International Conference on Computing in High Energy and Nuclear Physics (CHEP'07) IOP Publishing
Journal of Physics: Conference Series **119** (2008) 082001 doi:10.1088/1742-6596/119/8/082001

The Aragats data acquisition system for highly distributed particle detecting networks

S Chilingaryan[1], A Chilingarian[2], V Danielyan[2], W Eppler[1]

[1] Institut fuer Prozessdatenverarbeitung und Elektronik, Forschungszentrum Karlsruhe, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
[2] Cosmic Ray Division, Alikhanyan Physics Institute, Alikhanyan Brothers st. 2, Yerevan 36, Armenia

E-mail: `Suren.Chilingarian@ipe.fzk.de`

Abstract. For a reliable and timely forecast of Space Weather world-wide networks of particle detectors are located at different latitudes, longitudes and altitudes. To provide better integration of these networks the data acquisition system is facing a challenge to establish reliable data exchange between multiple network nodes which are often located in hardly accessible locations and operated by small research groups. In this article we want to present a data acquisition system for new establishing SEVAN (Space Environmental Viewing and Analysis Network) elaborated on top of free open-source technologies. Our solution is organized as a distributed network of uniform components connected by standard interfaces. The main component is URCS (Unified Readout and Control Server) which controls frontend electronics, collects data and makes preliminary analysis. The URCS operates fully autonomous. Essential characteristics of software components and electronics are remotely controllable via a dynamic web interface, the data is stored locally for certain amount of time and distributed on request to other nodes over web services. To simplify data exchange with collaborating groups we are using an extensible XML based format for data dissemination. The data acquisition system at Aragats Space Environmental Center in Armenia was started November, 2006. Seven particle monitors are located at 2000 and 3200 meters above sea level at a distance of 40 and 60 km from data analysis servers in Yerevan, Armenia. The reliability of the service was proofed by continuous monitoring of incident cosmic ray flux.

1. Introduction

GCR (Galactic Cosmic Rays, mostly protons and heavier nuclei), may be accelerated in our Galaxy by supernova explosions in jets ejected from black holes or by other exotic stellar sources. After traveling millions of light years in our Galaxy they arrive in the solar system as highly isotropic and stable flux. On the other side, our Sun is a very variable object changing radiation and particle flux intensities on many orders of magnitude within a few minutes. Because of Sun's closeness the effects of changing fluxes have a major influence on the earth, including climate, safety and other issues (see for example [1]).

Therefore, the solar flux of cosmic rays can be described as a modulation of the stable galactic cosmic ray "background". The Sun modulates GCR in several ways. The explosive flaring processes on the Sun result in ejection of huge amounts of solar plasma and in acceleration of the copious electrons and ions. These particles constitute, so called, SCR (Solar Cosmic Rays). The SCR reach the earth and initiate secondary elementary particles in the terrestrial atmosphere, increasing the counting rates of particle monitors by several percents. This effect is called ground level enhancement. Other, non-direct solar modulation effects influence also the intensity of GCR. The solar wind "blows out" the lowest energy GCR from the solar system, thus changing the GCR flux intensity inverse proportionally to the Sun activity. The very fast solar wind from the coronal holes, huge magnetized plasma clouds, and shocks initiated by coronal mass ejections are traveling in the interplanetary space and interacting with GCRs. On arrival at the earth the magnetic field of the plasma cloud deplete the GCR, measured as decrease of the secondary cosmic particles (so called Forbush decrease). [2]

Hybrid particle monitors at ASEC (Aragats Space Environmental Center [3, 4]) measure both charged and neutral components of secondary cosmic rays and provide a good coverage of different species of secondary cosmic rays with different energy thresholds. A multivariate correlation analysis of the detected fluxes of charged and neutral particles is used for analysis of geo-effective events, i.e. Ground Level Enhancements, Forbush decreases, Geomagnetic Storms and for reconstruction of the energy spectra of SCR [4]. The particle monitors are located in the two research stations on the slopes of Aragats Mountain at altitudes 2000 and 3200 meters above sea level and are connected with the data analysis center in Yerevan by means of a wide-range radio network. Additionally, there is an ongoing process of establishing a world-wide network of detectors operating at different latitudes, longitudes and altitudes.

