From out of the Caves and into the Light

Art Brooks,
Director
Administrative Data Processing
University of Missouri - Rolla

Institution information:

The University of Missouri - Rolla, founded in 1871 as the Missouri School of Mines and Metallurgy, is one of four campuses which constitute the University of Missouri system. While throughout its history, engineering and science has been the focal point of its educational efforts, the campus also grants bachelor level degrees in several liberal arts areas. UMR offers bachelor, master and doctoral degree in twenty-eight disciplines in three colleges and schools; the College of Arts & Sciences, the School of Mines & Metallurgy and the School of Engineering. Although its student population has consistently hovered around 5,000 in population, the campus has a history of ranking among the nation’s leaders in bachelor degrees granted in engineering and its graduates can be found employed around the world.

Paper abstract:

In 1986 the University of Missouri - Rolla began downloading corporate data to its local mainframe as a cost saving measure and to improve programmer productivity. After ten years of this process the campus has implemented what today is called a data warehouse. This paper will briefly cover those first exploratory years, review the data warehouse as it exists today and focus on a discussion of the trials and tribulations of today and into the future.
From out of the caves and into the light -- in those fleeting moments one finds to reflect on the technical progress which has been made since being first introduced to computing, that is precisely the emotion one feels. In comparing the manipulations of the punched cards of years ago to the marvels of life on the Internet, it is mind boggling. It’s as if we had leapt into another galaxy at warp speed; a trip for which one can not purchase a round trip ticket. We may feel at times like space explorers, but we are not time travelers. As information technology leaders we must be certain of our direction and confident of the tools of our trade. It is important to be daring and conservative while being meticulous and methodical. We must be looking ahead while remembering the lessons we learned in the past. Our biggest threat as information technologists is to allow ourselves to ignore the evolution in our profession, thereby staying with outdated methods and archaic tools. It is imperative we stay abreast of our times and stay current with our evolving tools. In truth we must come out of the caves and into the light.

There have been other papers which have preceded this treatise and undoubtedly there will be others which will follow. The purpose of this paper is to share some of our experiences and to demonstrate what can be done even though a sizable programming staff was not available. This is a testimony of the capability of the new technology as well as the evolutionary emergence of one campus from those caves.

The University of Missouri - Rolla, with a student population hovering around the five thousand mark, is the smallest campus of a four campus system. All primary administrative computer systems are run at a central computer facility for all four campuses and maintained by a core of central administration computing staff. Any support beyond basic system requirements falls upon the campus IT staff. Prior to the advent of PCs, that included such items as new program testing, writing JCL, determining job failures, handling back-ups/restores and writing programs for campus specific reports. In order to address those responsibilities, the Rolla campus had at that time a support staff of a director, a secretary, four programmers and up to five keypunch operators. Today, that staff has evolved to a director and six programmers.

While the basic system support responsibilities are still there, many other responsibilities have been added in recent years. There are now desktop computers to support, administrative LANS to maintain, software to install and users to train. We download data onto a local server and have created applications which were not covered by the core systems. In addition there are Novell and Unix servers which invariably need some attention.

Fighting the fire breathing dragon-- a sentiment felt when everything seems to be falling around one’s neck. How peaceful and calm it seemed to be when we were still in the caves and focusing on only one problem. In those days, when a user approached this office with a hideous request, one could reply with; “It can’t be done on a computer.” While that is not necessarily an accurate statement, the length of time it would take to accomplish the request made it the same as an accurate statement. Users at that time were computer illiterate enough to accept the notion some things couldn’t be done on a computer and the users were patient. The mood was more, “when you get around to it.” They accepted the argument that creating a computer program was a long arduous task and that results were frequently measured in months and years. Unfortunately, that was also fact. COBOL, PL/I, even newer tools such as MarkIV were labor intensive languages and nothing could be done quickly. A well performing computer program was one which took an extreme effort and left the programmer feeling a batch of the finest steel had just been forged.

Comparing today’s users with the users of twenty years ago is a stark contrast. Today, users know ANYTHING can be done on a computer. While few of them could actually write their own programs, most believe creating a program is little more than pushing a button. It seems strange that with computer literacy comes such a false impression. But, the average user today is also an impatient user. No longer will they accept a development time line measured in years. Not only do they demand their project be placed at the top of your development list, but they insist you concentrate only on their request. If you find it impossible to address their request in their desired time line, they will find other ways to get it done on their own
PC, regardless of whether that is in their own best interest. Then, when it is ‘in production’ and invariably you are at a critical time-line on a new application, they will call you and cry for help. Their alternate method has totally failed them and they need immediate assistance in recovering their system. Oh, ... no there is no documentation for their new system. It’s on a computer; a computer programmer should be able to figure it out without documentation.

