Tcl, The Glue for a New Generation of Nuclear Physics Experiments

Ronald Fox
National Superconducting Cyclotron Laboratory, Michigan State University
640 S Shaw Lane
East Lansing, MI 48824-1321
fox@nscl.msu.edu

CAEN Technologies
1140 Bay Street Suite 2C
Staten Island, NY 10305
ron@caentech.com

ABSTRACT
This paper will describe two case studies in which Tcl components played a key role integrating diverse detector data acquisition systems together into a coherent integrated data acquisition system. At the NSCL Tcl and NSCLDAQ were used to integrate the recently upgraded S800 Spectrograph Data acquisition system with that of GRETINA, a large segmented Germanium gamma-ray tracking detector. At Argonne National Laboratories, the CHICO2 detector system, using a tailored version of NSCLDAQ, was integrated with the Digital Gammasphere detector system. The work done to perform these integrations will be described as well as the experience gained from campaigns with these integrated systems.

1. INTRODUCTION

The detector systems used in experimental nuclear physics have been steadily growing in complexity. As a result, some detectors are required to have integrated data acquisition systems built with data flow architectures designed to meet the specific, specialized needs of those detectors. This is in marked contrast to the prior generation of detector systems which would often allow a laboratory like the NSCL to use a single general purpose data acquisition system to instrument all detectors in use at that institution.

In many cases, however these complex, large channel count detector systems are single purpose specialized detectors and, in order to get useful science, they must be run in conjunction with one or more other detectors. The integration of these diverse detector systems, each with their own, data acquisition system, each system with its own design goals, is one of the new and recurring challenges software development for nuclear experimental physics.

In this paper I will first describe the basics of a nuclear physics experiment. I will then describe a pair of case studies where the NSCLDAQ general purpose data acquisition, through its prominent use of Tcl/Tk was used to 'glue' pairs of diverse detector systems together to present an integrated system to the users performing experiments with them.

The first case study will describe CAEN Technologies work to integrate the CHICO II detector system with Digital Gammasphere. The two detector systems will be described along with their data acquisition systems. The integration will be described as well. The second case study will describe the NSCL's integration of the data acquisition system for its S800 charged particle Spectrometer with the GRETINA segmented germanium gamma-ray tracking detector.

The detectors will be described along with the data acquisition systems each uses as well as the integration effort and the role Tcl played in this work.

Finally, experiences with the integrated systems will be described as well as future work and what this implies for the data acquisition systems for the Facility for Rare Isotope Beams under construction at the NSCL.

2. BACKGROUND

This section describes how experiments in nuclear physics are performed. The NSCL Data Acquisition System (NSCLDAQ) is also briefly described along with some of the roles Tcl/Tk plays in that system.

2.1 Nuclear Physics Experiments

Since in general, atomic nuclei are much smaller than the wavelengths of visible light, microscopy is not an effective tool to explore the structure of and the forces that bind the nucleus together. Experiments on nuclei are performed by colliding nuclei with other particles whose quantum wavelengths are of the same order of magnitude as the nucleus itself.

The protons and neutrons within a nucleus are bound together by what is known as the *nuclear strong force. F*or elements heavier than hydrogen, the nucleus is a positively charged electromagnetic compound particle, and therefore affects the trajectories of other charged particles.

The charge distribution of nuclei can therefore be probed by the way electrons scatter off the nucleus because electrons will only feel the Coulomb force due to the nuclear charge. Other nuclei with energies that allow the repulsive Coulomb barrier to be crossed can map the strong force and hence the structure and energy levels of the target nucleus.

Producing projectiles with sufficient energies to interact with nuclei in a useful way for experiments is why nuclear experimental physicists require accelerator laboratories in which to perform their experiments.

When nuclei collide, they interpenetrate, and, depending on the total energy of the resulting system, and the impact parameter (a measure of how far apart the center of masses of the colliding nuclei are) they may break up (fission), glue together (fusion), or some combination of the two processes.

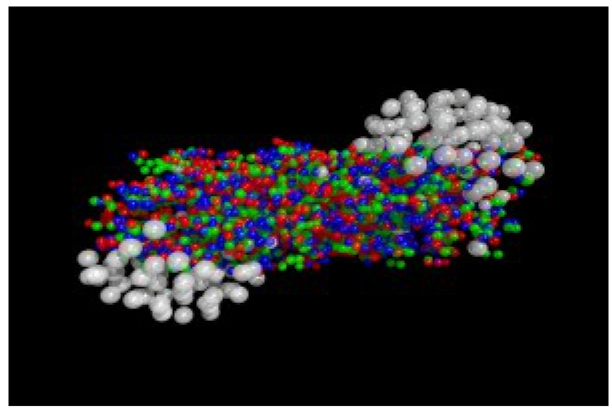

Illustration 1: A collision between two lead nuclei

Given high enough energies and sufficient numbers of protons and neutrons, the results of these collisions can be very complicated as shown in Illustrations 1 above and 2 below:

