GEOSPECIALLING

SUNDAY, AUGUST 03, 2008

« Passing a Large Mapguide Selection XML t... | Main | Autodesk University 2008! »

@MApp â€" a Developers take on an amazing RF Design and Drafting Application

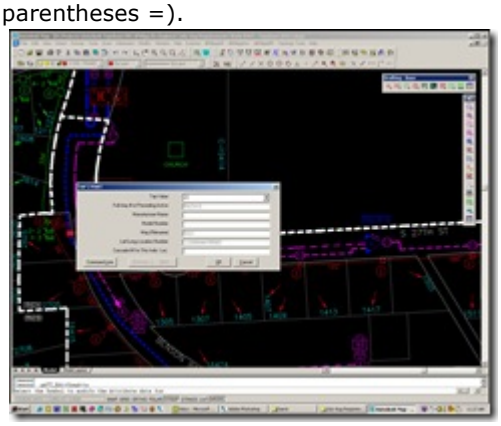
My friend and colleague **Evan Wager** recently wrote an article on the **history of @MApp** (sometimes known as ATMapp or ATTMApp). A strange twist of events also resulted in me working very briefly with the most "recent" build of the product. It had been many years since I really saw the application running - so I had a bit of a fresh eye. This made me a little bit nostalgic, and also quite proud. We had really made some amazing software when you got right down to it. Unfortunately, in "internet years" it has sadly become a dinosaur.

Long ago (it seems like a lifetime) I worked for a company called Kanotech. One of my first real development projects of scale was on the **@MApp product**. @MApp was a powerful **AutoCAD Map** extension built for AT&T/Comcast cable. My first task was to head to Seattle and learn how field technicians would walk the existing RF Plant (fancy word for cabling and RF equipment) and map the existing assets, on paper. The first task was devise a way to run @MApp on one of the early rugged **Hammerhead pen computers**. I walked the streets of the greater Seattle area with some of the field technicians and got my first introduction into the world of cable TV networks. At the end of this trip I had a collection of notes and enough information to design and implement the pen based version of @MApp. This was the beginning of my life with @MApp.

With the bigger picture in mind, there were many underlying goals for @MApp, one of the key needs was standards enforcement (and creation - Evan created the first solid standard which, though very much evolved is still in use today). Like many organizations back in the day (and likely even today) the mapping data being generated was garbage. Each division, each consultant, sometimes each drafter had their own set of "Standards". As Evan put it, the mapping data was "Only good for printing" - and even that was a stretch in some of the sample data we saw. Our top priority was to create a set of tools which facilitated drafting standards compliant drawings - as well as a method to test each drawing to ensure the drawing was truly compliant.

Under the hood, @MApp was all database driven. Layers, cable types and properties, even symbols and block attributes could be configured in the database. Generic lisp calls were defined to allow simple wrapper functions to be created to add new entities to the application. The application had

been created using an extensive amount of **AutoLisp**, VB6, **C++/ObjectARX**, Python, and SQL Server/Access databases. Let me tell you, I still have nightmares about parentheses =).



The drafting tools, in retrospect were good. Damn good. Obvious elements were there, select a cable, or equipment type using standard AutoCAD toolbar buttons. @MApp went a step further. It would create/set the layer, set the desired object snaps, draft the cable. Network connected equipment would snap to the cable, rotate, trim and physically connect itself to the model. In some circumstances, inserting a block/cable of a certain type also required an accompanying block. If required, the user would be prompted to insert these as well. In some cases - these blocks were placed automatically. Using @MApp - a user could draft very clean RF drawings without having to really know AutoCAD. Every piece of possible equipment was available from a comprehensive set of toolbars - and only one click away.

