Emerging Technologies Roadmap

A Decision Tree Guide to Understanding SAS® Strengths

Greg Barnes Nelson, STATPROBE Technologies, Cary, NC
Eric Brinsfield, Meridian Software, Raleigh, NC

ABSTRACT

Whether you are developing internal applications for corporate use or providing solutions for client use, one of the most daunting challenges facing technologists today is selecting the most appropriate technology and system design. Often the process is wrought with politics, gerrymandering, and a bit of black magic. Even within the SAS System, products compete for their place in the application architecture.

Based on over 35 years of combined experience in developing applications and an inexhaustible thirst for emerging technologies, we will share with you some of our experiences to try and help provide a framework to understand some of the most compelling technologies available today and emerging tomorrow. In this paper, we will outline which of today's problems the SAS system is uniquely qualified to solve and what types of problems SAS should be well positioned to handle in the future.

INTRODUCTION

As IT and analytical service providers, we are both faced with the challenge of selecting the best solution for solving a client's problems within the client's constraints and requirements. In the past, choosing software tools was easy for an experienced system designer, because options were limited. Today, however, we can find several solutions to any single problem that vary in cost, flexibility, speed of development, ease of use, robustness, and scalability.

Even within SAS software, we can find multiple tools or products to achieve the same objective. In addition, with SAS Institute's recent forays into open architecture with SAS/IntrNet, AppDev Studio and SAS/Integration Technologies, we find ourselves constantly researching and analyzing new and better ways to build software applications. For instance, now you can match the strengths of SAS software with the complementary strengths of other third party packages.

To help you select the right tools, we will first provide a brief discussion of SAS' evolution, covering SAS software's expanding capabilities over time, the growing competition, and the driving forces in today's business and technology. Building on that foundation, we will present a methodology to help organize your thoughts and choose the right tools for your software and database applications. Ultimately, we will attempt to reduce the decision making process to a decision tree, one which you can add your own branches and conclusions.

SAS EVOLUTION: PAST AND CURRENT STRENGTHS

We will mention only the major turning points in SAS software's evolution, which is an appropriate term, because even SAS produced a few dead ends, like SAS/DMI, System 2000, and SAS/CALC. We will focus on the success stories.

SAS' initial success hinged on two primary capabilities: easier data manipulation and statistical power. Compared to COBOL, Fortran, and PL/I, the SAS DATA step was a lifesaver to anyone who needed to work with data. The ability to read varying length records and merge data made SAS popular within the computer performance evaluation crowd and the analytical breadth became the tool of choice for statisticians worldwide. So, even though SAS was initially a mainframe, batch-oriented product, the foundation of powerful data management and statistics ensured SAS' continued success.

SAS Institute has consistently led the industry with innovative ideas and has adapted to new technologies and operating systems. For example, SAS introduced FSEDIT in 1981 to provide full-screen editing on IBM mainframes. Back then, that was a big deal. As new operating systems emerged, SAS Institute redesigned and rewrote the entire system several times in order to satisfy the demand. In other words, one of SAS Institute's greatest attributes has been its commitment to its customers.

Other factors that contributed to SAS software's continued success over time include:

- The macro language
- SAS/Graph with charting, plotting and mapping in the early 1980's
- The Display Manager, offering interactive development in Version 5
- A complete re-write with Version 6 to introduce true multi-platform support through it's Multi-Vendor Architecture (MVA)
- SAS/Access, availing new data sources for

analysis and reporting

- Relentless dedication to improving and augmenting their statistical prowess by hiring the best statisticians and developers
- SAS/AF, providing an application development tool within the SAS environment
- Providing true web-enablement through SAS/IntrNet and AppDev Studio in Version 6
- And finally, embracing an open standards approach to interoperability through SAS Integration Technologies and the SAS Business Intelligence Portal

In the beginning, SAS software was primarily a batch-oriented tool and was renowned for its ease of use in the mainframe world. When Windows arrived, SAS had to adapt to the new "point-and-click", fat-client architecture. Although SAS succeeded, the software never quite looked like Windows (according to many clients). Now that the Internet dominates and thin client is "in", the batch-oriented foundation of SAS is an asset when used in multi-tier architectures.

