

Channel Access Client/Server Interface for LabVIEW

Start of discussion by Holger Brand, H.Brand@gsi.de

GSI will build FAIR, an International Facility for Antiproton and Ion Research, http://www.gsi.de/zukunftsprojekt/index_e.html, and therefore we are evaluating possible SCADA solutions for accelerator and experiment control.

The two most important options are EPICS and PVSS/JCOP (CERNs favorite solution).

HADES, <http://www-hades.gsi.de/> becomes part of CBM http://www.gsi.de/zukunftsprojekt/experimente/CBM/index_e.html and a lot of members of the PANDA collaboration are coming from BaBar http://www.gsi.de/zukunftsprojekt/experimente/hesr-panda/index_e.html. Since both communities are using EPICS today, there is a good chance for using EPICS in the future.

Since our group, http://www.gsi.de/informationen/wti/dvee/exp_kontrol/exp_kontrol_systeme_e.html, developed an control system framework based on LabVIEW, <http://labview.gsi.de/labview/CS/cs.htm>, we would like to save our investments in that development. Last year we had some discussions with Jochen Klier, NI Germany, about the extension of the Data Logging and Supervisory Control Module (DSC) to large systems. I suggested to him to check the features of EPICS and PVSS for ideas on improving the DSC module.

On VIP 2004 in Germany I talked to Michael Santori, Buisness & Technologie Fellow of NI, about the plans of a native CA integration into LabVIEW. Probably I will have a discussion with the LabVIEW core developer team in Austin at the NI Week 2004. I also talked to Mathias Clausen, whether this activity is a part of the EPICS 2010 activities, but he did not know any details.

I would like to offer my participation within the task of integrating a native CA interface into LabVIEW.

I am interested to reuse all the existing LabVIEW instrument drivers, our CS framework and LabVIEW applications developed at external labs e.g. universities. Especially I would like to make LabVIEW RT a real IOC using pure LabVIEW because both existing interface are not suitable for this purpose as far as I know. The last point would hopefully end in a close collaboration with NI on LabVIEW developments.

My Starting Point

- Reasons for LabVIEW:
 1. Easy to learn.
 2. Graphical programming language, which makes it easy for people coming from electronics.
 3. Very close connection between programming and GUI.
 4. Good documentation and easy installation.
 5. Large number of existing instrument driver.
 6. Very good connectivity: ISA, PCI, PXI, VME, VXI, CAMAC, FieldPoint, Analog/Digital I/O, Time/Counter, Relais, Signalkonditionierung SCXI, Motion, Vision, RS232, RS485, GPIB, CAN, Profibus, Firewire, TCP/IP, DataSockets, OPC, DLL, Database
 7. It is easy to build your own instrument driver. (I remember my first device support module for EPICS!)
 8. Inherent multithreading and events! Semaphore, Occurences, Notifier, Queques, Rendezvous, Datasocket, OPC
 9. VI-Templates and VI-Server methods to enable object oriented programming
 10. LabVIEW RT and Datalogging and Supervisory Control
- But LabVIEW lacks of scalability to very large systems exceeding some 10000 PVs and some 10 nodes.
- Here EPICS comes into the game. We want to save out investment in LabVIEW and our "working horse" but we would like to use the scalability and features of EPICS. We guess that a lot of laboratory developments and prototyping will be done with LabVIEW. Especially small groups maybe do not want to start with EPICS in their home institutes. But of course they want to integrate their systems into the integrative SCADA system. This is of course in principle already possible with the existing LabVIEW/EPICS interfaces. But you cannot use them easily in LVRT systems.
- At this point I thought it would be a good idea to start cooperation with NI to use LabVIEW RT as platform for an IOC instead of VxWorks or RTEMS.
- Advantages for EPICS:
 1. Distributed IOCs and therefore no intrinsic bottleneck. Systems like LabVIEW DSC or PVSS II have an event manager and all events have to parse it.
 2. Channel Access, as alternative to NI-DataSocket, DIM, or pure TCP/IP
 3. I like the idea that the PVs reside on the IOCs with all their properties.
 4. Configuration database with substitution files for mass configuration.
 5. Event logging and alarming is of course one point.
 6. Lots of additional tools.
 7. A busy and powerful community.