Of all of the events which this office has experienced in the Information Technology (IT) arena, perhaps the most rewarding and maybe the biggest burden is data warehousing -- a collection of data from all of the core systems plus the data created by locally written systems. Today that is the focal point for nearly every programming activity this campus initiates. Some way, some how, each project today involves the data warehouse. It has grown to be the foundation for virtually every request emanating from the campus. Everyone wants to take advantage of the availability of corporate data for their activities.

Data warehousing, as it is known today, had a very inauspicious beginning on the Rolla campus. During the early part of the eighties, computing costs and demands increased at such a rate the campus reached an absolute limit. After years of continual subsidizing, the campus administration finally drew a limit on the burgeoning administrative computing budget. Since the budget was managed by the IT office, the responsibility fell on the director to manage the activity. Naturally, every data custodian on campus believed their system was vital to the campus and they could not possibly cut back their activities. Instead, they demanded their support be increased; the end result being a programming load which could not be supported and a computing cost which could not be funded.

After trying every possible option short of eliminating vital reports, the Office of Administrative Data Processing (ADP), began investigating alternative solutions. The campus had an IBM 4381 which was not being fully utilized and there was a database platform on the IBM consortium agreement called SQL. We knew nothing about it, but the computing strokes were free and the software was free, so what could be lost?

Calling for the Oracle -- Following a review of SQL’s potential by one of the ADP programmers, a meeting was called in September, 1986 to appraise the campus data custodians of the potential solution to the existing problem -- insufficient funds and insufficient programming staff. That meeting was so incredibly impressive. Everyone of these participants had the belief all they had to do was to talk with their Vice-Chancellor and the money would fall from the local money trees. At the conclusion of the presentation all of the data custodians, save one, were totally apathetic to the problem and to the solution. The one individual who was not apathetic, was the director of the department we hoped to use as the springboard into this new approach. That individual was completely indifferent. He did not perceive the problem nor did he understand the solution. However, he did take the perspective that he was not opposed to what we wanted to attempt since it appeared the project would not affect him in the least.

After having been bolstered by the enthusiasm from the data custodians, the meeting with the ADP staff was invigorating. The individual who conducted the SQL review was willing to do whatever would help the campus. One staff member was doubtful it could work and the rest of the staff were either in opposition or believed it was a total waste of time. Obviously, with such clear cut support and enthusiasm, the time was ripe to venture into this brave new world. One could easily relate to the emotions General Custer or Christopher Columbus might have had at some point in their historic ventures. After all, were they not totally supported by those on whom they depended?

The benefits of a democratic society -- With the decision made and a clear majority in support, MarkIV programs were written to extract a subset of the data from one of the core systems from which three SQL tables were created. To make sure there was not a ‘political’ problem, the ambivalent data custodian was involved in deciding what data items were to be retrieved from his system. Since the idea at that point was to merely replace the less critical reports, the information principally dealt with biographic and address information. Because the data being downloaded was not dynamic, it was decided downloading three or four times a
semester would be adequate for the anticipated needs of the campus. Having established those definitions and with two people assigned part time to this project, the first set of queries were written. Since the staff members assigned to the project were not familiar with the system with which they were working, it was decided to create an SQL table strategy which corrected the programmer perceived faults in the core system’s basic design and to replace many of the seemingly inane system codes with ones which the programmers better understood.

Due to the lack of adequate dating of fields in the originating database, it was concluded that maintenance of the downloaded data could not be accomplished by refreshing the existing data. The only alternative was to delete the tables with each download and to re-create the tables. That was not the approach we desired, but it was the only manageable one. It is the one decision made at that time which is still in affect today.

 Barely had the new effort been initiated than the lessons and basic assumption fallacies were learned. Even though the data was static, it became immediately clear the data HAD to be downloaded more frequently. It was also clear the first cut of the data definitions was not as brilliant as at first thought. Consequently, with the data custodian involved, it was concluded the data downloads had to take place once a month and the SQL definition was expanded in a second iteration.

