an enterprise-level software application; it allows hundreds of normal users to access the spreadsheet analytics that Hypo's small team of super users maintains. The normal users are the lending officers and loan-administration staff who manage thousands of individual loans. Unlike the super users, Hypo's normal users do not have—and do not need—direct access to the underlying spreadsheet. The rationale is that their primary goal is to input data and analyze the results. They can only submit jobs, via their Web-browser window, to a centralized cluster of computational servers that manage the data flow to and from the underlying spreadsheets.

For structuring new loans, the SFS allows lending officers to type new loan information directly into the system using the graphical user interface (GUI), bring up past loans for modification, or upload details (e.g., tenant lease schedules) from external files. The results include the Basel capital statistics, classic metrics (e.g., loan-to-value), and profitability measures (e.g., net present value and possible losses). One of the most important results is the probability of default for each year at a specific loan level. This allows the lender to quantify the importance of specific risks and remove any spikes in the PD profile by altering aspects of the loan structure (e.g., by incorporating hedging strategies or reserve accounts).

Super Users

The super users are the few (typically four) designated expert financial analysts in Hypo's central risk-management group who control the central model(s) against which all loans are processed. The super users have responsibility for maintaining the integrity of the spreadsheet models, and for uploading their tested and "signed off" versions to a centralized server.

With ESP's separation of the data layer from the spreadsheet core, it is straightforward to establish a set of "standard" tests; these comprise a collection of input and output data sets that the super users have approved and signed-off as being valid and correct. Thereafter, whenever a super user modifies a spreadsheet, the policy can be such that the spreadsheet can only "go live" when it has successfully passed the battery of standard tests.

Accessing Data from Hypo's IT Architecture

The super users are also responsible for generating reports about the entire portfolio. For monitoring the risk and capital for the existing live loans, the SFS is run in batch mode, and produces capital and stress-testing reports. In addition to manual data entry via the GUIs, the data on live transactions can be updated via links to Hypo's existing banking systems and data sources. This is important in reducing workload and eliminating the spreadsheet risk associated with manual "double entry" and/or "copy and paste" of data from the data sources into the spreadsheets.

For the live transactions already in the portfolio, Risk Integrated created feeds from Hypo's major legacy IT systems. By using XML data structures, this integration was completed within a few months—an unusually short time for complex IT integration projects by industry standards. An additional benefit of this integration was that reports from the SFS database greatly speeded Hypo's process of collecting data for pipeline reports, credit-committee applications, and portfolio-risk reports. The portfolio-risk reports include not only the Basel risk statistics, but also stress tests such as the response of the portfolio to specific changes in market conditions.

The Technology of the Enterprise Spreadsheet Platform

Overall Architecture

To gain user acceptance, the system needed to be fast, secure, reliable, accessible, and flexible. Hypo achieved this through the SFS/ESP client-server architecture, which centralizes the calculations in a computational cluster of servers working in parallel. This parallel architecture makes the system completely scalable, and capable of simulating a complex portfolio of thousands of real estate transactions within a few hours. It is based on a deployment of ESP/SFS on typical PC server hardware (3 GHz processor; 2 GB RAM), whereby a single real estate transaction (approximately 10,000 rows of Excel) embedded within the ESP/SFS C++engine, would take approximately 100 seconds to evaluate 1,000 Monte Carlo iterations with 200 correlated driver variables over 20 years into the future. A portfolio of thousands of transactions could be evaluated in a few hours on an SFS/ESP server farm comprising a few machines (or CPU cores in a virtual server environment). This is orders of magnitude faster than performing the identical calculations using stand-alone Excel—even when taking advantage of the multithreaded features of Excel 2007.

The SFS/ESP architecture provides a separation among the reporting, data, computational, and spreadsheet layers. The separation of the data layer from the spreadsheet layer enables data from external sources to be easily transferred into (and out of) the spreadsheets. For example, the creation of a series of links, which connected various data sources within the bank, minimized the need for manual input of data, and created of single point of consolidation for information from all the bank's major systems.

