MISYS
FINANCIAL SOFTWARE

The Misys view

Has the turning point to advanced risk management technologies finally arrived? David M Rowe argues it has.

Risk management beyond VaR and emerging **technologies**

Connect
Innovate
Expand

Contents

Executive summary

The need to supplement distributional techniques like value-at-risk with structural stress tests and scenario analysis is now widely accepted.

Unfortunately, persistent and seemingly intractable data and analytical fragmentation make implementing such analysis cumbersome and expensive. David Rowe argues that the application of information technology we all see every day can provide a solution, but it will require a qualitative shift away from the system architecture that has dominated enterprise computing for over 25 years.

1.0 Beyond the 1% threshold

The Global Financial Crisis that began over five years ago revealed serious shortcomings in the practice of financial risk management. Since the early 1990s, increasingly sophisticated mathematical and statistical models have been developed and deployed to measure exposure to financial turbulence. These were uniformly based on distributional analysis rooted in classical statistics and calibrated to comparatively recent samples of historical data.

Popular grandstanding to the contrary, such models have proven quite effective at measuring short-term fluctuations at the 99% confidence level. The problem with such models is that they say nothing about what 'lurks beyond the 1% threshold.'

It is now universally understood that distributional techniques like value-at-risk do not address the full spectrum of potentially adverse conditions facing institutions. Other tools such as scenario analysis and stress testing do not offer a foolproof solution to the analysis of possible systemic upheavals. They do, however, represent an important step in the right direction. Unfortunately, effective implementation of these concepts faces a serious practical obstacle.

First, stress tests involve shocks to market conditions that go well beyond the range of daily fluctuations used to calibrate and simulate today's VaR models. As such, accurate estimation of the impact of stressed market conditions requires full revaluation of the underlying assets. Market value changes based on extrapolation of Greek sensitivities are too unreliable to be used for this purpose. This means that the full terms and conditions of the assets under review are required for meaningful results.

The need for full transaction details aggravates the second challenge, namely the chronic failure of large institutions to solve the enterprise data integration problem. Despite decades of expensive efforts and vast quantities of corporate resources devoted to building and maintaining enterprise data warehouses, success remains elusive. Like Sisyphus and his stone, pushing the challenge of data integration to the summit of success only to find it repeatedly falling back to the foot of the hill is growing very old.

Death of the 'Golden Copy'

Attempts to assemble consistent enterprise-wide data warehouses have been based on the concept of Extract Transform and Load (ETL) technology. The idea was to develop flexible tools that allow business analysts to create the necessary metadata to support translation of information in multiple inconsistent formats into a structurally consistent central 'Golden Copy'. The essential shortcoming in this approach is inherent in the very name of the target result. A Golden Copy might sound valuable, obviously the name was chosen with that in mind. Nevertheless, 'Golden' or not, it was still a copy. Analysis applied to this central data repository was not dealing directly with system-of-record information.

"A Golden Copy might sound valuable, obviously the name was chosen with that in mind. Nevertheless, 'Golden' or not, it was still a copy. Analysis applied to this central data repository was not dealing directly with system-of-record information."

In particular, there has always been a variable degree of latency in transferring data from the systems of record to the central data warehouse. In many cases, the concept of creating an enterprise data warehouse was sold on the basis that it would reduce cost by eliminating diverse bilateral data transmissions between specific systems. This web or, more accurately, this hairball of bilateral system-to-system data flows inevitably developed piecemeal in response to emerging analytical and reporting requirements. Individual transmissions often were created by different developers at different times using a variety of programming tools, data formats and naming conventions and with documentation that differed widely in quality and accuracy.

These inconsistent data transmissions did, however, have one very important advantage. They were explicitly designed to meet the timing requirements of the receiving system.[1] Data in the central Golden Copy were often not available in time to meet these individual systems' processing deadlines. The result was that the purported cost savings from eliminating these bilateral transmissions never materialised. Too often central data warehouses would be useful for some periodic and strategic purposes but failed to be integrated into the daily time-critical information processing flow.

1. For example, counterparty exposure systems were typically tasked with providing updates based on the previous trading day's closing trades and market data.

"Properly designed, a SQL database allowed users to group, sort, filter, consolidate, matrix multiply and even mine the contents for hidden patterns using flexible run-time commands."