**Seeing through the dense fog**— From this meager beginning, it was clear the SQL solution did hold great promise for the UMR campus. It was confirmed this new platform could replace ALL of the hundreds of MarkIV programs which had been written for data extract over the years. With each execution of a MarkIV program there was an associated cost. However, with the SQL queries, a single download expenditure could provide the ability for countless SQL queries at no cost. Moreover, it was demonstrated a single programmer could create countless queries in a single day whereas a single MarkIV program could take several days due to the mode in which we had to operate. From a pure programming effort, a code reduction in excess of fifteen to one was observed. The need to reduce costs while addressing the burgeoning requests was clearly achievable with this approach.

An important realization from this beginning was that an individual did not have to be a degreed computer professional to create an SQL query. This method of reporting was so straightforward that many of the campus users could be trained to write their own reports. Such a solution held promise in addressing the dilemma of the suffocating number of demands for extraneous reports. It would also potentially allow the campus an avenue by which to enter the realm of the electronic informational system, a developing environment and an attractive administrative perspective.

In the final phase of development of this iteration, the ADP programming staff determined a set of rules had to be drawn before full production of local downloaded SQL tables could be initiated. The rules developed were as follows:

1. Only needed data would be downloaded and not entire systems.
2. The campus download efforts were to provide enhancements or extensions to core systems and would not attempt to duplicate existing services from the Central Computing Facility.
3. Access authorization to the SQL tables, download specifications and data interpretation were to be kept in the domain of the data custodian for that data.
4. Any activity associated with the SQL platform was the exclusive responsibility of the Office of Administrative Data Processing.
5. Levels of access had to be succinctly defined.
6. A schedule of downloads would be established, adhered to and made known to the data custodians.
7. Downloaded SQL tables would be autoloaded, eliminating the need for a programmer to work late nights and keeping maintenance out of the productive office hours.

In addressing the levels of access, three levels were clearly definable. The first level, requiring an absolute minimum amount of training would allow an individual to have only query access to SQL data. A second level of access, with additional training, would allow an individual to have query access and to submit stored SQL queries. The third level would, with still further training, allow individuals to write and store their own queries. This latter select group could fundamentally be
viewed as an extension of the ADP programming staff. In effect, they would become the increase in staff which had been so badly needed, but never seemed to arrive. (The first group is in fact the on-line inquiry users and not report users in the true sense. As it applies to reports, this group wants to merely ask someone else to run the report for them and have always been with us.)

During the first two years of downloading with SQL locally, progress was relatively limited. The staff had much to learn regarding SQL’s characteristics, table idiosyncrasies, terminology and system strategies. Frustratingly, continuing demands for services in other areas constantly took the staff away from the download activity. Additionally, in mid 1987 and then again in mid 1988, the programmer assigned the responsibility for the SQL effort left the employment of the university. In one year, two different programmers were assigned to the project and both resigned for work in other states. The staff had another individual who was talented in an extensive variety of computing activities and had some knowledge of SQL, but his talents were desperately needed to keep the other activities viable. Without the proper expertise and staff commitment, the SQL effort would die. Thankfully, a new programmer was hired in 1988 who had the ability and desire to carry the project through.

Rowing through molasses -- After three years of working with the download approach to report creation, the ambivalent data custodian decided to try his own hand at writing a query rather than relying totally on a programmer to do his work. Immediately, he saw the benefits this tool offered him, but he also raised issue with the design of the downloaded tables. Since he was familiar with the system from which the data was originating, he did not understand nor wished to tolerate the liberties which were taken with the SQL definitions. He also quickly realized the monthly downloads were not enough and the data being retrieved was grossly inadequate for his needs. Consequently, the third iteration of the SQL tables was born and the download frequency increased.

One of the beautiful aspects of the relational model of processing is its ability to handle major table (file) definition modifications without affecting the data. Making system changes in the more traditional IMS environment is a major undertaking; requiring data unloads, system redefinitions, and data reloads. In SQL, changes in one table can be made without affecting either the data in the table involved or any of the other logically associated tables. Major table redefinitions are basically inconsequential to the applications using them. Therefore, the new table definitions did not have a significant impact on the local activities.

