

The IBM I - A Different Roadmap

Not long ago I was reading an article about a session Steve Will gave on how to make the IBM i "sexy". Those who know me know that that would immediately start me thinking about how I'd make the system more 'sexy'. This article is the result.

The 1st thing we'd have to do is start with the next evolution of the system to give us a free hand to make it as 'sexy' as we want. For lack of a better name I'll call the new system an AI500 (Advanced Integration 500). The brand will be easy to remember and easy to run searches on.

So what would this new system look like? That's going to be the first question potential buyers will ask. Here are some of the possible answers.

Right out of the box it will do a few things.

As you load your applications onto the AI500 it will create two versions of every screen - a default, modern UI version, and the typical green screen. Users will be able to pick whichever version they want to use the first time they sign onto the system. A flag will be saved to their user profile noting their choice so they'll get that version from then on.

If they choose the modern UI then they'll also get a feature no other system has - they'll be able to move all of the fields to wherever they want to see them on the screen. This will be possible because the system will save user-specific screens. Not only that, they'll also be able to hide any optional fields so they don't have to see them. If developers add new fields to the screens down the road then a box will pop up when a user hits a screen with the new elements and the user can drag and drop them wherever they want. Any deleted fields will automatically disappear.

This personalized system will be the first obvious sign that the AI500 is not your mother's IBM i or your father's AS/400.

The next feature will be an internal, Google-like search engine. The information behind the search engine will be loaded as applications are initially loaded onto the system and updated as old applications are changed and new ones are added. Users will be able to ask questions in a browser and see several links in the results. A question like "How do I reverse an A/P batch?" could show you a link that leads to that application and a link explaining how to do it (the existence of this second link would depend on how good the documentation, either user-created or Watson-created, is). Users won't have to remember where seldom-used applications are anymore. And if you're a new user this will shorten the learning curve and cut down the number of sticky notes you need to remind you what you've been taught..

So what else can we do to make this new system 'sexy'? How about adding a feature that will let you turn any file into whatever EDI format you need using a data dictionary the system also built when the database was loaded. You could pick any physical file that was created and tell the system to turn the records into any type of file (XML, X12, EDIFACT, etc...) and a specific type of transaction set within that type of file (an 857 for example). The system would give you the option of sending it right away or scheduling it. (Credentials would have been defined ahead of time by the system administrator so the file would go where it was supposed to go.) Incoming files could also be unpacked into physical files or tables regardless of what form they came in as.

Want another feature? How about using an email address attached to user profiles to automatically email reports to them when they run them. All the system has to do is create a temporary file under the covers that matches the output specs or printer file the report uses, convert the temporary file into an Excel, PDF or some other type of file (an Airtable file for example), and then email the results to the email address on the user's profile. The system could automatically insert a pop-up question asking if the user wants this to be done - it knows which programs are going to create reports.

And here's a feature many small shops with 2-3 programmers would benefit from. In those IT departments you're usually a programmer, system admin, security officer, etc... The AI500 would have a PIP window (picture-in-picture) like TV sets have that you could turn on. The window would be a little box in the upper right of your screen.

If you're a user and turn it on it will show you any messages that turned on the MSGW alert. That way you won't have to stop what you're doing to bring up the messages on the full screen. Granted, that's a minor issue.

If you're a programmer though it will not only show you your messages but every time someone got a program message it would feed those to your window - after they responded to it. That way you'd see every time someone just pressed enter to get past the message (effectively a 'C'), keyed in a 'C', an 'I' etc... If they just pressed enter the screen would say 'Pressed Enter' beside the response. If they keyed in a 'C' it would say 'Keyed C'. This could help you in several different ways, the most important being that you could stop the user before they got hip deep in the big muddly so you'd have time to look into the problem. It would also let you know there were problems they weren't reporting - in real-time. Since it would be a small window you could keep coding without it getting in the way. If there were several people having problems the windows would be stacked one on top of the other. You could either drag them apart to see them or fix them one at a time. You'd know you were down to the last window when it was showing you your own message screen.

Adding a 'SNDBRKMSG' option to the windows would also let you click on it and type something like 'Stop what you're doing right now until I come to your desk.' to keep people from getting themselves into even deeper trouble

Blocking unwanted messages like 'printer out of paper' from popping up could be handled easily enough by adding a 'block' check box beside each message so those message types would be blocked for your user profile going forwards.

Another few new features would make the system more attractive.

Here's the 1st one. As the data dictionary is being built when legacy applications and files were moved to the AI500 it would also build Entity Relationship Diagrams (ERD) under the covers. These diagrams could be used later on when programmers are writing new programs to add to the system so they can see how they'd fit in but that wouldn't be their main purpose. Files at most companies aren't completely normalized. When applications are run a quick check against these new ERD layouts would be done to make sure two fields in two files that were supposed to have identical values don't suddenly have different values (blocking data corruption errors), child files wouldn't suddenly be orphaned by the deletion of a parent, etc. etc. In other words, referential integrity and constraints would be handled at a completely different level for legacy programs. Alerts would be sent any time some kind of violation was detected. Most of the time these would be caught during the development or QA phase but the Integrity Monitor would run in production too to protect real data.

Here's the 2nd one. The AI500 would have validity checking programs added to every command and opcode that could add, change or delete data. These would act sort of like exit point program where developers could attach handlers to them if they always wanted certain things to happen under specific conditions.

And here's the 3rd one. IBM already has an ECS line (electronic customer support) for each machine - it just hasn't used its imagination to make the aggregate number of lines benefit everyone. That would change starting with the AI500. IBM and its midrange base would come to some kind of consensus on what kinds of things it would be helpful to brag about - things like how many intrusion attempts were blocked ecosystem-wide today, how many transactions were processed ecosystem-wide, etc... As long as the confidentiality of the companies using the machines was protected and there was an opt-out feature for companies that didn't trust IBM then every company who let the ECS poll their machine(s) on a daily basis could have a new command to click on that

would give them up-to-date bragging rights for whatever everyone agreed was worth bragging about. It would be hard for any competitor to even come close to this nearly real-time reporting. The best part of all is that it would also apply to the machines companies are incorrectly calling an AS/400 now.

There you have it - a starter template for a 'sexy' new system. All IBM has to do is take the road less traveled on its roadmap and create the AI500. If it does then some of the 100,000 midrange customers it's lost might even be eager to come back home to the platform.

** Note: As far as splitting the UI from the operating system goes, making both a GUI and green screen possible, maybe whoever designed the X-Window system is available and can give IBM some ideas on how to do it - https://en.wikipedia.org/wiki/X_Window_System. If they weren't available then maybe the original designers of the MOVEX ERP package which ran on the AS/400 (iSeries) and did this could be enticed out of retirement to give the architects some tips.