Paper AD02

Real Time: what is it and what are we doing about it?

Richard Phillips and Greg Nelson
ThotWave Technologies, Cary, North Carolina

ABSTRACT

Real-time computing is the term used to describe systems which are subject to constraints in time. Typical data warehousing processes are, more often than not, batch processes that are allowed as much time as needed to perform the required transformations and loading processes. Business Intelligence (BI) applications, typically built on top of data warehousing systems, demand exceptional performance by the user and people work hard to optimize for performance, yet performance problems rarely incur any serious penalty as BI applications tend to surface strategic information about the company – not the kind of information that drives the company today. This is contrary to what we believe can be achieved.

SAS Software is applied in many contexts and different scenarios. Common uses include as the driver for an ETL process (to a data warehouse/mart/hut/superstore) and as a just-in-time reporting tool. What may not be apparent are the opportunities for introducing real time techniques in the heavy lifting world of batch processing and ETL, analytics and of course, just in time reporting and notification (business intelligence).

Leading industry analysts (Gartner, Bloor among them) agree that real time business activity monitoring is the missing link in most BI strategies. We will discuss the ways in which SAS Software is ready (and unready) for this next technological challenge. In this paper, we will take inventory of the possible uses of real time processes throughout the SAS system. Here we will focus on what critical business issues seem best solved by real time agents that cleverly attract the notice of systems administrators and those that create value for the business consumer.

INTRODUCTION

There have been numerous articles written on the concept of real time (for example, Nelson and Wright, 2005). For brevity's sake, we will describe real time systems in the context of data warehousing and business intelligence (including advanced analytics) as those components in our information architecture that would be better served by delivery of results faster than we have today. Realizing of course that there is no real measure of what "better" or "faster" means but in practice it is what we hear all of the time. A daily process could be made more useful if it were delivered intraday. An hourly update of the customer scorecard would be better if it were only updated a little more often. What we really mean is that information should be delivered when we need it – just in time. If the data warehousing load process fails because it cannot read an Oracle table, then we want to know about it right then (not at 9:00am when the users are ready to start using the data). If a customer is about to churn because of a negative experience on a web site coupled with dangling in a call center wait queue for 5 minutes- wouldn't it be nice to know that NOW – when we have a chance to affect that customer?

From a purist's perspective, real-time computing is the subject of hardware and software systems which are subject to constraints in time. Some kinds of software, such as many chess-playing programs, are allowed as much time as needed to perform a computation such as the next chess move.

A distinction can be made between those systems which will suffer a critical failure if time constraints are violated (hard or immediate real-time), and those which will not (soft real-time).

Hard real-time systems are typically found interacting at a low level with physical hardware, in embedded systems. For example, a car engine control system is a hard real-time system because a delayed signal may cause engine failure or damage. Other examples of hard real-time embedded systems include medical systems such as heart pacemakers and industrial process controllers.

Soft real-time systems are typically those used where there is some issue of concurrent access and the need to keep a number of connected systems up to date with changing situations. Example: the software that maintains and updates the flight plans for commercial airliners. These can operate to a latency of seconds. It would not be possible to offer modern commercial air travel if these computations could not reliably be performed in real time. Live audio-video systems are also usually soft real-time; violation of constraints results in degraded quality, but the system can continue to operate.

It is important to note that hard versus soft real-time does not necessarily relate to the length of time available. A machine may overheat if a processor does not turn on cooling within 15 minutes (hard real-time). On the other hand, a network interface card may lose buffered data if it is not read within a fraction of a second, but the data can be resent over the network if needed, without affecting a critical operation, perhaps without a delay noticeable to the user.

REAL TIME BUSINESS INTELLIGENCE

Throughout this paper we talk about the concept of real time and how we might enable our well engineered SAS applications to take advantage of this strategy for delivering useful information just in time. We want to emphasize in our discussions that what we mean here is well engineered, enterprise level applications. This is not to say that the one-off application that reports on the financial metrics of a department would not benefit from the real time strategy, but in reality these are often skunk works projects that don't get the time or attention they require in the first place. Software engineering practices such as software configuration management, change control, proper requirements analysis and a robust quality management approach is usually rejected as elements not required for "our" application. Our perspective is this – we believe that regardless of how bloated or agile a software project is, there are key elements that should not be forsaken – even if a lightweight approach is used to deliver the application.