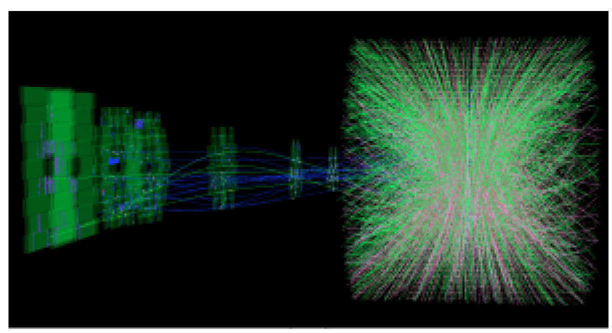

Illustration 2: The same lead-lead collision as seen by the ALICE detector at CERN/LHC

The showers of particles that are emitted by these collisions are also not directly visible. Complex detector systems are required to "see" them and determine important physical characteristics such as their kinetic energies, the time of detection their masses and flight direction.

The signals from these detectors are digitized, read out and transmitted to computer systems which perform online analysis and record the data for further study offline. The number of computer nodes that are involved in an experiment can be as few as one, and as many as 100 or more. The CMS detector system at CERN's LHC requires 100's of nodes running 1000's of programs and produces 1.5Mbytesof data for each recorded collision[1]. With collisions being detected many 1000's of times per second complex *trigger systems* are used to filter out data that is not of interest, lowering the rate of CMS events recorded to fewer than 100/second.

The increasingly lower cost of computing and innovations in detector instrumentation are rapidly pushing low and intermediate energy nuclear physics experiments such as those performed at the National Superconducting Cyclotron Laboratory at Michigan State University and ATLAS at Argonne National Laboratories to use larger and larger detector systems. These larger detector systems produce data flow problems that are difficult to solve in the general purpose data acquisition systems that have been used by these labs in the past. This in turn pushes detector designers to include special purpose data acquisition systems as part of their detector designs.

Special purpose data acquisition systems solve the data flow and data handling problems of specific detector systems, but do not address how to take data when there is a need to use more than one detector system in an experiment. For example the Gammasphere detector is, as the name implies capable of detecting γ-rays but without the addition of a charged particle detector such as CHICO[3], is unable to perform important particle-γ coincidence experiments.

An increasingly important problem, therefore, is the knitting of detector specific data acquisition systems into an integrated *meta data acquisition* system that can be used in a practical manner to perform useful experiments.

2.2 NSCLDAQ

The NSCL Data Acquisition system (NSCLDAQ) is a general purpose data acquisition system. This system has been described partially at past Tcl conferences, most recently at Tcl 2011[4]. This section provides a very brief overview of the system as it is today.

NSCLDAQ's core is a ring buffer data distribution system. Each ring buffer can have a single producer and a number of consumers. Access to consumer get pointers and the single producer put pointer is mediated by a pure Tcl server called the **Ringmaster**. Ringmaster maintains a persistent TCP/IP connection with all resource holders allowing it to release resources on client exit regardless of the cleanliness of that exit.

Distribution of data to remote systems is handled by setting up a proxy ring buffer in the remote system and an ordinary client in the local system. The local client sends data from the ring over a simple TCP/IP socket to the producer for the proxy ring. As ring buffers are identified to the system via a URI name space, remote ring access requests can be automatically detected and the set up of the proxy ring is transparent to the consumer.

A ring naming convention ensures that only one proxy ring is created for a given remote proxy ring in a local system.

Several loosely coupled and even uncoupled components as well as application frameworks provide support for various data sources and useful data consumers (for example event recording, status display and support for supplying data to non NSCLDAQ aware clients). Tcl/Tk figures prominently in all user interface applications and, in many cases also provides an extension and scripting language for these components.

One of the integrating components is a user interface that controls data source programs. This **ReadoutShell** provides an integrated start up of the event recording software as requested by the user. The ReadoutShell user interface is shown in illustration 3 below:

