Abstract

A discussion of the potential of using SAS® products, such as SAS/ASSIST® and SAS/AF® to build the user interfaces, the data management and the reporting capabilities of an end-user based Information System. The discussion focuses on how these SAS products interact with such issues as the sensitivity of data, data security, multi-platform environment, hardware, software, and the variety of end-user interests.

Section 1: Information System Issues

Those of you who have a computing system composed of numerous PC’s, MACs, DECstations and a huge IBM® mainframe, where you run all your production applications, wouldn’t you like to free up some CPU cycles on your mainframe from certain non-production applications and use the mainframe primarily for your production? As you know, these cycles are very expensive! And those of you who don’t have such worries how about your data? Wouldn’t all of you like to turn your data into information? After all, we are in the so called Information Age. What does this mean? What is the difference between data and information? Data does not become information unless used by those who know how to use it and need to use it. All of you have data - you wouldn’t be reading this otherwise, so, regardless of your computing horse power, I hope, I have something to offer to all of you.

So, what we have here is a need to turn your data into information and do that by distributed processing (if at all possible). How is that, you will ask. Well, for one thing, only the end-user can turn your data into information. And, consider this: the end-user may already have some processing power or, possibly, some way to access your computing system. Believe me, the end-user will be thrilled to hear that there is nothing new to learn to do the work! Let the users use what they know, just make it possible. And, how can you make that possible? Just make your data accessible by your end-user and give him or her some tools. If you are interested in increasing the number of your end-users, allow them to use your tools and let them also use their own data.

Let us call this system EURT standing for End-User Reporting Tools - that’s the name we use here at the University of New Mexico, Computer and Information Resources and Technology (UNM, CIRT).

The key words so far are: End-user, Data and Tools. Which one of these words is the most interesting? It really depends on what you are in the business for, but I can assume that you all have customers, and everything you do is primarily for them - trust me, or even better, look at your company’s mission statement. Let’s then start the discussion (or monologue, if you wish) from the end-user, the customer.

The End-User

Who is your end-user? Do you know anyone named John? Well, let’s take a more general approach. End-user is anyone who can access your computer system, and I really mean ANYONE! Too many of us have the tendency to consider as customers only those who “we want to serve”, and to a great extent that’s correct. In a restaurant one can see the “No shirt, no shoes, no service” sign, but that sign may or may not stop all of those who do not qualify, those whom you do not want to have as customers. The moment the person walks in the restaurant, he or she is a customer, whether a “wanted” or an “unwanted” one, and he or she has to be handled accordingly. Similarly, in our business, we have “wanted” and “unwanted” customers or end-users, if I may. In most cases, a policy sets objective criteria, which differentiate the customers into “wanted” or “unwanted”. Who handles the “unwanted” ones? In bars we have the “bouncers,” and, when needed, the local law enforcement agents. In our business we have the security administrators and the law enforcement, the range of which may extend to world wide dimensions.

It is crucial to define which are your “wanted” customers and which are not, and do that as early as possible. It will bring the scope of your project to sizes you can feel comfortable with, and it will allow you to handle your data security strategies from the beginning. At UNM the policy currently defines as legitimate users those who are in any way affiliated with the school. The sign “No UNM Affiliation, No UNM Computer Account, no Service” is not a bad greeting after all.

Data Sensitivity and Security

Do you have sensitive data in your installation? Let me ask that differently to avoid jargon. Do you have any data in your installation which should not be looked at by certain people? How will you feel if one morning you find that your accounts receivable files are gone, zip, out of here? How much damage would it do to you, if your competitor found out about your new product line before you announced it to the public? If you are in any situation similar to these, then you do have sensitive data. Also, consider this: a fire burns down your entire building, your computer room and melts your computer and all the backups you have stored on site.
Are you ready to recover from that God forbid awful disaster? Data security is most commonly thought to be related with avoidance of electronic intrusions, and we are not going to deviate from this norm, nevertheless, do find a way to hedge against these physical security issues, if you haven't already, and here we will concentrate on your electronic security.