So regardless of how lightweight or "enterprise" class an application is, the fact is that most BI or advanced analytics applications are built with some common architectural elements... they include some sort of ETL process, a data store and a reporting layer. We think of these as sourcing, storing and disseminating data.

Figure 1. A Process Flow for Typical SAS Applications

A properly engineered ETL system falls into the category of a soft real time system – the design takes into consideration what happens if, say, one of the source systems is offline, or if you run out of disk space during an update to your data mart. More robust applications also tend to take into account a quality step which allows business analysts to evaluate the data before it is published to a wider audience (we refer to this as publish and approve – and example of this is referenced in Anderson, Nelson and Wright, 2003).

The system itself is therefore resilient. But the definition of real time computing is more concerned with business impacts; your ETL job may stop without errors if a 'jolt' occurs – but is there a non-technical driver which would increase the impact? Are entire business units dependent on the nightly refresh of the data mart? Will their activities grind to a halt if their analytic data is out of date?

BI systems often suffer from the 'silo effect' despite the best efforts of SAS developers. SAS has some great tools to help in integrating disparate data sources and massaging data into a consistent 'one version of the truth'. Too often, though, the process breaks down after this point - many separate reporting shops churn out tables and graphs for small scale and tactical purposes, often with no relevance to any super-ordinate goals.

SAS Software, in particular SAS 9, provides a huge variety of tools for producing reports. With this dizzying array of 'stuff' it is important to keep in mind the (in)equation:

$$BI_{more} \neq BI_{better}$$

In brief, the problem is the old one; sorting the wheat from the chaff, the needles from the haystacks, the sheep from the goats.

A fairly recent concept in the arena of Business Intelligence systems is that of Integrated BI. Analysts use this term to refer to techniques which place business logic and rules as an integral part of business processes. These ideas of Business Activity Monitoring (BAM) involve identifying the 'wheat' (or, more amusingly, 'sheep') and automating the application of the rules representing the business logic to the flow of information within the organization.

In short – to get full value from our investment in BI projects, we need to close the loop between the analyses we conduct on business data and the operational processes and systems that data represents. The figure below depicts the importance of integrating data from lots of sources and delivering it as a unified picture that tells a story about the business. It means taking data from historical systems (data warehouse), operational system and analytically produced results.

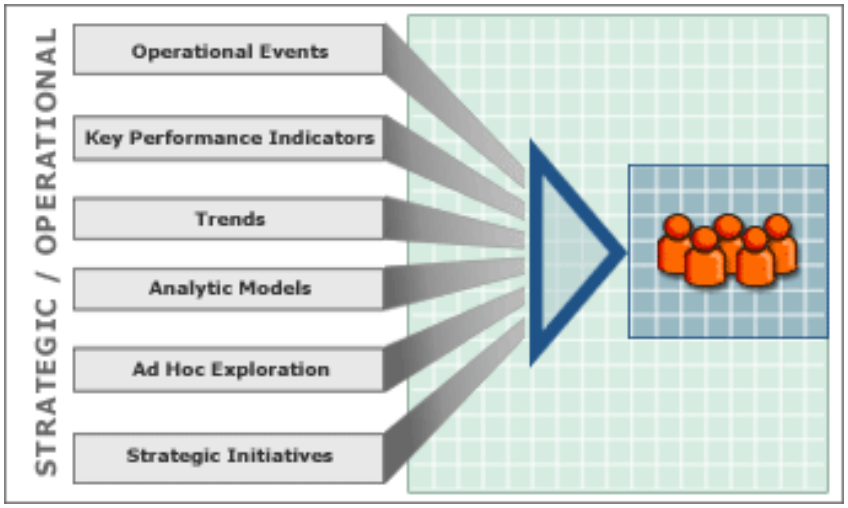


Figure 2. Real Time BI = Tactical & Strategic Analysis (Just in Time)

Integrating real-time information with persistent, historical facts and reference data is Real-Time Data Warehousing, an absolutely essential part of real-time analytics. Simply flowing instantaneous data to waiting devices is nothing new. Industrial process controls, telephone switches, even the thermostat in your house have all done this for decades. What separates real-time analytics from these process automata is the idea of *context*.

SAS TECHNOLOGIES FOR REAL TIME

For those of you that have used SAS for more than a few years, you may quickly lament/celebrate the fact that SAS is *a lot* of software. With strong roots in data management (data integration), analytics and reporting we wanted to begin to overlay what we saw as the opportunities for real time. To do this, we thot it might be helpful to first think about the activities involved in each of the three major components of a real time data warehouse: Data Integration, Analytics and Reporting.

