

HOST: Hi, and welcome to the “Did you say Mainframe?” podcast series. This is where we regularly interview IBM technical experts who can help you to understand important IBM mainframe hardware and software issues. I'm your host Nick Garrod.

Today we're going to talk to someone without a Hursley accent as I shall be chatting to someone from Australia about the CICS Configuration Manager that has recently come out.

Our guest today is Reuben Andrews from Fundi Software in Australia. Reuben is the Product Architect for CICS Configuration Manager . Hi Reuben, it's great to have you here and I shall try to refrain from saying g'day.

SME: Good day from Australia

HOST: Before we begin, I'd like to mention to our listeners that there is more information in [a white papers and a CICS Redbook.] I'll be giving out more information at the end of this podcast.

HOST Q1: So Reuben, why is predicting changes in cold start important, and how do you do it?

SME A1: The business growth means application change which leads to your cold start risk, and the risk is effected by the volume of changes you're making, you can minimise the risk by doing more frequent cold starts with smaller volumes of change. But the drive towards non-stop operation means for many sites, that they have to do less frequent cold starts which have higher volumes of change. The change might be changes in the CSD or the CPSM definitions themselves that we are trying to implement. But it could also be through changes to the run-time for the installation of temporary groups or setting set commands.

Now the paradigm of prevention, detection and recovery, we find that many sites are stuck in the recovery phase, so when something goes wrong at cold start, they are after the fact trying to fix it. If we could break that cycle, if we could predict what was going to happen, then we could avoid the problems in the first place. The business applications would run smoother, the IT department would receive its service standards and everybody would feel easy, and that's why we want to predict the changes at cold start.

HOST Q2: What are the conditions you actually want to predict ?

SME A2: Well there's three obvious conditions, you want to know “what's new?” that's going to go into your run-time. So these are things that you have in your CSD or your CMSP, but your run-time currently doesn't have them, the new things. The second thing is “what's different?” this is where you've changed something in the CSD of the CPSM environment but those changes have not yet occurred so you then your run-time is somehow different to what's going to happen after the cold start.

And the third obvious condition is the emission, the deletion of things. This is where something is in the run-time but it's not actually out there in your CSD and CPSM environment, so therefore it could go missing. So there are the obvious things, the new, the changed and the missing, but there is actually another set and the other set is really about anomalies that may happen which could effect the way that the intrigues are going to be used at cold start.

HOST Q3: Can you elaborate on those anomalies?

SME A3: Yes the anomalies relate to problems within the definitions themselves, an example of this might be, a definition which has actually duplicated and appears more than once through the groups and the group list, so if that occurs, one will win and one will lose, now which one is the winner and which one is the loser, perhaps it's a case of "first in" wins, it might be a case of where the last one wins. Another type of anomaly could occur where there is a definition that relies on some kind of a system resource or another type of definition but that other definition isn't present, so that could sometimes generate a warning condition, but sometimes its actually like a referential integrity error. On these types of anomalies, you might start off with a set of candidates that are coming from your CSD or your CPSM environment, but what actually happens is, not all candidates are actually eligible for installation, that's the type of anomalies that I'm talking about. What you want to know is at cold start time, if any of these anomalies exist, it's nice to know that because it means that the resources that were going to go into your run-time are perhaps short in list because of these other types of anomalies.

HOST Q4: So how does CICS Configuration Manager do this, and how is that any different to what customers are doing now?

SME A4: Well let me answer the last part first, how do customers do it now, by in large they don't! some customers have actually undertaken to do this but they find it quite extensive, time consuming and also system resource hungry, because what they tend to do is bring up a cloned set of CICS regions on the new sub definitions, then they dump the in memory images from the cloned set and the original production regions and then they somehow wash the data to remove any anomalies and then go through this tedious task of then comparing them, so you can imagine how labour intensive that is, especially when you consider that your using your best system programmers to do this analysis and comparison. So that's what the customers are doing, but what CICS Configuration Manager is doing is, it has collectors, and these collectors go out and they gather information about the runtime and they have collectors that go off and analyse the CSD or CPSM candidates of it could be a hybrid of both and then it take these two sets of collections, the runtime and the candidates, and feeds them through a comparator analysis to give you the output reports that your seeing. Now one of the interesting things about the way CICS CM does this is that we're talking about an immediate analysis, no need to stop and start your production regions, there's no need to have new or cloned CICS regions, it does it off the existing CSD or CPSM repositories in your existing CICS regions. It does this, because what it's doing is an interpretive analysis as opposed to actually having to rebuild or re-clone physical environments.

HOST Q5: You make it sound very simple, can you describe some of the complex hurdles you had to overcome to achieve this?