With the introduction of SAS/IntrNet, SAS/Integration Technologies, and AppDev Studio, developers can create applications that take advantage of the latest user interface technologies, while interfacing seamlessly with SAS software running on a server. The user can get the best of all worlds.

SAS remains a leader because of the software's powerful capabilities in data management, analysis, and reporting. The success of SAS in data warehousing and data mining exemplifies how SAS Institute can capitalize on its core strengths.

SAS EVOLUTION: THE COMPETITION

SAS software's competition has grown over the years for several reasons. First, SAS software is a much broader package now, which consequently leads to competition on more fronts. Not only does SAS have to compete in statistics, but also in databases, graphics, statistical quality control (SQC), operations research (OR), user interface development, and more.

In the current market of rapid innovation and fast delivery, niche players can quickly develop tools that compete with SAS in very small arenas. Even though SAS offers the tools that can be used to develop solutions in very narrow markets, the industry specialists can often deliver turnkey solutions that do not require additional development. In response, SAS Institute started developing vertical products, such as SAS/HR Vision and CFO Vision, which are based on other SAS products.

Another noteworthy reason for increased competition is convergence. Many of the big software companies are expanding into each other's turf. SAS entered the computing world from the analytical side and provided data management to ease that task. From statistics, they grew into other areas and a data warehouse powerhouse. Now, other large companies are also branching out into SAS' space. For example, Microsoft includes their OLAP services with their SQL Server database, which clearly competes in the same space with SAS/MDDB Server. Oracle has been expanding its analytical and data management capabilities for years, thereby increasing the competition with SAS.

SAS software competes well in their markets, but will need to stay nimble and convert more of its products to thin client. Overall, the thin client (Web browser) revolution should ensure SAS Institute's future success, as long as this technology goes well beyond just analyzing Web data.

SAS EVOLUTION: DRIVING FORCES IN BUSINESS AND TECHNOLOGY

Despite incredible advances in technology, the wide-spread adoption of best practices and methodologies, in addition to software development life-cycle tools, persistent challenges face us as technologists who try to solve real-world business problems.

As technologists, our goal is to bring technology to bear on problems where it makes sense. Where information can be processed to make it more understandable, computers play the role of the 21st Century plow-horse. Our desire to make our data cleaner, more accessible, or provide a richer context for decision making and delivery across the web or across the room has forced us to invest new tools and methods to conquer some common tasks. As with critical-thinking tasks in general, part of the real challenge is to know which problem we are solving. Often, this requires careful consideration of the factors that are most at play and which tradeoffs we are willing to make. Then, and only then, will the solution present itself.

Here, the authors don’t pretend to presume which challenges faced in corporate America are the same that are pressing to you, the reader. Based on experiences gleaned from numerous technology implementations, we have identified those that seem to persist over time and cross-organizational boundaries. We have summarized what we believe are the most pressing technology challenges facing us all.

Widespread Delivery of Information

Organizations competing in today’s economy are facing some major challenges, including:

- Global Competition

- Rapid Technological Change
- Product Obsolescence
- Organizational Downsizing
- Business Reengineering
- Empowerment
- Focus on quality and continuous improvement
- Measurement
- Interorganizational systems through partnerships, mergers and/or acquisitions
- Decidophobia caused by the overwhelming number of possible hardware and software solutions

As with many of these challenges come the tasks of getting information out to a wider audience, in shorter time, with fewer resources.

Everyone wants it on the web – whatever it is. We suspect that the reason is not because the web "browser" provides a particularly graceful interface or the graphical elements surpass those common to client/ server applications. In fact, quite the opposite is true. As application developers, we often find ourselves wanting for the event-handling methods of a true object-oriented, graphical user interface. Rather, the widespread adoption of the web browser in all its glory has more to do with the universal access to information behind the HTML.

