Paper 139-26

Avoiding eOverload:

Personalizing Web Content through Security, eIntelligence and Data Mining

Gregory S. Barnes Nelson STATPROBE Technologies Cary, NC

ABSTRACT

Today, the worlds of high-tech with high-touch are quickly converging. The landscape of this new frontier in business intelligence and customized content has challenged the traditional models of delivering knowledge to diverse constituents. To make the most effective use of these delivery methods, business needs detailed intelligence on how people use information from these sources. In this paper, we will explore the analytical techniques, technologies and tools used to answer real-world business questions, accelerate knowledge transfer, and foster more profitable relationships with customers, partners, employees and suppliers.

Specifically, we will explore the business and technology underpinnings of personalization. Then we will examine the analytic and technical approaches to understanding that data as delivered through eIntelligence, click-stream and log analysis techniques.

INTRODUCTION

This paper tells the story of personalization and measuring business value from a developer's perspective – that is, as a pragmatist who spends his time developing strategies for personalization and their implementation through web technologies. Here we discuss the concepts behind personalization and address some of the challenges that people will find as they unearth the hidden mysteries of web log and click-stream data.

According to a Forrester Research report, "As the general content of the Web gets broader, individuals will cease aimless surfing activity and gravitate toward sites that deliver products and services customized to their needs. Sites must plan now to respond to this expectation or risk being left behind as the Web changes to a personal medium."

In the first part of this paper we will describe the business and technology reasons for personalization. Although we will discuss this concept in much more detail later, we can think of personalization as an approach to delivering content dynamically, or "just-in-time", so that the content is specific to a user or group of users. Next we explore how we can begin to understand how people use information on our web sites through measurement. We will discuss various factors about the web site experience – from the perspective of the visitor as measured through web logs and click-stream analysis.

Although this paper is intended as a resource for people who have, or are creating, personalization strategies, we recognize that many questions go unanswered. Because the web has spawned numerous approaches to solving some of the problems of a stateless environment, it also creates problems as we attempt to monitor its usage. That is to say, the technologies that we use to make web sites more dynamic and personal often create complexities when we try and figure out what people have done on our site. So the story is really a story told from two perspectives – one describes how information is delivered and the other, how we describe or analyze patterns of activities when the content is dynamic.

THE NEED FOR PERSONALIZATION

The need for personalization is paramount in the world of eOverload. Information is coming at us from all fronts – radio, billboards, web phones, wireless PDAs, the ever-looming Internet… all trying to vie for our attention. Developers of web content, whether they are marketing, technical, or hobby, all want their content to be noticed.

Site differentiation is key in the world of *e*Overload. One author (Ramsey, 2001) has suggested the following as key to obtaining site differentiation:

1. Value added information
2. Frequent, useful content updates
3. Interactivity
4. Information personalized for specific users or user groups.

The first two of these is likely the responsibility of the business side of the house – or the content experts. Technology, however, can play a significant role in making sure that the visitor is engaged once they are on the site. Using client-side tools such as DHTML, JavaScript, and Java, we can make the site extremely dynamic and interactive. Finally, our last goal – personalization – allows us to create a true one-to-one relationship with a visitor.

Web personalization allows you to have a Web site that tailors Web content to a Web user's preferences and other profile information. In addition, a personalization system logs every Web page displayed to every user so you can develop a "clickstream" view of what they saw, when they saw it, and for how long. Just imagine what you could learn about your audience with a complete understanding of their Web usage.

In our net-centric world, we have seen these in action – everyone vying for your attention – sending invitations, gifts and even eCards. In the early days of web, advertising banner ads were unusual and effective. The thought of having 6 million people look at your ad each time they logged on to check their email was fascinating. The early marketers capitalized on this new media and found it very profitable. Just by measuring click-throughs (whether a person actually acted on the ad they saw), we could evaluate the effectiveness of an ad campaign. But soon, like the advertising we see on park benches and buses, the effect had diminished.

Unlike the simple site maps of even just a year or two ago, most modern sites are not static. For example, it is not uncommon that product taxonomies change regularly, marketing campaigns or new service offerings for content updates and even segmentation of content to different groups of users. Often content is personalized – the way that information is presented, the method of delivery and

even commerce models have been impacted our expectations of content (e.g., pricing models.[1])

So what do we mean by personalization? Is it just adding the "Dear Greg" on top of a web page? Personalization can mean a lot of things in the high-tech, high-touch world of eCommerce, here we adapt the following definition from Ramsey (2001):

> *Personalization means strategically targeting specific users or groups with content relevant to them, delivered at the time and manner most appropriate to them.*

By delivering dynamic interfaces to information so that the right people have access to the right information at the right time, we can solve some of the key challenges that marketing faces.

The Business Need for Personalization

As we take inventory of the types of models that we have in the web world, we can find at least four categories that describe our use of personalization technologies. Examples of these exist both on publicly available sites as well as in custom applications that sit behind our corporate firewalls.

For example, we may find personalization being used for:

- **Technical Benefits** – Information may be stored in a data repository, making updates more efficient and universal. In addition, we may find it technically more feasible to create content on demand, rather than having information stored in static files.
- **Security** – one method of creating secure access to content is referred to as conditional disclosure. Here sites provide customized views of information resources that are specific to a user or group of users. Information may be personalized to prevent access to certain content based on the role that they have been assigned to in the application.
- **Localization** – providing web content that is in the language of the intended audience is critical – especially for global sites catering to the needs of international audiences. We also think of localization in the context of portals that provide content for a specific geography or interest area (e.g., content subscriptions.)
- **"Relationship" Management** – perhaps, the most widely held reason for creating content on demand is so that information can be customized for a specific audience to establish a personal relationship with the organization (e.g., one-to-one marketing.)