DATA INTEGRATION

In the SAS world, data management usually involves accessing data from various source systems. These can include flat files, XML documents, RDBMS systems or other databases. SAS of course excels at getting at just about any data source known to man (and woman). The typical programmer would use tools like Base SAS (Data Step; SQL; Macros), SPDS and SAS/Access engines to extract, transform and load data into a data warehouse (or mart). With SAS 9, the programmer (and non-programmer) now has access to a suite of tools that help accomplish these goals – using such tools as Data Integration Studio, Data Flux, Data Surveyors, Information Map Studio and OLAP Studio.

As we think about the whole of ETL[1] processes in SAS we typically think of these as batch jobs that take periodic, non-volatile, snapshots of operational data, and clean, transform, and load the data into a repository.

[1] Note: we use the terms data management, data integration and ETL fairly loosely here to describe the steps involved in getting data ready to analyze/ report

REAL TIME DATA INTEGRATION: OPPORTUNITIES FOR SAS APPLICATIONS

For true real time data integration, this batch snapshot approach needs to be replaced by processes that continuously monitor source systems and capture and transform data changes as they occur, and then load those changes into the data mart in as close to real time as possible.

Real time data integration can occur at two levels within a system. The first opportunity for such data integration is within the data layer of the system, using traditional ETL processes for data access and extending these processes through real time data-driven triggers and embedded log processing. The second integration point is through software and processes which have no other purpose than to provide communication channels between system components – such as messaging services (*e.g.* MSMQ, JMS or TIBCO Rendezvous), web services via SOAP, or proprietary enterprise integration APIs such as SAP's BAPI.

These techniques can be used to form a notional 'data integration server', which can capture data changes from operational systems by monitoring either data events or applications events in the source systems.

The main challenge to SAS developers here is the learning curve associated with the SAS Integration Technologies middleware integration layer – SAS developers do not necessarily have the expertise necessary to build interfaces using this product, and enterprise software developers certainly have no expertise on interfacing with non-operational BI systems.

EVENT MANAGEMENT: OPPORTUNITIES FOR SAS APPLICATIONS

Traditional BI systems (as indicated by several diagrams within this paper) have a data flow architecture, characterized by the extract of large subsets from source systems and the wholesale processing of those subsets into the 'right shape'. The contrast we are trying to draw with event driven systems is that the latter are focused on detecting change at a detail level – in this case, change within source systems – and responding to that change. The exact nature of the response depends on the type of change that triggers it.

Many organizations use SAS analytics for modeling, simulation, statistics, and business intelligence. Compared to providing these capabilities on an ad hoc basis to individual analysts, there are a different set of challenges to delivering these results on a routine basis throughout an enterprise - a production system. ThotWave's event management solutions are designed to support production analytics via the capture and disposition of both technical and business events.

Technical events are the errors, warnings, and tracing information that allow support staff to keep production systems running smoothly. The SAS System detects many technical events and records them in log files. However, without an archive strategy for these log files and constant attention to their content, problems may well go unnoticed until the user community starts to have difficulties. The *thinking data*® Event System captures these conditions as they occur, records them, and provides proactive notifications on a subscription basis. In addition, ThotWave provides a powerful assertion-based API for defining custom events that have more context than standard error messages.

Business events are meaningful conditions in a stream of operational data. SAS has tremendous capability for reaching data wherever it resides and applying analytical techniques to recognize events in need of action. When combined with the *thinking data*® Event System, it's possible to manage these events, creating powerful Business Activity Monitoring solutions.

Often there is no clear distinction between technical and business events. Inconsistencies in data prepped for loading a data warehouse will require both technical and business users to correct the problem. The *thinking*

data® Event System provides an integrated architecture for handling both perspectives.

In the context of data management (or ETL processing), we see the following as areas of opportunities for introducing real time event management into your current SAS processes:

Data Quality Handling – As data problems are encountered in the ETL process, it is important to relay those issues to the right people. But often more importantly, relay the context of the problem along with additional information that helps the users solve the issue. Currently, the SAS programmer has at their disposal the SAS log information which may not be all that useful. What is required is a good way to handle the exceptions (notification, rules for what to do when the exception has occurred and a way to proactively test for those problems in the future).