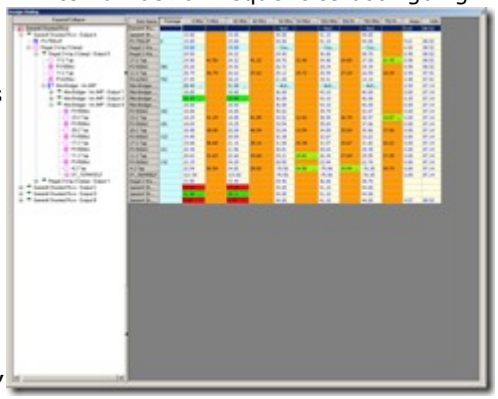
On insertion of blocks, the user was presented with a form powered by the attribute information stored in the databases. Some attributes were required, some were populated programmatically based on nearby objects or other conditions, and some were selected from lookups or manually entered by the user.

@MApp had a really cool connectivity process. This process would perform physical connectivity on the network, following rules defined for equipment inputs and outputs. In addition to this, rules were also defined to dictate what equipment could connect to other equipment. This connectivity, in my opinion was one of the most powerful components of @MApp. Connectivity information was stored on the entities. When a trace was done, we could follow the network and do all kinds of great reporting. I recall one situation where we knew that in situations where a specific combination of equipment was setup - it would cause service problems. Within a few hours, I had a batch process created that would process hundreds of drawings and spit out a report of nodes, and locations where this combination existed. The problems in the field were fixed even before customer complaints came in.

One of the key requirements for @MApp was a quality assurance process. We defined several "levels" of QA. Each level of QA required a password to execute. The first two levels of QA were for contractors doing mapping redraft, or drafting. These levels would let the consultants know that the required data had been entered and would "stamp" the drawing. The drawings were then submitted, and Comcast staff would run their password protected version of the QA routines - which would verify that indeed the contractor had done the required work. If not, the drawing was rejected and the contractor had to fix it. I can say with a great deal of certainty that Comcast, using @MApp is one of the few organizations with near perfect data. And the things they do with that data are incredible.

The final version of @MApp, @MappRF could have dominated the cable industry. Sadly, it was never completed. This release added RF design to the drafting. A library of equipment was created, as well as design parameters which were sub-selection of the equipment that let the RF designer know what equipment was available to be used. All of the equipment's operating parameters were captured. As the user drafted equipment, the actual equipment model would be selected on the attributes. This allowed us to increase the efficiency of populating attributes during the drafting phase all the while making the stored data even more accurate and complete.

One of the coolest bits of @MAppRF was what came to be known as the TRID. The TRID was a multi-threaded C++ control was created that contained a tree-grid hybrid. This form could be docked within AutoCAD or floating on a separate monitor. As the user drafted, the RF signal and power calculations were performed in real-time, without interfering with the drafting process. I recall being told, this project would be simple. RF design is just table math. And it is, but, mix in a multi-threaded C++ form running per entity calculations on a potentially infinite number of frequencies both going forward, and backwards - oh and sprinkle in a little bit of power draw calculations. Then tell me its simple. =)



If I could do it over again. Wow, what a difference the current technology would make. By far the single largest flaw @MApp has is its dependence on drawing based storage. Given the timeline, we didn't have a choice. **Oracle Spatial** was available, but immature. Going with Oracle at that time would have likely been more grief than it was worth. Seamless map access would be a must. Given the times we used a special piece of ObjectARX code called the SPE. This gave us spatial analysis abilities that rival even Oracle Spatial. One of the major downfalls here is that it required AutoCAD to run. Therefore - most of the calculations had to occur within AutoCAD itself. This became a problem as other applications could benefit from these operations - but couldn't take advantage of them without also being AutoCAD based.

Leveraging a multi-tiered architecture, a lot of the common functions could be moved to a more **SOA** like architecture. Now, though the TRID could still be a multi-threaded form in AutoCAD - the calculations could be done on the server side - thereby making the AutoCAD portion a presentation layer - instead of a business logic layer. Now the RF design calculations could be done in any user interface - instead of having to opening a drawing in AutoCAD.

And that my dear reader, concludes my epic tale on @MApp. Even this large novel like piece only scratched the surface of this project. It was my life for many years. I slept under my desk a number of times - but it was a great project. It was fun to think about it again and put together something to share with you. =)