In another paper (Barnes Nelson, 2001), we discuss four major "relationship" types in the web commerce world: customers, suppliers, partners and employees. As we think about these relationships, the rationale for using these dynamic interfaces seems apparent.

[1] "Amazon charging different prices on some DVDs", Rosencranz, L (2000)

Audience	Buzz-word(s)	Purpose
Customer/ Consumer	CRM – Customer Relationship Management B2C – Business to Consumer	Create a positive method of interaction, including recommendation engines, secure access to project information, cross-selling and key customer/organization information. (amazon.com, dell.com)
Partner	PRM – Partner Relationship Management Extranet	Establish communication methods, exchange of information (e.g., customer exchange standards), specifications (documents, CAD/CAM), inventory levels, etc. (ClinicalExchange.com)
Employee	B2E – Business to Employee Intranet	Provide customized views for the employee about his/her worklife. Examples include benefits information, expense/ payroll data, Knowledge management, document management and distributed team communication. (www.schedule.com)
Supplier(s)	SRM – Supplier Relationship Management EDI – Electronic Data Interchange Extranet	Building on the reengineering efforts in the late 1980's and 1990's, just in time ordering, business process mapping, electronic data interchange standards, etc. all can be provided through personalization engines. (www.perfect.com)

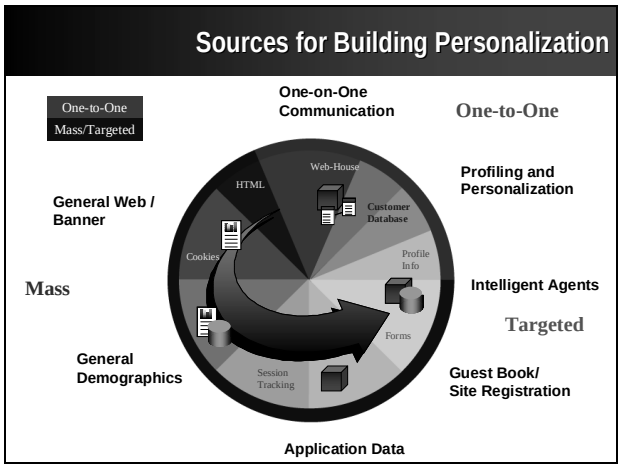
Table 1. The landscape of personalization techniques appropriate for types of business applications they intend to serve.

This table (Table 1) illustrates compelling reasons for creating interfaces that deliver information "just-in-time". In all of the cases cited here – on both sides of the firewall – content is being provided on demand. That is, there are few static HTML files that sit idly by waiting to be called. Instead there is usually some engine behind the scenes waiting for content to be requested. Usually, it is stored in a database or content repository so that information can be pulled "just-in-time". There are a number of vendors that provide content management solutions (for example, Vignette and Xpedio.)

Technologies for Personalization

As mentioned previously, there are a variety of ways that we can think about personalization. The diagram below examines this in the context of the audiences that are served and the goal of the interaction. As we move from Mass communication with web audiences to more personalized forms of interaction, we see an

increase in the complexity of the underlying technologies. As we move to one-to-one marketing approaches, a more robust personalization and profiling engine is required.



Here, we show a continuum of interaction regarding a web site visitor. As we move from the upper left around the circle to the upper right, we find the following trends:

- **General traffic patterns** - Static or Dynamic HTML is used to provide access to page views. The site may be used for informational purposes and has little or no personal interaction with the individual visitor. Gross analysis of patterns of traffic can be reporting on using web log analysis.

- **General demographics** - Cookies/ JavaScript provides the basic ability to monitor usage within a site (transient) and across visits (persistent). Cookies are used to store basic information about a visitor (computer – not a person) in order to provide some basic personalization.

- **Guest Book/ Registration** - Forms used to collect information in databases are a good source of gathering information about a visitor but often don't integrate well with web log data. Technologies such as collaborative filtering[2] can be used in response to form submissions to build dynamic interfaces. An example of this might include using collaborative filtering to recommend book choices based on what others have purchased in the past.

- **Specific demographics/ usage** - Specific utilization and repeat visit tracking can be accomplished when the site supports technologies such as session identification techniques such as JavaServer Pages (Sun), Active Server Pages (Microsoft) or SAS/IntrNet® (SAS® Institute). Here web server logs and application server logs can be combined to form a visitor profile. Repeat visits can be tracked as well as a comprehensive view of what actions were taken while on the site.

- **Profiling / Membership** - Customer provided authentication (userid and/or password) and personal/ membership data can be used in conjunction with intelligent agents to provide a personalized view of a web site. Intelligent agents can be used to filter information and provide the user with the content they desire. They work on behalf of the user observing their preferences and site interaction habits.

- **Integrated 360° View** - Internal databases that collect information from multiple touch-points such as call-center, sales, web, etc. are combined to create a true view of the visitor.

As you can see from this brief summary, there are a variety of technologies hard at work for us that provide these personalized, dynamic interfaces on the web. Now lets examine these technologies from the perspective of its participant.

Active versus Passive Participation

Up to this point, we have talked about personalization as a general concept. It may be helpful to understand it in the context of what the user or visitor does or sees. We differentiate here between some active participation in receiving dynamic information versus those that are provided by the application "auto-magically".

For anonymous users (based on IP address), we can simply evaluate click-stream or web log data to understand macro-level traffic patterns; for permission-based users (those providing information via cookies or server-side session state management), we can understand patterns of usage and repeat visits through more advanced analytics; and finally, authenticated users – those that give us some very specific information about who there are – we can provide very rich scenarios of behavior. Let us now examine these in turn.

Personalizing Content for the Anonymous User

For "un-authenticated user, we can indeed provide some form of personalization. From a technical perspective, we can do this through information provided from the browser, through cookies as well as through the use of forms.

