

Migrating Embedded Government Workflows to Better Software such as ArcMap

Compared to ArcTools and ArcView, the relatively new ArcMap software offers many useful benefits. ArcSDE is superior to ArcSTORM for multi-user editing. Oracle 9i is better than Oracle 8i. Windows XP and Active Directory are better than Windows 2000. The geodatabase offers custom topological relationships that were tricky to maintain in the older coverage format. VB7 has some big software development advantages over VB6. The whole computer world is in hyper-drive. At the same time, government budgets are getting tighter. The vendors are all in a frenzy offering fabulous improvements; jumping up and down on the waterbed around us. We are clinging to the waterbed with our fingernails. We are trying to provide stable and reliable workflows to city and county employees, who for the most part, don't like change. As the vendors all improve, we must keep pace. However, changing software embedded in workflows is not a trivial matter.

Introduction to this Train of Thought

Looking out over this topic of 'software migration' from inside the head of an amateur philosopher, it's difficult to know how to jump directly into the topic. Inside my head is Gilbert-the-frog, sitting on the edge of the train tracks. Many trains of thought are moving in different directions at different speeds. Like the Frogger video game somehow Gilbert-the-frog needs to navigate to the other side.

This first set of tracks to be crossed carries 'The Writing Style Train'. A big old heavy metal train can easily squash a tiny little philosophical frog. Should this paper be written in a straight and serious style, or in a convoluted fun style? Well, past empirical experience from writing papers has taught me two things. Serious papers are concise and to-the-point. They get the job done. However, when Gilbert the frog tries to write papers in a serious style he gets very little feedback from readers, implying that no one actually really read them. Conversely, when he writes in his own croaking voice he gets more feedback from readers, implying that the fun style is, in fact, better at getting the job done. So, off-we-go, let's just tense up our springy frog legs and hop past this first 'Style' train, blindly ignoring the convention..... Not the software-convention, Gilbert means, hop on, ignoring the convention of looking 'left-right-left' before crossing the train tracks.

Well that was easy. The next train of thought that needs to be crossed is what Gilbert calls the 'Big Picture Train.' Let's see if he can hop across this next set of tracks without getting squashed. Some of you are probably thinking: 'This is going to be a loooooong reading.' Well, relative to a newspaper article, yes. Relative to a book, no. Gilbert the philosophical frog has provided a convenient emergency exit from this article. The bottom-line of it was written for any, busy, bottom-line, goal-oriented folks who want to have all the points spelled out cleanly, neatly and quickly. Simply skip to the bottom line if you get bored.

The Big Picture Perspective

Companies and government organizations are moving slowly from paper workflows to computer workflows. City and county departments do not function totally in the paper world or in the computer world. As a society, we are still transitioning our work and our data from index cards, manila folders, paper files and drawers into digital information. The task will not be complete within Gilbert-the-frog's lifetime. Still, the public expects faster, more efficient computer based services.

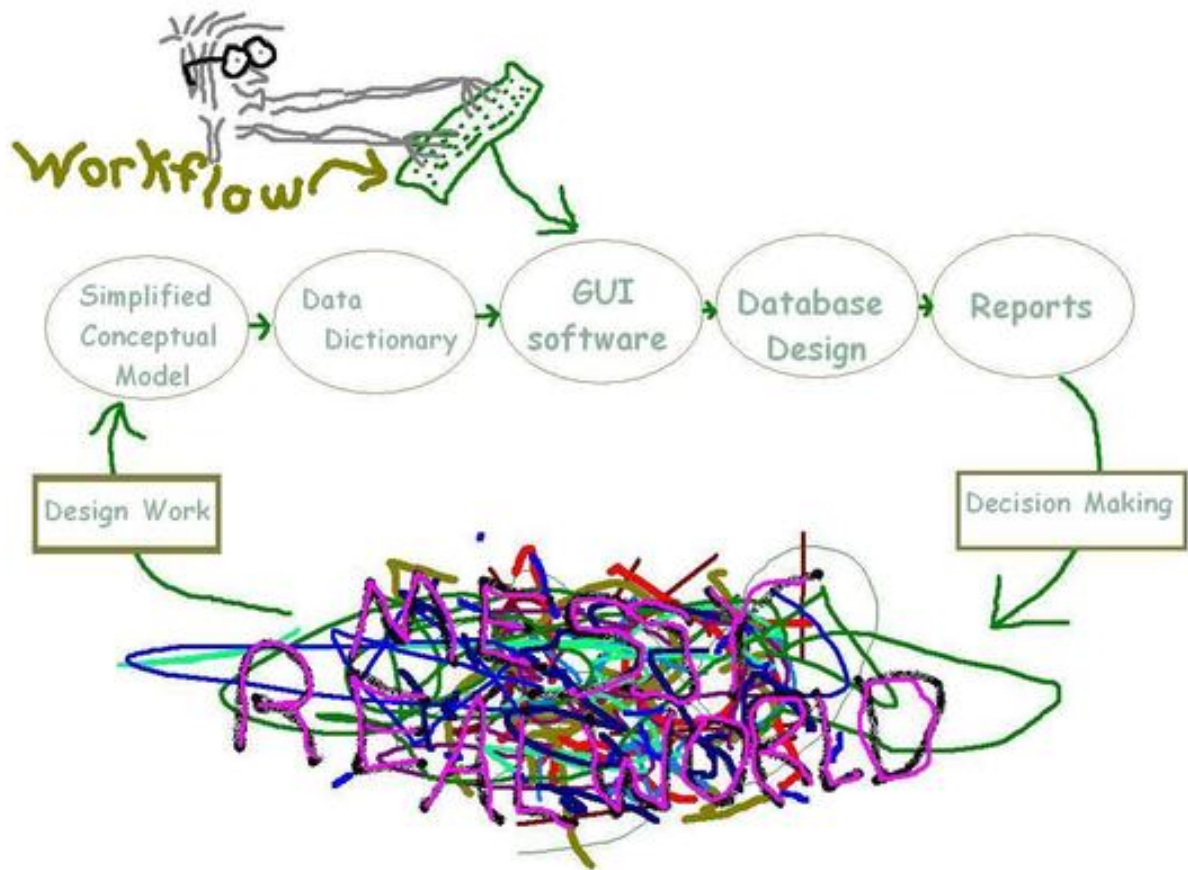
Younger folks in their thirties and forties, grew up using the first personal computers. Folks in their 50s and 60s did not grow up using personal computers. In general, older folks fear computers, more than younger folks do. Are the fears unreasonable? Not totally. The paper is there. You can see it, feel it, touch it and control it. It's understandable. For a lot of folks, computers in general represent an unknown risk. Computers crash. Software upgrades. Vendors go out of business.

What if we need to do some real world work but the computers don't work? There is some risk in moving a paper workflow onto a computer. We've all heard of or experienced some computerization horror stories. Sometimes the computers seem to make things more complicated. There is wisdom in changing workflows slowly to computers and running them in parallel with paper for a while.

On the other hand, imagine if you will, walking into a bank and seeing the teller write your withdrawal information into a paper ledger. Imagine getting a hand-written list of credit card transactions from your credit card company. If you walk into our county administration building and pull out the taxmap for your area, it's an old crunchy hand-drafted map. The deed for your house is in a manila folder. And the system works that way. The paper taxmap drafting goes on even in 2004, even in a county with one of the most successfully integrated GIS projects in the country.

This 'big picture' train is moving fast, exponentially gaining speed and momentum, just as the computer-database-IT-GIS world is exponentially gaining speed, and momentum. Gilbert-the-frog, is watching it pass, mesmerized, trying to calculate the timing between the cars. He has to spring through the tiny space of this moving train to cross this set of tracks.

In local governments everywhere, there is a struggle to computerize workflows and integrate databases across various departments. It is easy to become so engrossed in the technical details of implementing GIS that we forget this 'big picture'. Computers help us to store information about a complex, real, world.



Andy Barth, 2004

GIS designers look at the complex real world, and simplify it into computer database models (GIS layers) which are useful for spatial analysis as well as statistical analysis. The computer geeks of the modern world conceptualize reality into database designs, and also software GUI's. A software GUI is often a front-end for a database. People doing their daily work interact with the software GUI. Codes from a simplified model of reality (a data dictionary) are stored in a database, which is stored on the computer hard-drive. Databases are then analyzed and reports (and maps) are generated for decision-making purposes related back to the real world.

The GIS field is a subset of the Information Technology field. In another way, IT is a subset of the (much economically smaller) GIS field because IT is about databases which often lack real location. GIS is about complex location-oriented databases. IT is about complex databases, so Gilbert will refer to it as the GIS (slash) IT field. It's easy to get rapped up in the technical details of GIS/IT and forget the bigger picture. Computers are not decision-making tools, not really. A brain is the decision-making tool. Computers help store and index transactional and inventory records. Computers help us analyze simplified data, and make reports and visual graphics which help people to communicate their ideas. Right? The purpose of GIS/IT and workflow computerization is decision-making in the messy real world.



Organizations need to store their data in databases rather than on paper. The 'big picture' is that simple. Information Technology is primarily about how to store data into databases, and later, report that out for analysis and decision-making. People may have various excuses (or valid fears?) about not wanting to move paper workflows onto computers. However, in the long run, the simple fact is that computerized records are more efficient than paper records. Organizational change is necessary when people who have been doing their work on paper suddenly need to change and do the same work using a computer instead. This change sounds simple but it is not. Why not? It's expensive because computerization of workflows requires a lot of design work. An even larger difficulty is the inter-organizational sharing of data.



The 'big picture' train cannot be ignored. The transition from paper to computers seems like a slow process. But the world of computer technology and databases is a gigantic train with gigantic momentum. Ah, here comes another space between cars. It's time for Gilbert to make the logical leap, through this topic and onto the next set of tracks: money. This is another big, fast, scary train.