Why would someone want to access your data? Think about it. You have made and have probably advertised (as we have here at UNM) an information system where you provide data, tools and computing power to your end-users, but, "Hey... this data is not for you. We do not want you to see our data. You are not one of the privileged ones." Rejection. I would feel bad myself. You tell me that you are showing data to so many people but not me? Why? What are you hiding there? Since you have sensitive data, you cannot avoid such situations. So, you must protect your data, especially, if you are an institution, which is in the business of encouraging international access to your systems (such as an educational institution like UNM). You want researchers to access your system from every corner of the world. It makes you famous, and it makes you useful to society. So, you connect your system to the Internet and the Technet and the what-have-you-Net. Don't be unreasonable, people will look around. Just make your data reasonably difficult to access and don't advertise that "Hey! Here is my students' grades file - change it if you can." Reasonably difficult to access, you will ask, aren't we here trying to give data and tools to the end-user? Well, human readable files are more susceptible than machine readable ones. How is that for an answer? "Pretty bad," you should say because:

- not all end-users are data processing professionals; and
- not all users will use the same platform for data processing, and not all platforms are compatible for data transfers among them.

That's why, here at UNM we are currently providing flat files. Flat files help because:

- much less, or no data processing expertise are needed to access a flat file vs. an encrypted one, for example, and
- a flat file can be moved from one platform to another much easier than let's say an encrypted one.

As for the data protection, basic security is a system dependent issue, which can be enhanced with the security provided by the selected access tools. Caution! New software may introduce holes to your existing security system. In addition, a carefully designed file structure can enhance your security. A social security number or an address or a telephone number by itself is not sensitive data. Data becomes sensitive in context, if, for example, a name and the social security number are found together.

Hardware Hosts for Tools and Data

An invariant of an EURT system is that it will run in a multi-platform environment. You can only recommend a specific machine to run your tools (other than your IBM mainframe, since its cycles are expensive) and you can only recommend a set of tools to your end-users. Here at UNM, we decided to use a DECstation to be the host of the EURT tools. Notice that I didn't say EURT system, and that's because the system also includes the data. Where would you like to place your data? First of all, it is reasonable to assume that your data is production data, and, therefore, it is created and maintained on your production machine. Special massaging of your data may be needed to bring it to the form you want for use with your EURT tools. For example you could make flat files and organize them in some file structure. Which is going to be your EURT data host? Wouldn't it be reasonable to place your EURT data on the same host with your EURT tools? Let's look at the pluses and the minuses, and, to make the discussion less general, let's take the UNM example in which the EURT tools host is a UNIX machine, while the data originate at the IBM mainframe. Let's see if it makes sense to have the entire EURT system on one machine.

The pluses are processing efficiency and minimization of security risks during data transfers through the network. The pluses are the data capacity of the DECstation and, of course, the fact that UNIX is the "hacker's paradise":

(+) Processing Efficiency:

there is no doubt that processing time increases when data has to travel across platforms vs. it being right there and ready for use, especially in an interactive environment.

(+ ) Data Transfer Security Risks:

The data would have to be transferred from the IBM to UNIX on a regular, scheduled basis across the network. The scheduling aspect of this data transfer offers potential for minimization of security risks. Data can possibly be transferred encrypted and at hours when not too many people are on or watching the network. Both of these ideas, however, are questionable. Data encryption across platforms, particularly IBM and DEC's is quite complicated (ASCII vs. EBCDIC and different encryption algorithms, for example). As for the off hours data transfer, it is more of a statistics exercise. The true question is how frequently can you afford to have your data violated? Probably, one violation is one more than you can afford.

(-) Disk Space Availability

In terms of data capacity, simply put, the DECstation does
not have the space and the department does not have the
money. This is the UNM case, but, I would not be
surprised if you smiled and silently thought that “that’s the
situation in my department too.”

(UNIX Security)

About UNIX security, many would claim that “there isn’t
any,” or “what security?” Well, this is not exactly true.
UNIX does provide some rudimentary security and data
access controls, but UNIX was designed for educational
purposes, not for data processing. There is plenty of
literature on UNIX security and plenty of literature on how
to break it. Additional software, such as Kerberos for
network security, is available on the market, but not at
UNM.

The decision to keep the EURT data on the IBM mainframe
was rather easy. The space limitations and the
questionable security on the UNIX system compared
with the space availability and the existing and sufficient IBM
security were strong hints for the decision. The security
risks due to data transfer during data processing were
considered marginally higher than those of a scheduled
data transfer system. Possibly, the EURT tools could offer
some relief on the data transfer security issue, while a
carefully designed EURT file structure does give a security
advantage.

Variety in End-User Interests

We have said that EURT is data and tools allowing your
data, and tools allowing your
end-users to do their own data processing. What kind of
data are you going to give them? That really depends on
the scope you have decided for your EURT system. If you
allow your end-users the option to use their own data with
your tools, you will end up with more customers. But
remember that, in most cases, end-users have specific
needs for information and a lot more needs for curiosity
satisfaction. Give them data and they will use it. In a
university environment, for example, you have data of
interest, such as the Athletics Department budget, which
easily attracts attention. Security allowing, everybody will
look at it. Here is the moral: give all the data you feel
comfortable with; it will be used. If your EURT is
successful, be assured that your end-users will let you know
of which additions you need to make for future system
upgrades.