Browser Sniffing: Who is that computer behind the visitor?

Because the web is digital, it may seem obvious that we ought to be able to get a lot of information from those that browse our sites. It is true that as you traverse the web, we do leave a virtual trail. That is, depending on how you access information on the web, what browser you use, what technologies sites have for you and the ISP you use all seem to leave our digital footprint – and in some cases, lead directly back to you.

Common examples of how we might use this information can include:

- Browser and Version (e.g., Internet Explorer, Netscape)
- E-mail address (if provided in your browser software)
- IP address (depending on how you connect, this may be yours, your ISP or even your firewall)

These are just a sampling of the types of information that can be obtained by using just standard, out of the box settings on a web server. More sophisticated web servers can provide a much more complete profile of who you are (as represented by your browser). A complete listing of the information that is available to most web servers can be found at Brian Lavoie's site (Lavoie, 2001).

[2] Systems that allow for the presentation of information based on what others have chosen.

Cookies: Unsolicited Personalization

Cookies are simply a method of storing basic content about a visitor to a specific web page (or site) such that the information can be retrieved at a later time. Cookies are used to store information such as the user's name, the last time they visited the site or demographic information (i.e., personal and technical information – such as the browser). Cookies are usually stored on a user's computer without their explicit permission.

There are two types of cookies that can be used to store information about a visitor, depending on how long you want to retain the information about the visitor, transient and persistent.

Transient, or a session level cookie, is placed in the users browser and lasts as long as the browser is open. This serves as a temporary ID for the browser and to any application servers that request the cookie. This method is used to associate data from click-stream and application server log processing.

Persistent cookies are similar to transient cookies, except these can be read by entire domain (e.g., sas.com®) and can be used throughout applications across an organization if common conventions are used. Persistent cookies can be used to "persist" information across multiple sessions or visits.

There are many issues surrounding the use of cookies, but these are generally a safe and effective way to capture information about a visitor (computer). The main advantage of using cookies is that they are easy to program and can be implemented quickly.

Cookies also suffer from several disadvantages.

- There are issues about how and what information we collect about people – that is, the social/ethical/perception issues surrounding their use.
- Cookies have a limit to how much data they can hold.
- Most significantly, though, cookies lack portability: the statefulness of cookie data is tied to an individual computer, rather than an individual person. When that person visits your site from a different computer, they have no access to their personalization settings.

Transient cookies cannot be used to track multiple visits/ repeat visits. In addition, Users may turn off or destroy cookies. From an organizational perspective, enterprises that wish to use them across all web touch-points should provide some standards and management of cookie signatures. On some operating systems (e.g, Windows ME and 2000) that allow for multiple users, a clear understanding of who the user is and the interpretation of a *visitor* versus a *user* versus a *household* should be clearly understood. In addition, the same user may visit from multiple machines making it impossible to really track a specific person.

Forms: Anonymous Solicitation

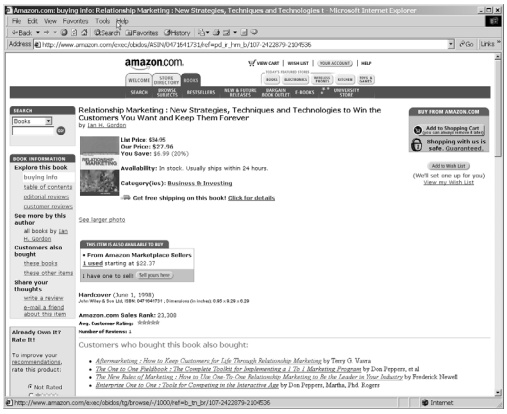
Forms, as you might suspect, gather specific information from anonymous visitors, and give something back – usually a request for information, a search or a subscription to content.

Many of us have seen this is action when we visit a web site to request product information and find that they have seemed to understand us by giving us recommendations of what others have purchased/ liked.

These sites have probably used some form of collaborative filtering. Collaborative filtering combines preferences and interactions of similar users and applies it to a new user/ request. This approach takes user built profiles in combination with system-generated models of how other "like" users look. The content is then provided to the user as part of their "personalized" view. There are at least three different types of Collaborative Filtering in use today:

- *Automatic Collaborative Filtering*, where a system modifies or customizes the interface and content offered based on understanding the preferences and usage patterns of other users within similar and behavioral characteristics.
- *Active Collaborative Filtering* is based on voluntary inputs from the community of users. ACF is based on permission-based marketing and can help determine which content is most relevant to users. As content is based on what users actually say they want versus modeled content.
- *Expert/ rules based filtering* is a highly sophisticated technique that allows the system to make deductions or inferences about a visitor based on built-in experiences and knowledge (usually stored in a database.)

The screen shot below from Amazon.com shows an example of collaborative filtering in action. Note the section displaying what others have purchased.



Like Amazon, most of the e*tailors on the market make use of third party application providers. Examples of vendors that provide collaborative filtering software include: Net Perceptions, Broadvision and Like Minds.

Anonymous Session Management

Because of the limitations of cookies for storage of data – either within a session or across sessions – vendors have developed server-side[3] solutions to manage this problem. Server-side solutions that allow for the persistence of information across multiple pages within a web site are referred to as session ids or session management solutions. Usually, these work in conjunction with cookies or authenticated forms of permission. When the visitor first

[3] Server side refers to the fact that the information about the session or user visit is stored/ generated on the server, not on the client (as in cookies).

visits a web site, the initial page sets a Session ID that allows the server to track session "state" or persistence across multiple pages or even multiple visits to a web site. Often these session ids are included as hidden fields within a page. As the information is requested from the server, the session id is validated and the page is generated on demand using a server side technology such as Micorosoft's Active Server Pages or Sun Microsystem's JavaServer pages.