Financial Issues for Local Government

People have been moving from our city into the suburbs and surrounding counties for over 20 years. The tax base is smaller, the schools are deteriorating, and the maintenance cost of repairing old infrastructure continues to increase. It's a cycle that many mid-western cities face. Surrounding farmland is converted into buildings and pavement. Over the last 20 years the proportion of developed land in the region increased by 140%, while population increased by just 15%. Local urban population declined during that period. The end result is that many local governments have decreased tax revenues and increased costs associated with suburban sprawl and urban decay (re-development). So, local governments need to be very efficient with their money.

Based on crime trends, how we deploy police to maximize our effectiveness and minimize cost? The pipes need to be replaced, but which pipes should we replace, and when. Who is working on a particular street, and how do we keep departments from tearing up the street, repaving it, and then having someone else come along and tear it up again? Our city has a high cancer rate. What causes this? Is there a pattern, how do we predict it and change it? What are the demographic trends of the future, how should we plan transportation and development?

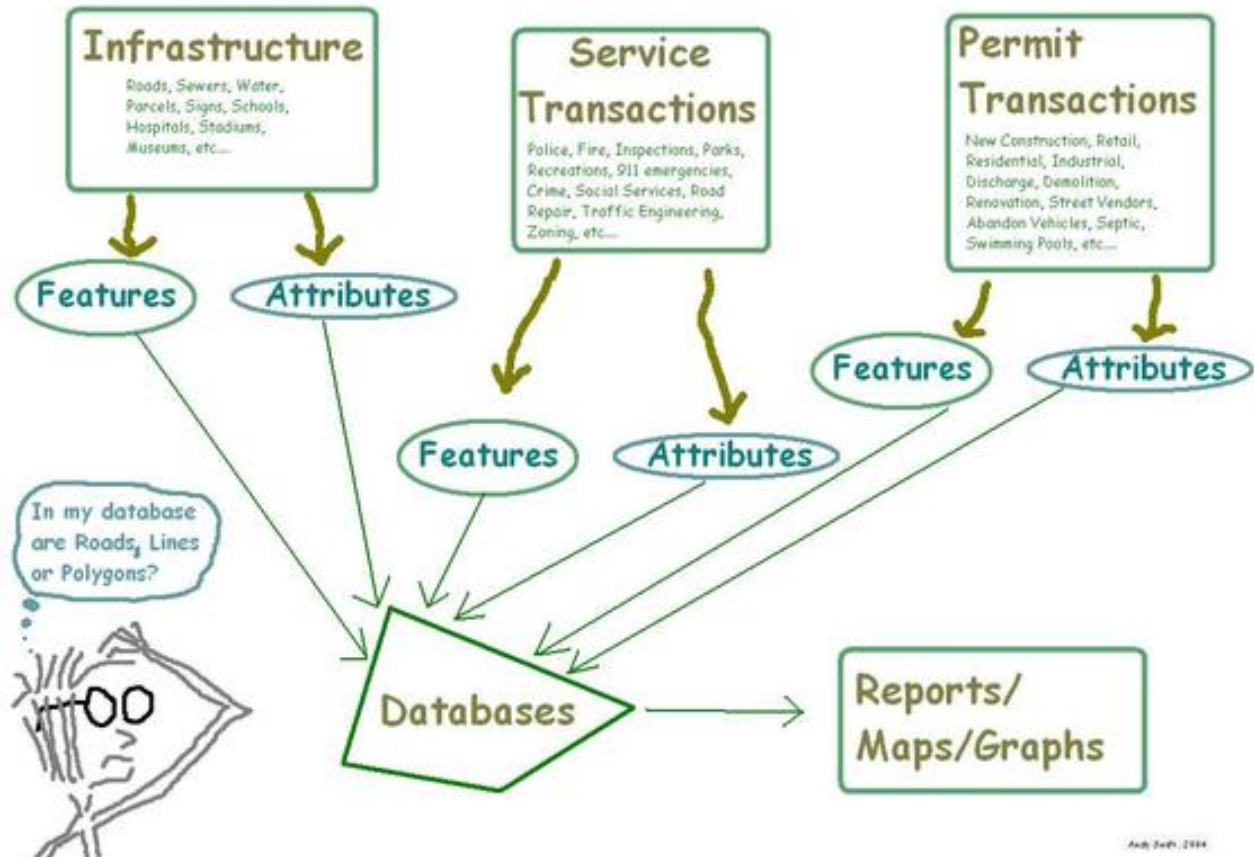
To plan for the future, community leaders need to ask hard questions and use computers to try to answer them and develop plans for the future. Cities compete for businesses, not just American cities but world-wide, cities compete. The efficient cities that provide more services for less money will attract vibrant populations, and growing businesses. GIS/IT databases are a more efficient long-term storage media than paper records.

The main functions of local government are to provide services and manage infrastructure. Community development and planning is implemented through the permitting process which is the main way that citizens have a say in how their community is developed by corporations. Therefore, CAGIS is deeply involved in integrating GIS with the permitting system. It is in the permitting process that many workflows originate. Data should be entered one time, at the source of its creation.

In order for database reports to be useful for decision-making, those reports need to be based on 'up-to-date information.' Obvious right? Well, it's a tricky thing. In order for the database to contain up-to-date information it needs to be imbedded into daily workflows of people doing the work. This means work needs to be done using computers rather than paper. If paper forms are used then the data must be retyped into a database, which is an expensive extra step. Data should be entered into a computer, one time, at the source, and then propagated down the workflow to all other people in all other organizations who need that data. This is the cost efficient way. It's easy to describe this utopian vision but it's difficult to implement.

Local Government data stores information about public infrastructure, public services and permitting transactions. Infrastructure can usually be modeled as lines or polygons. Service transactions and permits can be modeled as point locations.

Local Government Records



A main function of GIS is attaching locations to records in a database. A workflow should capture not only the nature of a transaction but also the time and date when it occurred. Usually folks just use addresses as a location. But addresses are only a surrogate for real location. The spellings of addresses and names are tricky to standardize and use as a primary key.

Many organizations omit the extra cost of capturing the location of records when they are input into the database. I'm willing to bet that fifty years from now many organizations are likely going to realize that the location is necessary to use the data for the kind of questions they want to answer. GIS software can help integrate data across departments but it is not a quick fix to bad enterprise information technology design.

Addresses are important for interdepartmental database integration. Addresses and locations are the key to how various city and county departments join together expensive and separately maintained customized databases. In many cases the only common field that could be used as a primary key across departmental workflows is:

name and address. That is why citizens end up writing it ten million times on various paper forms.

When departments and organizations initially design their databases they don't always plan into the future. Past database designers have not always planned ways to join inter-organizational databases. Therefore, CAGIS has spent many years working with city and county departments to try to re-design and standardize address formats so that databases can be used together. It seems like it would be a simple thing, but somehow, for large organizations to standardize and share data is difficult.

The money train is approaching quickly, hurry Gilbert, hop onto the next line of reasoning....

Sharing Data and Inter-departmental Efficiency

Even if it is not their intention, sometimes people get territorial about data. One department sees their data in terms of what they need it for. It is very beneficial for inter-departmental data sharing to have a neutral territory and a schema for sharing data together.

Besides, think about how hard it is to organize files on your own machine. It's so easy for a department's network drives to get cluttered with files. Often departments are on different networks, sometimes different operating systems, in different buildings. It is critical to have an agency in the middle with the role of managing inter-organizational data.

Local Government Databases



It is not necessarily recommended to have a separate GIS/IT department that handles GIS/IT data and software. It may not work to put a bunch of computers in a GIS/IT lab and staff it with experts. Organizations cannot (should not), expect some outside department or consultant to solve all their database, workflow problems. Each department needs to have folks using the GIS/IT data on a daily basis. For example, new database records are generated when a subdivision is approved, permitted and built. Each department needs to have trained staff to update their own data and help users in that department understand and use their data.

The consortium solution works well because GIS/IT specialists staffed by the consortium can work with GIS/IT specialists from each department. Does this seem like a lot of GIS/IT specialists to hire? Don't forget the big picture. Organizations everywhere are moving from paper to computers. It requires a lot of commitment to make this shift.

Gilbert-the-frog is squatting on the tracks. Quick, hurry again and jump away from the ego-train. It'll run you over. Organizations are usually driven by people with strong personalities, who know what they need, who will do it their way, who don't want to