System Management Events – not all of the events we encounter in an ETL process are a result of bad, missing or inconsistent data – sometimes we run into system errors such as write permissions to a file system, read permissions on a database, structural changes in a table or out of disk errors. These are critical to not only test and notify the appropriate people, but log the events to establish a history of service level agreements.

Recovery & Restart – when bad things go wrong, we need a way to restart our processes without having to clean out the staging tables or starting the process from the beginning. A logical approach to recovery should be implemented.

Scheduling/ Dependencies – most scheduled jobs that we find in the SAS world are simple %includes of all of the programs that have to be run in sequence. More advanced schedulers take into account the dependencies among jobs but rarely do we see intelligence built into SAS Data Integration processes that allow for the SAS developer to model out what sequence of SAS jobs have to be run and which are dependent upon one another. This also impacts our ability to recover and restart the SAS jobs if something goes wrong.

Exception Handling – as Ralph Kimball pointed out (Kimball and Caserta, 2005) "exception handling should not be a random series of little ad-hoc alerts and comments placed in files, but rather should be a system wide, uniform mechanism for reporting all instances of exceptions thrown by ETL processes into a single database."

Completion Status and Proactive Notification – throughout the entire system, it is important that we notify the right people that something they might be interested in happened. For example, if a number of new customers just churned, that might be a signal to management that something bad is happening. Technical data such as the number of new records added might be helpful for the system administrator to prevent potential problems.

Change Data Capture – in a typical data warehouse, we don't simple wipe and load the data every day. We want some sense of history that the operational system may not have. We need to know what happened over time and when things changed, how that affected things. Kimball refers to this as change data capture and in his book (Kimball, 1996) describes the concept of slowing changing dimensions and the various types of changes that can occur. Since change is an important component of understanding the context of our data, the metrics about what changes occurred during an ETL process are critical events that should be captured.

Metadata Creation – when we think of metadata, we often just think of the business or technical metadata. Rules which tell us how a field is used or its lineage. Often we miss the critical process

metadata about what happened when. By building in steps that allow this metadata to be written as the process occurs, users can safely rely on the results because they are informed of what happened, when it happened and how many records where written.

Logging and Eventing – In addition to being able to "listen" for data changes, one of the other areas in data management that should be improved within the SAS environment is logging and eventing. 'Logging' is the general term used to describe the systematic recording of specific types of data processing events. 'Eventing' is the specific application of logging technologies to business logic, applying user-defined rules, models and tests to flows of data. SAS now provides many logging mechanisms – these include the SAS log itself, server logs produced through the log4j standard java API, the SAS Integration Technologies event broker interface, the SAS/AF® SessionEvent component... the list goes on. What is lacking is, believe it or not, a 'SAS log' which behaves in precisely the way we need in order to interface business logic with a messaging service.

ANALYTICS

Advanced analytics is one of the hallmarks of the SAS system and has been for nearly 30 years. Of course analytics can mean a lot of things to a lot of people. While we won't try to describe all of the imaginable tasks one could perform with SAS as an analytic tool, we do know that variety of tools that these analysts have at their disposal. These include SAS/STAT, SAS/OR, SAS/ETS, SAS/QC, Enterprise Miner, Text Miner, Data Flux, Forecast Studio and High Performance Forecasting. Suffice it to say that over the years there have been numerous, imaginative applications where SAS has been used to unearth some previously hidden gem.

ANALYTIC SERVICES: OPPORTUNITIES FOR SAS APPLICATIONS

One trend that we see doesn't have to do with what problems people are striving to solve with SAS analytics, but rather how they use the tools to solve those problems. What we are seeing is that people want to automate or "productionize" these applications. Historically a statistician would develop a model as an analytic exercise and then send the results off to users to consume. When the model required changes, the statistician would make the changes and again rerun the model. Over time, what was once a one-off model is now being used over and over. Smart statisticians quickly became smart programmers – turning their creations into macros that could be parameterized for use in different contexts. When the parameters changed, the model could be self-validated without having to be rewritten. When exceptions to the model were encountered, notes in the logs were captured or emails were sent notifying the users.

This critical transition from ad-hoc analytics usage to production, routine runs of these programs represents a huge opportunity for real time computing. We see the opportunity as this: creation of personalized alerts or notification for exceptions. By using the analytics platform as a service, we can now envision operational applications which call out to these analytic services to run a very specific model just in time. For example, in the customer intelligence space, we've seen the this application model customer call center scripts based on not only historical data, but real time events while the customer is on the phone.