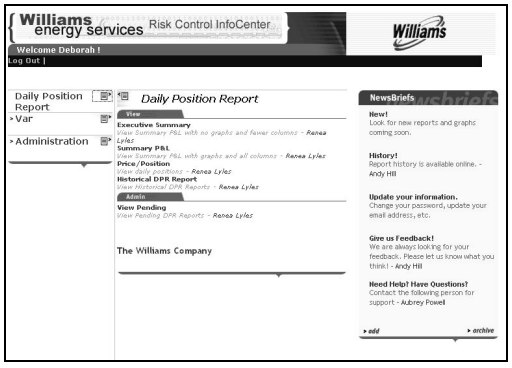
There are a few problems that developers may find when they wish to utilize these approaches. Namely, users that "bookmark" pages may find the page unusable at a later visit because it may have contained a hidden variable (sessionID) that has expired. In addition, caution must be taken to ensure that sessionIds can be used between applications in an organization/ web farm. Since many organizations use different tools and technologies – even within their own external web pages – users may find that they are not "remembered" even when directed from the customer service area of the web site to technical support.

Understanding the Authenticated User

Authenticated users are those that provide some sort of validation or verification of the individual. Authentication may be as simple as extracting the userid from the host (for example, the Microsoft NT userid is available to dynamic applications that request this information.) These applications represent the most reliable method for remembering users and providing content management services. These applications are often most appropriate within the firewall (intranet) or as secure connections between organizations (extranet).

Typically, authentication can come in several forms. We described one scenario of obtaining the domain userid from the operating system, but you may also have a custom logon screen, which requires a userid and/or password. In addition, SSL (secure sockets layer) can be used to authenticate a user to ensure that they are who they say they are.

We think of these as user supplied authentication where the user provides some level of information in order to receive content appropriate to them. Examples of these include commercial portals (myYahoo.com; myAOL.com); industry portals (ClinicalExchange.com); or corporate portals (Plumtree, Viador, Hummingbird). In each of these cases, the visitor is known and the content is based on their preferences (content subscription) or privilege (security). The screen shot bellows shows us an example of an Intranet page that offers conditional disclosure based on security roles.



Technologies for Authenticated Users

All of the approaches described earlier for managing and presenting content to the anonymous user can be used for the known – or authenticated user. For example, we can store information from page to page or even from visit to visit through the use of both client-side (browser, cookies) and server side (sessionID) solutions. In addition, since we know something about these people, we have the ability to store information about their likes, dislikes, demographic information, purchase history, favorite reports, etc. As a result, the possibilities for how we generate content for these audiences are endless. Most often these solutions rely on enterprise database management scenarios that allow for the integrated of corporate data and web commerce data (e.g., web log and click-stream data).

EXAMPLES OF PERSONALIZATION

As we discussed, there are a variety of approaches to personalization. In this section, we will examine these in the context of SAS®. We now take a complete example from beginning to end – showing how we can use SAS to understand whom the user is – from anonymous, permission based and authenticated.

First, we will show how you can get a variety of information about the visitor just by using information cleaned from the browser. Second, we will show how we can set a cookie so that we can use that within a session (from page to page) and then again on a repeat visit. Finally, we show an example of how we might authenticate a user and then present information to them based on who they are.

Please note that these examples are simplified and are not meant to show how SAS can be used fully to exploit techniques for personalization.

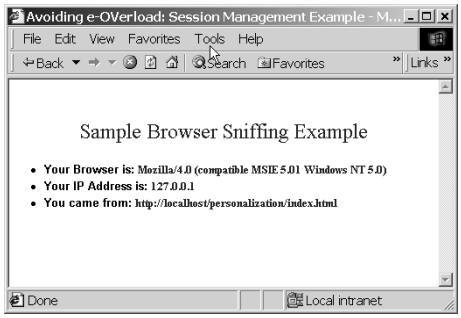
Reading Browser Information

At its simplest form, understanding a visitor in terms of their browser, gives us some really good information about who a user is – as represented by their computer/ browser. In this example, we set up a simple page that shows us some information like what browser they are using and their name if available.

The goal of this paper is not to show users how to set up and configure SAS/IntrNet®. But a word about some of the variables that are available to us through the broker is important. The broker – or application dispatcher – is a CGI (common gateway interface) program that allows us to communicate with SAS through a web request. By default, many of these variables (about the browser) are commented out. Lets take a look at some of the browser/ client specific variables. This partial list is taken directly from the broker.cfg file.

```
# User's IP address
  Export REMOTE_ADDR        _RMTADDR
# Username if authenticated
  Export REMOTE_USER        _RMTUSER
# Browser name
  Export HTTP_USER_AGENT     _HTUA
# Referring page if known
#  Export HTTP_REFERER       _HTREFER
```

The following screen displays the results of a simple program that uses these variables in formatting the screen. Specifically, we have identified the name of the remote user and their browser version.

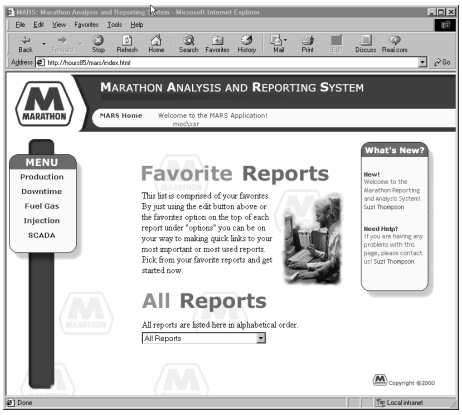


Of course, we could accomplish the same effect through the use of JavaScript:

```
<SCRIPT LANGUAGE="JavaScript">
document.writeln(parseInt(navigator.appVersion));
document.writeln(navigator.appName);
</SCRIPT>
```

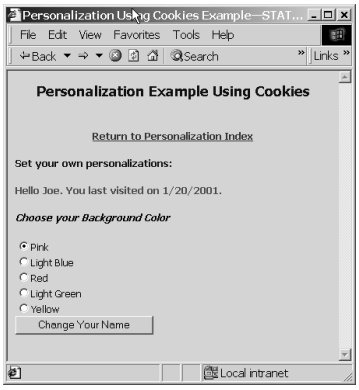
However, in dynamic SAS applications, there may be situations where getting this outside of the browser and passing it on application logic/ code, would be useful. A good example of this is for applications that might pull the current userid (from the Microsoft Windows NT login) is pulled from the operating system environment variable and identified in the first screen the user comes to.