In the simplest sense, ESP represents a secure spreadsheet-management framework whereby the spreadsheets are available only to a select few designated experts (super users) in the organization, thereby offering a high degree of source code protection (Grossman 2007). ESP provides sophisticated tools that monitor, assign, and track changes to those spreadsheets. A full audit trail is available for tracking the versions of the spreadsheet models against the users who are submitting changes; this eliminates sources of error that are frequently associated with the proliferation of models that lack versions and are scattered around the bank, and helps organizations in addressing operating risk and Sarbanes-Oxley compliance (SOX). The objective of the Sarbanes-Oxley Act is to avoid accounting error and fraud. It requires each company to archive all business records for five

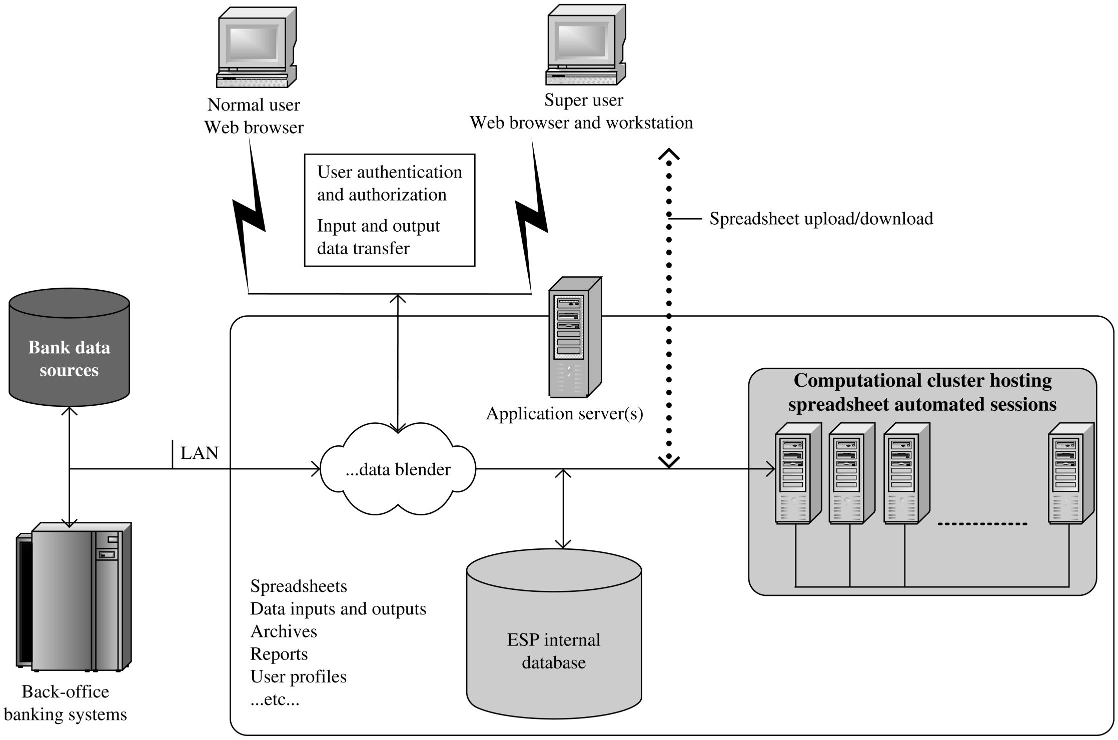

Normal user
Web browser
Super user
Web browser and workstation
User authentication and authorization
Input and output data transfer
Spreadsheet upload/download
Bank data sources
Application server(s)
Computational cluster hosting spreadsheet automated sessions
LAN
...data blender
Spreadsheets
Data inputs and outputs
Archives
Reports
User profiles
...etc...
ESP internal database
Back-office banking systems

Figure 2: This diagram shows Risk Integrated's ESP architecture. Spreadsheets, which are centrally stored, are editable only by designated super users in the organization. Normal users can interface only indirectly with the spreadsheets using a Web browser to enter data and retrieve results from the spreadsheet computational layer. The calculations are performed in automated spreadsheet sessions that the system manages.

years and the external auditors to report on the internal business controls. This affects all public U.S. companies and other companies with a U.S. presence.

More generally, ESP is a secure computational framework for mission-critical enterprise spreadsheet applications.

As Figure 2 illustrates, when a normal user makes a request to assess a loan, the central server retrieves the super user's spreadsheet model, automatically spawns service sessions of the underlying spreadsheet application, and "runs" the computations against the inputs that the normal user submitted. Normal users cannot download the spreadsheets or open them interactively. In addition, the spreadsheets cannot reside on normal-user (client) machines, thus eliminating a major source of spreadsheet risk caused by the introduction (inadvertently or otherwise) of errors within the core of the spreadsheets. Depending on the specificity of the spreadsheet in question, the data, which the normal users submit via their Web browsers, could be prescreened (before being sent to the computational servers) using technology that is built into the Web application to validate data in specific fields. This minimizes another source of spreadsheet risk, namely, the entering of "nonsense" data.