REPORTING

Finally, in the area of reporting we wanted to highlight what we thot were the opportunities for using real time strategies in our SAS applications. Traditionally, the SAS programmer has used a number of tools for productionized reporting. These have included Base SAS (Data Step; SQL; Macros; ODS), Stored Processes (create), SAS/IntrNet, AppDev Studio, Enterprise Guide, Integration Technologies, SAS/AF, SAS/FSP. Of

course the modern SAS user and report consumer can now use some of the SAS 9 approaches which include Web Report Studio (and Viewer), Information Delivery Portal, Add-in for MS Office and stored processes as surface in EG, WRS and the portal.

LOGGING AND EVENTING: OPPORTUNITIES FOR SAS APPLICATIONS

Of course the most interesting of applications we think can also benefit from the addition of real time "componentry". The area that we think have the biggest opportunity is when we think about reporting by exception. Reporting by exception is a departure from the way most people do reporting. They simply generate the report (as needed or in batch) and put it somewhere that can be accessed.

ThotWave has developed an easy to use library of SAS macros which allow SAS developers to very quickly integrate the business logic tests they are familiar with, and the enterprise messaging facilities associated with real time data warehousing activities.

Using tools such as this, ETL can be integrated with the BI layer by providing an opportunity for business rules discovered during traditional BI activities by traditional BI experts to be imposed at an earlier stage in the information flow.

As an example, a user may (using typical BI systems) create a particular report (run a SAS program) on a weekly basis. The report is complex and takes some time to produce, however, once printed; the person is able to see important information relating to the returns and complaints activity for the week just gone. Further analysis tells the user whether there are any significant trends developing that need to be fed back to the manufacturing department.

With a properly integrated BI solution, the same user defines the report once (registers their existing SAS code as a Stored Process), and then specifies the conditions under which it should be executed; say when complaints reach an unacceptable level, or when a problem is flagged with a particular batch of product. Simultaneously, an alert is generated and pushed to a member of the manufacturing team, giving them an early warning of a potential problem.

Not only is time being saved in needless, repetitive report production, but BI is being used to intervene with operational processes in order to reduce the costs of shipping poor quality goods (cost of handling returns, losing customers, etc.).

The key component in the scenario above is the ability to determine when the conditions that will trigger the report execution are met. It's not desirable to end up with another BI report that indicates which BI reports should be run. Rather, the condition for the trigger needs to be set in some repository of event triggers, and "automagically" checked during, say, an overnight process:

```
/* Check for exceptional data. */

proc sql;

       create table WORK.Extreme_Risks as

       select        RISK_OBJPATH,

                     RISK_CE format=2.,

                     EXPECTED_ANNUAL_LOSS format=dollar15.

       from &ETLTGT..RISK
```

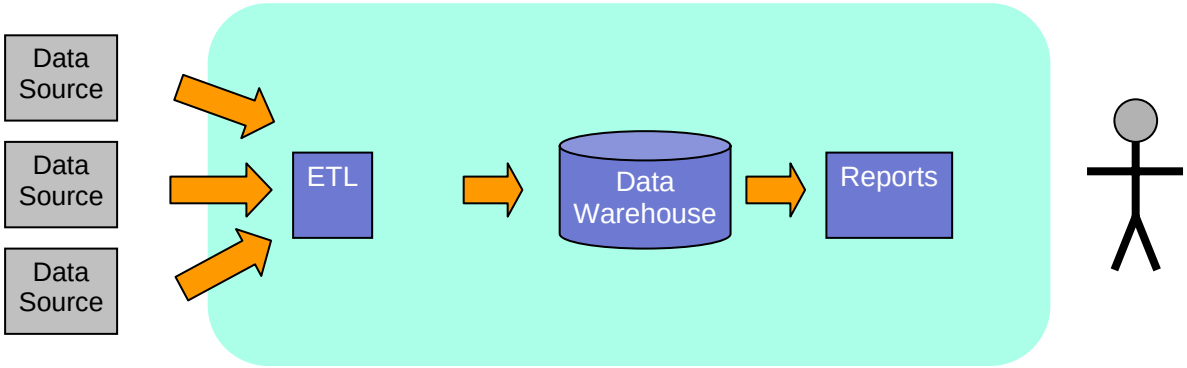
```
        where RISK_CE < 3
                and EXPECTED_SEVERITY = 6
                and PER_FILEID = 1
        order by EXPECTED_ANNUAL_LOSS DESCENDING   ;
quit;


%assert_empty(  WORK.Extreme_Risks
                LEVEL=INFO,
                TYPE=EXCEPTION,
                MESSAGE=Extreme Risks,
                ATTACHDATA=WORK.Extreme_Risks)
```