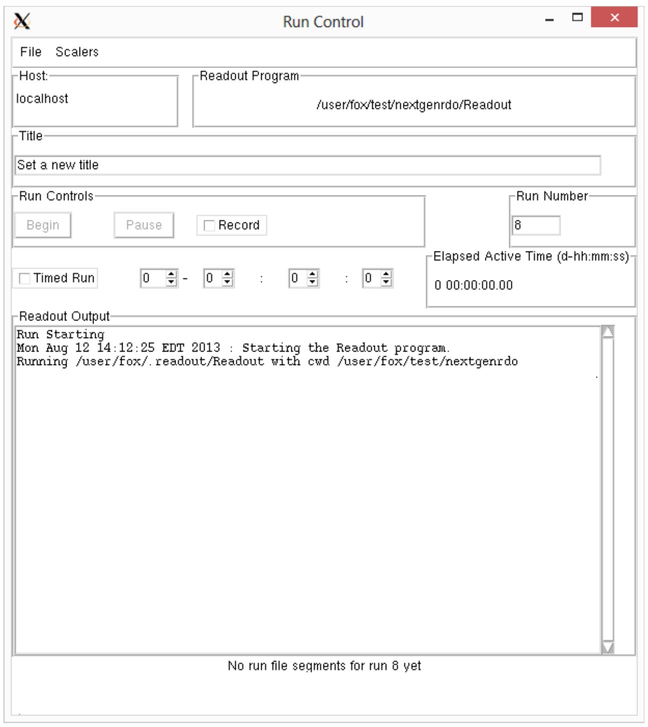

Illustration 3: ReadoutShell's user interface.

An important feature are the extension points provided by the ReadoutShell. It turns out that these will play a key role in the knitting together or diverse data acquisition systems.

When ReadoutShell starts it looks for a user extension script named **ReadoutCallouts.tcl**. If found that script is sourced at the global level and can provide several **proc**s that are called at well defined points of ReadoutShell's operation. ReadoutShell itself exports an API these extensions can use. The extension points are:

- **OnStart** – Invoked when the Readout program is started by the ReadoutShell.
- **OnBegin** – Invoked just prior to beginning a data taking run.
- **OnEnd** – Invoked just prior to ending a data taking run.
- **OnPause** – Invoked just prior to pausing a data taking run.
- **OnResume** – Invoked just prior to resuming a data taking run.

Another important, however non-Tcl difference we will come across is implicit in the nature of the ring-buffer implementation:

- Ring buffers are named entities
- An arbitrary number of ring buffers can be instantiated.
- There is not really any preferred set of clients (producers or consumers) to any ring buffer. A ring buffer client is just a program that uses the Ring buffer API
- The ring buffer software provides a pair of clients that make the production of ring buffer to ring buffer pipelines easy to construct. These clients are used by the ring buffer API set up data transmission from host ring to remote proxy ring:
    - *ringtostdout* – copies data from a ring buffer to stdout
    - *stdintoring* – puts data on stdin into a ring.

It turns out that *ringtostdout* and *stdintoring* can be used as the source sink of data from UNIX pipelines of programs.

3. CHICO2 AND GAMMASPHERE

The CHICO2 detector is a charged particle position sensitive detector. It has been optimized to work with both the GRETINA and GAMMASPHERE γ-ray detectors. We will see more of GRETINA in the next section.

CHICO2's geometry allows the GAMASPHERE detector to fully close around CHICO2. CHICO2 also fits within the GRETINA detector frame. Furthermore the CHICO2 position resolution has been optimized to work well with both GAMMASPHERE and GRETINA's position resolutions. [5] provides a summary of CHICO2 and its design goals. Illustration 4 below shows the Chico2 detector with GAMMASPHERE.

GAMMASPHERE is best known in the popular press as the nuclear physics detector that made a cameo appearance in the Universal Studios movie *The Incredible Hulk.* The scale model of GAMMASPHERE created in that movie is

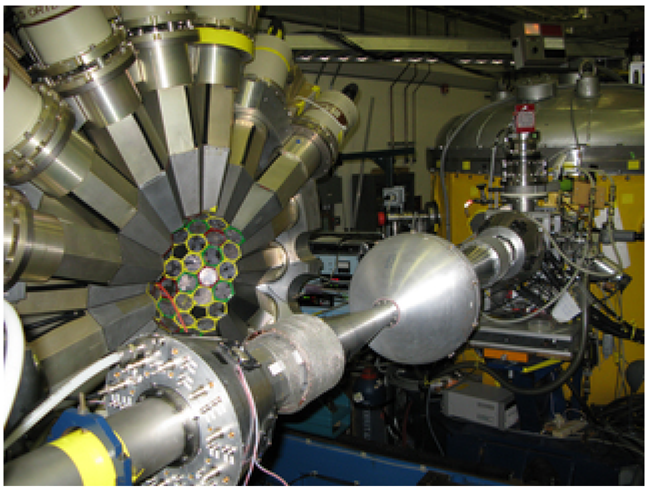

Illustration 4: Chico2 and one hemisphere of GAMMASPHERE

on display at the time this paper is being written at Universal Studios in Orlando, FL.

GAMMASPHERE (the real one) is located in an ATLAS beam line at Argonne National Laboratories (ANL). Prior to using CHICO2 with GRETINA, the detector had several physics runs scheduled for Spring/Summer 2013 with GAMMASPHERE.