The web has indeed become the great equalizer. Regardless of where the data came from or how it got there, the web displays it. As a technologist, business questions arise that typify this challenge. For example, "How do I get data out of a legacy system and into a format that everyone can use?" Answer: the web. or "How we disseminate this report to all 1000 sales reps each quarter without having to spend a fortune in printing and mailing?" Answer: the web. As we will discuss later, the answer may not always be "the web" – but probably.

Multiple Views of the Same Data

For the past four decades, organizations have been shoving data, records, files, contacts, accounts and every conceivable bit of information into some electronic form. The challenge clearly has not been how we get data into computers – although our friends that spend their careers writing transactional systems might disagree. Rather the challenge is how you get it out and make it of use to those that want or need to use it?

Those having written COBOL extract programs or Mark IV reports to access data tucked away in the bowels of a mainframe computer recognize this challenge. Creating processes that access and report on data can be fairly straightforward. Reusing the lessons learned from others and making the computer work smarter for us to that end is the challenge.

There are a variety of tools and technologies, both in the mainstream and in the labs that promise this reusability. That is, to be able to extract, transform, and display information over and over again for different purposes.

Application Integration

Enterprise Application Integration (EAI) is a fairly common buzzword at the time we write this paper. To the point, EAI has to do with getting data and applications to talk with one another. It's all about creating standards that allow for the magical transformation of cross-organizational and inter-company systems into a finely tuned machine that processes data on one end and creates knowledge on the other.

If EAI was a passing trend, it wouldn't be listed here as a challenge, nor would it be easy to solve. It is our contention that most of the technology work that goes on in organizations has more to do with integrating applications and data than bringing new technologies or innovations on board to solve unique problems.

Information Optimization

Once extracted, transformed, analyzed, and disseminated, information has to be useful. We spend an inordinate amount of time trying to boil information down into some mystic sauce that can be spread evenly over our decisions. After all, how does one really digest the 20 GB of raw web data coming in from our 40 web servers each day?

Helping provide information – not data - is a tremendous challenge that has forced "knowledge-workers" and "IT" to bridge the proverbial information gap. What has not been realized, however, is how we adapt our human capacity for understanding and learning – different for each of us – into one simple data byte that provides the kind of depth and relevance that each of us needs to make decisions.

Technology Optimization

Finally, the last challenge that we will discuss here is how we take advantage of technology in new and interesting ways to solve as many problems in a millisecond as possible. Fortunately, we have experienced a revolution in computing power that has enabled us to solve much more complex problems in much shorter times because of the pure, raw computing power at our disposal.

The challenge, however, still persists in at least two forms. The first is much more insidious than the second: our ability as humans to write programs in such a way as to optimize the number of instructions that a computer requires to solve the problem. This is easy to see if you've worked with a programmer who has been forced to work within the constraints of 640K of memory or to solve a hugely complex merge without being able to sort the records using available memory or physical disks.

The second form of this problem clearly contributes to the first – that is, computers are getting so fast, we don't need to worry about the same problems we did 20 years ago. However, these large-computing challenges still exist and, our belief is that they will continue to grow at a faster pace as the volume of data we are collecting continues to grow at exponential rates. This problem is one of physical resources. For example, in a real-life scenario, we faced a client who wanted to analyze all possible records in a marketing database and because of system constraints, there just wasn't enough time to physically process the data. In this case, sampling wasn't an option and all records were required as part of the analysis. In these cases, programmers will continue to face challenges head-on – either behind the keyboard or in budget meetings.

"BOILING THE OCEAN": COMMON THEMES

> *If you want to "boil the ocean" you might get a LOT of seafood, but you'd better be patient!*

In this paper, we have identified five major challenges that face us as technologists in our ultimate task of defining an appropriate architecture. Our goal has been to try to identify some themes that may persist across all of these challenges so that we might help provide some context for decisions that are made in the architectural phase of a software development project.