The 'other end' of this report distribution model is equally interesting. Individual events (using the ThotWave Event System) are created with context – they have attached properties such as 'level', 'type' and 'category' – which form the basis by which users subscribe to be notified of such events. This further abstracts the report producer from the report consumer. Customers of BI departments should be able to specify, for example, an interest in Events relating to 'Sales of Widget A' and receive automated reports based on the application of real business rules in a timely manner.

PUTTING IT ALL TOGETHER IN SAS

Often, the BI process around Data Warehousing projects is represented at an early stage in the process as a clean and clear one-way flow which serves to get the high level message across.



It's important to realize that the message represented by diagrams such as this *is* high level. Typically, marketing literature contains this kind of appealing image – also, too typically, the literature is taken as literal truth and is used to inspire an implementation project. After all, the figure above fundamentally represents the intent of what we need to achieve – right?

It should by now be fairly clear that the promise of true enterprise-scale BI involves not only having enterprise-strength tools and technologies, but also in implementing them in such a way that they operate on the same playing field as other enterprise applications.

Think through this scenario; consider your most important SAS application. If (heaven forbid) the proverbial meteorite struck your headquarters building tomorrow (naturally, during the hours of darkness - therefore leaving only fabric and equipment destroyed and with no loss of life), your disaster recovery strategy would smoothly roll into operation. At what point in the bootstrap of your business would your aforementioned SAS application be recovered? Would it feature high in the list, along with order processing and payroll systems? Or is it not actually on this list at all?

SAS is a powerful, high-RoI product which should have the potential to provide your operations with the best business intelligence you can muster. Business intelligence should be both the compass and the map for guiding the enterprise. Using techniques and ideas discussed above, we can sketch a BI system which ensures that the *right* intelligence is available to the *right* consumers at all points in the operational information cycle. 'Consumers' here includes people and processes; just enough information reaches interested parties, and BI services are available to *operational* systems in real time, enabling the in-stream, context-driven processing of individual transactions.

A BI System incorporating real time ideas is illustrated in figure 3.