Illustration 5: Gammasphere to the right of its best known user; Dr. Bruce Banner.

CHICO2 has a data acquisition system based on NSCLDAQ. This author provided that system under direct contract to Lawrence Livermore National Laboratories (LLNL) which, together with Rochester University, built the CHICO2 detector. NSCLDAQ system allowed CHICO2 to be tested at Rochester University prior to its shipment and installation with GAMMASPHERE at ANL.

GAMMASPHERE recently upgraded its own data acquisition system to a fully Flash ADC continuously Digitizing System. The name of this system is “Digital Gammasphere” (DGS). This system is described in [6]. A block diagram from the talk slides for that reference, which focuses on the electronics is shown as Illustration 5 below:

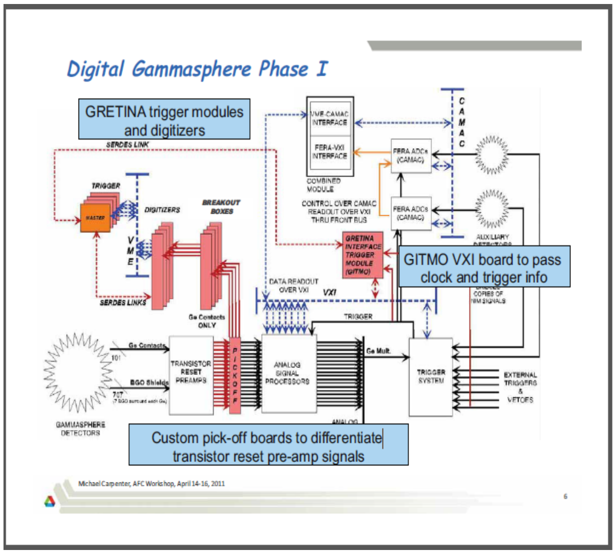

Illustration 6: Block diagram of digital gammasphere

Only features important to the integration with CHICO2 will be described below.

3.1 GAMMASPHERE Run Control

DGS's run control is managed via an EPICS[7] process variable. A second EPICS process variable reports the run state. The DGS run state machine only provides Running and Halted states while the NSCL run state machine provides an additional Paused state.

Unlike many data acquisition systems, of its sort, DGS does not build events online but accumulates event segments in a set of files (one per contributing computer). Offline analysis software (which may be run as soon as a data taking run completes), builds events and produces initial check spectra. The NSCLDAQ on the other hand interfaces with the NSCLSpectTcl[8] analysis software to provide online spectra.

3.2 The CHICO2 Data acquisition System

CHICO2's data acquisition system is an implementation of the NSCLDAQ using the VM-USB readout and SpecTcl described in [10]. As such it uses the ReadoutShell software described previously.

A block diagram of the CHICO2 electronics is shown below in Illustration 7

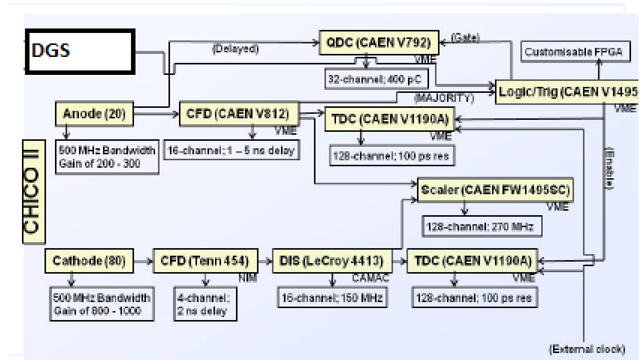

Illustration 7: The CHICO2 electronics diagram.

With the exception of the CAEN V1495SC Scaler module and the CAEN V1495 logic module all modules were already well supported by that software.

A Driver module was written to support adding the V1495SC module to the Readout configuration file making it available to the NSCL Scaler display (a pure Tcl application as well). We will come back to the role and support provided for the V1495 logic modules in the section below.

3.3 Integrating the two systems

Integration therefore involved finding and implementing answers to the following questions

- Who controls the run and how?
- Where is event data recorded and how does it get there?
- How is the event trigger determined and passed between the detectors?
- How are timestamps produced in CHICO2 such that they are meaningful to DGS data and can be used to build events in offline analysis.

The simplest problem to solve turned out to be that of timestamps. While the Gammasphere time stamp and trigger distribution system module was a VXI format module[9] (VME with an extra connector making for a 9-U format), the module designer J.T. Anderson had just completed a VME (6-U) format module. A driver module for this device was incorporated in the NSCL VM-USB readout software to provide the capability of adding this module to the Tcl configuration file that drives the readout. The SpecTcl 'unpacker' for this module simply ignored the data packet produced by that module.