Summary

In the following section I want to summarize the responses of

- Bob Dalesio, ldalesio@lanl.gov
- Jeff Hill, johill@lanl.gov
- Joseph Xu, jzxu@aps.anl.gov
- Willem Blokland, blokland@ornl.gov
- and contributions to the tech-talk list.

Already existing LabVIEW-CA Interfaces

- **ActiveX CA Support**

Labview can access EPICS through the channel access client interface. This allows LabVIEW to read/write EPICS values that are available from EPICS database servers.

- <http://www-wnt/hsc/LabviewCAServer.htm>, my own documentation for HADES referring to the documentation of Kay Kasemir, kasemir@lanl.gov
- <http://ics-web1.sns.ornl.gov/~kasemir/axca/index.html>, Official EPICS site.

This package is not longer maintained. Last build with EPICS base R3.13.7 and LabVIEW 6i.

- **LabVIEW Shared Memory Interface to EPICS IOC**

http://www.sns.gov/diagnostics/documents/epics/LabVIEW/SNS_LabVIEWEPICS.html

- The Shared Memory Interface links LabVIEW variables to EPICS IOC Process Variables (PVs). Data acquired and processed by LabVIEW is available to the IOC to communicate to an EPICS based control system. LabVIEW and the IOC can also send interrupts/signals to notify each other that data is available.
- The Channel Access Client for LabVIEW supports the use of LabVIEW as a display environment. You can obtain data synchronously or through a monitor.

Both interface libraries are not running on LabVIEW RT, as far as I know.

Serious Issues

- **Timestamps**

- There are some serious issues in the integration, however. When you are mixing data between EPICS and LabVIEW, you have to consider the possibility to correlate the data. The timestamps from Windows and the timestamps from EPICS are not likely to be useful in a mixed system without serious modifications and hardware.
- Get meaningful timestamps on the data. This could just be a time server that runs in the EPICS portion. This would mean that the time stamp for something running in Windows could be significantly off. For LabVIEW RT, I don't know what the time specifications are.

- **CA-Server/Client**
 - EPICS has all variables available to the network by default, LabVIEW requires that they be explicitly published to the network. So, a policy decision needs to be made here. It would be easier to only allow access to those variables that are explicitly made available. It would be consistent with the way LabVIEW now works. Having all variables available would probably require too much for a change to LabVIEW.
 - Then the major part of the work is the connection of the data through the channel access server. For this, the metadata issue will need to be resolved. EPICS expects metadata for each value to support control and display. This will be the bulk of the programming, I suspect.
 - The LVRT uses Pharlap as its RTOS. In EPICS, VxWorks is like Pharlap and EPICS like LV and LVRT. To make LV talk EPICS, one of the easy way is to add EPICS Channel Access (CA) extension to LV or LVRT. But that is not integration but rather bridging. Any thoughts?
- **Documentation**
 - As for protocol documentation I am sorry to admit that none exists. However, the news isn't all bad because steps have been taken to remedy the situation, and a document will probably be available soon.
 - It should be possible to call the CA client and server win32 DLLs directly from within a LabVIEW process. The CA client programming interface is documented in the "CA Reference Manual" available now in HTML format with every R3.14 EPICS release.
 - The portable server interface, based on GDD, is guaranteed to evolve. The documentation for the portable server is at this URL:
<http://lansce.lanl.gov/lansce8/Epics/ca/ca.htm>. Beware, that the CA client documents at this address are old. Please instead use the "CA client reference manual" supplied with the R3.14 releases.
- **LabVIEW for Channel Access data display demo**

For all the people who are interested in using LabVIEW for Channel Access data display I [Willem Blockland] have created a little demo to show how LabVIEW could be used. I think it would help deciding on the specs to give to NI when they will integrate CA and LabVIEW. It is built on top of Andrei Liyu's CA client and requires virtually no programming to get a display with live data. It is very EDM like in that you just put an indicator like a waveform graph on the front-panel, give it the name of an EPICS PV, save and close the panel. You can then run the panel by selecting it from a list in the start screen utility. This utility, creates a background task that will scan the PVs and send data to the indicators on the front-panel. It is in the early stages and supports only a limited number of LabVIEW's graphs and indicators (waveform, chart, table, single indicators of float type). You can also do settings and add program code to analyze the data or save to disk. Currently, it can only scan PVs not monitor. The code should work on Windows, Linux, and Mac OS X and requires LabVIEW 7.