In the third iteration of the system downloads, it was decided the best table design was one which emulated the IMS file structure of the system at the Central Computing Facility. That data scheme was one already familiar to the data custodian and would eliminate occurrences of report irregularities which the second table definition inserted. Surprisingly, it was found very easy to replicate an IMS structure in a table oriented scheme. That hierarchical system could in fact be replicated precisely in the relational mode. In the redesign each SQL table was defined to closely imitate an IMS segment in the core system.

At this time, the data download definition was increased to the extent that nearly all of the system’s data was now to be downloaded to the UMR campus mainframe. The number of tables increased from the original three in the first iteration to thirty-one in this design. (This did not count ‘frozen’ files and other tables created for auxiliary reporting) These newly defined tables included not only basic data, but also items which in the original definition were considered inconsequential. Additionally, nearly all codes were returned to the system’s original values. The information downloaded now contained a data history since 1980 as well as information on active individuals. With concurrence from the data custodian, the ADP staff decided to download all dynamic data after midnight every Monday morning. More static data were downloaded at intervals relevant to the data characteristics or upon demand of the data custodian.
After the expanded definition of the local data was complete, the data custodian became very active in writing his own reports. Instead of referring requests for reports to the Office of Administrative Data Processing, he proudly began handling many of these requests himself. At his request the ADP staff created a table of the dates of the downloaded data to allow him to determine the currency of his data in the various tables; a very valuable addition.

As the initial data custodian was becoming an advocate of the concept of downloading data, a second data custodian approached the Office of Administrative Data Processing to begin downloading some of their system data. They had been inundated with requests for reports from their system and they had experienced budget associated staff reductions. As a result they could not meet the demands of the campus users. Their desire was to “empower” the Deans offices, and the Office of Vice Chancellor for Academic Affairs with the ability for them to write their own queries; thereby relieving the pressure for many of their requests. As a result of this request, a cooperative training session was designed with the ADP office conducting the technical training and the data custodian conducting the data training.

On the bleeding edge -- From this beginning, the UMR data warehouse was born, providing capabilities which far exceeded the original scope. While there had been efforts in the past to write reports from these two totally independent core systems, those efforts had never been successful. MarkIV just simply could not handle the file coordination between the two systems, which had different types of record keys. Now, with the downloaded SQL tables, those reports were as easy as saying the word ‘join’. A significant void had been filled by a technique which would be used commonly to jointly report from many of the campus’ locally resident administrative systems. The immensely popular concept of downloading data was proving to be capable of what was desired and much more.

In order for the Office of Administrative Data Processing to make progress with the SQL efforts, a closure was placed on all Mark IV efforts at the Central Computing Facilities. The statement was made that (with few exceptions) any Mark IV program which needed to be run, modified, or written would be written in QMF against the local SQL tables. This statement did not meet with welcome acceptance by the ambivalent data custodian, but the move had to be made. Despite intermittent complaints from that individual, the decision was adhered to and new reports were created without affecting the schedule of the requesting users. Rather than embarking on rewriting ALL Mark IV programs in a single massive effort, new QMF queries were written against local SQL tables as requests for the Mark IV programs were received. Consequently, the conversion took well over a year, but did not have a noticeable impact on the workload in ADP, nor did it adversely affect the campus user community.

Confirmation of the Endeavor-- After ten years with the relational environment the Rolla campus of the University of Missouri now frequently downloads portions of all of the core administrative systems. Dependent upon the needs of the campus user community for the specific data, tables are refreshed nightly, weekly, monthly or upon demand. The original three defined SQL tables have increased in number to over twelve hundred. Additionally, new applications have been written for; equipment inventory, student admissions (prospective students and applicant tracking), a departmental/student telephone tracking and billing system, and a chemical tracking system. Due to the success of this endeavor and the requirement for the chemical tracking system, a Unix server was acquired in 1993 and Informix was selected as the relational database platform.

With the acquisition of a dedicated Unix server, an effort was made to identify a software product to disseminate to the anticipated user community to replace the QMF facility available from the VM system. After reviewing the options, PowerSoft’s PowerViewer (now called InfoMaker) was selected as the software product to provide this solution. This choice was made because it provided an extensive range of abilities at an affordable price.