Diagnosis of the problem

In my view, the central culprit in this sad saga is the almost unconscious assumption that relational databases must be the central paradigm for storing and retrieving data. Progress in effectively organising data and analytics at the enterprise level demands eliminating, or at least significantly reducing, the role of relational databases.

Let there be no mistake, relational databases organised around variations of Structured Query Language (SQL) were an important advance in their day. The SQL protocol became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organisation for Standardisation (ISO) in 1987. Soon thereafter commercial database management system designed around this protocol began to appear. I can still remember being very impressed with the ingenuity of it all. Properly designed, a SQL database allowed users to group, sort, filter, consolidate, matrix multiply and even mine the contents for hidden patterns using flexible run-time commands.

This breakthrough in design was truly the culmination of decades of trial and error development of database management systems. It is important to remember, however, that SQL was developed within the technological constraints that prevailed almost thirty years ago. It is the epitome of understatement to say that there have been some significant alterations in those constraints in the intervening three decades. In light of this, a major reevaluation of our business data storage assumptions is long overdue. Fortunately, combined advances in several areas of information processing point to a radically different paradigm for the future.

2.0 A relevant personal experience

Perhaps it is the cynicism of age, but after decades of living with Moore's Law I am not easily bowled over by what technology can do. I did have such a reaction, however, in late 2011. I had been asked to update a ten year old paper entitled Organisational Balance for the EDP Audit, Control, and Security Newsletter. The idea for the original essay had been inspired by an op-ed essay in the Financial Times by their long-time columnist Peter Martin. Martin's essay dealt with the place of Jack Welch in corporate management history and was written in the immediate aftermath of his retirement from General Electric. Having failed to save a copy of that source of my inspiration I turned to Google to see if I could locate one on-line. I executed an advanced Google search for all the following words: 'Peter Martin Jack Welch Financial Times retirement'. In less than a second the results appeared with a PDF of my 2001 column.

My first reaction was to be mildly angry as I thought, "I don't remember giving Google permission to index my hard disk," since I assumed it was my local file they had found so quickly. Then I looked closer. The file they had located was the copy of my column stored on my consulting firm's website. In less than a second, Google had determined that this file contained all seven of the specified search words and was the most uniquely relevant document on the entire internet!

Working out the implications

This somewhat startling personal experience led me to rethink the entire paradigm of enterprise data storage. We too easily accept the view that enterprise data are far too heterogeneous to be stored in an unstructured format and still be analytically useful. Recognising the degree to which Google has succeeded in creating – and maintaining – a viable index for everything on the internet brings a radically different perspective to the magnitude and complexity of the data for even the largest institutions.

Of course, the 'Look what Google can do!' perspective is easy to push too far. We also need to recognise that web searches need only be suggestive, not definitive. We want most enterprise data handling procedures to be far more precise than this. On the other hand, we should not benchmark alternate enterprise data paradigms against perfection (even if this is our ultimate goal) but against the current sorry state of enterprise data capabilities. The sensible middle ground is to recognise that corporate data do require more formal structure than the total anarchy that prevails on the internet. Despite that, far less rigid structure is required than is currently demanded by the dominant relational database framework.

“A common enterprise risk management problem is to calculate the current value and potential future exposure for all trades with a particular legal entity.”

The prescription

The way forward is to move progressively toward storing system-of-record data in self-describing documents in a document store. This need not require a common semantic grammar for all files, although this is desirable insofar as it can be maintained. Multiple grammars can coexist as long as they have documentation that is complete, up-to-date and readily available to relevant applications. Clearly semantic inconsistencies need to be addressed, but the idea is to deal with the required semantic translation on an as-needed basis at the core of a loosely coupled, radically modular and fully parallelisable environment.

The central challenge of relying on a store of self-describing documents is how to access the ones required for specific tasks in an efficient fashion. This is accomplished by means of two high-level reference files that, because of their simplicity and structural stability, can take the form of relational databases. The first of these is a Document Name Server or DNS. This is a direct analog to the Domain Name Servers used on the Internet. The DNS holds simple records containing a document ID, which can be a simple integer, and a corresponding memory location – nothing more.