The results of the computations are time-stamped and archived in the central database for auditability and reporting before they are sent back to the user's browser for display. Because the normal user has no direct access to the data flow, this minimizes the risk that arises from the unsanctioned manipulation of the results of the spreadsheet calculations.

ESP is complementary to other spreadsheet solutions that audit and track the creation and modification of spreadsheets. The spreadsheet auditing solutions ensure that the analytics embodied in the spreadsheet by the super users are correct. ESP makes the analytics available to everyone in the organization with the stability and infrastructure of an enterprise computing system.

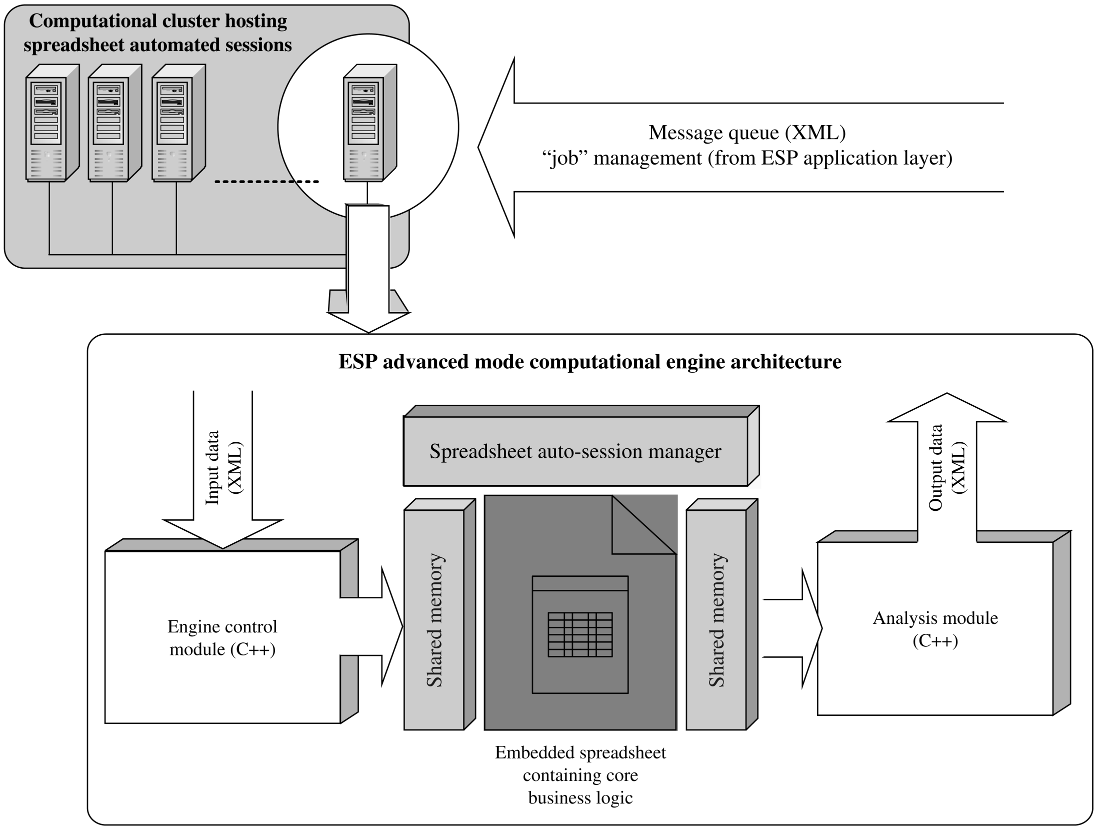

Computational cluster hosting spreadsheet automated sessions
Message queue (XML)
"job" management (from ESP application layer)
ESP advanced mode computational engine architecture
Input data (XML)
Spreadsheet auto-session manager
Output data (XML)
Engine control module (C++)
Shared memory
Shared memory
Analysis module (C++)
Embedded spreadsheet containing core business logic

Figure 3: This diagram shows ESP advanced mode as used by the SFS for Monte Carlo simulation. The auto-spawned spreadsheet session is embedded within a (C++ coded) computational framework via shared memory, with XML-based input and output data feeds to and from the application layer via queued messaging.

Detailed Architecture

As Figure 3 illustrates, ESP incorporates a code layer that enables the spreadsheet to be embedded within a robust computational engine.

