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, Alikhamyan 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 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 the 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 interact 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 this paper is to describe the new distributed data acquisition system and its data representation used also for other applications. In the next section an overview of the ADAS (Aragats Data Acquisition System) architecture is given. The third section presents in more details the architecture of readout software and 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 intercomponent communication is released on top of high level standards. The new extensible XML based data format is used for the data storage and exchange.

The data acquisition system is constructed from uniform autonomous components providing standard web service based control interfaces. The component is called URCS (Unified Readout and Control Server) and takes care for the readout of experimental data of one or more underlying detectors, detector control, preliminary analysis and distribution of the data to other system components. The URCS server is not dependent on any external information and can operate without connection to the rest of the data acquisition network for a long periods of time. To prevent information loss the collected data is stored in a local data storage and distributed to the clients upon request. The dissemination of the data is established by means of the of the Web Services. They provide structured access to the collected data and, therefore, facilitates cooperation with remote system components giving a chance to correlate the obtained data with the data collected by other components of the detector network. Along with treating the experimental measurements the URCS server provides a set of control interfaces for both detector electronics and URCS software behavior. On the basis of these control interfaces the web frontend provides the operator with a full set of remote management capabilities.

In addition to the URCS servers the ADAS incorporates alarm services, data storage subsystems running 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. And 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.

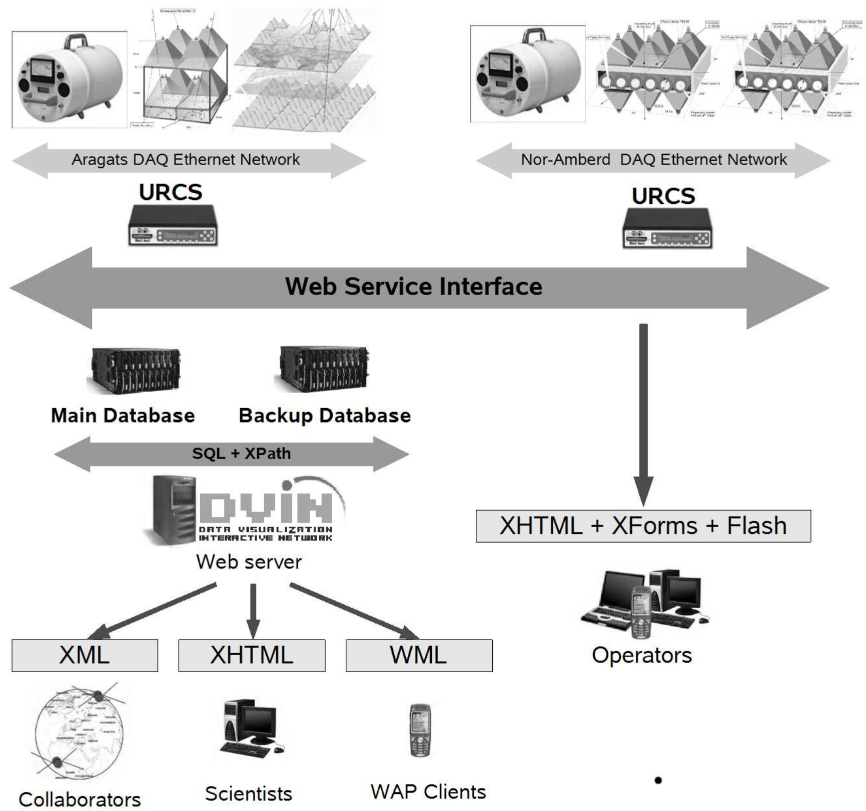

Figure 1. The figure represents a layout of the new ASEC data acquisition system. Several detector arrays are operating at Nor-Amberd and Aragats research stations. The detectors are controlled by URCS which are installed on each station. The data dissemination and detector control is facilated by web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

2.1. Embedded Software

The readout electronics of ASEC particle monitor is based on ARM7 micro-controllers from Philips. The embedded software is implemented using double buffer 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 on each of the detectors as well as coincidences/anti-coincidences between preselected detectors. 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 detected by each of the detectors. The full information on individual events satisfying desired requirements may be also selected.

Embedded software expects what the host system is periodically issuing requests in order to retrieve the data. The double buffer architecture is used to relax timing demands. While the current data is prepared in the first buffer, the data of previous operation is available from the second one upon a driver request. After the data in the first buffer is ready, the buffers are switched. The data consistency is assured using CRC16 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, spectra being of interest, etc.) to the embedded system, establishes time synchronization and issues initialization command. On each iteration 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 thresholds 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 the same Minibox M100 (VIA C3 533MHZ, 512 MB RAM [7]) computers based on VIA Eden platform at all research stations. The computers are equipped with Gentoo Linux based operating system which is used in conjunction with the 2.6 family Linux kernel optimized for the real-time applications. The major advantage of the platform is the complete absence of any mechanical part. The system has passive (fanless) cooling. Instead of a hard drive, the CF (Compact Flash) memory card is used. A small LCD keypad embedded into the computers is used to represent current system status, notify operators about critical failures and provides a way for basic system management.