My Proposal

- The first thing I thought of was to be independent of NI. Maybe it is a good idea to build the portable CA-Server and CA-Client libraries into a DLL with LabWindows/CVI. This DLL could be then called from LabVIEW RT. I still do not know the details, but I guess that this results in registering callback functions to be called from the DLL via VI-Server methods. In that case the LV developer would have to provide VIs which could be called from the DLL via VI-Server methods. In principle it should also be possible to use Notifier and Queues as interface between the DLL and LabVIEW RT, so that the execution of the LabVIEW VIs is event driven. This solution would be reasonably consistent with open source. The following not.
- The other possibility would be a close cooperation with NI
 - NI could develop something that optionally provides the CA-Server and CA-Client native with LabVIEW RT. In that case the LV developer would have to provide VIs which could be called from the LV engine via VI-Server methods.
 - NI could also extend the functionality of their Data Logging and Supervisory Control Module to use the EPICS configuration database for setup and to use CA instead of the Lookout protocol.
 - NI could implement CA similar to the propriety DataSocket protocol for communication.

These are my first ideas. I am happy to get so much feedback in which direction to go.

Some selected opinions

- **Bob Dalesio**

It sounds like a great idea. I think there are several people that will be interested in the work. It will be particularly useful for places that have universities pull up small experiments at the end of their accelerators - or telescopes.
- **Joseph Xu**

I'm with the APS Controls group and assigned to investigate the needs for LabVIEW/EPICS integration.
- **Date Brew**
 - In terms of future needs on the client end, LabVIEW w/ SCA/Win32 works about as well as I would want. At the server end, as I mentioned, a CA interface would be nice as far as integrating positioners and detectors controlled by LabVIEW into our beamline software. The main requirement I can think of for a future CA interface is that it be more scalable than datasocket. With datasocket, one has to wait for client writes to individual datasocket items with a separate element on the diagram for each datasocket item. This makes it difficult to scale up to a large number of datasocket items. I've resorted to combining individual related variables into an array or LabVIEW "cluster" and sending/receiving the whole thing at once even if only a single element changes.
 - I haven't used the ActiveX interface to epics, but there are events, which are called occurrences, and threads available in LabVIEW itself. It's

pretty straightforward to set up a parallel structure in a vi that sits and waits for events from other areas of the diagram, or to launch a vi that sits and waits for events from other vis. There are also semaphores, queues, and some other types of synchronization. There is also support for threads, but not very flexible. There are about 5 threads to which vis can be assigned ahead of time. Within each thread LabVIEW does cooperative multitasking, to which there are limitations (for example I think any called dll takes over its thread until it's finished).

I've used the LabVIEW "datasocket" server, which I think is based on activeX. Basically I have a small server application that launches a few subvis executing in separate threads and waiting for new values in some datasocket items, which are passed off to queues in other subvis that handle the input. This is a pretty small scale server.

- **Tom Meyer**

I like the idea of using a LabVIEW front-end to EPICS since it combines the strength of LabVIEW (its user interfaces) to the strength of EPICS (its rapid execution).

Has anyone looking into the idea of a CORBA interface from a Unix or Windows machine running LabVIEW to an IOC running VxWorks?

- **Paul Sichta**

I have a few LabVIEW systems that use the Simple Channel Access (SCA) package from Timossi, et al. I use SCA as a CA client only, so the LabVIEW programs do a handful of gets and puts to an IOC at about a 1 second rate. The LabVIEW systems are basically stand-alone; We use the CA interface to couple the subsystem into our experiment's shot cycle and for (central) alarm handling. From my viewpoint, LabVIEW is attractive because of it's extensive device support, often supplied for free by the hardware vendor. For our simple EPICS/LabVIEW interfaces we don't need performance.

Links

- EPICS Homepage: <http://www.aps.anl.gov/epics/index.php>
- EPICS Tech-Talk: <http://www.aps.anl.gov/epics/tech-talk/index.php>
- NI Homepage: <http://www.ni.com>
- GSI Homepage: <http://www.gsi.de>
- ECoS Homepage:
http://www.gsi.de/informationen/wti/dvee/exp_kontrol/exp_kontrol_systeme_e.html
- FAIR Experiment Control:
<http://www.gsi.de/zukunftsprojekt/experimente/Controls/index.html>
- GSI Discussion forum: <http://forum.gsi.de> -> Controls

Darmstadt, 3. Juni 2004

Holger Brand