At a very early stage it was determined that it would be much simpler to use the NSCLDAQ ReadoutShell's extension to control DGS than it would be to modify the DGS run control panel to control NSCLDAQ; since the NSCLDAQ ReadoutShell is a pure Tcl application it was a relatively simple matter to use the EpicsTcl[11] package to set the DGS run control PV and monitor the state of the DGS PV in a text widget. These were all implemented via the ReadoutShell's ReadoutCallouts.tcl extension script. The ReadoutShell's API directly supported the addition of a DGS status strip with the run state.

The actual event trigger was a bit more interesting. Due to the timing considerations. Chico2 trigger decisions was a hardware discriminator and quite fast relative to the GAMMASPHERE trigger decision. This was because the GAMMASPHERE trigger decision is the result of FPGA based analysis of the digitized waveforms (note this is done in parallel with digitization but nonetheless sufficient information must be digitized to provide for a digital constant fraction discrimination).

Furthermore, DGS required a DGS trigger accept to be received in one or more of its GITMO or Myriad modules to accept the digitized data as an event. The necessary logic and delays were encapsulated in firmware programmed by Carlo Tintori of CAEN Technologies. This firmware was embedded in a CAEN V1495 FPGA module.

A pure Tcl control panel was written that interfaced with the slow controls server of the VMUSB Readout program. The control panel controlled internal delays that lined up GAMMASPHERE's trigger with CHICO2 triggers. Output trigger widths were also controlled, and most importantly, the control panel allowed internal test points of the FPGA firmware to be gated to module outputs where they could be monitored. Without this it would have been much harder to set the internal module timing.

As we have previously written, DGS does not actually do online data monitoring. Data from each of its I/O controllers (IOCs), are recorded in a separate file. Once data taking is complete for a run (an experiment in general consists of many runs), offline analysis is done immediately to build both events and spectra from the events fragments in these files.

We decided that, while the CHICO2 DAQ system could locally record data, it would be advantageous to also provide the the ability to send CHICO2 data to DGS as if CHICO2 were an IOC. Unfortunately, the protocol used to send data from IOCS to DGS is quite complex. We decided instead to use a simple push protocol.

A ring buffer consumer for CHICO2 event data was started whenever a run began. After connecting to a very simple Tcl catching program, the data consumer wrapped each CHICO2 event inside a DGS header and sent them to the catcher. The catcher opened an event file with the appropriate name and saved data to disk without interpretation. The sender was written in C++ and the catcher was a pure Tcl program.

A small library of functions was provided to DGS (in C) that decoded the CHICO2 events and supplied them to the caller. This made the integration of CHICO2 event fragments with DGS event fragments a straightforward bit of programming. That work was done by the DGS collaboration.

3.4 Experimental experience

After completing the integration, a set of commissioning test runs and physics runs were performed using the ATLAS[12] accelerator at ANL. The commissioning with stable beams went smoothly. Analysis performed were able to demonstrate particle γ coincidences between Chico2 and DGS without any difficulty.

The physics runs using unstable beams from the CARIBU[13] ion source were also smooth from the data acquisition standpoint. Unfortunately the beam intensities on target were much lower than originally predicted, and unforeseen analog contaminants were present in the desired beams. As a result, the physics runs did not get the hoped for statistics.

ANL Atlas is currently down while a power transformer damaged in a lightning strike is replaced. When scheduled running resumes, GRETINA will have been installed at ANL and the CHICO2 DAQ system will be integrated with that system in a manner similar to that described in the next section.

4. S800 AND GRETINA

GRETINA[14] consists of a number of segmented high purity germanium detectors. The signals from each crystal's central contact and each segment are continuously digitized at 100MHz. FPGA based digital signal processing of these waveforms coupled with complex deconvolution algorithms allow the detector to achieve position and energy resolutions that are better than achievable by a single segment in isolation. Compton tracking allows suppression of γ-rays that either did not originate in the detector or that scattered out prior to depositing full energy in the detector. This detector was constructed by a large collaboration largely based at LBNL.

In the Fall of 2011, after commissioning runs at LBNL, GRETINA was transported to the NSCL to begin a year long experimental campaign. The experiments performed required looking at coincidences between radioactive isotope projectile-like fragments and the γ-rays emitted by those fragments as the decay from the excited states they populate on collision with the nuclei in fixed targets.

The NSCL device used to identify and detect the projectile fragments was the S800 spectrometer[15] and its associated detector package.

This section will describe the GRETINA data acquisition system, the S800 data acquisition system and the work done to fuse those two very different systems into a coherent whole.

4.1 The GRETINA Data Acquisition System

GRETINA is an example of a new breed of waveform digitizing data acquisition systems. A block diagram of the data acquisition system is shown below in Illustration 7 below.