Section 2: Solutions and the SAS Software

Thus far we have talked about everything except for the
EURT tools. What would you like these tools to
accomplish? How about these suggestions:

1) they will provide an easy to use interface allowing
every end-user with or without any data
processing expertise to access your data (possibly
his or her own data as well) and convert it to
information,

2) they will be able to function in a multi-platform
environment, where the data reside on site A,
while the processing occurs on site B, and

3) they will allow (ideally) different end-users to have
access to different subsets of the data.

Why did UNM select SAS as its EURT tool? Why would
someone select a statistical analysis system for online
interfaces and data processing applications? The answer
is relatively simple. SAS can do all data processing
functions as good as or even better than any other
programming language. It has simple coding, and its
statistics influences allow it to easily handle any size of
data. SAS has a great advantage: it is easily portable and
has the same look and feel regardless of the platform it
runs on. It provides a friendly end-user interface
(SAS/ASSIST), which produces and executes SAS code,
while it requires minimal knowledge of the SAS language.
It provides an elaborate graphical representation of data -
big advantage for understanding information, and it provides
the means to handle the multi-platform nature of your EURT
system. It provides packages which allow remote access
of your data, and it has no security holes known to us. Do
you just install the different SAS packages and you are
done? Well, not exactly.

The User Interface

It was mentioned earlier that SAS/ASSIST provides a
friendly end-user interface and it requires minimal
knowledge of the SAS system. Here is the UNIX SAS 6.07
look of the initial SAS/ASSIST screen running on the same
machine, but accessed through an X-terminal and a VT100
interface:
What is the difference? A few $1,000 and the use of point and click communication, let alone the looks and the high/low quality of graphics. Today, everyone, you would say, has a PC and runs Windows, or has a MAC and has great graphics. Consider this for an answer! SAS runs on the DECstation, not on your desk-top. You access the DECstation with some terminal type emulation, most likely a VT100, and your mouse is dead. A VT100 is a terminal with no mouse attached to it. You simulate an X-terminal environment, but your PC is not an X-server, and you are even worse off than having a VT100 emulation. Is this a SAS/ASSIST problem? Not really, but the user friendly interface is not as friendly any more. How about, SAS knowledge? Think of the user who can turn on a PC and do some word processing, and that's about all that user can do with a computer. Am I unreasonable? Go and ask your help desk people if they have ever encountered such a user. Do you expect that this user will be able to get to the initial screen of SAS/ASSIST and from there end up with a report which requires the merging of three files, one of which needs to be transposed? Let's not go too far. Let's assume that he or she wants to display one of your EURT files. Look at the screen that will appear after she logically selects the "Data Mgmt", "Edit/Browse" and "Browse data in tabular format..." SAS/ASSIST options:

The terms "Libref," "Catalog," and "Formula" will flush in front of his or her eyes. "Huh? What do I do now? I don't understand! They said I can go ahead and use this thing - no sweat, it's user friendly. Am I stupid?" There is nothing more discouraging to the unfamiliar user than five immediate questions at the first screen of a user friendly/no-experience required interface. What is your problem in this case? Terminology! Whether the screen said "Libref" or "Please tell me the name that SAS knows for the path to the file you want to use," the uninitiated user will run for help. At least, the English prompt is easier to remember when the user asks for help. I have heard that SAS version 6.09 offers a solution to this. It allows you to change the default SAS terminology and use your own. To what extent this is true, and, if so, how much better will your terminology be compared to the SAS one, remains to be seen.

You may say that in this discussion I started from the low end of end-users in terms of computer knowledge. Take an expert IBM-MVS SAS programmer, who is using batch SAS, let's say for the last eight years. His reaction will be almost identical. This programmer knows a different term: DDNAME. "Here you go! DDNAME! That I can understand. What is this Libref thing?" I think I've made my point.

Let's say that the end-user is brave and selects the "Active data set" button - that is what she needs to do to select a data set to use. A new window pops up with a list of all of your EURT files you have made available to your end-users. Two things you should keep in mind. Your EURT files must be described in two eight characters long words and they will appear on the selection list only if you have defined a "Libref" and have made it available to the specific user. The naming of each EURT file is an art, and I cannot offer any help here. How can you make different files available to different users? I am still scratching my head on that one. One approach is to define a default profile for all users, one which assumes no special privileges.
Then, on a case by case basis, add Librefs to individual user profiles. Another idea is to utilize your current data security scheme. Here at UNM we like the IBM security system much better than the UNIX one, and we are going to rely heavily on it. This means that we will utilize two levels of security: when SAS accesses a data set it consults with the SAS host security; because the data actually resides on a different host, security will be checked there as well. Let's see how this remote access of your data will work.