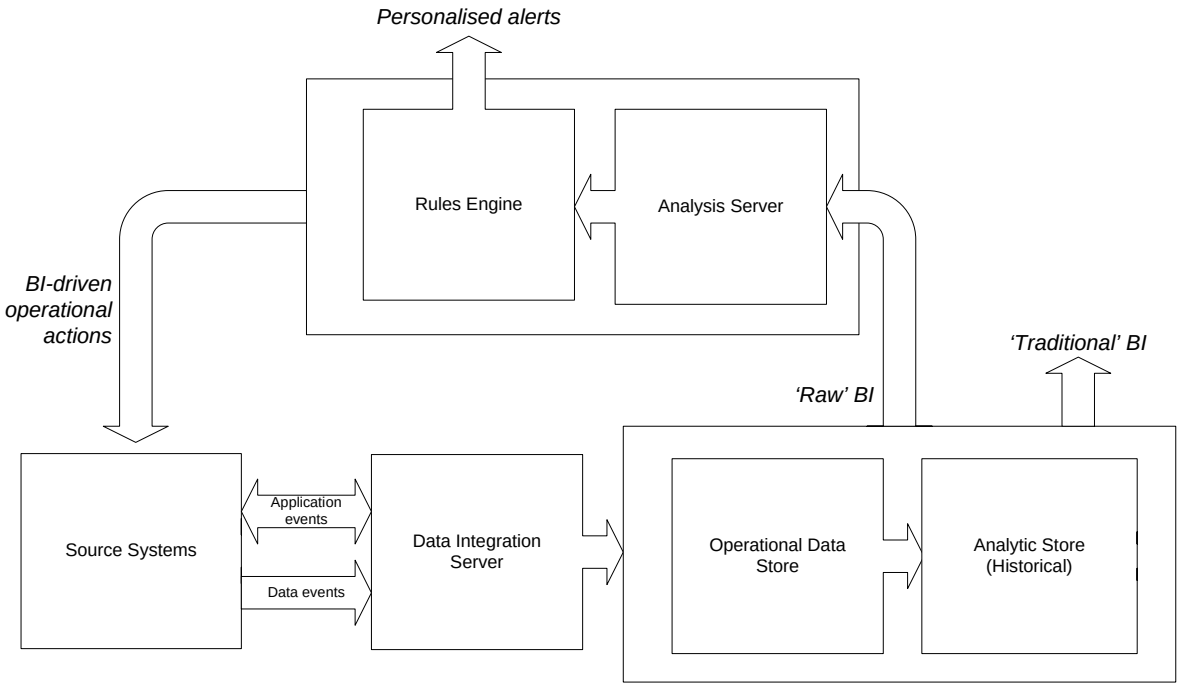


Figure 3. a real time decision support system

The 'new' parts of this system need some explanation:

A *data integration server* captures and transforms operational data, and loads it into a near real time data store - a data warehouse whose data currency is close to that of operational systems. One way this technology could be used is to build an operational data store that integrates customer data from all the customer touch points throughout an organization. Note that latency in the operational data store may vary from a few seconds to many hours, depending on business requirements. The important concept is that this operational data store is built not by periodic bulk extracts, but by transactions passing through the regular operational processing cycle. Various species of transactions trigger various corresponding changes in the operational data store, which may involve the passing of fragments of raw data back and forth. The operational data store could be used in a front-office application such as a call centre to provide current customer status

information for making day-to-day operational decisions. An example would be to identify high-value customers who are to be given priority service or a better discount.

An on-demand analysis server reacts to requests from users and applications for business analytics for tactical and strategic planning and operational decision making. The analytics produced by the server can also be used as input to a rules engine. An analysis server may be implemented using the SAS modules and components of your choice, together with some mechanism for communicating with the outside world – such as the middleware technologies discussed above. The main objective of the analysis server is to convert raw analytics stored in a data warehouse to actionable analytics that put the warehouse information into a business context that can be acted upon. For example, the raw analytic, *widget sales today = $20,000*, could become the actionable analytic, *today's widget sales were 30% lower than forecast*. This actionable analytic may then indicate that some kind of business action needs to be taken. The combination of raw number *and its context* makes the metric meaningful to BI 'consumers'.

A rules engine is used for operational decision making. The engine employs actionable analytics and business rules to generate and deliver personalized alerts and associated business intelligence to business users, or to generate and deliver action messages for processing by operational applications. The alerts and messages may contain notifications, warnings, recommendations, or suggested actions for handling a particular business situation. If widget sales fall by 30%, the appropriate business users could be informed, or the rules engine could generate an operational message to lower the price of widgets by 10%.

TOOLS AND TECHNIQUES

That's all very well – but how do you actually do it in SAS?

The short answer is – you can't, entirely. SAS software for the most part lacks a true application server architecture which would make much of the real time wizardry talked about above relatively easy.

There are things we can do, however, to leverage SAS throughout a real time BI architecture.

The track record of SAS software as a data integration server needs no particular discussion here – this functionality is well proven and extremely well tested and documented.

Perhaps the most obvious new role that SAS can fill effectively is that of the analysis server. The SAS 9 Intelligence Platform provides a reusable, metadata-centric set of general purpose SAS 'servers' – the Workspace and Stored Process servers. Real time demands for analytics (which are not directly served in the data layer of an application, say by OLAP) are a good fit for the SAS model of Stored Processes, particularly when used in the context of web services – which provide a generic API for metadata defined Stored Processes to any requestor. Stored Processes served through the BI Web Services component of SAS BI Server communicate through XML streams, making them easily interoperable with other applications and therefore useful as part of an integrated BI application.

SAS Integration Technologies provides software adapters to allow SAS applications to write directly to enterprise integration middleware. The notion of middleware has emerged as the architectural trick of choice to allow heterogeneous software to interoperate – it provides a transport for 'Application X' to notify 'Application Y' that, for example, *my work is done*, or *widget 0036 stock is depleted*. The applications of this technology are only limited by the capabilities of the message issuer and message consumer.

The keynote here is that while you may be tempted to create an entire, automated, closed-loop real time decision support system using SAS and only SAS – you probably shouldn't. With the increasing tendency of

middleware products to cleave towards common standards, SAS applications should be no exception. (One of the issues being that it should not take a SAS programmer to fix a SAS application, syntax errors aside). By adhering to open standards for application messaging, you increase the chances of success when interoperating with other applications.