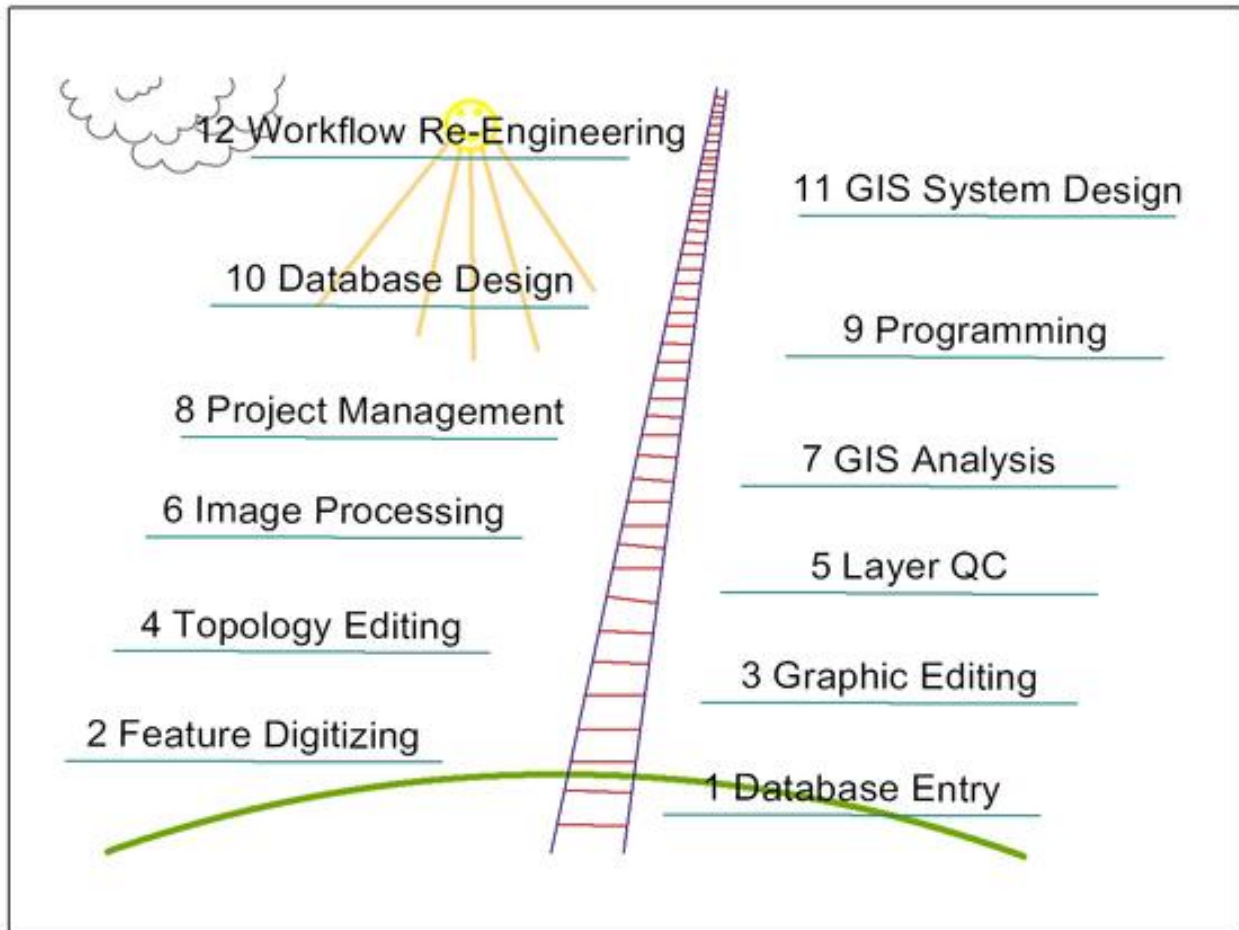
release any control of their data to people from another organization.... Spring to a higher level of thinking little froggy.

Why do GIS/IT projects sometimes fail?

Anyone who is a high-level manager and who hates computers is not going to like my answer to this question. I'd guess that GIS/IT projects usually fail because they are managed, (funded) by people who do not understand information technology. Generally, generalizations are flawed; this one is probably no exception.

Don't misunderstand Gilbert on this point. He's just a frog. It's pointless to become upset over the opinions of a frog. Experience using computers is a different skill than experience reading and managing other people. Naturally, understanding and managing humans is a critical skill for any business or organization. People are infinitely more complex than computers. However, on a GIS/IT project, trouble arises when the more 'people-experienced' managers pull rank and ignore the concerns of the more 'computer-experienced' managers. If you don't believe the frog then look harder into your own experience. Gilbert-the-frog isn't trying to sell anything. He's just hopping through the train-yard, trying to avoid being squashed.

GIS/IT professionalism is a craftsman-like trade. Designing and building databases and software is similar to carpentry. Anyone can nail together some boards to hold boxes and tools in the basement. Becoming a master carpenter requires years of dedication. GIS/IT skills are like a ladder.



I've noticed a tendency among GIS students, for example, to want to do analysis work or programming work but not graphic editing or database entry work. This ladder needs to be climbed step-by-step, each person at his own speed. It is dangerous to take on a 'high-ladder' task without understanding the issues and skills from the bottom part of the ladder. The tendency for GIS students (and IT managers?) ((and myself)) to run up this ladder too fast is a major contributor to false expectations and failed GIS/IT projects.

However, the idea of 'failure' is a bit fuzzy. I'd be willing to bet that many GIS/IT, projects which are viewed as 'failures', are not really. Rather, they are probably under-funded or crammed into an unrealistic time frame. I've noticed a tendency by everyone, even myself, to consistently underestimate the amount of work necessary to do high-quality GIS/IT work. It's time-consuming to design a GIS database and data dictionary properly. It's time-consuming to talk to all the people involved in a workflow and map the entire thing out, looking for ways to computerize it. It's time-consuming to capture vector features from high-resolution imagery. It's time-consuming to edit and assure good topological relationships between vector features in a database and to assure good vertical integration between various GIS layers. It's time-consuming to design software that is easy to use to input data into the databases and also report it back out. It's time-consuming to do quality control on a newly created or edited GIS

layer. It's time-consuming to train people to use GIS/IT software on their daily work flows, rather than paper. Time-consuming tasks are expensive.

Come to think of it, writing about this darn topic is time-consuming, so is reading it. Let's poke the frog across the next set of tracks.... user customization.

The Customization Conundrum

Local government leaders and politicians want to use computers to answer complex questions about the world. Employees want simple, cheap, push-button software. Everyone is a software client. Clients all want software that is cheap, easy to learn, and simple to use. When a client talks about what they want to do, our role as technology experts (or GIS/IT consultants) is to listen to what the client wants, hear what they say, then translate it into a plan.

At this point, we have a basic clash. Being able to answer complex questions about the world requires complex databases. Complex databases are, well, complex. The clients also want simple GUIs so they don't have to spend a lot of money training folks on new and difficult software. The clients don't want to spend years doing complex data entry, reformatting past records. They don't want to pay for sending clerks to months of software training.

Folks managing information technology have to understand about a critical trade-off involving cost, quality and customization. Once you start down the road of heavily customizing menus and workflows to make daily work easier, then every vendor upgrade in the future will be dangerous. It's hard to deal with all the software, and hardware upgrades and changes that happen underneath all the pieces of customized software.

People want easy one-button tools to do daily tasks. In local government, many workflows now depend on highly customized software tools. When daily workflows depend on certain software's then vendor upgrades to the underlying software could cause problems for the customizations. Because other pieces of the computer workflows are connected, a small 'upgrade' by one software vendor could have a ripple effect that causes another piece of the workflow to function in a different way.

To make simple software workflows that update complex databases in real time, across various departments, you need customized menus and databases. Customization is expensive. Once you get it running; then the operating system changes; then database software changes; then GIS and other vendor software changes. Out-of-the-box software doesn't integrate well. How do you keep the workflows and the software that people use in their daily jobs running in a stable way? It's like trying to thread a needle on a waterbed with the whole computer family of vendors jumping up and down around you, doing 'upgrades'.



On the other hand, simply using out-of-the-box software for everything has its disadvantages too. It takes time and effort to understand out-of-the-box GIS software for editing work, for GIS analysis and for long-term layer management. Some of us have made it our career to understand GIS software, programming languages, and database administration. Clerks doing data entry can't be expected to have that level of personal involvement in their jobs. Clients and casual users understandably want the software to be simple, just a few button clicks. There is a big trade off. Either the software users have to be highly trained to use out-of-the-box software or else you have to heavily customize the software to make it simple for untrained users.

Actually, either way is expensive. Moving a workflow from paper to computer is expensive. In the long-run, GIS/IT is worth the cost. It's just easy to underestimate the time and cost to implement computerized workflows. Omitting the location of database records is a popular way in the IT field to cut costs. However, down the road, I suspect that many traditionally non-GIS databases such as financial, retail, insurance, federal and hospital databases will wish that their database had been designed initially using real locations integrated into the design rather than using addresses.

Go little froggy, jump away from the heavy customization train, then hurry across in front of the next train, the lazy-always-want-everything-to-work-right-out-of-the-box train. Either train could easily squish an unwary frog.

This train yard is busy. There are many trains of thought zipping around. The poor little frog just wants to write about 'Migration to ArcMap Software' but the talk itself is becoming its own migration saga. The frog stops to consider for a moment: 'Maybe this is why the ArcInfo to ArcGIS conversion process is called a 'migration' rather than a 'software upgrade'. Hop, Hop.

A Single Workflow Migration Example

A long time ago, our city, in acknowledgement of its deteriorating infrastructure and a host of other problems established a huge effort to get a handle on things and make recommendations for improvement. A big study was done by the Schmale Commission. A list of recommendations was generated, including the suggestion to create an electronic/automated mapping system. Our GIS consortium was created based on the calculated savings of having one electronic map, always up to date, accessible by all, rather than separate paper maps stored in 137 locations around the city (in various states of repair, accuracy, scale).

Back in the mid-eighties, various departments like Water Works and Metropolitan Sewer District were drawing their maps on mylar, making copies and sending their copied mylars across town in order to share map data. Back then most of the office work was done with paper workflows. These notions about multi-departmental database integration hadn't been thought of back then. Why? Well, as you know, back then, computers were more expensive and slower. So were databases and software.

As an example of a workflow affected by software changes, I'm going to share a few details from a long project concerning computer generation of taxmaps for one of our important members: the County Engineer's office. Property ownership boundaries for our county parcel taxmaps are drawn on paper maps and changes are still done by hand drafting.

Original Paper Taxmaps, Part of Original Paper Workflow.