Accessing Your Data Across the Network

SAS is offering the SAS/CONNECT® package for remote site processing. There is a small problem, though, which makes this package rather non-applicable to your EURT system. Let me quote the SAS/CONNECT manual:

SAS/CONNECT software provides two major capabilities:

You can copy files between the remote host and the local host.

You can run local SAS programs on the remote host, using data residing on the remote host without transferring either the program or the data. Output from the remote host is displayed on the local host.

It sounds good, but this is exactly what EURT is designed to avoid. You do not want to transfer files to your local host (discussed earlier), and you do not want to use the expensive mainframe CPU cycles for your EURT processing (also discussed earlier).

The approach we are trying here at UNM is a network link of the two systems through the Network File System (NFS) client/server. The data resides on the IBM mainframe as plain and simple flat files*. The local UNIX host, where EURT runs, has one or more defined but empty directories. NFS maps (or to use the NFS term, "mounts") the IBM data sets to the UNIX directories, and UNIX thinks that the files are as local as they can get. That's exactly what EURT is designed for. SAS uses the data as if it were local UNIX data. There are three things you should keep in mind. The IBM NFS server is definitely using up some CPU cycles, the data travels through and increases the load on your network, and your users must have accounts on both hosts.

The big advantage of maintaining your files on the host which regularly provides the security you consider acceptable is that you can allow all end-users to use the same SAS profile, which defines all of the EURT Librefs, and let your remote host security control who can access what. NFS provides additional security. For example, you can define and reinforce that the NFS access is Read-only, therefore, preventing corruption of your EURT data. NFS also allows you to specify which host is authorized to access your data. On the other hand, by making available all of your Librefs you advantage what kind of data exists and you offer challenges to hackers to prove they can get to your data.

Would it be a problem to give accounts to your users on both hosts? At UNM this is not a problem. Any UNM student, faculty or staff can create his or her own account on all UNM hosts. The system looks up the UNM databases, recognizes the person, creates the requested accounts and automatically assigns appropriate security levels to each one of them.

Data Dictionary

One way or another you have made your data available to your end-users. How on earth, could your users know what kind of data do you provide? Even if your user was brave enough to select the "Active data set" button, how descriptive can you be when you only have two eight characters long words to name each data set? An EURT data dictionary application would not be a bad idea, and what would you use to develop it in? How about SAS/AF?

First of all, let's talk about this EURT Dictionary. You have a number of files with a number of fields in each one. Your end-user may have an idea about which file(s) s/he wants to use, or may know which fields s/he wants to use but does not know in which files to find them. Your dictionary should:

- give extensive information on the contents of all of your files,
- give extensive information on all the fields and where they can be found,
- provide a means for access on either a file or a field basis,
- provide password protected dictionary administration functions, and
- provide a double password protection on who can change the password of the dictionary administration functions.

SAS/AF was wonderful in designing and implementing this application. I had all the freedom to use my own jargon, and the more one learns about the SAS Screen Control Language, the more power to him or her. You can write an application and demonstrate it and no-one would believe that you are not executing a fancy PC application, but rather you are running an application on a mainframe or a workstation. SAS/AF's limits are pretty much only defined by your imagination.

Pheewww! Have we covered everything you need to know about your EURT system? Almost. There is one more thing. It would be truly unreasonable to expect that any
Some of you may remember the security hole introduced to UNIX through... e-mail!

By the way, do you know what is a Libref? Here is the SAS online Help definition:

Library or libref
A SAS file's libref (an abbreviation for library reference) is a name that is associated with the physical name of a SAS data library.
When a new SAS data set is created, the libref points to the SAS data library where the file is to be stored. When you reference an existing SAS data set, the libref indicates the data library where the SAS System should look for the file.

Incidentally, from this phrase you should be able to figure out the stage of the EURT project at UNM: conceptually everything is in place, now we need to implement it. This is the best time for this paper, when all the excitement, the disappointments and the head-scratching is alive and fresh in memory.

As mentioned earlier, we have decided to keep our EURT files in a flat file data format. NFS can handle storing UNIX SAS data sets on the IBM platform and UNIX SAS can use them, but the usage of flat files through SAS VIEWs is a better approach and the data is more up-to-date.