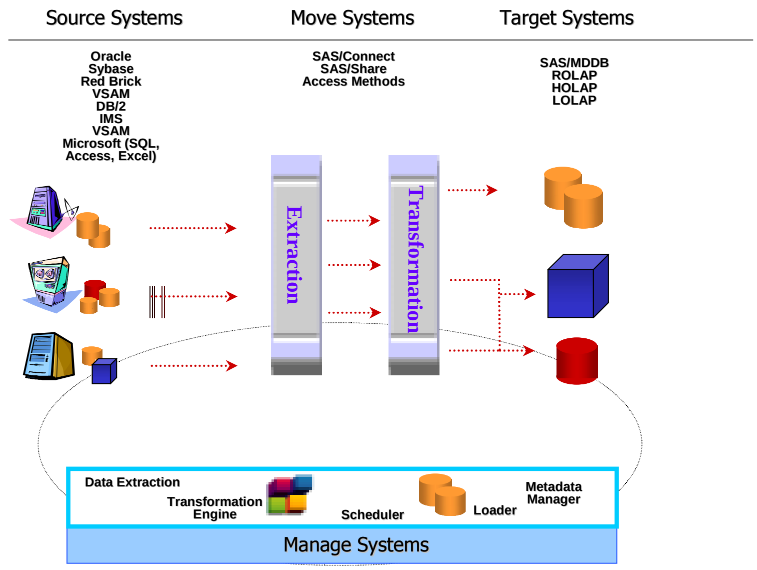
Albeit a simple model, we can break down most, if not all architectural problems to a three simple letters: S-M-T – or Source-Move-Target. SMT refers to the idea that every decision support project that we have identified typically has these three simple components.

The **Source** refers to where the data is coming from. In the simplest scenario, it might be a SAS data set. In a more complex, multi-tiered data warehouse, the source may refer to an intermediate MDDB or an Oracle table.

The **Move** in S-M-T describes the business and technical environment that enables data to flow from the source to the target system. This includes any transformations that need to take place as data are moved from their origin to their target.

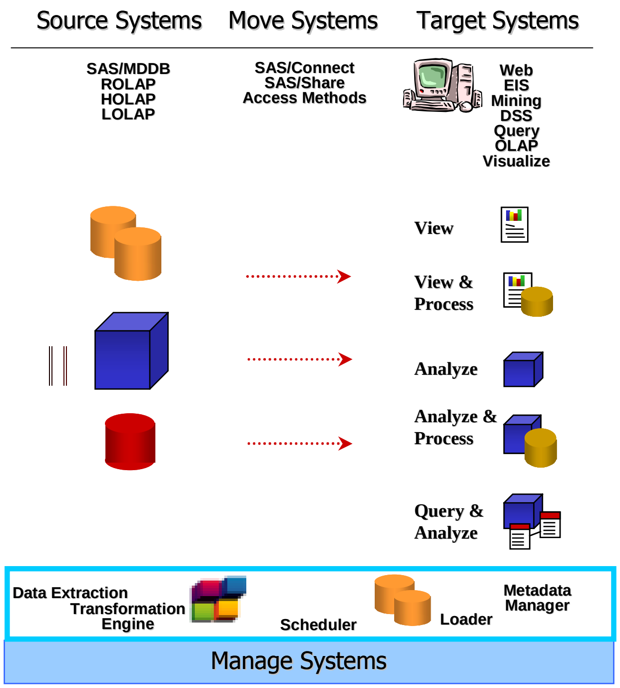
Finally, the **Target** in our methodology implies a platform where the newly integrated data reside. This includes the physical methods of data storage as well as the mechanisms used to access information.

It is important to note that there may be multiple S-M-T paths within any given architecture. For example, let's take the case of a simple data warehouse architecture (shown here).