After this length of time:
1. Only a scant handful of campus written Mark IV programs are still in production.
2. The costs for many reports to end users have been reduced by eliminating the computing charge from the Central Computing Facility.
3. It is now possible for a campus user to have a requested report complete in only an hour or two rather than several days. (Such spontaneous requests are not encouraged, but it is a significant boon to be able to do so under unusual circumstances.)
4. Programmer productivity has been impressively improved.
5. Requests for reports from disjoint systems are now being processed.
6. User access to corporate data for reporting purposes is now possible without jeopardizing the core systems and without interfering with production activity since all local activity is restricted to locally maintained tables and not at the site of the live data.
7. Campus expenditures at the Central Computing Facility have been reduced with further reductions projected.
8. Computing activity greatly increased without the addition of a large cast of computing professionals.
9. The Office of Administrative Data Processing has advanced from being an office totally dedicated to writing miscellaneous reports and maintaining JCL to one which actively develops new applications for the campus.

To date:
1. Of the corporate systems (accounting, alumni, financial aid, grants, library, loans, payroll/personnel and student) only the library does not have data downloaded to the campus or considering the move at this time.
2. In excess of sixty campus personnel have InfoMaker installed on their desktop units to allow them to create their own reports.
3. Since 1987, members of the faculty have been able to download an electronic class roll to a CMS account or to a PC to create their grade book for the semester. This routine accesses the current student course enrollment table.
4. In 1991 an electronic bulletin board was made available to the campus community (students and employees) via Gopher to provide them an electronic view of a variety of campus public information. From the student system's portion of this electronic information system, an individual can review the course catalog since 1980 and the current schedule of classes. (Both of these items are downloaded from corporate systems.) (In 1994 the ability to view course equivalencies was added to their access list.)
5. Also in 1991 electronic telephone directories were provided to the campus. From two separate directories a user can retrieve the current local address/phone for the student requested or the current office address/phone for the selected employee.
6. From this beginning the students today have available from the Web: access to their degree audit, telephone billing information, Cashier’s billing information, a “share-a-ride” feature, electronic surveys, student council balloting, teacher of the year survey and co-curricular transcripts. Until the features were added to the core system application, it was also their only on-line access to: address information, biographic data, cashier’s billing detail and telephone registration schedules.
7. Students and faculty alike have access to our MSDS (Materials Data Safety Sheet) from the chemical tracking system via the Web.

**The sound of the wings of the bees** -- In reflecting upon the operation in existence today compared with what this campus had ten years ago, one could erroneously conclude matters were well in hand and the operation was working smoothly. Sadly, that is no where near the case. While downloading data to create a data warehouse in order to distribute corporate information was a saving advancement a decade ago, today it is frequently a significant drain on a very small programming staff. Certainly, we have succeeded in empowering the users and have provided a mechanism by which reports could be written which we would not have had time to produce. With one user who has created over four hundred queries, another one with over seven hundred queries and with over sixty people with InfoMaker installed on their desktop, plus access to the data from the Web, maintaining the currency of the data warehouse has become a constant effort. No longer can we take the refreshing of the data lightly. If a download fails, we can not shrug our shoulders and say “wait until
the next one.” This has become an additional production environment and must be addressed the moment there is a problem. With the number of jobs now being run to refresh the local data, download failures have become an all too frequent occurrence. Since people are depending upon the tables to do their ad hoc reporting, they need the tables and they need them complete. It has added to our load.

The requirement for data warehouse support would have been difficult enough to maintain if the level of user activity had remained constant. However, with success came other realized needs and users began considering alternative reporting which did not involve the traditional mainframe. As a result, never, never have there been so many users coming with requests for services from so many different direction as we have today. Due in part to the popular new concept of “empowering the users”, literally everyone today wants computer services and virtually all of them deal with the data warehouse in some manner.

**Shooting one’s own foot** -- All too often those individuals touting the concept of “empowering the user” do not mention it is a risky venture. As computer professionals we are expected to know our programming language, our system and its idiosyncracies well enough to know how to write a report correctly. We also have been trained to think through the request, test our programs and to verify their validity. The casual user exercises none of the processes we as professionals do. Such users hastily conceive what they want, throw some ideas into the “too easy to be true” query tool, run the report and widely disseminate it. Their confidence in the accuracy and the validity of the resultant is in the fact that a computer produced it, therefore the product must be right.