The second essential resource is a Document Index Server or DIS. This holds records of variable length with key/value pairs in the header and a constantly updated list of all document ID's containing that specific key/value combination. A common enterprise risk challenge will serve to illustrate the power of this environment.

An example

A common enterprise risk management problem is to calculate the current value and potential future exposure for all trades with a particular legal entity. Unfortunately, virtually every financial institution of any size trades with a specific legal entity in multiple locations on multiple systems and/or multiple versions of a common system, across multiple product categories. These trades are thus embedded in various relational databases with different table structures and inconsistent counterparty naming conventions. Thus, the first problem is to locate all the documents corresponding to trades with a specific legal entity.

Let's assume IBM Corporation is designated as 'IBM' in the corporate customer database. We know, however, that local trading systems often have other identification strings than those used at the enterprise level. To deal with this we need to create legal entity cross reference documents for each counterparty in each trading system that specify the corresponding identifier in the corporate customer database. Let's assume the form this document takes is as follows:

Relevant keys

DocuID means Document ID, typically an integer value

DocuType means Document Type

GLEI means Global Legal Entity Identifier

SSI means Source System Identifier

LLEI means Local Legal Entity Identifier

Relevant values

Two relevant valid values that can correspond to the DocuType key are:

'LECR' meaning Legal Entity Cross Reference and

'Trade' meaning a description of the terms and conditions of a capital markets trade

With these conventions, an example of a Legal Entity Cross-Reference Document would be as follows

DocuID = 47

DocuType = LECR

GLEI = IBM

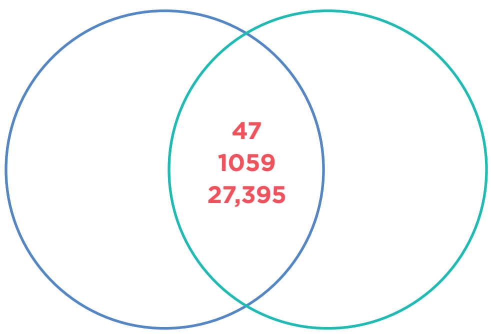
SSI = Summit2

LLEI = Int Bus Machine

Our first task is to locate all the relevant Legal Entity Cross-Reference documents related to IBM. That is, all documents for which DocuType = LECR and GLEI = IBM. To do this we extract two records from the Document Index Server, first the record for DocuType = LECR and second the one for GLEI = IBM. This gives us two lists of document IDs as shown in Figure 1.

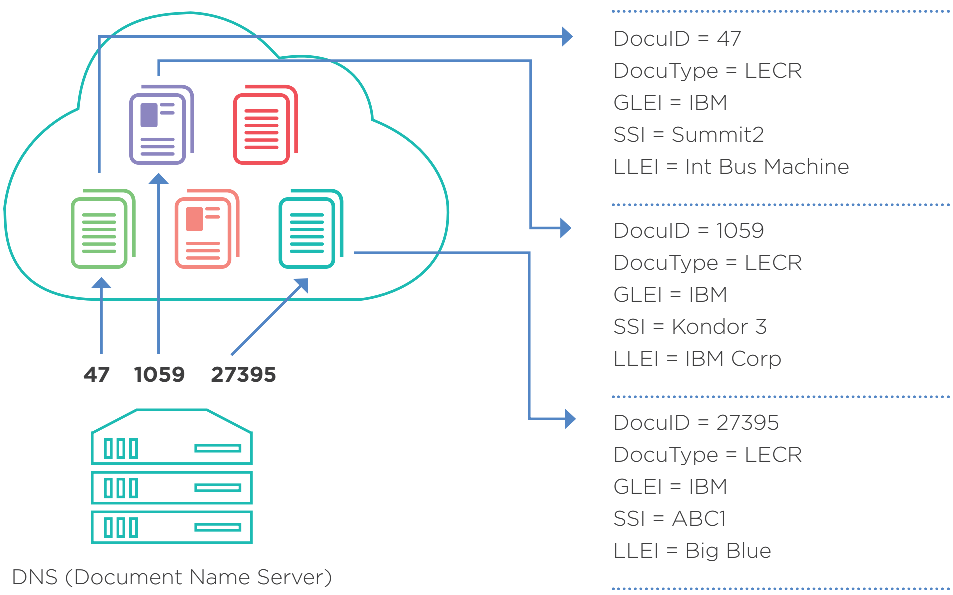
Figure 1

DocuType = LECR	GLEI=IBM
N(1)	M(1)
N(2)	–
N(3)	M(35)
47	47
N(5)	M(37)
N(6)	–
1059	1059
N(8)	M(405)
–	–
27,395	27,395
–	–
N	M



It is a quick calculation to compare the two resulting document ID lists and discover that there are three common values. We now know that we need to examine the three documents corresponding to document IDs 47, 1059 and 27395. We send these values to the Document Name Server which retrieves the three documents from the document store and brings them into memory as shown in Figure 2.