SME A5: Sophistication inside CICS CM hides some of these complex issues from the customers. So for example, CICS CM is order sequence aware, either using CSD lists with groups, then it is possible that a group can be duplicated through different lists or you might have the same resource, define and multiple groups, in which CICS CM needs to be able to track which one which one is the winner and which one is the loser and that's made more complex by the possibility that some resources might be ineligible for installation because of referential integrity errors, if that occurs, the true winner has to take into account ineligible resources because the winner might cascade to a later version of the definition for example. If you consider the CPSM environment, you might have installed a resource assignment, and these allow you to filter particular resources within CICS systems, or apply override values. So when you look at the definition, what you see is not a true reflection of what could actually be installed in the run-time and CICS CM has to be able to track through that. Some customers might actually be a hybrid case where they have some definitions inside the CSD and some definitions inside the CPSM and they are actually drawing definitions from both into the CICS run-time, so in a case like that, its actually the merger of the two sets of definitions and that has to be tracked through as well. If you consider the run-time, then there's the issue about "how do you gather the information from the run-time, if you are a CPSM managed site then CICS CM can use CPSM services to go off and gather that data, but many customers might not have implemented CICSplex SM, so CICS CM actually provides it's own collectors, which can reside within that CICS region and collect the data. We can manage either the CSD based sites or the CPSM sites, or the hybrids, its all covered. Another interesting thing about the run-time resources is that when CICS actually installs definitions into the run-time, it interprets attribute information and produces like a reduced in memory operational set, so you're actually trying to compare apples from the CSD file or CPSM repositories to pairs which reflect the runtime resources and CICS CM understands that and is able to navigate through to find out, what's the same, what's new, what's missing and what's different.

HOST Q6: What have you done to make this easy to use?

SME A6 Let me mention four things, filtering, state information, what if analysis and something beyond cold start. When we first developed this facility we cold started a CICS region and immediately ran the cold start analysis report, and of course what we saw were quite a few resources in the CICS run-time that weren't reflected in the CSD and CPSM. These were dynamically installed resources, or auto installed resources and six of who represent noise, because you look at your output reports and you can see these definitions are missing, but this is absolutely normal for this particular CICS region, and what we did was, we build in a filtering capability and the ability to create a baseline filter file, the filter file has descriptions of specific or generic conditions that we don't want to report. Once you produce your baseline file, you can use that to feed into the future analysis and remove that noise from the report and then you can remove other conditions as you move forward and you've accounted for them, and you are now only focused on the things with real significance to you.

Another thing that we realised is that state information like in the CICS run-time “is it enabled? Is it disabled? Is it local? Is it a remote resource?” these are things that might be of interest to you, they appear in the report but they appear in different column areas of the report and can be report managed separately, so that increases the usability and flexibility that you have with your reporting.

Another thing is the what if analysis, the easiest way to describe that is “I have my CSD lists, but I wanted to put a new group into the list, what would happen if I added a new group to the list?” so when we go and we specify the inputs to say “where do I collect the resource candidates from?” you have the ability to change that collection list, to say “oh and add this group to it”, or use a set of groups and remove one which would have appeared in the list, and now you can play this what if game to see what will happen, and you can actually go and say “I don’t want to use the set of lists which are normally used for the CICS region, I want to use a different set of lists which I’m now setting up, what would happen if I used that instead?” and compare those sets of candidates to what’s in the CICS run-time.

The last thing I want to talk about is going beyond cold start analysis, you see this cold start analysis is about collecting candidates collecting CICS runtime and then comparing them. But we can go beyond that, we can collect one CICS runtime to a second CICS run-time and compare those. You can have two CICS regions which are supposed to be like clones to AOR’s running the same sort of definitions “But are they actually the same? Did I do some changes in one that I didn’t do in the other? So are they truly reflective of each other?” so now you can collect one run-time, collect a second run-time and compare those. You can say “I want to collect one set of resource candidates, using one set of lists or values to another set of candidates and compare one set of candidates to another set of candidates. So weather its candidates to run-time or candidates to candidates or run-time to run-time, you can do all of that, and I just think that it makes the whole facility that much more easy to use.

HOST Q7: So Reuben, in summary, please sum up what we have discussed today?

SME A7: Predicting cold start outcome improves the business operations, it addresses questions like “at the next cold start, what’s going to be different?” “if I change the group order, what’s the impact?” “when I edit a resource, but is the one that actually CICS installed?” and “are my similar CPSM resources also in the CSD?” and if so “which one wins?”. And with CICS CM the deployment analysis reports including cold start compare, just augments all of the other goodies that are already available in the product, things like standards enforcement, resource based security checking, audit tracking and reporting, analysis including cross CSD and CPSM displays with search and side by side compare of groups and lists and a very powerful change control capability, which allows you to promote from work to development to test to production, including the automatic transformation of attributes like data set names, group names .etc, there are some rules that you can define. This capability helps you to prevent having problems in the first place. So really CICS CM with the augmenting of the product through deployment analysis has become even a better choice for CICS resource management.

HOST: Thank you, Reuben, that was very interesting.

HOST: Well, that wraps up this podcast discussion. To find out more about CICS Configuration Manager, please go to the description for this podcast at:
<http://www.ibm.com/software/os/systemz/podcasts/websphereonz/>

Join us next time as we talk about another important mainframe topic. For now, this is Nick Garrod saying “Thanks for listening”.