This expected phenomenon will exist in even the most perfect environments, but only through proper training of the users can the extent of occurrences be reduced. Due to the sophistication of today’s tools, technical training is not nearly as critical as the data training. It is imperative the users not be allowed to have the software installed for their usage until they have completed required classes in the use of the product and the data characteristics of the system(s) they will be using. Easily used documentation is essential and follow-up training is valuable. Without a sound approach to distributed processing, it is tantamount to “shooting one’s own foot.”

While the proponents of “empowering the users” fail to emphasize the risks, they also frequently assume the staff to support a successful implementation is in place. As with any heavily used application, the daily and occasional users will encounter problems and will need help. From the programming perspective, the effort required to support those user problems is several times more involved than doing the work themselves. As a result, programmers will find a certain portion of their day consumed with resolving user created dilemmas. Therefore, a distributed system emphatically requires a larger staff than the traditional method of staffing. There is definitely no problem with job security for the programmer.

In disseminating the software, one can not expect every requesting individual will become a user of the software. Unfortunately, there is not a reliable method by which the non-users can be weeded out. Of the sixty or so InfoMaker installations on this campus, something under a half of those staff members actually use the software. Do not believe any of them were coerced into getting InfoMaker. Everyone of them requested the software be installed on their system and everyone of them had the best of intentions to use it. However, once they went through training and realized it takes more work than calling someone else to run a program for them, they quickly procrastinate writing their first query. Everyone of them realize they have so much work to do they honestly do not have time to write their own reports. Consequently, those who do not use InfoMaker, call the programmer to run the reports as they always have.

**Up to our necks in quicksand** -- With years of successful experience, there came a conviction that the campus had a sound basis for a distributed system and the staff could handle the foreseen growth in the activity. However, last year the DBA and creator of all of the table domains and routines left the campus. The knowledge drain can not be described. Suddenly, we find scores of routines which are outdated and need to be replaced; reports run, but the source not known; a triannual software update created havoc with many routines making
download success questionable and users crying for tables to be re-loaded, modified or created. The UMR operation had always been run on a bare bones staffing, its mere existence proof of the power and success of the local data warehouse. However, that foundation was clearly shaky from the beginning, but it was all we had. Since at no time had there been more than one and a half FTE directly assigned to this endeavor, that loss has had a rippling effect throughout the operation. The situation in which UMR finds itself is recoverable, but unfortunately, the users have had to suffer an intermittent disruption of services as we recuperate from the blow. Until the fragile staff can be rebuilt, we will be struggling through quicksand.

This is not to say we should not have ventured into the data warehouse venue. With the locally available data we have created the ability to write reports in terms of minutes and hours rather than hours and days. Clearly, as a result of the efforts of the programming staff and those active InfoMaker users, we have hundreds of reports which would not have been created otherwise. We have been able to extend our programming efforts by “empowering” some of the users. The point is that if a campus is entertaining the idea of venturing into this arena for the first time, there are heavy costs to this venture in addition to tremendous benefits. One must have their eyes open before taking the plunge. However, as computing professionals, no matter what the situation may be, we can not allow ourselves to remain in the caves and keep our scope narrow with antiquated tools. We must keep abreast of the current technology in our profession.

Rubbing two sticks or lighting a match? -- “Empowering the user;” a state of the art term we hear repeatedly with the newly touted concept of process re-engineering. It is a fashionable cliche for an age old phenomenon. Throughout history man has experienced, promoted the concept of empowering the user. It is at the heart of engineering and the very essence of invention. Every generation, every age has made its own contribution to this notion. We could trace it back to the introduction of the wheel, to the invention of the plow, even to the discovery of controlling the physical phenomenon of fire; a breakthrough given to us by the caveman. An unlikely divergence from late twentieth century computing? Perhaps not.

Clearly, there had to have been a single day in history where the first ancient man found the way to start and control fire. It undoubtedly was first introduced by stealing a burning stick from a fire caused by nature. They had to have religiously guarded that wondrous flame since it was irreplaceable if it went out and the naturally ignited fire had been extinguished. With time, someone found they could generate that fire by rubbing two sticks together. Today we consider that to be archaic when all we have to do is light a single match. Consequently, we think nothing of the work required to ignite a fire, but how many of us have ever tried to start a fire by rubbing two sticks together? Sound simple? It’s not. This had to be a skill which was learned, then taught only to a select few. Understanding human nature, there had to be just anyone capable of understanding the concept of rubbing two sticks together nor could just anyone be trusted with the phenomenon which was vital to the cave society’s existence. This was a special skill and conceivably placed the firestarters near the top in their uncomplicated society.