In the following example, we show an example of personalization based on what we know about the user instead of what the user tells us. Here, we allow the user to save reports (a.k.a. My Favorites) within the application – but not force them to logon with yet another userid and password.



Forms, Cookies and JavaScript: Remembering People

In this next example, we want to provide some familiarity by setting a cookie when they first come to our site. Once visited, upon return visits, we either say Hello <name> (if the cookie exists) or simply "Hello" if there is no cookie set. We also show how you set the color of the background color and then remember that color when they return.



As we discussed previously, cookies can be very useful – especially when we combine it with more advanced techniques such as forms. In this example, we will have a user fill out a form, then remember this information so that it can be used the next time they visit. In this case, we take advantage of JavaScript to check whether or not the cookie is set, then depending up whether the value is available, we display information relevant to them.

Server-Side Session Management

Up to this point, we have relied on the browser and its ability to remember people from session to session by storing information on the client. There may be times, however, when you either cannot or do not wish to rely on the browser or cookies to store this information. Recently there have been numerous technologies that have been introduced that allow us to remember visitors from page to page and even from visit to visit. We can do this through the use of session management techniques.

Session management can be accomplished using a variety of technologies – some of these include: Microsoft's Active Server Pages, Sun Microsystems JavaServer Pages, Haht Software's HahtSite. Indeed, all of these server-based solutions could even exploit SAS data and compute services. For simplicity, we will focus on two methods that are provided for with SAS technologies: SAS/IntrNet sessionID and session management using Java through SAS' AppDev Studio®. Note that with both of these approaches, we could use either client side (cookies) or server side (sessionID) – we have decided to limit our discussion here to just server-side approaches.

SAS/IntrNet: Application Dispatcher Session Management

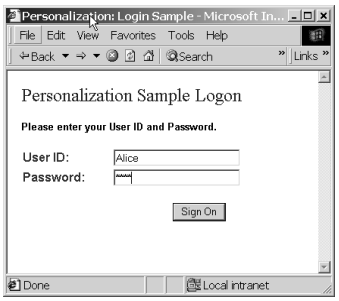
Like most web application servers, SAS/IntrNet affords us the luxury of being able to manage state within a web application by persisting information through a sessionID that resides on the server (or in cookies). Sessions allow us to save information such as temporary

datasets and variables just as we would typically save data in WORK libraries and in macro variables.

Macro variables can be saved for use in subsequent requests by prefixing them with the word SAVE_. Similarly, datasets can be persisted if you use the library SAVE instead of WORK.

Let's explore an example. In the series of screens that follow we perform the following actions:

1. The logon page passes the userid and password to the SAS application dispatcher through a POST method. (logon.sas)



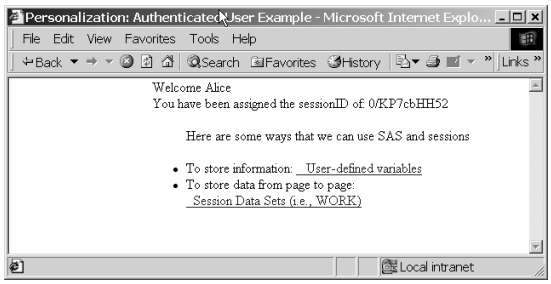
2. The logon.sas program reads the userid and subsets the sample dataset using the userid in the where clause. That dataset is saved using the SAVE.* method – which signifies to SAS that we want to save the data beyond this page.

```
Data save.mydata;
  set pdata.class;
  where upcase(name)=upcase("&userid");
```

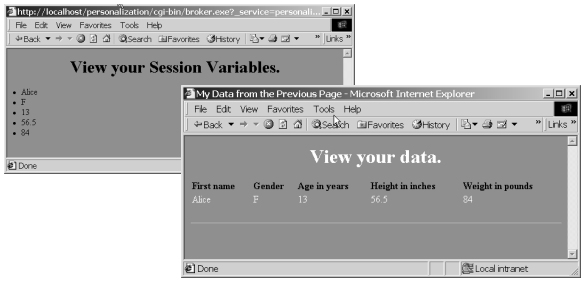
3. In that same step, we assign several macro variables based on the values being read in. These macro variables (SAVE_varname) are also available to subsequent calls.

```
call symput('SAVE_NAME',name);
call symput('SAVE_SEX',sex);
call symput('SAVE_age',age);
call symput('SAVE_height',height);
call symput('SAVE_weight',weight);
```

4. Finally, we build two hyperlinks dynamically so that we can see our sessionID in action. The sessionID is added to the link so that we have access to the SAVE. Datasets and the save. Macro variables.



And view the two subsequent pages where we pull the information based on the session information (as passed through the sessionID).



By take advantage of the server to manage session information, we have reduced he potential problems usually associated with using cookies. In addition, we have eliminated most of the effort in constructing construct complex URL's with parameters in order to pass information from page to page.

In addition, we can now also take advantage of the session to store information about what people have selected and/or saved through by intercepting each REQUEST and programmatically processing the information contained in the request – including writing the session information to a log. We would do this by modifying our appstart.sas program (the Application Dispatcher program that provides the service for our SAS/IntrNet programs). Here we would add a REQUEST statement – specifically an INIT program that would run each time a request was made.