In this case, we will have a Source-Move-Target mapping for each of the data sources listed on the left (VSAM, Oracle, MS Access, etc.) In fact, as we move data from left to right in this model, we have multiple targets for a single source data element. Note that the move systems take into account not only the physically movement of bits across application or even system boundaries, but also what transformations and analysis take place, and also the scheduling and metadata load processes.

We will reiterate that this architecture may consist of multiple S-M-T mappings. For each source, there will exist a minimum of one mapping definition. But equally important is the fact that targets in one context – such as a SAS data set or a SAS MDDB – may become the source system for another S-M-T mapping. In the diagram below, we illustrate this concept by showing how these same data sets, MDDB cubes and SAS data marts have now become source systems for our new targets – the web, thin-clients, intelligent clients, etc. The move systems in this context are the interfaces between what the user sees and how the data gets there. For web-based systems, this may be as simple as generated HTML from SAS processes to more complex dynamic applications using JavaServer Pages or Java with SAS/IntrNet.



The strength of this heuristic is that it allows the architect to break down a complex design into some reasonably manageable components. Once architected, decisions regarding platform (UNIX vs. NT); database (Oracle vs. SAS data sets); network topology and connections (Email vs. FTP vs. SAS/Connect) and exploitation strategy (SAS/EIS vs. Enterprise Reporter vs. Web) can more easily be attained.

UNDERSTANDING THE FACTORS

In our experience, we can think of S-M-T methodology as a roadmap, of sorts, that allows us to construct the architecture without being blinded by the complexity of the components. As we think about our challenges once again, let's think about them in the context of the S-M-T approach.

Challenge 1: Widespread Delivery of Information

The first challenge that we described above was one of getting the information into the right hands – regardless of physical location and the experience of the end user. In the language of our S-M-T approach, we see this as a problem primarily of Target. That is, how are people going to access the information (our Target) from Source systems?

Since the focus of the business question is how to disseminate information to a widespread audience, a practical approach to the design of the architecture is to actually move backwards from the Target. Once we have analyzed the tools and techniques that may be appropriate to the way in which people consume the information we are to provide, we can begin to understand the kinds of data that will need to be sourced in order to deliver it to the target system(s).

As we take this macro view, our architecture begins to take shape. Then, our focus can drill down into the details about where the data is coming from, how is has to be moved through the network and even where it might reside in either temporary or permanent data marts.

Challenge 2: Multiple Views of the Same Data

The second challenge we described above has to do with being able to reuse information in different contexts. In our methodology, this has to do with the entire S-M-T mapping process. That is, as we define business processes that require data to be extracted, we should be contributing to organizational knowledge by managing the metadata about the extractions and any implementation of business rules throughout the S-M-T process. Indeed, even the access mechanisms such as the application logic used to display a report or an analytic can be catalogued for future reuse by other candidate applications.

Challenge 3: Application Integration

As we described earlier, Enterprise Application Integration (EAI) is primarily concerned with getting data out of legacy, or even modern, operational systems to allow for the free-flow of information between systems. In the context of our S-M-T methodology, EAI just another source and we have to figure out what the target is going to be. The target might be another operational system – such as the case of extracting data from a customer database into a billing system. More often in the world of decision support, our target systems are web-enabled reporting systems.

Challenge 4: Information Optimization

Recently, one of the authors prepared a paper specifically describing emerging SAS technologies appropriate to this challenge (Barnes Nelson, 2000.) In this paper, he described the various messaging systems that get the right information, to the right people, at the right time. That is – making information matter. How we get information out of the bowels of corporate databases, ERP systems and transactional systems makes little difference to those needing to make decisions. Information optimization is all about creating target systems that get people the right kind of information – making it matter.

Challenge 5: Technology Optimization

The fifth, and final challenge we describe in this paper looks more like a technical challenge than a business problem. However, in the language of S-

M-T, the technical architect's job is perhaps the most critical when it comes to the Move systems. We haven't spent a great deal of time discussing this, but the move systems contribute greatly to the success of a good architecture and expose its weaknesses perhaps more clearly than any other component. We suggest that the move systems comprise most of the mysticism of systems architecture. That is, these are the things that glue everything together but are not well recognized in the software sales cycle. Most people talk about how we are going to get at the data and how we are going to display it. As it has been described here, this challenge implies that poorly designed systems – those that don't have the right glue - most often lack good move systems.