The aim of the paper is to describe a new distributed data acquisition system used at ASEC. In the next section an overview of the ADAS (Aragats Data Acquisition System) architecture is given. The third section presents the architecture of readout software in more details. The last section explains the data representation which is used for information storage and exchange.

2. Aragats Data Acquisition System

The ADAS (Aragats Data Acquisition System [5]) is developed having in mind the distributed nature of GCR detection networks often consisting of multiple detectors located in hardly accessible places. The most attention is devoted to the possibility of autonomous operation, error recovery and remote management capabilities. To simplify cooperation of research groups and open a way for integration with other particle detection networks the inter-component communication is released on top of Web Services. The extensible XML based data format is used for data storage and exchange.

The data acquisition system is constructed from uniform components communicating over well-defined interfaces. The main component is called URCS (Unified Readout and Control Server) and takes care for readout of experimental data, preliminary analysis, and distribution of the data to other components. Along with treating of experimental measurements the URCS server provides a set of interfaces for controlling both detector electronics and URCS software behavior. Web Services are used for that purpose. On the basis of these control interfaces the web frontend is constructed. It provides the operator with a full set of remote management capabilities.

In addition to the URCS servers the ADAS incorporates alarm and data storage subsystems which are executed on servers in the main lab. The alarm service is used to issue e-mail notifications about severe conditions of Space Weather or/and electronic failures. The data storage servers are periodically inquiring the data from all URCS servers and storing it in a database on reliable servers in the main lab. Further, the stored data is analyzed by off-line software and made available for the physical analysis by means of DVIN (Data Visualization Interactive Network [6]) interface. The Figure 1 presents the overall system design.

The rest of the section provides detailed information on the various ADAS components.

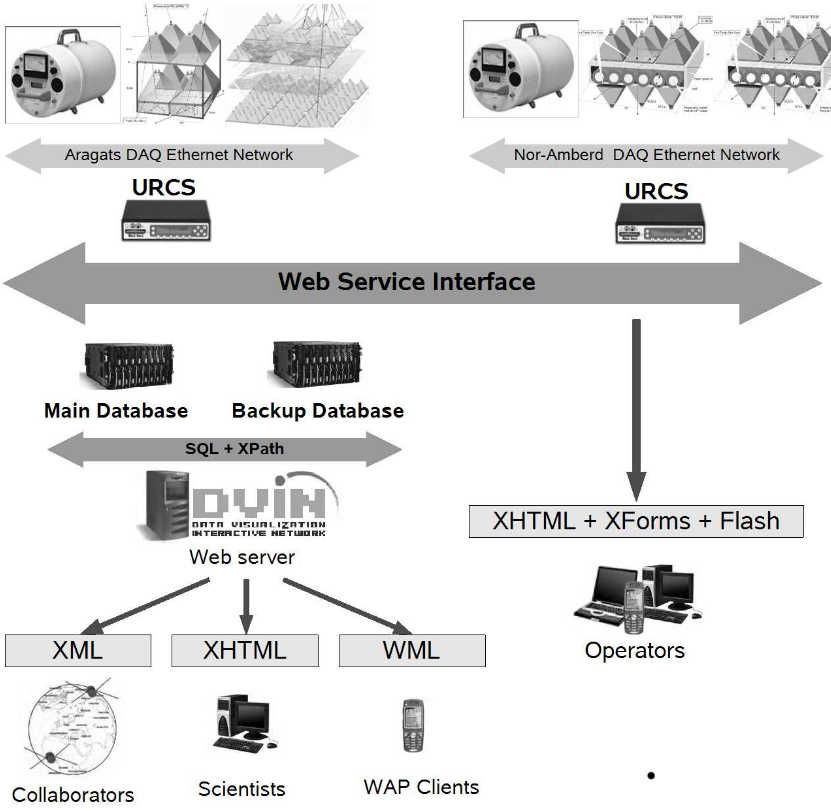

Figure 1. The figure represents a layout of the new ASEC data acquisition system. Several detector arrays are operating at Nor-Amberd and Aragats research stations. The detectors are controlled by URCS which are installed on each station. The data dissemination and detector control is facilitated by web services. The DVIN is used to distribute the data to end users.