Active Server Pages, JavaServer Pages and AppDev Studio

JavaServer Pages was introduced by Sun Microsystems in response to Microsoft's Active Server Pages as a technology to allow for the generation of HTML-based content in web applications from Java. JavaServer Pages is a special type of servlet (Java server program) that generates content on demand using the Java programming language. Similarly, Active Server Pages is a server side language that produces content from the server. Active Server Pages uses VBScript primarily as it's programming language. However, we can also take advantage of COM/DCOM for storing program logic on the server.

In each of these languages, we have the ability to maintain session state or persistence, using the session object. The session object, much like SAS/IntNet, allows us to persist information from page to page. We do this by establishing a sessionID when we first start out application. From there, we simply pass the sessionID from page to page and it "remembers" information despite the "connection-less" state that HTTP protocol provides.

THE PROBLEM WITH PERSONALIZATION

In the first part of this paper, we explored some of the business and technical reasons that "personalization" is now part of our every day vocabulary. Now we will discuss some of the challenges that we face in measuring the effectiveness of these strategies.

Earlier we spoke of these technologies from two perspectives. The first, our ability to generate content on demand so that visitors to our web sites have a much more personalized experience. The second is technologies that we use to understand what people do with that information. So now we turn to from technologies and approaches that deliver web content to those that help us understand its effect. That is, did we accomplish our objective with this web site? Do

people buy more? Do they stay longer? Do they come back to the same places? Will our servers survive the traffic?

The Need

Today's business and technology landscape is replete with reasons for personalizing content. In order to measure the effectiveness of these approaches, a methodology is needed, which allows us to determine the effectiveness of a site – in terms of the business value. Today's analysis techniques for reporting about site traffic, such as the most popular pages – tell us little or nothing about things that drive the bottom line. The business decision makers want reporting against profitability, customer satisfaction and return on investment.

e-Marketers who must make decisions about how to organize and present content on the web need to know such things as: What pattern of behavior by a web site visitor is followed by a high-margin product purchase. And which changes to the web site facilitate such behaviors.

Market for web activity analysis solutions, profiling data and auditing services has exploded. The business need requires an intelligent system capable of aggregating and analyzing activity on infinitely variable content in a manner that is meaningful.

Behavioral Indicators

The question of why people buy the things that they do has been at the forefront of the minds of researchers for decades. In the 1950's, folks like Louis Cheskin attributed the reasons to the way it made people feel. However, others found that attitudes and behaviors often are incongruent (see for example, Aizen & Fishbein, 1975). If we can really measure what people are doing, rather than merely getting their opinion on the subject, is much more reliable. So it may be that we measure people's traffic patterns through our web sites because we can, but it is also that is gives us a tremendous amount information about behavioral patterns which can then be used to make business decisions that directly affect the bottom line.

Web Log Analysis

One of the most popular techniques for analyzing web activity involves reporting on web logs. Web logs helped us understand such things as: Where did people (visitors) come from? Where did they jump off? Where did they go when they were there? How long did they spend at each place? Did they come back?

Web activity reporting consists of:

- Basic traffic statistics (hits, page views, visits)
- Navigation patterns (referrers, next-click, entrance and exit pages)
- Content requested (top pages, directories, images, downloaded files, stickiness - measuring depth and persistence)
- Visitor information (domains, browsers, platforms, time of day)
- Fulfillment of the site's objectives (purchases, downloads, subscriptions)

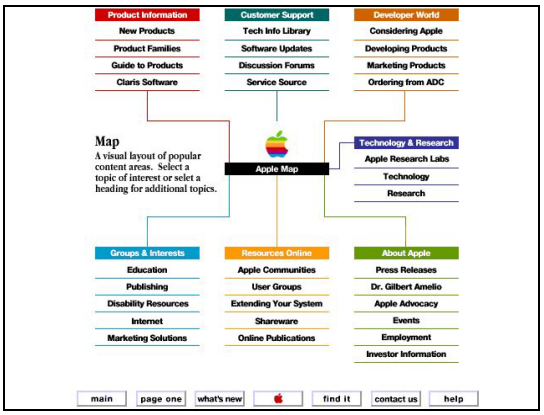
Answering questions such as these are critical as companies are pouring huge capital investments in the web-front experience. Help with decision making about content activity – gives companies quantifiable measurements as to the type of content that is being requested.

Analysis tells an organization about their visitors – just like as presenters, we try – as we may – to understand our audience so that we can tailor content to the audience – tracking helps us understand who is it that is requesting information. Questions like: audience composition, how do they match our models of a customer, how does it change over time, what improves visitor retention or session depth?

***e*-Intelligence**

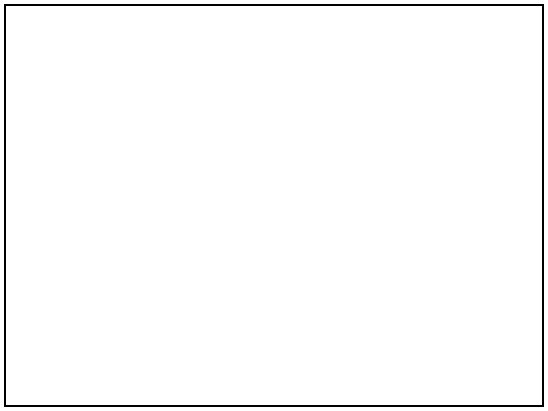
One of the most common techniques for understanding how people have interacted with a web site is through the use of web log analysis. That is, web servers produce a tremendous amount of data about who clicked on what, where. These techniques were adopted early in order to unearth the hidden mysteries of performance, traffic flow, on-ramps and off-ramps and buying patterns. However, in this context, we must move beyond traditional web log analysis and focus on how we can use the data coming from our web sites when the content is continually changing through personalization techniques. The holy grail of marketing is, after all, delivering your message in a manner that is compelling enough to create a personal relationship between an individual and the organization providing the message.