Using this method eliminates various aspects of spreadsheet risk. For example, the following are applicable to the commercial real estate Monte Carlo simulation application that we presented:

(1) Complex numerical-method algorithms (e.g., matrix computations, which are central to many financial applications) do not risk being programmed in Excel (or VBA). Rather, best-of-breed compiled libraries (such as LAPACK for matrix computations) could be linked via the ESP advanced-mode interface.

(2) Because the ESP computational and data-management framework is separate from the Excel application, proper run time monitoring could be invoked. For example, if the spawned Excel session "hangs," the ESP framework could detect this, shut down the spreadsheet session, and inform the user of the unsuccessful computation. This eliminates any operating risk associated with accepting the results of incomplete spreadsheet computations.

(3) Spreadsheet-application computations are notoriously slow; this is painfully evident when attempting simulations. By contrast, using ESP increases the speed because it performs the bulk of the numerically intensive computations in the compiled C++ code; only the loan logic uses Excel. Typically, the ESP simulation framework is orders of magnitude faster than using Excel. Providing such performance advantages mitigates the risks associated with a user's temptation to "run just a few Monte Carlo iterations" (i.e., to accelerate the Excel-based simulations).

(4) For any calculations involving pseudo-random numbers, the issue of testability becomes more severe

because the random iterations might change from run to run. ESP enables full control of the pseudo-random "seeds" such that input and output data sets can be fully replicated; this contrasts with stand-alone Excel, which does not allow such control.

(5) With the separation between the data and the spreadsheet, ESP enables any important parameters (e.g., the number of Monte Carlo iterations or macroeconomic assumptions) to be "locked down" such that the computations are performed consistently across the portfolio.

Concluding Remarks

The implementation of this system has allowed Hypo to generate the results needed to comply with Basel II at the advanced level. To fulfill its Basel requirements, Hypo has presented the system to the German federal banking regulator (BaFin); its application for capital treatment under the advanced approach is proceeding as planned with the SFS.

The SFS, which has improved management reporting, has also provided insights into structuring new loans to make them less risky and more profitable. It has improved the efficiency of Hypo's internal processes and the way in which Hypo does business. The approach provides all the advantages inherent in an enterprise computational framework; yet, because spreadsheets remain at the core, it preserves the flexibility that business analysts demand.

The development and deployment of the system has also provided some general lessons for the implementation of complex analytics in enterprise systems:

(1) Retaining the central analytics in spreadsheets made implementing the project easier. Any changes by the expert financial analysts could be instantly uploaded into the enterprise system without requiring a translation from the prototype model into a programming language. In addition, because the experts had faith in the systems analytics, this reduced the time required for testing and debugging greatly. Keeping the analytics in spreadsheets allowed the financial analysts (the domain experts) to work within the computing environment in which they are most effective.

(2) The solution used the software tools for the applications for which they were most suited. The intensive processing required for creating, storing, and analyzing the Monte Carlo simulations was performed in C++ and kept outside of the spreadsheet. The financial analytics in the spreadsheet were then treated, in effect, as a callable routine.

(3) Keeping the data separate from the analytics allowed Hypo to make changes to the models without modifying the data structure. XML provided a flexible protocol with which to collate data from disparate sources and keep it separate from the models so that a change to the central model did not affect the data. The use of XML to gather the loan data required for the risk engine also created a rich central database that Hypo has since exploited for many other reporting purposes beyond risk calculation. One of these applications stress-tests the portfolio in which all the loans are subjected to a single downturn market condition, without invoking Monte Carlo simulation.

(4) By using spreadsheets inside enterprise-level systems, it is possible to eliminate many of the common problems of spreadsheets. ESP allows data validation in the Web interface, strong version control, accountability by date and time-stamping, and the archiving of analytical results. Only super users are allowed to open the spreadsheet; normal users interact via their Web browsers, passing data in and out, without any direct access to the spreadsheet.

(5) Building a robust infrastructure for the deployment of spreadsheet-based business logic has made it possible to create and distribute spreadsheet applications that would not have been used otherwise. The bank now has the ability to quickly create analytics and use them as applications in all areas of operations, not just risk management.

Today, Hypo has users in its global offices who use centralized, secure analytics that the bank's expert analysts maintain and trust.

References

Bank for International Settlements. 2005. International convergence of capital measurement and capital standards: A revised framework http://www.bis.org/publ/bcbs118.htm.

Grossman, T. A. 2007. Source code protection for applications written in Microsoft Excel and Google Spreadsheet. *Eur. Spreadsheet Risks Interest Group, 8th Annual Sympos.*, Greenwich, UK, 81–92.