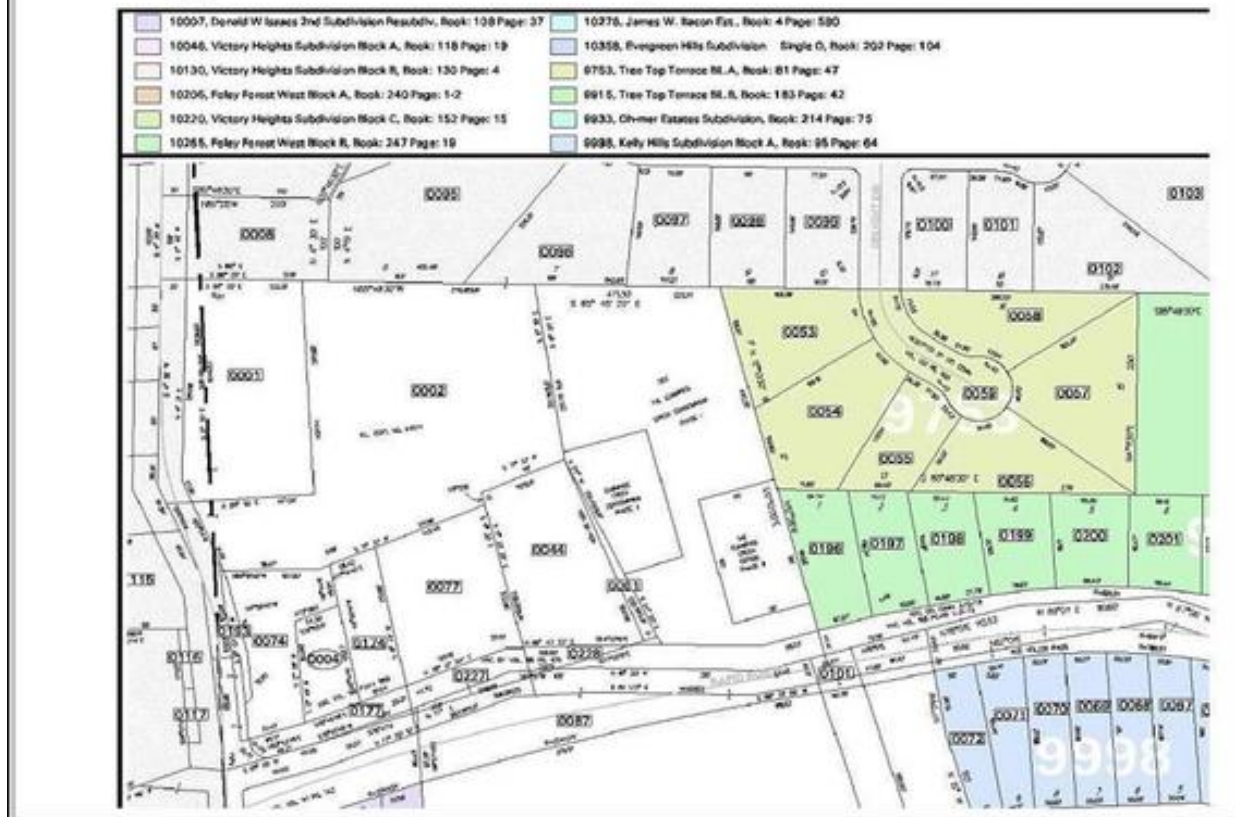
Back in the mid-eighties, our county engineer's office got into the GIS program mainly for the purpose of computerizing the drafting of the county taxmaps. Back then people viewed GIS as simply computerized mapping. About ten years ago, our county implemented a computer database for parcel mapping. Since then, they've had to maintain both the computer-based map and the paper-map system. They could have printed basic taxmaps long ago but there is a strong client desire to have the computer taxmap format mimic that of the hand-drafted maps. To solve that problem, CAGIS wrote a custom printing program in 1999 so they could plot taxmaps using GIS software and databases, which look similar to the hand drafted maps.

A Computer Printed Taxmap with Parcel Lines and Subdivisions



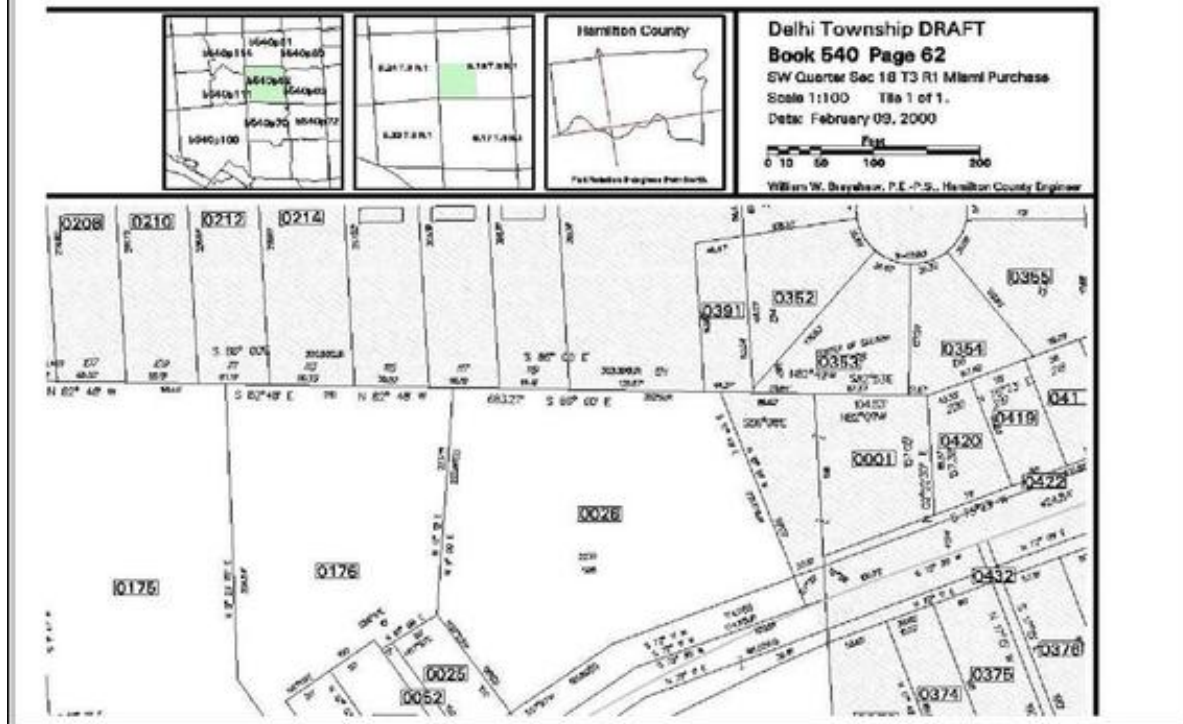
Even for simple computer mapping, desktop computers in the late 80's and early 90's simply couldn't handle the software or the databases. The original GIS data and software was on a VMS system. Hardware and software change has been amazing over the last 15 years. Early in the formation of the enterprise GIS we switched vendors and adopted ESRI software. This involved moving from VMS to Unix machines. The story about reformatting the GIS databases into a Unix database called ArcSTORM from a the earlier vendor's software is an ugly story. It was a long hard effort, accomplished by some folks that I consider to be among the top few percent of GIS professionals in the country.

TaxMap Detail Showing Shaded Subdivision Boundaries



Eventually, the folks at CAGIS got an Unix ArcSTORM server working smoothly to serve the needs of the Engineer's office (and other departments). Then when the UNIX workstations were 7 years old, and starting to wear out, we had to decide to buy new UNIX machines or try to shift to PCs running Windows NT4.0. At that time the leading software vendor for geographic databases (ESRI) didn't provide an ArcSTORM database that runs well on NT4.0. But new UNIX workstations would have been very expensive.

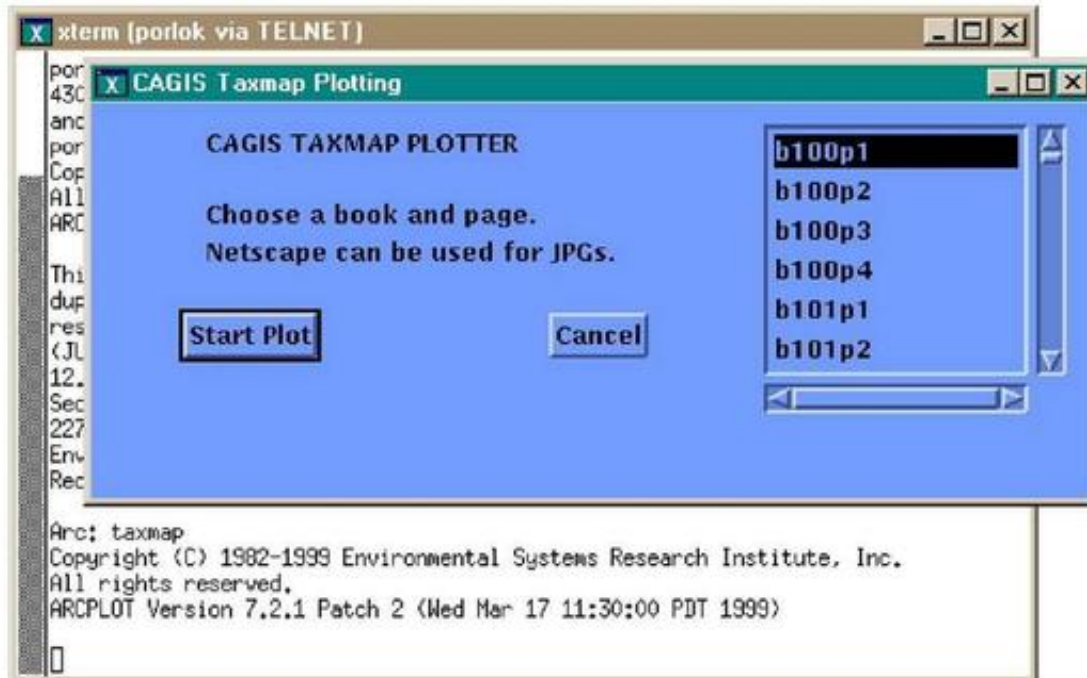
TaxMap Detail Showing Parcel Numbers and Public Land Survey



A main consideration was that ESRI, over the last 4 years, has shifted their software from Unix to the Windows platform. Since the vendor was moving to the Windows platform we decided that it would be unwise to try to keep our graphic editing workstations on Unix. Well, editing transactions between a Unix ArcSTORM server and NT clients is an unsupported configuration by the vendor.

You may ask: "How is it possible that the vendor could switch platforms and not support clients that try to switch platforms with them?" Well the vendors' phone representative has the fun job. He gets to say to his client: "That software configuration is not supported, sir. We cannot support systems that have been customized by the client."