2.1. Embedded Software

The readout electronics of ASEC particle monitors is based on ARM7 micro-controllers from Philips. The embedded software is implemented using double buffered client-server architecture. At first the devices are initialized in a dummy mode waiting for the control from the host system. After the initialization request the software starts operation in the standard mode accepting signals from the ADC's (Analog Digital Converter). In order to reduce the amount of data transferred between embedded and host systems the first stage of data processing is performed on the embedded system. The embedded software counts the number of events registered by each of the sensors as well as coincidences/anti-coincidences between preselected sensors. The coincidence information is used to restore the direction of incident particles as well as their type. Optionally the host system may demand creation of various conditional and unconditional spectra of particle energies. The full information on individual events satisfying desired requirements may be also selected.

Embedded software expects what the host system will periodically issue the data retrieval requests. The double buffer architecture is used to relax timing demands. While the current data is prepared in the first buffer, the data of previous integration interval is available from the second one upon a driver request. After the data in the first buffer is ready, the buffers are exchanged.

The data consistency is assured using CRC (Cyclic Redundancy Check) checksums carried along with data. Besides the data retrieval the host system may control various parameters of the underlying electronics.

As a part of startup procedure the host system passes desired configuration (counting intervals, specification of coincidences, and spectra of interest, etc.) to the embedded system, establishes time synchronization and issues initialization command. On each of iterations the time synchronization between host and embedded systems is checked. In the case of minor synchronization errors a time correction procedure is performed. If the error exceeds defined threshold the embedded system is reinitialized and time synchronization is re-obtained.

2.2. Frontend Computers

In order to improve the system stability we are using identical Minibox M100 (VIA Eden: C3 533MHZ, 512 MB RAM [7]) computers at all research stations. The computers are equipped with *Gentoo Linux* and running kernel (2.6 family) optimized for real-time applications. Major advantage of the platform is the complete absence of mechanical parts. The system has passive (fan-less) cooling. Instead of a hard drive, the CF (Compact Flash) memory card is utilized. A small LCD keypad is embedded into the computer case. It provides basic management capabilities and primarily used to represent current system status and to notify operators about critical failures.

The particle monitors are connected to the frontend computers by means of the USB and Ethernet interfaces. The old electronics which is only equipped with UART interface is connected using 1st Mile UART-Ethernet converters.

As it was mentioned above, long distance wireless links are used to connect the research stations with the main lab. Because of extremely long distance and rough weather conditions on the top of Aragats Mountain the quality of connection may considerably vary during operations. Therefore, to ensure reliability at least one frontend computer is dedicated for each research station.

2.3. Software Installation and Upgrade

The usage of CF cards drastically simplifies the process of software installation and upgrade. The installation can be performed on any computer equipped with CF card reader. The setup application asks several questions about the system configuration (Name, IP address, Type of Hardware etc.) and, then, installs software and configuration files on a provided CF disk. Then, upgrade is done by replacing the currently working CF card with a newer one. This operation is very simple and might be performed within several minutes by the technical shift looking after the stations.

2.4. Unified Readout and Control Server

The URCS (Unified Readout and Control Server) server is an autonomous component of the data acquisition network and consists of multiple interacting components. First of all it is a URCS daemon (daemon is a system service in UNIX world) which takes care of communications with underlying electronics. Hiding detector details from other URCS components it provides a uniform way for the detectors monitoring and control. The detailed architecture of the URCS daemon is provided in section 3.

The URCS servers are executed on the frontend computers and are able to operate without connection to the rest of the data acquisition network for long periods of time. To prevent the information loss the collected data is stored on the local Compact Flash card and served to the clients upon request. The amount of time the data remains stored on the server depends on the detector data bandwidth and may be adjusted by the operator.