APPLICATIONS FOR REAL TIME IN BUSINESS

What follows is a list of real-world examples where we have applied some or all of the ideas in this paper. Every one of these examples represents a case where we've created or taken existing BI applications and, by codifying the rules and logic and using real time computing techniques, made BI available to operational processes.

Financial Services	Credit card fraud
	Risk Management (Market, credit, ops)
Energy	Forward price projection
	Dynamic triggering
Retail	Pricing optimization, replenishment, markdown and inventory management
Government	CDC monitors Rx for epidemic outbreak
	Anti-terrorism port traffic
Consumer Products	Promotional effectiveness
	Spot promotions
E-Commerce	Targeted incentives at touch point
	Collaborative filtering/customization
Finance	Instant financial close
	Continuous planning
Executive	Real-time dashboards
	BAM
Healthcare	Service level optimization

CONCLUSION

There is an outdated view within end-user organizations that BI is a specialist activity that is typically undertaken by only a handful of business analysts. The rest of the business must then rely on these 'gifted' individuals to supply the necessary insight and information to allow decisions to be made and forecasts revised, etc. Such opinions need to be urgently updated, for whilst this was once true, things have progressed rapidly over the past few years.

Real time computing is a discipline with many well-tested techniques and well known constraints. As BI professionals we tend to approach problems with a batch processing, offline mentality. We hope we have demonstrated that the time is right to apply some of the principles found elsewhere in IT to SAS applications, and thereby increase the value of those applications to the enterprise. Particular opportunities lie in the area of data capture, analysis, and reporting.

Real time data capture involves a move away from the batch processing ETL model and towards a transactional approach to data warehousing.

Real time analytics follow; this is the application of reusable business logic to a real time data flow, possibly transaction by transaction but equally possible at an aggregate level.

Real time reporting stands alone and is possibly the easiest to implement. This requires reusable business or technical logic and a distribution channel.

In order for Business Intelligence to evolve from offline historical analysis to a key part of online operational decision making requires two changes:

- Data sourcing must change from batch snapshots to real time data integration

- The BI solution architecture must change from a data flow architecture to an event-driven architecture

REFERENCES:

Anderson, Julian, Nelson, Gregory S. and Wright, Jeff. 2004. "*A Day in the Life of an Analytical Warehouse*". Invited paper presented at the annual convention of the International SAS Users Group (SUGI), Montreal, Canada.

Kimball, Ralph. The Data Warehouse Toolkit : Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, February, 1996

Nelson, Greg and Wright, Jeff 2005. " *Real Time Decision Support: Creating a Flexible Architecture for Real Time Analytics*." Invited paper presented at the annual convention of the International SAS Users Group (SUGI), Philadelphia, PA.

Raden, Neil. Exploring the Business Imperative of Real-Time Analytics. Hired Brains, Inc. October, 2003

Butler Group. White Paper - Why Business Intelligence Projects Fail to Deliver ROI, December 2004

BIOGRAPHY:

Richard Phillips, Lead Global Consultant

Richard Phillips holds a B.Sc. (Hons.) in Ergonomics with a specialization in Human-Computer Interaction. Over 15 years working as a SAS developer, trainer and consultant in clinical research, general insurance and higher education provide him with a varied background and unique insight into enterprise-scale software deployments.

Greg Nelson, President and CEO

Greg has just celebrated his 20th year in the SAS eco-system. Starting out as a Social Psychology student doing statistical analysis then quickly moving into applications development. Greg is the President and CEO of ThotWave Technologies where he supports an entire organization focused on helping customers leverage their investment in SAS. Prior to ThotWave, Mr. Nelson spent several years in consulting, media and marketing research, database marketing and large systems support. Mr. Nelson holds a B.A. in Psychology and PhD level work in Social Psychology and Quantitative Methods.

About ThotWave

ThotWave Technologies, LLC is a Cary, NC-based consultancy and a market leader in real-time decision support, specializing in regulated industries such as life sciences, energy and financial services. ThotWave recognizes the difference between simply accessing data and making data work for business, and works at the juncture of business and technology to help companies improve their operational and strategic performance. Through products, partnerships and services, ThotWave enables businesses to leverage data for faster, more intelligent decision making.

CONTACT INFORMATION:

Your comments and questions are valued and encouraged. Contact the authors at:

Richard Phillips rphillips@thotwave.com

Greg Nelson greg@thotwave.com

ThotWave Technologies, LLC
2504 Kildaire Farm Road
Cary, NC 27511
(800) 584 2819
http://www.thotwave.com