The particle monitors are connected to the frontend computers by means of the USB and Ethernet interfaces. The old electronics operating over standard 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. Therefore, in order to ensure reliability independently from the failures of connection to main lab a single frontend computer is dedicated for each research station.

2.3. Unified Readout and Control Server

The URCS (Unified Readout and Control Server) server is a completely autonomous component of the data acquisition network. It is operating on the frontend computers and used for detector control and data readout. It is not dependent on any external information and can operate without connection to the rest of the data acquisition network for a 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 requests. The amount of time the data remains stored on the server depends on the detector data bandwidth and may be controlled by the operator.

The URCS server is a complex software consisting of multiple interacting components. In the first place it is a URCS daemon - a readout software which take care of communication with underlying electronics. The communication is performed by means of dedicated drivers while most of the software is the same for all supported detectors. The daemon reads the data from underlying detectors, makes preliminary analysis if necessary and stores it in the files on the local file system. Furthermore, it hides detector access details from other URCS components providing a uniform way for the detector monitoring and control. The detailed architecture of the URCS daemon is provided in the next section.

2.4. URCS Control Interface

The communication with remote components are carried out by means of web services running on a Apache web server. These web services hide the details of the URCS daemon providing a very simple interfaces for both the data dissemination and control capabilities to the external world. The data access is well structured. Each underlying particle monitor has its own address space and may provide to end client one or more independent data sets. The data channels in all data sets are described by metadata properties. These properties may include information on the registered particle types, particle energy range, particle coming direction, etc. The set of properties describing all data sets belonging to a certain particle monitor are collected in the detector description and are available to the clients upon a request as well. The information of the detector description is partly filled by the physicists during the detector setup and partly generated by the URCS software in run time.

The client applications are able to request the latest data from the desired data set or the data for certain historical periods.

2.5. URCS Operator Frontend

The operator web frontend is a URCS component providing a uniform way for the remote control of URCS servers and underlying electronics. By means of the interface the operator is able to perform a full range of monitoring and control operations. It is possible to view various aspects of the current URCS operation, modify actual configurations, start and stop readout daemons or access the URCS log files for the desired period.

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 historical data is available in XML, HTML and/or CSV (Comma-Separated Values) forms. 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, metadata properties specify special conditions demanding the operator intervention. If such condition is met the interface will signal an alarm to the operator.

The web frontend is used as well to control the URCS configuration, including configurations of the underlying electronic devices. All system configurations are expressed in XML terms and, therefore, the uniform XForms (XML Forms) based interface is used to control all detectors and URCS itself. For each detector only the XForms representation providing mapping between UI (User Interface) elements and XML nodes in configuration is specified. The XForms engine processes user interactions and submits the altered XML configuration to the URCS server. The URCS server processes global configuration options and passes the individual information for the URCS drivers. To provide XForms functionality in XForms incapable browsers extra application ("FormFaces") is used.

2.6. URCS Installation and Upgrade

The usage of CF (Compact Flash) cards drastically simplifies the software installation and upgrade. The installation can be performed on any computer equiped with CF card reader facility. The installation software asks several questions on the URCS configuration (Name, IP address, Type of Hardware etc.) and, then, install all required system files, URCS software and configuration files on a provided CF disk.

To upgrade URCS software on the running system it is only needed to replace current CF card with a newer one. This operation is very simple and can be performed by the technical shift looking after the stations.

2.7. Alerting Service
In order to detect automatically 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. In case of inaccessible servers or servers with inappropriate status, 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 too.

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 used to issue notifications as well.

2.8. Data Storage
The data is stored by means of two powerful servers working in parallel at the main lab. The dual-core AMD Athlon X2 4800+ systems equipped with 4GB of memory and two Serial-ATA 400GB hard drives are used. The hard drives are as a mirroring raid configured. 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 maid 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 slow embedded hardware selected as a platform for frontend computers and communicating with multiple detectors equiped 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 the architecture of the URCS daemon is based on a multi-level abstraction model. The performance problem is handled by dividing the detector drivers in two pieces. The tiny time critical module is executed with real-time priority and used only to facilitate the communication with hardware and to buffer the responses in the memory while the main part of the driver is executed with lower priority and used to process the buffered data. The next subsections provide details on the abstraction and threading models as well as giving more details about the architecture.

3.1. Abstraction Layers
The core of a URCS daemon layout consists of several levels of abstraction. The abstraction library lies in the deepest level, providing an ability to run data acquisition software under multiple operational systems. Currently, the Windows NT family and Linux systems are fully supported in both 32 and 64 bit environments. However, support for any POSIX complaint 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 data acquisition electronics and makes it easy to support new protocols without massive changes in the 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 this abstraction layer. This possibility is assumed to establish connectivity with 3rd party control software by means of standard slow control protocols, like OPC XML-DA (Open Process Control XML Data Access [8]). At the moment the data is stored locally in files.