It could be reasonably concluded the cave family established levels of responsibility as it related to building and maintaining their fires. In all likelihood, anyone in the tribe collected the firewood under the direction of the firestarter. He presumably had apprentices which helped sort the wood by size, type and dryness. Building the stack of wood for the fire could have been assigned to his more advanced apprentices as the construction of the pile can be critical. They had to know wet wood would not burn, too many large pieces would not catch and some varieties of wood created huge amounts of smoke. When it was all ready the firestarter rubbed his two sticks together and transferred his small flame to the pile of wood. The final product, the one the entire cave society depended upon, was the result of one very skilled person.

For eons man relied upon the skill of the firestarters to create one of the necessity of life; fire. Over the centuries there were other techniques introduced, but all of them relied upon the special skills of a few people. It was NOT something just anyone could do. During moments of history these skilled individuals had to be viewed as heros to the masses. Their whole society depended upon their talents; their skills with “magic.”

Just think what happened to those magicians, those heros, when some creative individual “empowered the users” by
inventing the common match stick. With that simple creative stroke of genius, any person, any age could start a fire. What
seemed like a profession to carry them into eternity was suddenly terminated by an invention which required the simple flick
of the wrist. Their day in the limelight was terminated. Of course, today, we have to be concerned with people starting fires
from ignorance and from maliciousness. We also have an entire profession, firefighters, whose job is to extinguish human
as well as naturally caused blazes. However, that is progress.

A divergence from the topic of distributing information? Totally unrelated? Consider the brief history of computing to date.
Is it not very analogous? As computer professionals, just like the firestarters, we at first were the sole entity entrusted with
producing the needed results and to interact with the “magic box.” Other people could utilize the fruits of our efforts, but
no one else was skilled in the procedures required to do our work. For several decades it was safe from infiltration of the
masses who were computer illiterate. They weren’t capable of doing the work themselves and were entirely dependent on
our skills, our good judgement and our good nature to get their work done. If we didn’t do it, what could they do about it?

The advent of the PC is very much the same as the advent of the match stick. It put in the hands of the masses the idea they
could do all of those wondrous things which they had been told could not be done in the past. Believing ANYONE could
program a computer, the users are now vulnerable to every slick talking salesman, cute television advertisement or any
brilliant idea their minds can conceive. Only when they can’t convince their computer to do what they command or their
software to operate as promised, do they contact the computing professional. There are so many problems out there, today’s
programmers simply can not handle the demand.

From the administration’s point of view, what is becoming a crisis is either not understood or not fully appreciated. After
all, if it is on a computer, especially one of TODAY’S computers, it surely is little more than “pushing a button,” or “rubbing
two sticks together.” Right? Haven’t you heard those advertisements on television or read numerous articles on the subject?
Many of them make it sound so simple. ... If one compares today’s modern client/server concept with the mainframe focus
of twenty years ago, it is not that much different from the firestarters and the match.

The match in its appearance seems so simple; not the least bit complicated, no moving parts. As a user all we have to do
is strike one on a surface and we have instant fire. However, have we ever stopped to think what it took to empower us in
this fashion? Just to create a match; a tree has to be cut down, taken to a lumber mill, shredded into tiny pieces and sent to
the match factory. The manufacturer must dip each stick in the phosphorus mixture for the head, package it, distribute to
the distributor or warehouser and finally place it on the retailer shelves for the consumer. Behind the scenes there are
business executives, accountants, advertising people, secretaries and even janitors. Certainly, the invention of the match
created jobs and made life easier, but how many people did it take to replace that one firestarter? How quickly we take
progress for granted. How quickly we forget what it takes to achieve that progress. That same phenomenon exists with
distributing information systems. There is work behind the scenes required to empower the users in producing their own
reports. We must not allow ourselves to be so caught up in the aura of our goal to ignore the requirements to achieve that
goal.

Where are they today? -- As society has progressed, one might conclude the ancient firestarters have perished from the earth, but they have not. Certainly, around the home the miraculous match has eliminated the need for a skilled firestarter, although occasionally a user burns a house down from careless use of a powerful tool. But, still today we employ people whose primary responsibility in part or whole is that of the firestarter. We will find them maintaining the boilers, deep in the “caves” of our modern buildings. The high tech equivalent of those early professionals can be found in our nuclear power plants having similar responsibilities using modern machinery and techniques. The firestarters have not disappeared, they still play an important role. It’s just they are not the valued commodity for the society as it was when fire was the responsibility of a sacred few. As IT professionals, we risk the same fate if we do not take action.