Addressing this topic in more detail, both authors have shared their experiences with system design, client-server architecture, and thin-client optimization in earlier presentations (Brinsfield, 1999).

SAS AS A SOLUTION: CHOOSING THE RIGHT TOOLS

In mathematical programming or operations research, the *principle of optimality* states that an optimal path is composed of optimal sub-paths (Stuart and Law, 1977). If you utilize S-M-T to decompose your problems into sequential sets of sub-paths from source to target, you can optimize the overall architectural solution by optimizing each S-M-T sub-path.

By applying the principle of optimality to an S-M-T representation of your intended system, the task of selecting SAS, non-SAS, or hybrid solutions will be much easier and certainly less daunting. You can simply focus your energy on one sub-path at a time.

SAS Institute provides a wide variety of products and solutions. SAS software can do just about anything that any combination of other software can do, but whether SAS provides the optimal pathway in a sub-path varies between situations. In other words, the optimal solution for one problem does not necessarily apply to other similar problems.

For this reason, we recommend building a set of questions that help you complete a problem-specific decision tree.

S-M-T: THE SAS DECISION TREE

As we have outlined here, the Source-Move-Target methodology really takes the complexity out of any large system design and creates a component view of the architecture comprised of a number of S-M-T mappings. Put together, these help complete the roadmap. As with any roadmap, there is no way that we can guide anyone through a series of questions to arrive at the perfect model. However, there are some questions that we, as experienced developers, system architects, and designers have asked ourselves as we help our clients through the Analysis and Design phase of a decision support project.

In the following sections we describe the kinds of questions that are relevant, the kinds of answers we might expect and the implications of these answers on our architecture – including SAS technologies that might be appropriate. These, in no way, imply that there won't be other, very specific questions that should be explored – but these should serve as a general guide to exploring the right architecture.

SOURCE SYSTEMS

Questions	Possible Answers	Implications	Possible Technologies
Where does the data reside?	Any possible platform.	SAS runs on over 22 different platforms. This may mean having SAS on a particular platform or simply having the right conduit to reach through and access it.	• With SAS, data can be accessed through native engines such as Oracle, Sybase or through general standards such as ODBC or JDBC. • In addition to native or gateway engines, SAS may not have to reside on the data platform if movement systems are constructed using technologies such as FTP, HTTP or even e-mail.
How many different data sources will need to be accessed? What are they? What versions are they? Under what operating systems do they reside?	Any possible data source (e.g., Oracle on NT Server)	Because SAS can access just about any data source known to man, this is information used in deciding platforms and products. Columns may need mapping if different data sources store comparable	• SAS/Access products may be required as well as other connectivity products that can communicate between SAS sessions running on multiple hosts.

Questions	Possible Answers	Implications	Possible Technologies
		information with different names or data types.	
What access methods can be used to get at the data?	A variety of engines and access methods such as SAS/Access to Oracle, DB2, Sybase, SQL Sever, VSAM, ODBC, DDE, OLE DB.	Is there a native method to access the data, or will transformation and/or movement routines need to be constructed?	• SAS/Access engines are appropriate for most native data sources.
How is the data organized?	Relational, hierarchical, flat files	Significant work may need to be done to read/ access the data and convert it into something usable.	• Again, SAS/Access engines allow views into most relational databases. • SAS DATA step processing may be required for other data sources or for advanced manipulation.
How clean is the data?	Very clean (accurate) – dirty	Operational systems do a good job of making sure the data gets into the system cleanly, or do they have to clean out records and/or fields that are meaningful?	• SAS DATA step or SQL processing may be required for file clean-up.
What data integration has to occur?	Merging with overlay data, driver files, match-merge for record clarity, etc.	Can it be integrated with other sources on the fly, or does it need to be staged somewhere for further processing?	• Again, SAS DATA step or some other processing may be required. • SAS/Connect can be used to help move data to a staging area for further processing. • SQL Pass-thru can be used to create temporary native relational database tables for merging before movement occurs.