The *Device* abstraction is a top most system component. It is used to get the data from the connected hardware, perform required preprocessing and store it. Each of the *Devices* is associated with a single *Connection* to communicate with underlying electronics but multiple *Writers* disseminating the data.

Multiple *Devices* are supported from one URCS server. The list of devices along with their settings is provided by the system integrator by means of the XML configuration file.

3.2. Threading Model

The URCS daemon uses multiple readout threads and one for processing. For each *Device* the thread with highest system priority is executed. This thread controls the underlying data acquisition electronics sustaining strict timing demands. It also polls the electronics for data and stores it in the intermediate ring buffer. The single buffer is used for all *Devices*. The reference to the device delivering the data is stored along with the data.

The processing and dissemination is performed by a single thread running with lower priority. It processes the ring buffer record by record and for each record executes the processing routine of the appropriate *Device*. It passes as well the data to the appropriate *Writers* upon the request from Device processing routines. In order to prevent overall server hang-ups due to *Writer* delays a built-in timeout restricts a maximal amount of time to data storage and dissemination. If the 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 URCS daemon communicates with the data acquisition electronics using a Ethernet interface by means of UDP or TCP protocols. The Ethernet segment connecting to the detectors is isolated from the outer world in order to avoid unauthorized access and data corruption. Only a frontend computer running the URCS server is connected to this segment.

In order to obtain the network address the connected devices are issuing a discovery request by means of the DHCP (Dynamic Host Configuration Protocol). The frontend computer accepts the request and assigns an IP address from the dedicated pool.

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

3.4. Configuration

The operation of a device is controlled by means of an XML configuration. The configuration is initialized from the supported URCS configuration file and mapped into the server memory using a DOM (Document Object Model) representation. The operators are able to adjust configuration using provided web interface.

The configuration structure is completely device specific. The DOM in-memory is passed to the device driver. It is up to the driver to process configuration and extract required information. The device configuration consists of several parts. One part is controlling the driver operation. It includes connection properties (type, address, timeouts), a list of writers to use for the data storage, properties of the data preprocessing algorithms and etc. Another part controls the detector hardware operation and is passed by the driver to the detector's embedded software. The configuration structure is described by the XML Schema Description. Both the current configuration and this schema description are used to generate XForms entries providing control interfaces.

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 daemon leave a emergency message in the log. In the last case it would be automatically restarted by a special system daemon which is monitoring status of all URCS components and restarting them in the case of a detected problem.

4. Data Format

To simplify cooperation of research groups, to enable data processing automation and to open a way for integration with other particle detection networks the the new extensible XML based data format is invented for the data storage and exchange. Maintaining compatibility with the data analysis tools developed for older data acquisition system it provides extensible detector description.

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 standard metadata describing the whole detector. It contains detector name, it's type, information on participation in various international detector networks (like Neutron Monitor network [9]), contact information of maintaining organization, geographical location, etc. 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 values. The first two components are preliminary filled during the detector setup and the data layout is automatically generated by the URCS software. Still additional properties may be specified manually on 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 layout records are considered to handle cases when the data set structure is changing during the detector operation. In a way similar to one used by the old data acquisition system at Aragats the data is represented by "space" delimited ASCII strings. These ASCII strings are enclosed in the XML structure providing basic information about the enclosed data and referencing appropriate *Data Layout* sections of the detector description with the information on the values meaning. 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 by 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 end and duration of the data integration time slice (both the timestamp and duration are represented following the encoding rules defined by the *ISO-8601* specification). 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 in the main lab downloads the data from the URCS server 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 will be 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 can be reconciled in a single document for data exchange 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 from the data set.

5. Conclusion

In this paper a new data acquisition and control system for highly distributed particle detecting networks is proposed. The system has a modular layered architecture and is designed to work in the distributed environments. 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 intercomponent communication is released on top of the web services. New extensible self-describing data format is invented for the data storage and exchange. To achieve better reliability the control software is running on embedded computers without disk and fans to avoid the limited lifetime of moving mechanical parts. The optimal performance is achieved by a multi-level abstraction model of the readout software. The tiny time critical hardware drivers are executed with a real-time priority and used to facilitate the communication with hardware. The main part of the software is executed with lower priority. The control and monitoring subsystem is implemented using AJAX based dynamic web interfaces and is available to the operators by use of Internet browsers.

Since November 2006 ADAS is implemented at ASEC on the two research stations on the slopes of Aragats Mountain. After 10 months of operation the architecture of the system is ready for implementation in SEVAN world-wide particle detector network. In the next step the SEVAN network will be installed at Croatia, India 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] URL `http://www.mini-box.com/`
[8] Opc xmlda 1.01 specification URL `http://www.opcfoundation.org/`
[9] Morall H, Belov A and Clem J M 2000 *Space Sci. Rev.* **93** 285