In the university environment today, we are faced with a very real threat. As information technicians, we can not take the
perspective that we can ignore the inventions of data warehousing, client/servers and PCs. We can not stay in the caves and
continue to rub two sticks together because that is what we have always done AND it gets the job done. We must be out in the forefront in technology and to the best of our abilities keep abreast of it. In our business, change is virtually an everyday occurrence. It took centuries for the firestarters to have to learn new skills. In today’s technological world, we are progressing factorially every decade, requiring the computer professional to be in a constant learning mode. There is no choice for us. If we stagnate, the users today will simply pass us by like water rushing past a bolder in the middle of a stream. Their end results will not be to our satisfaction, but they will not wait for the computer profession to gear up. Today’s society has misplaced the word “wait.”

It must be well understood by our administration that empowering the users has a price, a very big price. There is no disputing the fact data must be distributed, shared and available for all who have a need for it. However, administration must understand the one programmer who previously supplied all of the reports for one department can no longer be the sole support. Now, the user must be trained, there are help desk responsibilities, download work( if that is the approach), countless LANs to support, numerous data warehouse servers to support, complex telecommunications to maintain, Web activities, software to evaluate and still produce reports for those who can not produce their own. If a distributed system is going to succeed, it must have more than one person to carry the load. It must be an entire campus commitment and to be as successful as the administration wants it to be, there must be sufficient staff to support it. They must also understand the desire is appropriate, but there is a huge risk in providing casual users with the power to create reports from systems with which they are only faintly familiar. The degree of that risk is inversely proportional to the ratio of support staff to users.

**What has been Learned:**

1. The relational technology is vastly superior to more traditional methods. From all perspectives it is a very exciting environment.
2. The complex level of security in SQL and its design flexibility are extremely impressive.
3. A single SQL table or set of tables can be defined by the DBA to present different images to different users depending on the requirements of the user.
4. Users can be trained to write reports from administrative systems, striking down the antiquated position that database reporting should be reserved for the select few.
5. The most critical ingredient in the success of user written queries is the training in data characteristics. The matter of SQL concepts is secondary.
6. Computing directors must have patience in dealing with data custodians. They have legitimate reasons for conservatism and an acquired fear of new techniques which can threaten the stability of their operations. This does not mean capitulations on the part of the computing director, but a combination of encouragement and understanding. Assurances have little affect on users when their existence is involved. A director of computing must continue to explore and promote new computing techniques, but realize his advancements are not going to come quickly. This is especially true if there is insufficient staff to carry the improvements to quick fruition. Change comes slowly.
7. Data custodians can be expected to vacillate with time, especially if the idea was not their own.
8. To the users the relevance of the future is perceived through the perspective of the present time and present problems.
9. Progress does not come with a continuous advancement, but rather ebbs and flows.
10. Creativity must be accompanied by optimism, enthusiasm and a great deal of planning.
11. User expectations can frequently be anticipated as unrealistic and short-sighted.
12. The user community must make the mental adjustment and acceptance of the new methodology before the technology can be successful.
13. Computing directors must remember where there is a will there REALLY IS a way.

**The leading question** -- Knowing what we know today, would we do it again? Without hesitation the answer is yes! It is certain one would prefer to initiate a venture of this magnitude knowing the staffing was there to make it an overwhelming success. However, after years of requesting and
forecasting the needs for more programming staff, it was understood none could be expected from the administration. Because they are not experienced in computing activities, administration as a general rule does not relate to the effort it takes to write a single program. To most of them it is “magic”, so just have the magicians perform more magic. Even though this has not been supported to the level it justly deserves, it is felt it to be better to have a cake to eat without utensils, than to have no cake at all. Waiting until the staffing was right would have us deep in the recesses of our computer caves today. As computer professionals, we must take a leadership role. Without taking the responsibility to direct the path of the water, we become the bolder in the middle of the stream around which all of the water is rushing. At some point that bolder is worn down and with time can become a mere pebble in the bottom of the stream. Whether it is a pebble in the stream or a Neanderthal in his den, we must come out of the dark recesses of our caves and into the light.