Figure 2



We now know how to formulate a query to locate all the trade documents for trades with IBM. Specifically, we need to find documents such that:

(DocuType = Trade and SSI = Summit2 and LLEI = Int Bus Machine)

OR

(DocuType = Trade and SSI = Kondor 3 and LLEI = IBM Corp)

OR

(DocuType = Trade and SSI = ABC1 and LLEI = Big Blue)

The relevant document ID's for each of these three queries are derived by finding the common elements in three DocuID lists from the DIS in a repeated process similar to that shown in Figure 1.

3.0 The power of agility

Now comes the real payoff from this type of enterprise data structure. Assume the bank in question buys a competitor who has been trading with IBM using a system from vendor XYZ. It has always been a big and expensive headache to integrate such new trades into enterprise-wide data and analysis for such things as counterparty credit exposure. In the proposed architecture, integrating such trades will require that they be exported into the document store. If the conventions for this are not consistent with those for existing systems, then new translation binders will need to be written to extract the data for submission to specific processes such as valuation or payment processing, as discussed below.

Assuring that these new trades are reflected in the enterprise analysis will require creating new Legal Entity Cross-Reference documents for each counterparty in the new XYZ system. One of these will correspond to IBM and could look as follows:

DocuID = 167344

DocuType = LECR

GLEI = IBM

SSI = XYZ1

LLEI = IBM Inc.

assuming this system refers to IBM as 'IBM Inc.' Now when we undertake the search for all documents containing both DocuType = LECR and GLEI = IBM, the intersection of the document ID lists from the Document Index Server containing both these two key/value pairs turns out to have four values rather than just three. The search for the relevant trades will automatically be augmented so that it becomes:

(DocuType = Trade and SSI = Summit2 and LLEI = Int Bus Machine)

OR

(DocuType = Trade and SSI = Kondor 3 and LLEI = IBM Corp)

OR

(DocuType = Trade and SSI = ABC1 and LLEI = Big Blue)

OR

(DocuType = Trade and SSI = XYZ1 and LLEI = IBM Inc.)

Once the document store is updated with the trades and the relevant Legal Entity Cross-Reference documents, searches for all the trades with IBM will automatically incorporate the additional trades from the new system!

> “It is important to be clear that this architecture is not magic. Like all other forms of data representation, self-describing documents cannot be guaranteed to be in any fixed semantic style.”

Semantic translation

It is important to be clear that this architecture is not magic. Like all other forms of data representation, self-describing documents cannot be guaranteed to be in any fixed semantic style. This implies that semantic translation will always be required to bridge the gaps and inconsistencies that divide independently created applications. Views differ on how best to handle this and a single approach is unlikely to be best for every analytical situation.