2.5. URCS Control Interface

The communication with remote components is carried out by means of Web Services running on Apache web server. The data access is well structured. Each underlying particle monitor has own address space and may provide to the clients one or more independent data set. From the specified data set the client applications may request the latest available data or the data for desired historical period. The data channels in all data sets are described by metadata properties. These properties include information on the type of considered particles: charge, energy range, incident direction, etc. The set of properties describing all data sets belonging to a certain particle monitor are collected in the, so called, detector description and are available to the clients upon request. Detailed information about data format and detector descriptions is given in section 4.

2.6. URCS Operator Frontend

By means of the URCS Frontend the operator is able to perform a full range of monitoring and control tasks. It is possible to examine various aspects of the current URCS operation, modify actual configuration, start and stop readout daemons, or access the URCS log files.

The operator is able to browse the data stored on the URCS servers. The current data is presented in a fully annotated fashion using associated detector descriptions. The older data is available in XML, HTML and/or CSV (Comma-Separated Values). The continuous data quality monitoring is feasible by the provided AJAX (Asynchronous JavaScript and XML) interface which is depicting various aspects of the most recent data by means of SVG (Scalar Vector Graphics) charts. Additionally, the metadata properties specify conditions demanding the operator intervention. If certain condition is met the interface will signal an alarm to the operator.

In addition, the web frontend is used to adjust the URCS configuration, including the configuration of underlying electronics. All configurations are expressed in XML terms and, therefore, the XForms (XML Forms) interface is used to provide a control interface. The XForms entries are generated by URCS daemon (see section 3.4) and providing mapping between UI (User Interface) elements and configuration nodes. The XForms engine processes user interactions and submits the altered XML configuration to the URCS server. Because the XForms specification is poorly supported nowadays a FormFaces application is used to provide XForms functionality in XForms incapable browsers [8].

2.7. Alerting Service

In order to detect hardware failures and issue alerts on sudden disturbances of Space Weather a simple monitoring application is running on the server in the main lab. It periodically inquires all registered URCS servers for the current status and the count rates. If a server with inappropriate status is found, a notification e-mail is delivered to the responsible person. The count rates are checked for the limits stated in the detector description and in case if they exceed specified ranges the notification message is sent as well.

In reference [4] a strong relation between inter-detector correlations and transient solar events was demonstrated. Therefore, correlations between time series obtained from the different detectors are also considered to issue notifications.

2.8. Data Storage

The data is stored by means of two interchangeable servers (AMD Athlon X2 4800+, 4 GB memory) working in parallel at the main lab. To ensure data safety the servers are equipped with two 400 GB Serial-ATA hard drives which are organized in a mirroring raid. These servers periodically inquire the data from all URCS servers and store it in a MySQL database. The detector descriptions are separately stored in the same database and made available for the off-line analysis software.

3. Unified Readout and Control Server Daemon

The URCSD (Unified Readout and Control Server Daemon) is the main system component interfacing the detector electronics. It reads the data from detector, makes a preliminary analysis if necessary, and stores it in the desired format on the local file system. Furthermore, it provides monitoring and control interfaces which are establishing a uniform way to access and alter detector configuration, issue driver specific commands, retrieve operation logs. More interfaces may be added in future.

Running on the embedded hardware and communicating with multiple detectors equipped with heterogeneous electronics the URCS daemon takes up a challenge to provide fast and effective architecture minimizing amount of the detector specific code. In order to handle these goals a multi-level abstraction model is laying in the base of system design. The reasonable performance is achieved using effective threading model. The detector drivers are divided in the two pieces. A small, time critical module is executed with a real-time priority. It is used to facilitate communications with hardware and only buffers the responses in memory. The processing of the buffered data is executed with lower priority and performed by second part of the driver. The next subsections provide more information on the URCS architecture and bring details of the abstraction and threading models.

3.1. Abstraction Layers

The architecture of a URCS daemon is based on several levels of abstraction. The abstraction library lies in the deepest level and provides an ability to run data acquisition software under multiple operational systems. Currently, the Windows (NT family) and GNU/Linux systems are supported in both 32 and 64 bit environments. However, support for any POSIX compliant system may be easily added.