With these technologies, comes a price. That is to say, the more dynamic we make the content, the harder it is to track and monitor what content is being viewed. For example, given a typical web site, we are usually aware of the possible destinations.



Of course, the paths that one could take within this site could be enormously complex. However, technologies that allow us to map these pathways and make decisions about site design, buying patterns and visitor information are fairly straightforward. For example, WebHound™ (SAS Institute), provides compelling views about our web site, allowing us to determine which areas are most often visited, which areas lead to the purchase decision most often as well as the ability to understand complex patterns of traffic flow within our site.

Visualizing this content and the pathways are often the most useful technique. Below, is a sample screen from one of these tools (Astra Site Manager.)



These software tools are designed to aid their masters in the visualization and management of large, complex web sites. More often than not, however, these tools are designed to understand and capture information about data that is slowly changing.

The Challenges

Most often, the first stop on our way to understanding what people have done on our web sites begins with web log analysis. As it's root, web log analysis is a stream of data that is generated when a user does something on your site.

If we examine a typical web log, we see that it is simply a trail of activity. The unit of analysis of this data is a computer (represented by an IP address) and a request (for an object such as a page or an image).

```
202.188.93.150 - - [26/Dec/1999:04:31:32 -0500] "GET /images/1sdex.gif HTTP/1.1" 200 967 "http://ecommerce.nc
202.188.93.150 - - [26/Dec/1999:04:31:32 -0500] "GET /images/tac.jpg HTTP/1.1" 200 1596 "http://ecommerce.nc
202.188.93.150 - - [26/Dec/1999:04:31:35 -0500] "GET /images/gvr.gif HTTP/1.1" 200 41 "http://ecommerce.ncsu.
202.188.93.150 - - [26/Dec/1999:04:31:35 -0500] "GET /images/spacе.gif HTTP/1.1" 200 37 "http://ecommerce.ncs
202.188.93.150 - - [26/Dec/1999:04:31:36 -0500] "GET /topics/images/thumbnails/tn_businessmodel.gif HTTP/1.1"
202.188.93.150 - - [26/Dec/1999:04:31:38 -0500] "GET /images/logos/wachovia.jpg HTTP/1.1" 200 8848 "http://ec
202.188.93.150 - - [26/Dec/1999:04:31:38 -0500] "GET /images/gvr350.gif HTTP/1.1" 200 57 "http://ecommerce.nc
202.188.93.150 - - [26/Dec/1999:04:31:39 -0500] "GET /images/lquote.gif HTTP/1.1" 200 147 "http://ecommerce.n
202.188.93.150 - - [26/Dec/1999:04:31:39 -0500] "GET /images/rquote.gif HTTP/1.1" 200 151 "http://ecommerce.n
202.188.93.150 - - [26/Dec/1999:04:31:40 -0500] "GET /images/vertical.gif HTTP/1.1" 200 390 "http://ecommerce
202.188.93.150 - - [26/Dec/1999:04:31:41 -0500] "GET /images/edge3.gif HTTP/1.1" 200 100 "http://ecommerce.nc
202.188.93.150 - - [26/Dec/1999:04:31:41 -0500] "GET /images/horizonal.gif HTTP/1.1" 200 256 "http://ecommerc
202.188.93.150 - - [26/Dec/1999:04:31:42 -0500] "GET /images/l_elbow.gif HTTP/1.1" 200 87 "http://ecommerce.n
202.188.93.150 - - [26/Dec/1999:04:31:42 -0500] "GET /images/small_logo.gif HTTP/1.1" 200 2278 "http://ecomme
202.188.93.150 - - [26/Dec/1999:04:31:44 -0500] "GET /images/ban_logo.gif HTTP/1.1" 200 813 "http://ecommerce
202.188.93.150 - - [26/Dec/1999:04:32:37 -0500] "GET / HTTP/1.1" 200 3476 "http://partners.mamma.com/SurfFast
202.188.93.150 - - [26/Dec/1999:04:32:50 -0500] "GET /images/ec.jpg HTTP/1.1" 200 0 "http://ecommerce.ncsu.ec
202.188.93.150 - - [26/Dec/1999:04:32:53 -0500] "GET /ec.html HTTP/1.1" 200 7762 "-" "Mozilla/4.0 (compatible
202.188.93.150 - - [26/Dec/1999:04:32:54 -0500] "GET /images/ec.jpg HTTP/1.1" 206 0 "http://ecommerce.ncsu.ec
```

There are several challenges that we face as we deliver dynamic content to users. These include:

- **The problem of the dynamic "path"** – One of the challenges that we face when we deliver personalized content to a user or a group of users is that our notion about the "path" that they have taken to get to a certain point has changed dramatically. Each visitor may have a completely different path that makes traditional analysis difficult, if not impossible.

- **Making complex URLs Meaningful** – You may have noticed that while visiting a web page, the URL (the address of the query string in your browser) that appears as a long string of made-up numbers, letters and special characters. Often these are used to pass critical information to the server as you request specific information. An example of this can be found in even the simplest web application (this one is from the SAS Multidimensional Viewer.)

  http://10.133.1.7/sascgi-bin/broker?metabase=SASHELP.MBEIS&_program=sashelp.webeis.mddbrpts.scl&_DEBUG=0&VMDOFF=y&_service=default&mddb=SASHELP.PRDMDDB&css=%2Fsasweb%2FIntrNet8%2FMRV%2Fcss%2Fdefault.css

  As requests like these get made – no matter how similar – they appear in the web log report as different pages even though they may be the same request with a slight variation. For example, in the following query, we are requesting a sales report for the North East Region, sorted by County. In another request, we ask for the same data, only sorted by State. As these get processed by the web reporting tool, they are reported on as two separate pages.