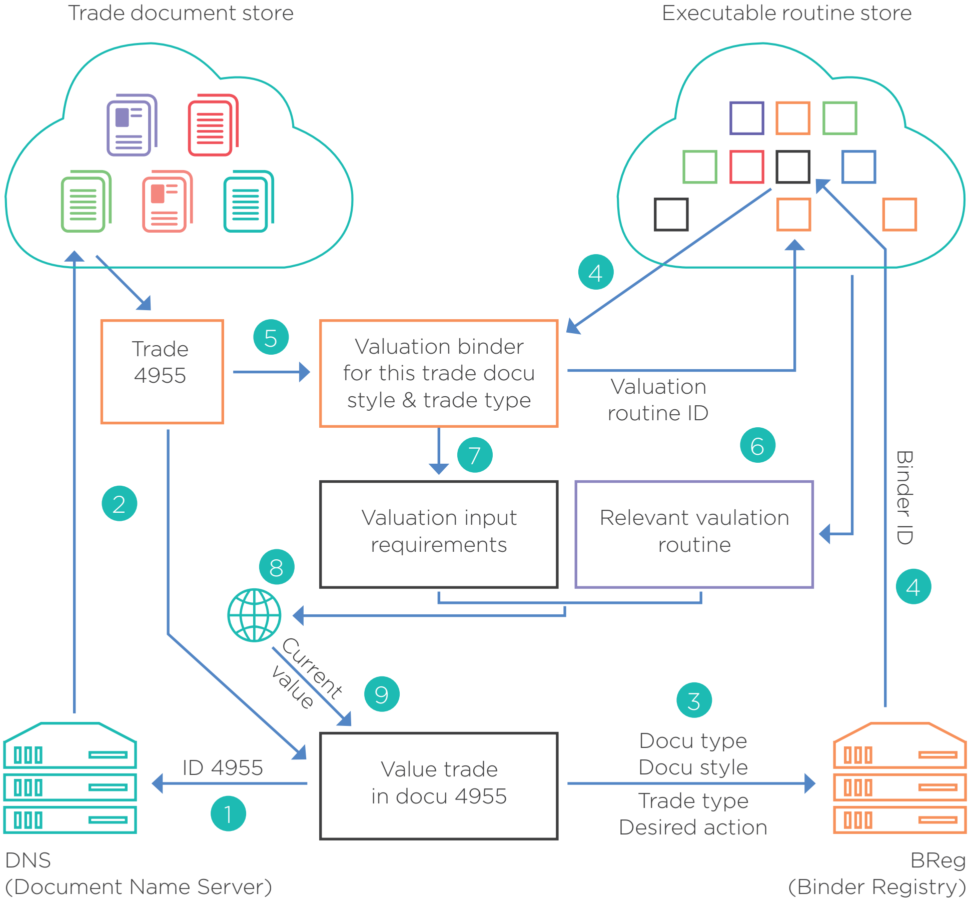
It is often necessary to perform standard actions on a given document. For a trade document, examples include:

- valuation - calculating the value of the trade given current market conditions and
- generating a payment instruction.

The trade details required to perform these actions are well defined and the associated routines can specify rigid rules on how these values are to be supplied. When such actions are required, one approach is to revert in all cases to the source documents created by the systems of record. Each trade document will have both a document type, ‘Trade’ in this case, and a sub-type such as ‘IRS’ for Interest Rate Swap. It also will have a document style that identifies the semantic framework to which it conforms. With this approach, a unique translation routine, commonly known as a binder, will be invoked for each unique combination of:

- document type
- document sub-type
- document style AND
- task to be performed

Valuation process schematic



1. The illustration begins when an application at the bottom needs to initiate a new valuation of the trade represented in self-describing document 4955. It sends the trade document ID to the Document Name Server.

2. In step 2 the document name server identifies the document, brings it into active memory (if it is not already there) and provides access to the originating program.

3. In step 3, the program sends the Document Type ('Trade' in this case) the Document Style (e.g. 'FpML') the Trade Type (e.g. 'Swaption') and the desired action to be performed ('Valuation' in this case.) to the Binder Registry.

“The financial crisis demonstrated that internal transparency is essential to prevent insolvency in a highly stressed environment.”

4. In step 4, the Binder Registry server determines the correct executable binder consistent with these four parameter values and loads it into memory.
5. In step 5, the binder accesses the trade document.
6. In step 6, the binder loads the valuation routine for the specified trade type (e.g. ‘Swaption’) Note this is normally not dependent on the document style of the trade document.
7. In step 7, the binder extracts the data required as input to the valuation routine and submits them in the exact format required by that routine. Thus the valuation routines can set their input requirements and formats without reference to how trades are stored. The binders take care of the necessary translation.
8. In step 8, the combination of the executable valuation routine and the trade specific data inputs are packaged and sent onto the grid for execution with available resources. (All the nodes will have access to current market data resources and lower level routines that may be called by valuation logic are provisioned to the executing nodes on the grid as required.)
9. Finally in step 9, the current valuation is returned to the originating program.

As illustrated in the above sequence, the binders extract the details from the trade document that are required to perform the desired action and pass them to the appropriate routine in the specified format. Thus, valuation of similar trades stored in different formats (i.e. different document styles) will call different binders but the necessary details can be passed to the same valuation routine.