Original Menu for UNIX-Based Taxmap Printer, Customized in AML



In this case, we were able to find a good solution. We designed an additional custom software component to shuttle these edit transactions back and forth between the editors windows machines and the Unix ArcSTORM database. In addition, we had to modify hundreds of customized AML Unix macros so that they would run on Windows. To do this by hand would have taken months, so we wrote a Visual Basic program to help convert text strings in the Unix AML's into an NT compatible format. Several valuable things were accomplished here.



Windows workstations are cheaper than Unix workstations. Therefore, we saved significant money by handling this project the way we have. And, we saved the immediate expense of having to rush to redesign or modify our database or editing tools. Departments have not had to retrain their graphic editors on new software. We were able to preserve the old familiar editing menus and make them function in the Windows environment.

Since implemented, this system has been running smoothly for 3 years. We have less support calls from the editors about database checkout/checkin problems than we had three years ago on the old Unix workstations.

There was one other more important benefit. Years ago when we moved from VMS to Unix, I mentioned that we adopted ArcStorm as a main database. At that time, ArcSTORM was the latest and greatest software product available from the vendor. Well, often the first release of any new software contains many bugs and flaws. Over time, the vendor upgrades and fixes the bugs as they are discovered. Because we adopted that technology when it was new, it was difficult to make it work initially. Now that the bugs are worked out, it has been running smoothly for years.

```

porlok.r2w - Reflection for UNIX and Dtpfal
File Edit Connection Setup Macro Window Help
SchemaEdit: lt
Listing transactions for Database: property (3 found).
33913 CHECKOUT_COMPLETE herbert
34112 CHECKOUT_COMPLETE herbert
34114 CHECKOUT_COMPLETE vdixit
SchemaEdit: describetrans 34112
Description of transaction 34112.
Owned by: herbert
Library: parcels
Started: Mon Aug 11 15:33:44 2003
Status: CHECKOUT_COMPLETE
Current Phase: 0
Data Sources
Type: Layer
Name: parcpoly
Feature Class: region.parcel
Private Coverage: /arcstorm/hdasfk/herbert/parc-570100
Status: CHECKOUT_COMPLETE
150, 13 VT400-7 -- porlok via TELNET 01:52:45 Hold

```

But the same thing is happening now. Several years ago, the vendor released a totally new database format called ArcSDE, which is a spatial database that sits on top of an Oracle database. ArcSDE geo-databases were a brand new product, just released by the vendor, designed to be the solution to all our woes. Our older database formats and editing menus are clunky because they are based on 15 year old technology, but they worked, and we decided the best thing for our client, the County Engineers, was to wait until the new software stabilizes.

The County Engineer's office still uses their original and familiar software menus, to maintain graphic changes to the parcel and land ownership boundaries. They still use the same ArcTools menus to access the AML macros that we developed to print out taxmaps. Technologically they are able to print computer taxmaps, but they do not have the extra staff or manpower to do the extra work. It takes time and effort to clean up line-work, annotation, and labels so that the maps look as good as the older hand-drafted work.

The next steps in this long project will be to work with the departmental GIS experts to redesign the back-end databases and move them from ArcSTORM format to ArcSDE format. Then we will have to design a few software tools to help with the graphic edits

using ArcMap rather than ArcTools. Hopefully, ArcMap software will prove to be more efficient and will require less staff time to print nicer taxmaps.

The end of the train yard is in sight. There are many other tracks to cross. There are many other consortium member departments besides the Engineer's Office, each with its own unique 'migration' issues that could be described in detail. One example is enough. The frog is getting tired, maybe he needs some coffee. Let's spring into action and cross this vast expanse of empty tracks while no trains are coming...

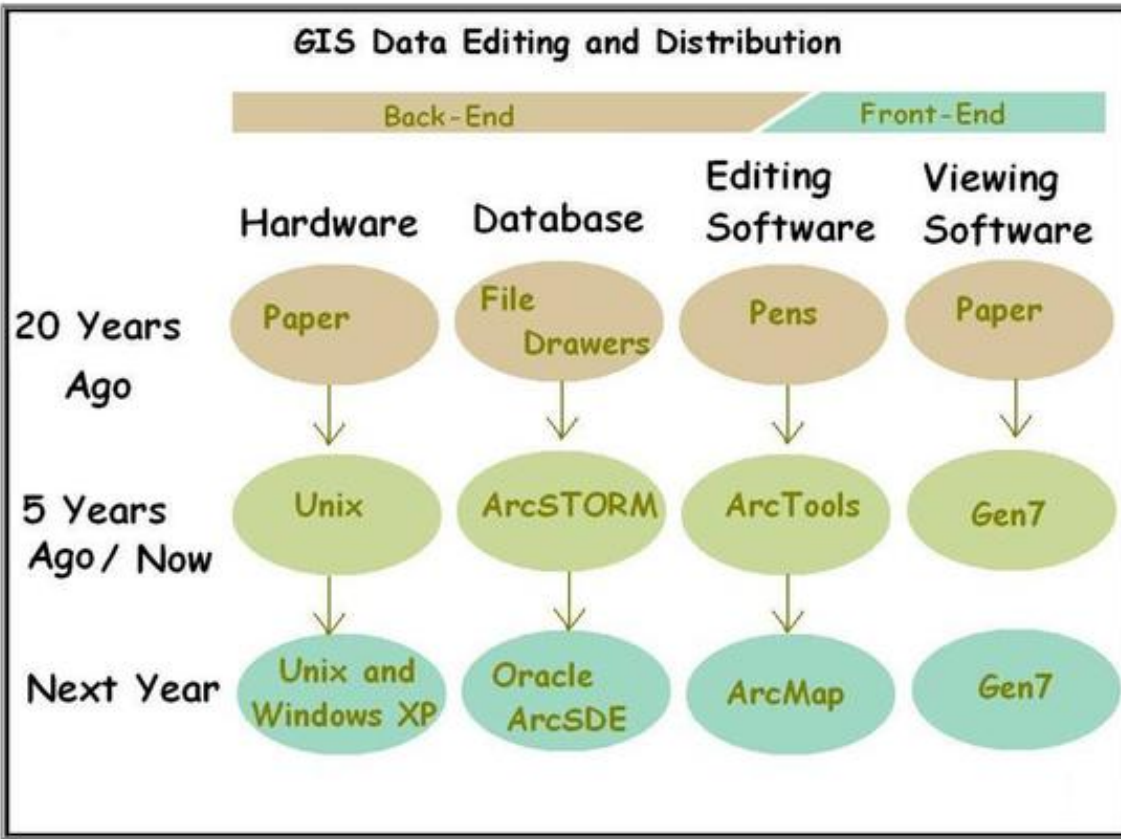
CAGIS Plans for Upgrading to ArcGIS

Once a paper workflow has been moved into customized computer software, upgrades need to be approached with caution. Upgrading from Microsoft Word 2002 to version 2003 is a simple installation process, most Word documents don't need to be modified. All the user needs to do is click 'install', click 'next' a couple times, then click 'okay.' The upgrade is done.

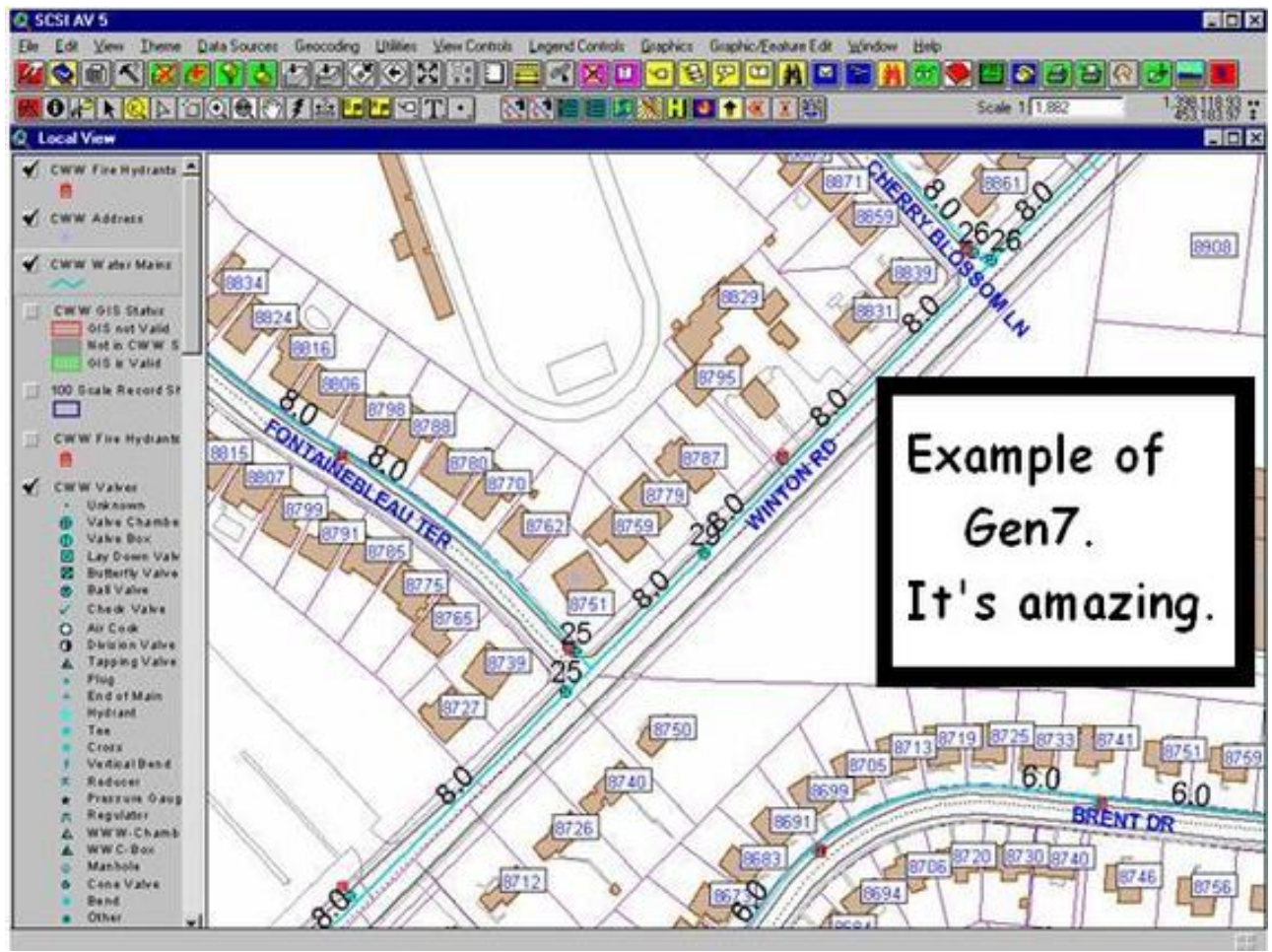
Likewise, it is technically simple, to upgrade Microsoft Visual Studio 6.0 to Visual Studio 7.0. It's just an installation process. But after the 'upgrade', (oh my gosh), it's probably easier to rewrite many VB6 programs from scratch than it would be to convert them into VB7 projects.