MOVE SYSTEMS

Questions	Possible Answers	Implications	Possible Technologies
What is the networking architecture?	DecNet, TCP/IP, Novell IPX/ SPX, AppleTalk, Banyan Vines, Windows NT/2000	We need to be aware of how we are going to access data on one system from another. In addition, special architectural constraints may force us to transform data in different ways (e.g., EBCDIC vs. ASCII)	• SAS/Connect can be used to help bridge architectures and take care of translation issues. • SAS/Secure can be used to secure the wire as the movement is made across physical machines.
What SAS products and versions on are installed on the client? Server?	Any possible configuration of SAS products and platforms.	It is often easier to work within the constraints of an existing architecture or support infrastructure than to introduce new hardware and/or operating environments.	• SAS runs on 22 different platforms – anything is possible.
What is the client/ server configuration, if any?	SAS/ Connect, SAS/Share, remote access via FTP, URL universal ODBC.	Work that has been previously done on architecture setup such as accessing relational databases and communicating across machines may contribute to an easier implementation.	• Since SAS runs on so many platforms and can have so many different configurations – again, anything is possible.
How do we get the data from source to target?	E-mail, file transfer, SAS/Connect, URL, tape	Physical challenges may exist that can be overcome by introducing more efficient methods such as processing data on a remote machine and transferring the subset of data across the wire.	• A variety of configurations for compute, application and data services can exist. Deciding on the most appropriate configuration can be calculated as a series of tradeoffs (network speed, bandwidth, processing power, security, off-loading requests, etc.).
What transformations have to occur? (e.g., summarization, calculation of business rules)	Possible calculations such as defect per unit, profitability, oil consumption.	As with integration, the complexity of the transformations may require that data be staged outside of the extraction process for complex calculations.	• Special calculations can be aided by products such as ETS, QC and OR.

TARGET SYSTEMS

Questions	Possible Answers	Implications	Possible Technologies
Who are the users and what technology(s) are they comfortable with?	Novice users, business analysts, SAS programmers, executives	The target (access) systems should be relevant for the audience.	• Analysts: SAS, Enterprise Guide, web-based query tools (Java). • Executives/ Novices: Web-based reporting tools (SAS/IntrNet, JavaServer Pages).
Do we have the support and infrastructure to deploy this globally on the web?	Yes, IT and/or a department has the technical talent to take over the technology. No, the system must be maintained with existing staff or outsourced.	Introducing technologies into organizations that have a good fit within the technical organizations that support them is key to a long-term success.	• Organizations who don't have a strong support staff with knowledge in SAS, may find base SAS easier to learn than say, SCL. • Organizations may have a strength – such as Java or ActiveServer pages – SAS/IntrNet applications can be built with a variety of complementary technologies.
Can the information be adequately protected if delivered in (a) paper, (b) client/ server or (c) through a web interface?	Yes/ No.	Security – even if never discussed in the context of requirements always needs to be discussed and risks evaluated.	• If security is critical in a web-based application, existing standards may dictate which specific technologies can be used. • If systems require delivery of personalized content – choose web technologies that can provide that such as SAS/IntrNet with session variables (V8) or Java & JavaServer Pages to deal with server-side persistence. • If encryption is required between machines, SAS/Secure and SAS/Share may be needed.
Where does the data live (or can be moved to) that will need to be surfaced through these interfaces?	Any possible data source.	Since SAS can access just about any data source, this is information used in deciding platforms and products.	• SAS/Access products may be required as well as other connectivity products that can communicate between SAS sessions running on multiple hosts.
What types of analysis will need to be done – canned reports; lightly summarized, ad-hoc analyses, query tools, or will power users require the full power of SAS (or Enterprise Guide)?	A variety of reporting options may be discovered.	Reports that already exist in one form (e.g., paper) are the ideal candidate for review and analysis to determine S-M-T mappings. Be careful of expecting too much from the web or novice SAS users – interfaces that require a tremendous amount of interactivity also requires a tremendous amount of development and time.	• Can the tasks be handled with off-the-shelf products such as Multidimensional Report Viewer or Enterprise Guide? • Specialized analyses may require special SAS products such as QC, OR, IML or ETS.
Has the application (a) scope; (b) requirements, (c) architecture been defined (are there assumptions in place already)?	Yes/ No/ Partially	There may have been a tremendous amount of work that has already been completed in defining the S-M-T mappings for your report.	• Some architectural decisions may have already been determined – limiting your choices.