The biggest advantage of this approach is that these actions always revert to the ultimate source for the trade details, or at least to a mirror image of the ultimate source. There is no question of keeping a translated copy in sync with the system of record. Nevertheless, in some cases the overhead required by this recurring translation may prove too great even in an environment of computational plenty. If paying the recurring translation cost is too onerous, it may be advisable to maintain a parallel document store in a format that complies with a common semantic framework. Of course, it still is necessary to execute a binder to extract the data required by the action to be performed, such as valuation, so the gain may not be great.

Risk calculations and intermediate results

Risk management calculations are generally more computationally burdensome than those required by the front office. This is because they often require multiple valuations under alternate hypothetical market conditions and, in the case of potential future credit exposure, on multiple future dates. Depending on the number of alternative market conditions to be evaluated and on the number of future simulation dates, the necessary number of valuations may be thousands or even tens of thousands of times greater than is required for an end-of-the-day mark-to-market process.

The usual way to handle this problem is to restrict the frequency with which current market conditions are updated to once or perhaps three times per day. The dramatic decline in random access memory (RAM) cost means that a massive volume of detailed simulation results can now be stored in active electronic memory. This allows risk estimates to be adjusted incrementally by performing full valuation on only the new trades. Low level simulation results still need to be re-aggregated, but this is dramatically faster than re-simulating all constituent trades, especially if the simulation results are stored in a denormalised format.

4.0 The target and the transition

“There will be some modest latency as the master index absorbs and reflects the new key-value pairs that it encounters, but this is minor compared to the serious update bottlenecks usually experienced by a massive data warehouse.”

Any transformation of the magnitude described here cannot happen overnight. It will involve a prolonged process extending over many years. The good news is that this new architecture can co-exist perfectly well with existing technology and can provide real benefits from the beginning. As noted earlier, projects to construct enterprise-wide data warehouses have been a common characteristic of corporate life since at least the early 1990s. These have generally failed to deliver many of their promised benefits due to lack of timely updates for many purposes. In part this was the result of most warehouse updates being performed in a serial batch process. In terms of the current architecture discussion, however, this approach also suffers from the same pervasive problems of inflexibility and the risk of regression failures as do all relational database applications. When the data warehouse has to be modified to accommodate new requirements, this inevitably causes problems for applications that already use the tables subject to change. The broader the use of the data warehouse, the more severe this regression failure problem becomes.

All this is tied directly to the monolithic structure of the data warehouse. By contrast, a modular document store allows inclusion of new enhanced documents with additional key value pairs without intruding into existing processes that use pre-existing documents. It also lends itself naturally to incremental, event-driven, real-time updates. There will be some modest latency as the master index absorbs and reflects the new key-value pairs that it encounters, but this is minor compared to the serious update bottlenecks usually experienced by a massive data warehouse. It also self-corrects itself as the master index application regularly prowls the document store for new or updated files to be referenced in its records.

Eventually we can expect various ‘silo’ applications to transition to use of a document store for their native information environment. In the meantime, ETL tools can be used to create such documents in much the same way that they are now often used to pull data into an enterprise data warehouse. The difference is that a well indexed document store is a far more flexible and adaptive environment. It can be queried effectively in unanticipated ways and can be modified far more reliably, with far fewer regression problems, than is true for a complex relational database.

The enterprise document store can offer immediate benefits in terms of timeliness and flexibility and these benefits will grow progressively as more and more data from more and more systems are added over time.

Conclusion

Swim with the current, NOT against it

The proposed architectural shift toward flexible self-describing documents leverages multiple trends in technology. These include:

- advances in search and indexing
- massive expansion in the scope for parallel processing
- increasing granularity in computing capacity
- elastic computing power supplied on demand
  - CPU cycles priced by the fraction of a second
  - analogous to water or electricity rather than dedicated hardware
- radically modular executable building blocks replacing closed monolithic applications.

"The dominant perspective that has governed computing technology throughout the past 50 years is one of computational scarcity. With the looming advent of the internet of things, we arguable are moving into an era of computational plenty."

The dominant perspective that has governed computing technology throughout the past 50 years is one of computational scarcity. With the looming advent of the internet of things, we arguably are moving into an era of computational plenty.