Installing ArcMap and ArcSDE is not hard. However, GIS databases and customizations are time-consuming to 'migrate' or 'upgrade'. Not only are there conversion issues from Avenue and AML customizations, but also geodatabases require planning related to layer design and topology issues.

Because CAGIS is an older GIS organization, ArcMap and ArcSDE should be approached with caution. The disadvantage to having GIS on front counters, embedded in workflows involving customer service, is that any software changes or 'upgrades' have the potential to disrupt front-counter service. Conversion to this new technology will be a gigantic commitment. CAGIS already has a highly customized front-end software named 'GEN7.' There is no question that the system we currently have of distributing shapefiles and Gen7 ArcView projects works better for our organization than 'out-of-the-box' ArcMap with ArcSDE.



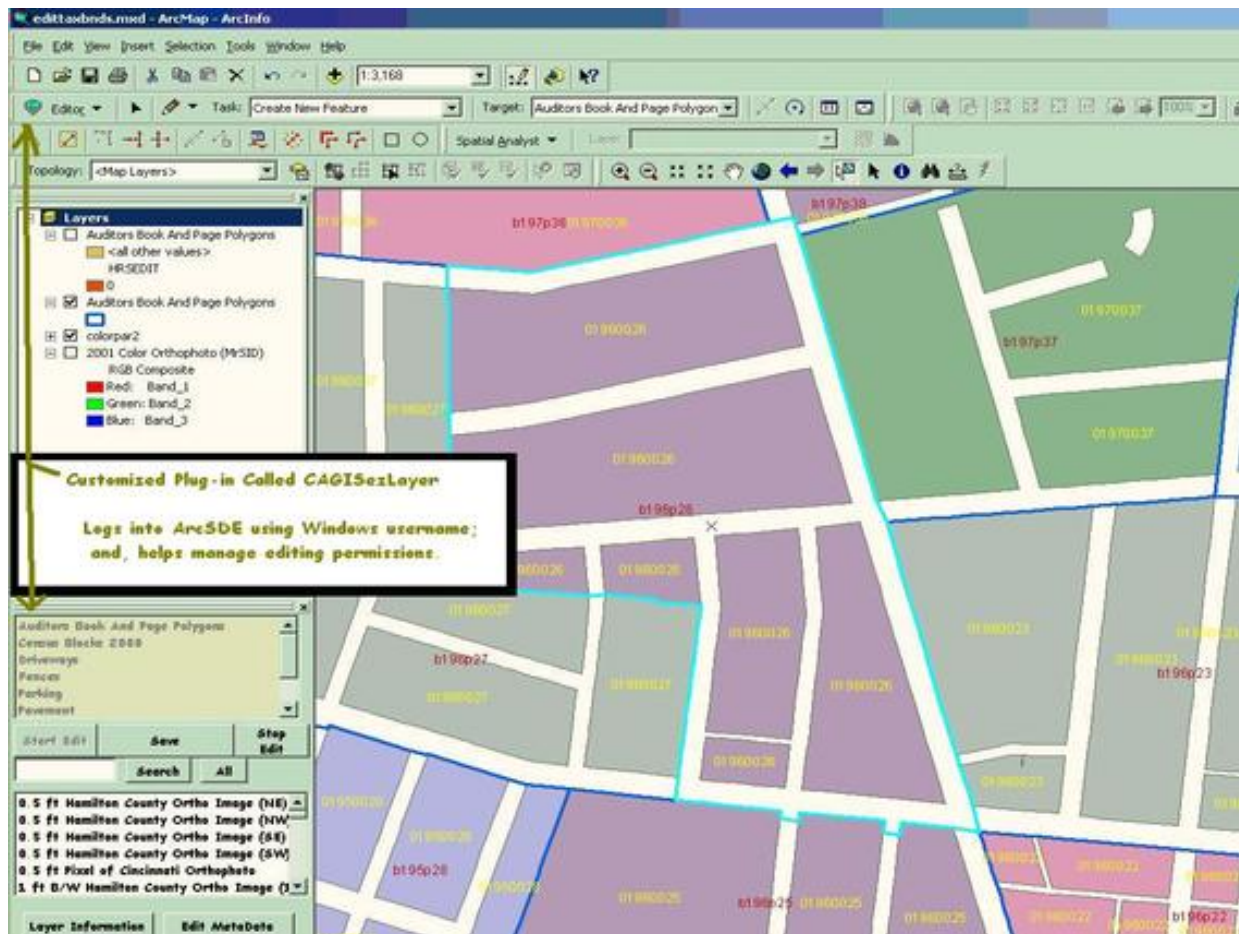
In the same breath, I can say that I like ArcMap and ArcSDE. Version 8.0 was useless; version 8.1 was superior to ArcView for cartography. Version 8.2 was more stable. Version 8.3 is the first version that has workable topology rules, and it's great for editing, even though we do not trust it enough to move it into full production yet.



ArcMap will not and should not immediately replace our ArcView3.3 application, Gen7. Our plan is to slowly replace our (back-end) ArcTools editing workflows with ArcMap editing.

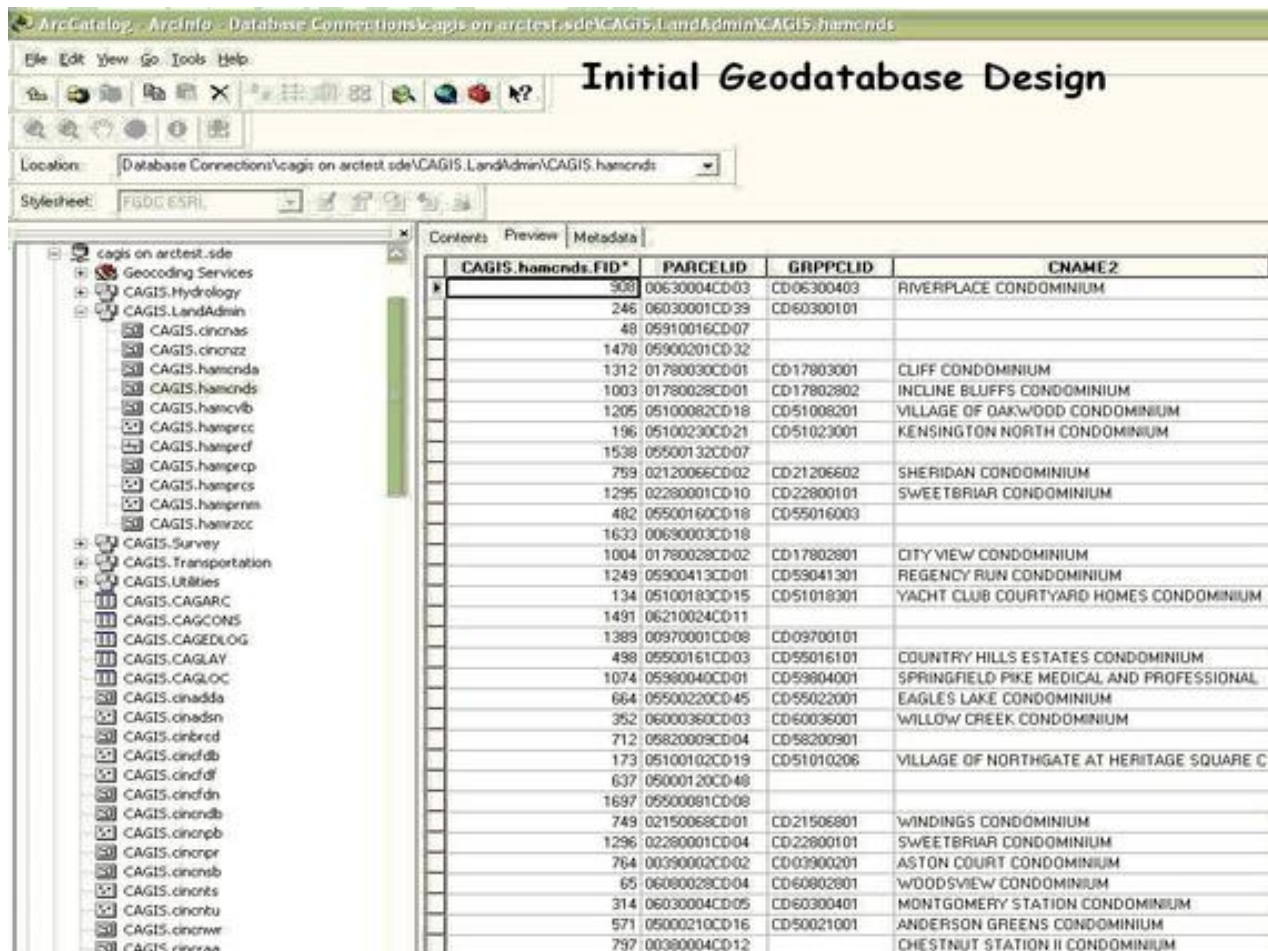


The transition, realistically, will take several years. We have been working with ArcMap, evaluating it and testing it since the Beta version. We have had a small ArcSDE test server with an 18gigabyte Oracle database of all or GIS layers running for quite a while. We've been successfully using it for editing projects in with ArcMap. There are still some bugs in the ArcMap software. Also some customized design and a lot of testing needs to be done. But, for the most part, ArcMap has been working great so far. I consider it to be a big advance over ArcTools.