- **Multi-source data integration** – as data from multiple servers are aggregated to provide a single, consistent view of the web activity for an organization, several problems can arise. For example, there are issues such as time-synchronization (multiple time zones) or tracking session variables across servers that may have different sessionId management standards. In addition, because of the pure volume of data/ interactions, it may not be feasible to report on all of the data that we would like.

- **Integration of Application Server and Session logs** – Mapping what's possible to what the user actually experiences often mean combining web server information, session logs and application server logs to create a logical mapping of the user experience.

- **IntraPage Analysis** – In order to make a web page more dynamic and interesting to users, we have adopted the use of JavaScript, XML, Java applets, ActiveX controls and other plug-ins. However, since non-HTTP protocols are not tracked in the server logs we have no way of understanding these client-side interactions.

- **Changing Hierarchies and Product Taxonomies** – Because content on the web can be changed often and even for each user, we have to take into account how we are going to report on this data in reaction to changing requirements of the site (movement and restructuring of pages, reorganization of products and product categories, visual representation of products, etc.)

Traditional web tracking and reporting tools are not prepared to deal with these complexities. Web server logs are designed to track the query string – not session or intra-page activity. Most often, web log analysis tools use IP matching to report on web activity. IP Matching consists of mapping time-contiguous server log entries from the same host ID in a short period of time.

Since IP Matching can only provide gross statements about traffic patterns, there are three major issues with this approach. (1) This technique tends to be inaccurate with web sites that have a sizeable number of visitors since IP address can be reused within a short period of time. (2) IP addresses may be used by the same user within the same session for the same user. (3) Firewall protection yields single IP addresses for a group of users in an organization (e.g., AOL).

These approaches are good for sites were there are few dynamic components because we are analyzing information after the visitor has left (i.e., web logs), there are no special requirements on the web server or the browser (except that the web server can generate the logs.)

Dynamic sites, on the other hand, have complex reporting requirements if we want to understand the entire picture. For example, we may find ourselves combining the information from a

web server with the application log to get a true picture about what has happened.

CONSIDERATIONS FOR E-INTELLIGENCE

If we really want to understand what has happened on a web site, it is critical that we understand several things.

First: We need to understand the difference between an object requested, a page and a visit. We typically look at objects as individual components that have been requested on a page (e.g., image, html). However, more meaningful content for analysis includes both the "Page" and the "visit" or Session. For analysis of site flow and pattern analysis, aggregating information at a page level is key. Integrating information about the context in which a page is viewed is critical for truly moving beyond the "river of clicks" and onto descriptions of behavior. Here, we combine server log information with application level content such as session state information.

Data must be mapped and aggregated at a level that is meaningful in the context of the business: server typically logs clicks and keystrokes (object level) – but basic analysis begins with page level experiences – product experience.

Mapping what's possible to what the user actually experiences often mean combining web server information, session logs and application server logs to create a logical mapping of the user experience.

Sometimes, a group of pages should be aggregated to understand the "super-ordinate" goal of a promotion or campaign.

Data Mining

From a technology perspective, data mining has certainly blossomed in its relevance for helping us understand the vast amounts of click-stream data. By applying data mining techniques such as market basket analysis and other association techniques, marketers can find a virtual gold mine in their data. Second generation mining techniques, applied to the web, evaluate not only end of line tracking – composed of tracking behaviors of a person in a store/ web site – but also what to display next. Good examples of these include Amazon.com's recommendation engine. The movement from traditional analysis (what happened) to "what's going to happen next" has taken eIntelligence to new way of thinking about the question "why do people buy the things they do?"

CONCLUSION

In this paper, we have explored the world of personalization in terms of its business drivers, the technologies for generation and techniques for measurement. The world of dynamic applications – where content is delivered to users based on who they are, what they like and what they have access to – is quickly evolving. No doubt this world will continue to change – both as a result of the technology but also because of the social and political landscape that governs its acceptance. As technologies such as XML and wireless communication become commonplace, our ability to deal with the information overload will become more important.

Equally as important is how we will deal with the measurement issues as we see the heightened visibility of privacy. Clearly this evolving landscape of politics, business processes and technologies will continue to raise questions along the way. However, personalization coupled with tracking provides us with a powerful toolbox that allows us to understand people – not just the technology – of our "webscape". Personalization can help your find out what makes your audience "click", what works and what doesn't.

BIBLIOGRAPHY

Fishbein, M., & Ajzen, I. (1975). Belief, attitude, intention, and behavior: An introduction to theory and research. Reading, MA: Addison-Wesley.

Lavoie, B. (2001) "Web Characterization Project", http://wcp.oclc.org/

Ramsey, C. (1.15.2001) "Managing Web Sites as Dynamic Business Applications" The Internet Industry Portal, http://idm.internet.com/articles/200006/wm_d.html.

Rosencranz, L. (9.5.2000) "Amazon charging different prices on some DVDs," Computer World on-Line, http://www.computerworld.com/cwi/story/0,1199,NAV47_STO49569,00.html.

Barnes-Nelson, G.S. (2001) " Collaborative Commerce: Portals for Decision Support and Knowledge Management," *Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference, Long Beach, CA*.

CONTACT INFORMATION

The authors may be contacted as follows:

Gregory S. Barnes Nelson
STATPROBE Technologies
117 Einburgh South, Suite 202
Cary, NC 27511
Internet: greg.barnesnelson@statprobe.com
Personal: gregbn@ix.netcom.com
Web: http://www.statprobetechnologies.com

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