Computer pioneer Grace Murray Hopper famously berated her students about the folly of wasting a microsecond, which she represented by a 1000 foot coil of wire, in contrast to a nanosecond represented by a single foot of wire. In the largest sense, it is never wise to waste any resources unnecessarily, however cheap they may be. That said, as an economist I am acutely aware of the importance of relative prices.

What is in desperately short supply today is openness, flexibility and inter-operability in most enterprise computer software. This is because the obsessive focus on static efficiency seeking to minimise use of CPU time and storage capacity has embedded our data in multiple complex and incompatible formats.

The discussion above about storing intermediate results for risk calculations is an example of situations where economising on computing and storage resources will continue to play an important role. Nevertheless, even in that example, some relaxation in our focus on static efficiency is necessary if we are to make any real progress toward greater adaptability and improved enterprise-wide data access and analysis.

To make progress toward a more holistic and flexible information environment we need to establish the strategic goal of transitioning to a radically different underlying architecture than the one that has characterised corporate computing for almost 30 years. Fortunately there is a viable transition process that combines existing database technology and ETL tools with growing use of more flexible information storage based on modular documents and advanced indexing techniques. This will pay immediate dividends in terms of improved access to enterprise-wide data while leading progressively toward an environment that is far more flexible and dynamically efficient than anything we can hope to achieve in our present business-as-usual mode.

“To make progress toward a more holistic and flexible information environment we need to establish the strategic goal of transitioning to a radically different underlying architecture than the one that has characterised corporate computing for almost 30 years.”

About the author

David M. Rowe, Ph.D.
Senior Strategist – Risk and Regulation, Misys

David M. Rowe is Senior Strategist – Risk and Regulation at Misys where he serves as a spokesman for the firm on risk management issues and provides input on the strategic direction for risk management software applications. Dr. Rowe is a frequent contributor to Risk magazine, where he has written the monthly Risk Analysis column since late 1999, and has appeared at numerous conferences and seminars over the past 20 years.

Immediately prior to joining Misys, Dr. Rowe was operating his own risk management consulting firm, David M. Rowe Risk Advisory. Earlier in his career he worked at SunGard as Executive Vice President for Risk Management.

Before that he spent more than 25 years in the banking and economic forecasting industries, most recently as senior vice president of the Risk Management Information group at Bank of America in San Francisco. In that role, Dr. Rowe had executive oversight and responsibility for the design, deployment, maintenance and operation of market and credit risk systems for the bank's global FX, derivative and securities trading activities.

Dr. Rowe's previous positions include:

- Chief Financial Officer of Security Pacific Securities, Inc.
- Executive Vice President and Director of Research for Townsend-Greenspan & Co. (Alan Greenspan's economic consulting firm prior to becoming Chairman of the Federal Reserve) and
- President of Wharton Econometric Forecasting Associates in Philadelphia.

His industry positions include:

- Board member of the International Association of Financial Engineers (IAFE)
- Former Board Member and current member of the Finance Committee of the Professional Risk Managers' International Association (PRMIA)
- Member of the editorial board of The Journal of Operational Risk
- Former Member of the Editorial Board of the GARP Risk Review

Dr. Rowe holds a Ph.D. in econometrics and finance from the University of Pennsylvania, an MBA in finance with a concentration in money and banking from the Wharton Graduate School of Finance and Commerce, and a BA in economics with distinction from Carleton College.

For feedback or comments on this paper, please send an email to david.rowe@misys.com

About Misys

Misys is at the forefront of the financial software industry, providing the broadest portfolio of banking, capital markets, investment management and risk solutions available on the market. With more than 2,000 customers in 130 countries our team of domain experts, combined with our partner eco-system, have an unparalleled ability to address industry requirements at both a global and local level. We connect systems, collect data and create intelligent information to drive smarter business decisions. To learn more about how our Fusion software portfolio can deliver a holistic view of your operations, and help you to solve your most complex challenges, please visit **misys.com** and follow us **@MisysFS** on Twitter.

Corporate headquarters
One Kingdom Street
Paddington
London W2 6B
United Kingdom

T +44 20 3320 5000

25 / 0514

MISYS
FINANCIAL SOFTWARE