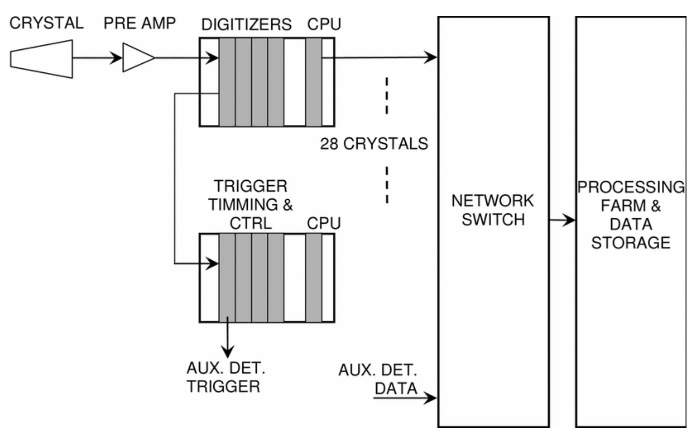

Illustration 8: Block diagram of GRETINA DAQ

The processing farm to the right side of the picture contains several hundred server level compute nodes that:

- Determine the interaction points in the detector crystals from waveform information.
- Put event fragments from the crystals into a time ordered list
- Save this ordered list online to a modest Network file appliance.

The entire system is controlled by a number of EPICS[16] process variables. The event reorderer (which GRETINA calls the Global Event builder), provides a tap that allows computers external to the GRETINA cluster to obtain sets of consecutive samples from the Global Event Builder.

The GEB is more properly an event fragment orderer. The samples it provides have the form of event fragments that are wrapped in headers that contain time-stamps from the GRETINA digitizer 100MHz timebase and payload sizes as well as fragment type descriptors that allow analysis software to know roughly what to expect in the fragment payloads. These event fragments have monotonically increasing timestamps within a data taking run.

4.2 The S800 Data acquisition system

The data acquisition system for the S800 spectrograph at the NSCL is shown below in illustration 9 below.

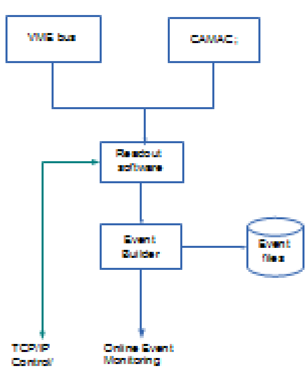

Illustration 9: S800 Data Acquisition system block diagram

In this illustration, the Readout programs and event builder are co-resident in a single computer system. There is nothing in that architecture that requires that. The Readout software makes use of a VM-USB[16] (VME/USB list mode bus adapter) to interface with the VME based electronics and a CC-USB[17] (CAMAC/USB list mode adapter) to interface with the CAMAC based electronics.

Some of the modules in both the VME and CAMAC buses are field programmable gate array modules (FPGA) with custom firmware to autonomously readout external digitization hardware. One of these FPGA modules includes an externally clocked counter to produce an event time stamp and logic to support synchronization of the initialization of that counter with other timestamped systems.

The S800 system does not use any NSCL DAQ software. One of the monitoring programs used by the S800 support group is, however NSCLSpecTcl.

4.3 Merging the two systems.

A successful merger of the S800 and GRETINA data acquisition systems requires that:

- Coordination of the time-stamp synchronization between the S800 and GRETINA be properly managed.
- Data from the S800 be provided to the GRETINA GEB in a format acceptable to that software.

This is sufficient to allow GRETINA to produce ordered fragment files which can then be analyzed offline. In order to try to get some physics information online during the run, we need the following as well:

- The ability to get data from the GRETINA GEB sampling output tap.

- The ability to merge fragments that occur within a predefined coincidence time window into single events.
- The ability to provide those built events to analysis software.

Since the appropriate analysis software existed within the context of the NSCL Data acquisition system, making the built events from GRETINA a data source for NSCLDAQ is sufficient to meet these needs.

Additionally it was deemed highly useful to provide a coherent run control for the detectors as would later be done for the Chico2/Gammasphere case study already described.

4.3.1 Event Data from S800 to the GEB

In order to send data from the S800 to the GRETINA GEB, we must:

- Accept data from a connection to the S800 TCP/IP monitor port.
- Extract events from the block structured data provided by the S800 monitor port
- Extract the time stamp from each event
- Wrap the event in a GRETINA GEB header and lastly
- Use the GRETINA GEB API to post each event to the GRETINA GEB.

We also wanted to be able to monitor the S800 events in isolation prior to submitting them to the GRETINA GEB. This suggested performing the first two steps of the process above, inserting the results in an NSCLDAQ ring buffer and performing the last two steps in an NSCLDAQ ring buffer client.