In the beginning of testing ArcSDE we realized that we had so many GIS layers that they were becoming confusing to manage. We developed an inventory of shapefiles to help sort out old layers and unnecessary shapefiles. We made dbase format database containing layer names and basic metadata on each layer. On top of that DBF was a simple GUI for folks to type in some basic layer information. The basic layer info was later loaded programmatically into the SDE metadata format. That worked well for what it was; a way to programmatically load shapefiles into SDE in a systematic way.

One difficult aspect of implementing ArcSDE and ArcMap has been making it easy to set layer permissions. After our inventory of GIS layers we found about 1000 shapefiles on the main server. About 500 of these fall into the category of temporary 'working' shapefiles, generated by software, behind the scenes. Another 100 layers are older versions. This leaves about 400 valid and useful layers that we'll load into ArcSDE. Naturally a smaller number of layers, like 100, are used commonly. Although we have about 1000 ArcView licenses and people using the software throughout the organization, we only have 26 ArcInfo users who edit graphic data. Our ArcSDE database will contain approximately 15 million vector features. We loaded them programmatically onto a test server. Soon we'll load them onto a larger, production server and then upgrade the test ArcSDE database to version 9.0.



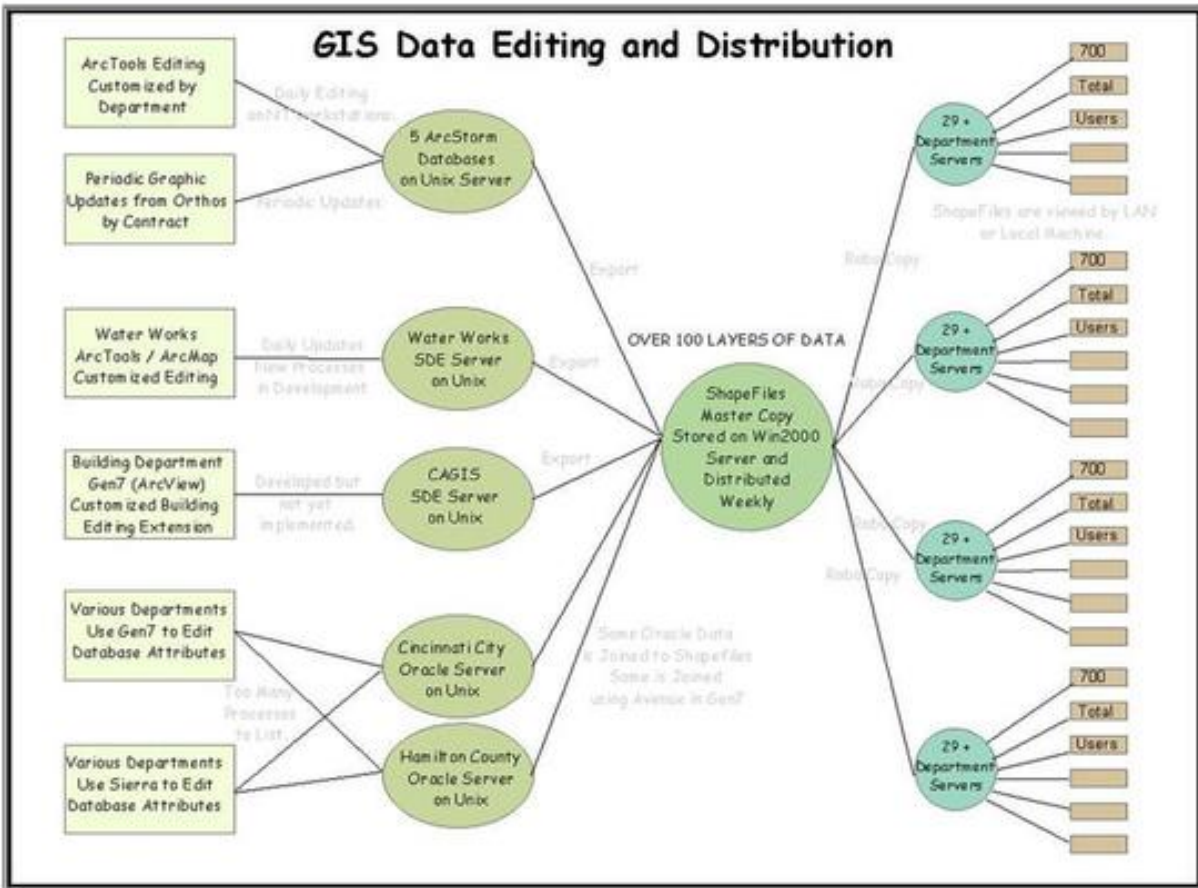
It's not a trivial matter to set 'read', 'write' and 'update' permissions on 400 layers for each of 26 editor accounts. It's wiser to do it programmatically rather than manually.

The ArcMap customization that we have envisioned will use an SDE standalone table to help manage these 400 various GIS layers. This table will have some basic layer information relating SDE layers to shapefiles, for import and export. It will also contain user login names to allow editing or viewing permissions for each layer. A single enterprise-wide .dll plug-in will be used in ArcMap to manage multi-user editing and layer permissions in the same way for all departments. So there would be a central customized layer information table in SDE on Oracle. This 'layer table' will contain only very basic layer information which will be copied into ESRI's metadata format. More detailed layer metadata will be entered using ArcCatalog's standard tools.

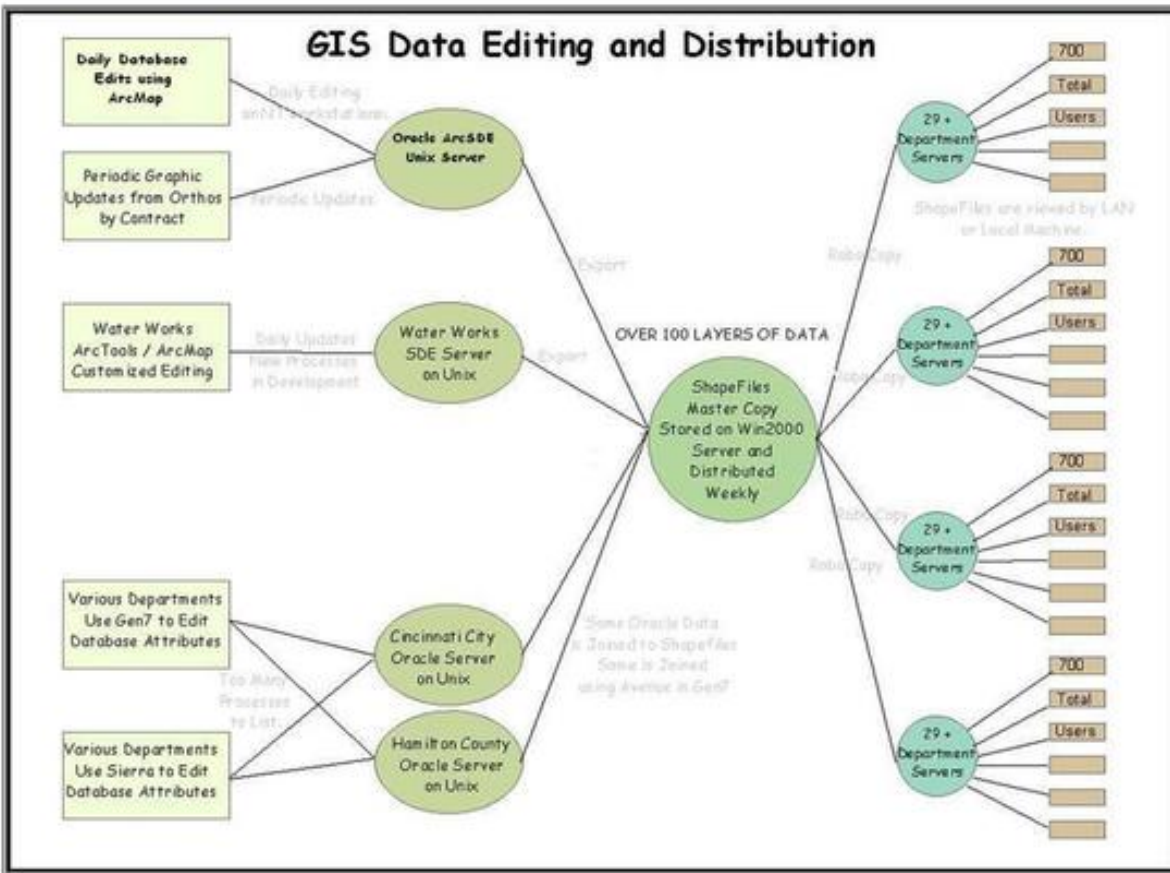
Our customized ArcMap plug-in will log in each user to ArcSDE based on the windows login name and set viewing and editing permissions accordingly. It will also tag any edited features with the username of the editor and date that the feature was edited.

It will be possible to load the bulk of our shapefiles into ArcSDE as simple feature classes. However some departmental data may need considerable re-design to utilize features of the geodatabase format. Ideally, CAGIS staff will work with the GIS expert

of each department to redesign layers to match as closely as possible with ESRI's Data Models. It is critical for folks using the data to be involved with the 'migration' issues and to be involved in designing and testing the new editing tools and databases.



Even though ArcMap is much better than ArcTools, it's also radically different. The main difficulty so far has been to get people excited about being involved with data migration issues, database design, testing and debugging. Departmental users need to be involved. After the back-end databases are redesigned and tested then we can try to switch this one piece of software inside our Data Distribution system.



The current customized software configuration at CAGIS works, for now, but we are planning for the future and working to implement ArcMap and ArcSDE GIS editing.