As we think about the answers to these questions, we start limiting and defining the architecture. The questions of approach, architecture and methods of access, transformation and information delivery becomes almost self-documenting as we build our requirements. It is worth noting here, that while we have described the bulk of our model, there exists an entire management layer that addresses issues such as security, refresh strategies, scheduling, load balancing as well as the management of the delivery mechanisms, which account for everything from print queues to wireless transmission of data.

While we have made every attempt to create a usable taxonomy that should carry over into your own work, we don't want to over-simplify the process and the value that an experienced system architect can have.

EVALUATING BUSINESS PROBLEMS IN THE CONTEXT OF S-M-T

Today, at SAS Institute, one of the primary initiatives has to do with the acquisition, storage/ warehousing and exploitation of data collected through vast web services. The S-M-T mapping for this business problem is an example of a technology fitting well within the analysis and decision support realm of The SAS System.

Whether the problem domain is collaborative commerce, business intelligence, or the next big thing, SAS Institute provides a solid foundation for solving a variety of problems. As we peer into the future of SAS and evaluating e-business strategies – be it digital exchanges, corporate portals and other e-commerce applications – the basic questions of integration, reusability and web-enablement become the architect's play-ground. Here our methodology plays an important role in defining fit and function.

CONCLUSION

SAS brings a tremendous amount of power and flexibility to both client/server and web-based applications. As we have seen in this paper, SAS software brings a number of compelling benefits to both developers and the organizations they serve. For the developer, SAS offers a rich toolset for communication across application and organizational boundaries. We are able to structure data in the context of meaningful business processes, benefit from technology reuse and leverage the architecture for its intended purpose. As we are challenged with evaluating new technologies and how/ where they fit into our organizations, sometimes breaking these problems into manageable chunks – such as the methodology presented here – makes the evaluation process much more digestible.

ACKNOWLEDGMENTS

The authors would like to sincerely thank several individuals for providing their thoughtful review of this manuscript. Specifically, we would like to thank Cathy Brinsfield for her insights and editorial massaging.

REFERENCES

Barnes Nelson, G.S. (2000). *Messaging Systems: SAS® Tools for Internet-Based Communications.* Paper presented at the Annual Convention of the International SAS Users Group. Indianapolis, In.

Brinsfield, E. (1999). *SAS Client-Server Development: Through Thick and Thin and Version 8.* Paper presented at the Annual Convention of the Southeast SAS Users Group. Mobile, Al.

Dreyfus, S.E. and Law, A.M. (1977), *The Art and Theory of Dynamic Programming*. Academic Press.

CONTACT INFORMATION

The authors may be contacted as follows:

Gregory S. Barnes Nelson
STATPROBE Technologies
117 Edinburgh South, Suite 202
Cary, NC 27511

Internet: greg.barnesnelson@statprobe.com
Personal: gregbn@ix.netcom.com
Web: http://www.statprobetechnologies.com

Eric Brinsfield
Meridian Software, Inc.
12204 Old Creedmoor Road
Raleigh, NC 27613

Internet: merecb@meridian-software.com
Web: http://www.meridian-software.com

For the latest version of this paper, please refer to: www.statprobetechnologies.com/articles.html