The serial nature of the operations described above along with the existence of the *stdintoring* and *ringtostdout* components of NSCLDAQ suggested the use of UNIX command pipelines for both of these processes. We could imagine the insertion pipeline would be something like:

```
netcat | evtextract | stdintoring
```

Initially a structure like this was tried. What we found, however was that UNIX netcat was too heavily and intelligently buffered to support the low data rates of secondary beam experiments. Furthermore, netcat did not guarantee the pipeline received the end of run indicator block before the start of the next run pushed it through.

Therefore this structure was used but with the netcat like functionality we needed re-written in the Tcl script shown below:

```
while {![eof $fd]} {
    chan copy $fd stdout -size 8192
    flush stdout
}
```

where the fd variable above was the socket connected to the S800 monitor TCP/IP port.

Similarly the ring client is a pipeline of ringtostdout, a program that extract timestamps from events, wraps them and pushes them to the GRETINA GEB. In order to make the time stamp extraction generic, this is actually done in a shared object that is specified on the command line that starts the program. This allows the output pipeline to send data from any NSCLDAQ data source to the GRETINA GEB.

The pipeline structure described above also allowed us to test each component in isolation. Integration testing actually occurred prior to the arrival of GRETINA using data files from the S800 and a set of stubs for the GEB API.

4.3.2 Data from GEB to NSCLDAQ

In order to get good gamma spectra from GRETINA it is necessary to correct the gamma-ray energies for the Doppler effect that arises from the fact that they are emitted from the remainder of the projectile which is moving forward from the target. To do this we need to know the velocity of that projectile-like fragment, and that requires building events from the S800 and GRETINA to detect particle-gamma coincidences.

In the context of the SeGA (segmented Germanium array at NSCL), Doppler correction code already has been written for S800/SeGA particle gamma coincidences. If we can provide S800/GRETINA built events, that code can also be used to Doppler correct the gamma rays GRETINA sees.

The GRETINA GEB includes an interface that allows clients to selectively receive a sample of the data it has ordered. The sample fills a buffer with a time-contiguous set of fragments.

Unfortunately, what we get from the GEB are not built events, but ordered fragments. To determine if there is a particle-gamma coincidence in the per-existing NSCLSpecTcl code we need to:

- Accept data from the GEB sample tap.
- Glue together fragments that occur within the specified coincidence window of the first fragment of a sequence
- Format the result as NSCL ring buffer event items.
- Insert those items in a ring buffer where NSCLSpecTcl can retrieve them for online analysis.

This again suggests a Unix pipeline of the form:

```
tapcat | glom | stdintoring
```

Where

- *tapcat* is a command that reads data from the GEB sample tap and sends it to stdout.
- *glom* is a program that accepts GEB data on its stdin, glues (gloms) together fragments into built events which are emitted in NSCLDAQ ring buffer format on its stdout.
- *stdintoring* is a component of NSCLDAQ that takes ring items on stdin and drops them into an NSCLDAQ ring buffer..

This pipe structure allowed the building program (*glom*) to be tested on event file data. Furthermore, with GRETINA moving to Argonne National Laboratory to continue its experimental program at ATLAS, *tapcat* will be used by them as input to their own GRETINA online analysis software.

4.3.3 Consolidated control

Consolidated run control was done essentially in the same way as Chico2, using the call out script facility of the NSCLDAQ run control software. In this, case, however neither detector system uses an NSCLDAQ readout program.

Therefore what we had to provide was:

- A call out extension that knows how to manage GRETINA runs.
- A call out extension that knows how to manage S800 runs.
- A call out extension that knows in which order the S800 and GRETINA must be started and stopped and is able to tell GRETINA when to start the S800/GRETINA time stamp synchronization process.
- A dummy NSCLDAQ readout program that ignores all run control commands sent to it.

The three call out extensions are all pure Tcl components, with the GRETINA script relying on EpicsTcl, since EPICS is used to control GRETINA's runs. The dummy NSCLDAQ readout program is really just

```
cat >/dev/null
```

5. RESULTS

CHICO2's integration was successful but unfortunately satisfactory beams from CARIBU could not be achieved before ANL/ATLAS suffered its power transformer. The next projected CHICO2 runs will actually involve a planned integration of CHICO2 and GRETINA which is being installed at ANL/ATLAS.

The GRETINA/S800 integration resulted in a year of successful physics runs. The on-line Doppler corrected gamma spectra made possible by this integration, provided invaluable information about the detector and experimental resolution.

During the year of running the code which wed the two detectors together did not express any defects. Furthermore the adoption of the *tapcat* program by ANL/ATLAS for its installation of GRETINA demonstrated the value of breaking up the data flow into a set of simple filter elements. While this concept is well known to Unix shell programs I am not aware of other uses of it in the main data flow path of a nuclear physics data acquisition system.