The frog is watching this train. The 'upgrade train' is moving slowly. Rather than, hopping, Gilbert-the-frog is walking across these tracks. Frogs can walk, but its awkward looking. Just one more giant leap of reasoning and this story will land on exactly the opposite side of the territory that it intended to cross at the beginning. It will land at the end. Gilbert just needs to avoid being pulverized by one last train: the migration-conclusion train.

Conclusion (back to the big picture)

Migration from older GIS software to newer ArcGIS software is sure a tricky issue for our organization because we have so many software customizations from different vendors working together. Whether or not to eventually upgrade from ArcView3x and ArcTools to ArcMap 9x is not really an issue. Rather, how to make the migration will take some careful planning. The much larger issue that completely overshadows discussions about software versions is the larger migration from paper to computers and the challenge of integrating computer database workflows across organizations.

Computer technology to manage local government data and share it efficiently between departments exists. Software and hardware are not the main barriers to re-engineering and computerizing local government paper workflows. Money is often touted as the main barrier. Gilbert-the-frog is skeptical; money is not the main barrier. The main barrier is will and understanding.

Often people fear the idea of moving work from paper to computers. Change involves both risk and extra expenses. Any older paper workflow and its newly designed computerized replacement simply must be run in parallel until everyone is comfortable with the quality of the computerized workflow. It costs extra money to design, test, and implement a computerized workflow while continuing to run the paper workflow. Maybe the computerization will take 1 year, or 5 years, or 10 years, or 100 years. The speed and quality depends mainly upon the commitment of the leadership in any department or government agency.

Corporations are more profit-driven than are local governments. Yet most large corporations invest heavily in information technology, because computers are more efficient than paper for managing transactional data. Governments have more pressure to decrease spending and avoid risky technology projects. Yet governments also have a long term responsibility to manage data and do work in a cost-effective way. Long-term efficiency requires that computers and databases be shoe-horned into government workflows with whatever budget is available.

The will to change from paper to computer is a function of fear. Some of the fear can be alleviated by zooming out on one's perspective to a longer timeline, and a smaller scale. Let's zoom way out.

A long time ago, human culture shifted from oral histories to paper libraries. Oral record keeping worked great for small tribes. However, larger, modern nation-states now require paper to track legal transactions related to land ownership and public infrastructure. Sure, computers are not necessary for many organizations to make decisions. In the same vein, tribal herdsman and farmers don't 'need' to be literate or educated to survive. Just because education is not strictly necessary doesn't mean that education itself isn't valuable. Just because paper records still work today, doesn't mean necessarily that organizations should ignore the value of computerizing their recordkeeping. Wise people, a long time ago, decided it was worth the expense to record public information onto papers and store those papers into indexed drawers. It was a more expensive system initially than the passing the information from mouth to mouth.

Nowadays, like it or not, humans culture is transitioning from paper record keeping to computer record keeping. If the Baby-Boom generation cannot figure out how to cost-effectively make the transition then it will be left to the X-generation to figure out. If the X-generation cannot accomplish it then the task will be left to the Y-generation. It's possible to defer the responsibility from parent to child, generation after generation.

Eventually though, local government officials everywhere will need to take pride in their role as the managers of public data, and realize the benefits of databases.

Currently, only about 10% of the world's population (the wealthiest) has access to and knowledge of computers. As a wild guess, probably over 90% of the world's local government work is recorded and transacted using paper workflows. However, who would deny that the GIS/IT train is gaining momentum?

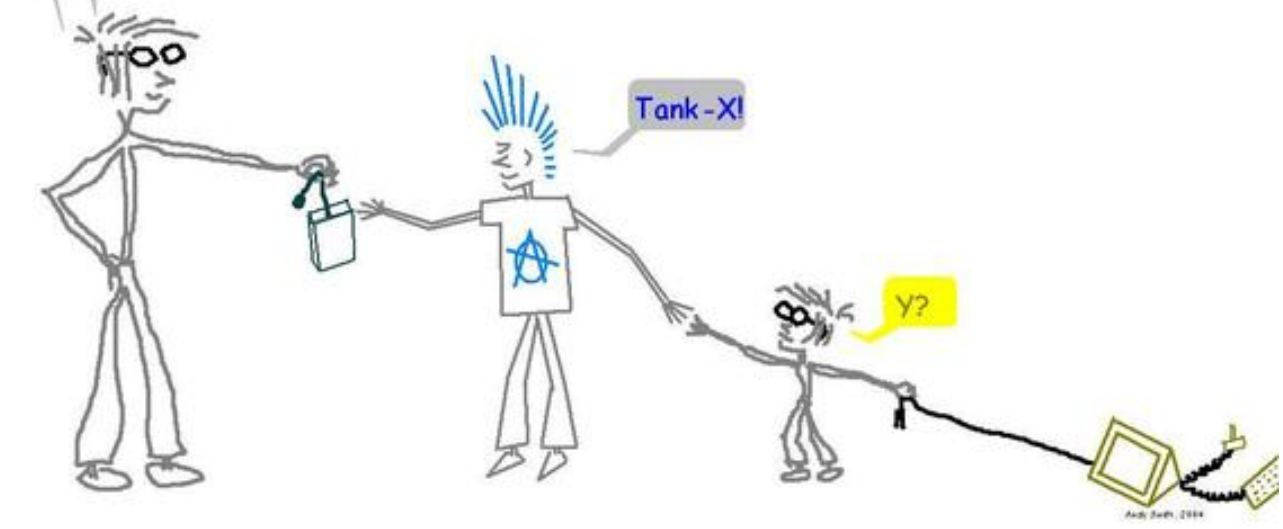
Rather than fearing the technology of the unknown future, a wiser understanding of the role of information technology professionals is to view them as the craftsman of the first database libraries. Many GIS/IT experts realize the long-term importance of the databases that they are designing. Hundreds of years from now, if our species can hold itself together that long, current government, financial and scientific databases may be analogous to the early libraries and the Dead Sea Scrolls.

This point has the disadvantage of sounding somewhat philosophically melodramatic, naive and idealistic. I am not trying to argue that financial issues be ignored for GIS/IT projects. Money is a major factor in every aspect of business and government. Money cannot be ignored, no matter how far we zoom our perspective out and philosophize about the evolution of human communication and technology.

However, this conclusion also has the advantage of being correct from the higher (long-term) perspective. Human culture is transitioning slowly from oral cultures to written cultures to computer culture. Understanding, wisdom, knowledge, science, literature, philosophy, book libraries and now computer databases of information are gifts that parents pass on to their children. I realize that this way of looking at databases is not a common view. Face it, and help build it.

Errr... Sorry... about... all the Enrons, and all the left-over nukes, and genetically modified organisms, and the health insurance industry, and social security, and all the endangered species, and the saving-and-loan scandal, and the world political situation, and the ozone holes...

But here... this is my hard drive containing all the scientific GIS data I was able to download, and my chronological database of philosophical sentences recreated from the writings of western philosophers from Plato to Wittgenstein... You can take it from there... Good luck!



From a practical standpoint, the reason to migrate from paper to computers is economical not philosophical. In the long-run, it's hard to imagine how any paper record keeping system could possibly economically out-compete a well-designed, well-integrated interdepartmental GIS database in terms of cost efficiency. Long-term cost efficiency is the main reason to migrate from paper to computer. And, it is the reason to migrate from old software to newer, better software.

Now Gilbert-the-Frog has attempted to cross this final train of thought. We have to ask, 'Why, did the frog attempt to cross the train yard?' The answer, in Gilbert's case, was that he was feeling a little down, tired, and he needed coffee, maybe some excitement. There is a coffee shop on the other side, you see.

Did Gilbert-the-Frog survive this last long leap of logic across the conclusion tracks? Maybe he is sitting safely on the other side smugly flicking his tongue into a triple-shot latte, flicking his challenging tongue at the world. Or maybe, Gilbert is lying squashed on the tracks of his own reasoning. After all, it's entirely possible that computers actually aren't all that incredibly cost-effective compared to paper. Maybe computers and databases are not really even necessary or useful at all. Maybe they are not even good. Maybe they are evil. It's certainly possible that statements about the evolution of

human communication are entirely irrelevant. In fact, Gilbert would be willing to admit that this entire article is complete rubbish, and that we are all, basically, wasting our time, accomplishing nothing useful, except hopefully a few laughs on a topic that bores most people.

Friendly thinking frogs wishing to discuss issues in this article can find Gilbert online at: koko_wadoko@hotmail.com. Naturally, Gilbert is available for consulting, presentations and parties.



Author

Andrew Swift is a programmer for CAGIS, the Cincinnati Area GIS. He received a Masters in Environmental Science from Indiana University in 1996, and a BA in Philosophy in 1991.

Acknowledgements (errr..... disclaimer?!?!)

Blame for my writing style shall be passed to no one, it's my fault. However, great respect and thanks for many teachings in the field of GIS should be credited to: Rich Allen, Tim Eisler, Jay Erndt, Raj Chundur, John Coulter, Kass Green, Shaoli Huang, Dave Railey, JC Randolph, Tim Turinsky, Mike Sullivan and Barb Quinn.

Acronyms

Many acronyms are used but not explained. Sorry. Simply drop any unknown acronym into an internet search engine. If you do not have an internet search engine or a computer then you are very unlikely to care about the meaning of the acronyms, or this entire article.

References

CAGIS INTEGRATED GIS AND PERMIT MANAGEMENT SYSTEM

<http://gis.esri.com/library/userconf/proc97/proc97/to450/pap444/p444.htm>

GIS Involving the Community: The Hamilton County Environmental Priorities Project,

<http://gis.esri.com/library/userconf/proc98/PROCEED/TO550/PAP510/P510.HTM>

GIS, Philosophy and Geography, Andy Swift

Internet Published, 2004, koko_wadoko@hotmail.com

Using GIS to Successfully Create Organizational Change in Cincinnati

<http://gis.esri.com/library/userconf/proc01/professional/papers/pap136/p136.htm>

The bottom line is; for busy bottom-line dudes; avoid long readings.