The next abstraction layer is built by, so called, *Connections* and *Writers*. The *Connection* abstraction provides a uniform way of accessing underlying electronics and makes it easy to add support for new protocols without applying massive changes to code. The current version of the software supports devices connected through UART, USB and Ethernet interfaces. The *Writer* abstraction provides an ability to save data in the different formats. The multiple ways of bringing data to the client applications is another possibility enabled by *Writer* abstraction. For example, in the future versions of ADAS software the OPC XML-DA (Open Process Control XML Data Access [9]) protocol is supposed for data exchange with commercial control software.

The *Device* abstraction is a top-most component. It encompasses all code which is required to support detectors of certain type. The *Device* code includes the detector driver, configuration parser, preprocessing algorithms and etc. Each of the *Devices* is associated with a single *Connection* to hardware and multiple *Writers* which are disseminating the data. Multiple *Devices* are supported by single URCS server. The list of devices along with their settings is provided by means of the XML configuration.

3.2. Threading Model

The URCS daemon executes multiple readout threads and one thread for processing. For each *Device* one readout thread with highest possible priority is started. It controls the underlying electronics sustaining strict timing demands. The received data is stored it in the intermediate ring buffer. The single buffer is used for all *Devices*.

The processing and dissemination is performed by a single processing thread which is running with lower priority. It processes the ring buffer record by record and for each record executes the processing routine of the appropriate *Device*. Then, upon a request the data is passed to the specified *Writers*. To prevent deadlocking due to *Writer* delays a built-in timeout restricts a maximal amount of time allocated for data storage and dissemination. If processing is not finished in the assigned time slice, the record is postponed and processing of the next record from the buffer is started.

3.3. Detector Network

The communication with the detector electronics is handled by Ethernet interface using UDP or TCP protocols. In order to avoid unauthorized access and data corruption the Ethernet segment connecting to the detectors is isolated from the outer world.

Usually, all operating detectors are listed along with their IP addresses in the URCS configuration. However, it is possible to specify the IP range and default configuration in order to enable device auto detection. In this case the URCS server will probe all IP addresses in the range using discovery command. The identified detectors will be initialized using default configuration.

3.4. Configuration

The XML file is used to specify the initial URCS configuration. During the application start it is parsed and mapped into the server memory using a DOM (Document Object Model) representation. The configuration structure is defined by the XML Schema Description. The current configuration and schema description are used to generate XForms entries which are later used in control interface to provide operators with possibility to make configuration adjustments.

The URCS configuration consists of several parts which are describing all *Connections*, *Writers*, and *Devices* present in the system. These configuration parts are component specific and passed in in-memory DOM representation to the appropriate software components. It is up to them to process the configuration and extract required options. The configuration of *Device* may additionally contain the hardware specific subsection which will be passed by the driver directly to the detector's embedded software.

3.5. Error Handling

The URCS server allows auto-recovery from system failures. In the case of a hardware failure the problem is logged and the controlling driver performs the hardware re-initialization. Most of possible software problems are handled internally. If a non-recoverable error is encountered the software is terminated and an emergency message is written to the log. Later it is automatically restarted by a system service monitoring URCS components.

4. Data Format

The ADAS data is consisting of two components:

(i) Collected data along with several properties characterizing the data (including the data timestamp, data quality, etc).

(ii) The detector description providing detailed information on the detector and collected data.

The detector description consists of three main components: *Global Detector Description*, *Detector Geometry* and *Logical Data Layout*. The *Global Detector Description* provides metadata describing the detector in general (detector name, its type, information on participation in various international detector networks, like Neutron Monitor network [10], contact information of maintaining organization, geographical location). The *Detector Geometry* describes the detector component parts as well as their positions and dimensions. The multiple *Data Layout* sections indicate the physical meaning and acceptable value ranges of the data. The first two components are preliminary filled during the detector setup. The *Data Layout* is automatically generated by the URCS software. However, additional properties still can be specified manually during the setup stage.