The generic data flow architecture of NSCLDAQ proved more than equal to the task of handling the data flows required of the GRETINA and S800 detectors.

The use of a Tcl based Run control program made it quite easy to extend the software to control detectors with control paths that were not originally planned for.

Future planned work includes the coupling of the CHICO2 detector with GRETINA for physics runs scheduled at ANL ATLAS in 2014.

In 2022, Michigan State University under a grant from the Department of Energy will be commissioning the Facility for Rare Isotope Research (FRIB). During the construction period, discussions are already underway about what detector systems would be useful for experiments performed at this facility.

In parallel with detector discussion groups, a committee including the author is discussing what we believe will be the high level requirements of a data acquisition system at the FRIB. We believe that the trend towards larger, more complex detectors will continue. We also believe that these detectors will, for the most part, have their own continuously digitizing data acquisitions systems, such as DGS and GRETINA do now.

We are therefore firming up the specifications of what might best be called a *meta-data acuisition system* one that is designed from the start to couple together two or more detector systems to build integrated experiments. The case studies described in this paper demonstrate that the NSCLDAQ systems has many of the attributes an FRIB metaDAQ system needs.

7. REFERENCES

[1] G. Bauer et al. *Monitoring the CMS Data Acquisition System* presented at the 17th International Conference on Computing in High Energy and Nuclear Physics (CHEP09), published in **Journal of Physics: Conference Series** 219 (2010) 022042

[2] M.P. Carpenter *How far from stability can we go using Gammasphere and the ANL Fragment Mass Analyzer?* Presented at CAPTURE GAMMA-RAY SPECTROSCOPY AND RELATED TOPICS: 10th International Symposium 30 Aug-3 Sep 1999 Sante Fe, New Mexico (USA) published in **AIP Conf. Proc.** 529, pp. 175-183.

[3] M.W. Simon*, D. Cline, C.Y. Wu, R.W. Gray, R. Teng, C. Long *CHICO, a heavy ion detector*

for Gammasphere **Nuclear Instruments and Methods in Physics Research A** 452 (2000) 205-222

[4] R. Fox et al. *Tcl at the NSCL: a 30 (15?) year retrospective* presented at Tcl 2011 October 24-28 Manassas, VA available online at: http://www.tclcommunityassociation.org/wub/proceedings/Proceedings-2011/RonFox/nscltcl.pdf

[5] CHICO2 is described at: http://www.pas.rochester.edu/~cline/Research/Chico2.html

[6] M.P. Carpenter et al. *Digital Data Acquisition System for Gammasphere* Presented at the 2011 Fall Meeting of the APS Division of Nuclear Physics East Lansing, MI October 26-29, 2011 published in the **BAPS** Volume 56, Number 12

[7] S.A. Lewis *Overview of the Experimental Physics and Industrial Control System: EPICS* Online at: http://csg.lbl.gov/EPICS/OverView.pdf

[8] R. Fox, C. Bolen, K. Orji, J. Venema *NSCLSpecTcl Meeting the Needs of Preliminary Nuclear Physics Data Analysis* Presented at Tcl 2004 available online at: http://www.tcl.tk/community/tcl2004/Papers/RonFox/fox.pdf

[9] J.T. Anderson *FPGA and VHDL Developments for DGS and DSSD* internal presentation to the ANL GAMMASPHERE Group September 20, 2012 online at

[10] R. Fox, K. Paräjärvi, J. Turunen, H. Toivonen *A Domain Specific Language for defining Nuclear Physics Experiments.* Presented at Tcl 2008 Manassas, VA October 20-24, 2008 available online at: http://www.tclcommunityassociation.org/wub/proceedings/Proceedings-2008/proceedings/nuclearDSL/stukdsl.pdf

[11] R. Fox *Tcl/Tk Tools for EPICS Control Systems*. Presented at Tcl 2007 New Orleans, LA September 24-38, 2007 Available online at: http://www.tclcommunityassociation.org/wub/proceedings/Proceedings-2007/proceedings/Gui/epics.pdf

[12] For a description of ATLAS and its capabilities see: http://www.phy.anl.gov/atlas/

[13] For information about CARIBU see: http://www.phy.anl.gov/atlas/caribu/ATLAS_Cf_upgrade.pdf

[14] GRETINA proposal A. Machiavelli et al. At http://radware.phy.ornl.gov/greta/Gretina_prop_partial.pdf

[15] *The S800 spectrograph* D. Bazin et al. **Interactions with Materials and Atoms** B V204 May 2003 pg 629-633.

[16] http://www.wiener-d.com/sc/modules/vme--modules/vm-usb.html

[17] http://www.wiener-d.com/sc/modules/camac--modules/cc-usb.html