The data collected by each of the detectors are divided into one or more independent data sets. Each data set is represented as a sequence of data vectors associated with the acquisition time (time series) and one or more *Data Layout* records in the detector description. The multiple layouts are considered for handling cases when the structure of the data set had been changed during the detector operation. For compatibility reasons the data vectors are encoded using space-delimited ASCII strings. These ASCII strings are enclosed in the XML structure providing basic information about the data and referencing appropriate *Data Layout* section in the detector description. Example:

```
<Data installation="installationid" layout="layoutid">
  <Time>2006-02-25T16:50:00.0000000+04:00</Time>
  <Duration>PT30.0000000S</Duration>
  <Quality>100.00</Quality>
  <Value>1846 2760 1956 1848 1763 </Value>
</Data>
```

This example illustrates the representation of a single data element in the ADAS format. The *installation* and *layout* attributes are referencing the appropriate layout in the detector description. The *Time* and *Duration* elements are indicating the start and duration of the data integration time slice (both the timestamp and duration are encoded using rules defined by *ISO-8601* specification [11]). Special conditions encountered during the data acquisition are described using *Quality* element. Usually, this element indicates hardware failures resulting in partly or completely inaccurate data. The *Value* element holds a data vector in the space delimited ASCII representation.

The data storage subsystem downloads the data from all URCS servers and stores it in the MySQL database. For each data set a separate table is created and for each attribute and element (*installation*, *layout*, *Time*, *Duration*, *Quality*) an individual column is used. All values are represented by individual columns as well. Such mapping allows easy and fast access to the data, while the original XML form could be easily recovered. The description is not transported together with the collected data but available upon request from the URCS servers. However, the collected data and detector description is reconciled in a single document while exchanging data with collaborating groups.

Using the described approach the legacy application can easily extract ASCII strings from the data set and use them in the old fashion. The new applications are considering the XML description in order to extract the appropriate data subset automatically.

5. Conclusion

In this paper a new data acquisition and control system for highly distributed particle detector networks is described. The system has a modular architecture and is designed to work in the distributed environments. The special attention is made to enable the possibility of autonomous operation, error recovery and remote management. To simplify cooperation of research groups and open a way for integration with other particle detection networks the inter-component communication is released on top of Web Services. New extensible self-describing data format is designed for the data storage and exchange. To avoid the limited lifetime of moving mechanical parts the control software is running on embedded computers without disk and fans. Proposed abstraction and threading models provide required performance of the readout software. The time critical drivers facilitating the communication with hardware are executed with a real-time priority. The rest of the software is executed with lower priority. The control and monitoring subsystems are implemented by means of dynamic web interface and are accessible by operators using standard Internet browsers.

Since November 2006 ADAS is in operation at Aragats Space Environmental Center. Seven particle monitors are located at 2000 and 3200 meters above sea level at a distance of 40 and 60 km from data analysis servers in Yerevan, Armenia. After intensive tests the architecture of the system is considered ready for implementation in SEVAN (Space Environmental Viewing and Analysis Network) world-wide particle detector network. As a next step the system will be installed in the host countries of the SEVAN network (in 2008 in Croatia, Bulgaria, and India; in 2009 in Slovakia, Costa-Rica, and Indonesia).

References

[1] Carslaw K S, Harrison R G and Kirkby J 2002 *Sci.* **289** 1732

[2] Munakata K and et al 2000 *J. of Geophys. Res.* **105** 27457

[3] Chilingarian A and et al 2003 *J. of Phys. G* **29** 939

[4] Chilingarian A and et al 2005 *Nucl. Inst. Meth.* D **A543** 483

[5] Chilingarian S 2006 *Universal Data Exchange Solution for Modern Distributed Data Acquisition Systems and Its Implementation for Cosmic Ray Monitor Networks* Ph.D. thesis

[6] Chilingarian A and Yeghikyan A 2005 *Proc. of Int. Symp. on Solar Extreme Events (Nor Amberd, Armenia)* p 483

[7] Mini-box: Pico size computing URL `http://www.mini-box.com/`

[8] Formfaces URL `http://www.formfaces.com/`

[9] Opc xmlda 1.01 specification URL `http://www.opcfoundation.org/`

[10] Morall H, Belov A and Clem J M 2000 *Space Sci. Rev.* **93** 285

[11] International standard iso 8601 URL `http://www.iso.org`