Frascati Physics Series Vol. XXXVII (2004) pp. 95–100
Frontier Science 2004, Physics and Astrophysics in Space
Frascati, 14-19 June, 2004

YODA++: A PROPOSAL FOR A SEMIAUTOMATIC SPACE MISSION CONTROL

M. Casolino[a], M. P. De Pascale[a], M. Nagni[a,b], P. Picozza[a]

[a] *INFN, Sezione di Roma II - Dipartimento di Fisica, Università di Roma "Tor Vergata", Via della Ricerca Scientifica, Roma, Italy*

[b] *corresponding author, nagni@roma2.infn.it*

Abstract

YODA++ is a proposal for a semi-automated data handling and analysis system for PAMELA (PA) space experiment. The core of the routines have been developed to process raw data downlinked from the Resurs DK1 satellite (housing PAMELA) to the ground station in Moscow. Raw data consist of the scientific data and are complemented by housekeeping and orbital information. Housekeeping information will be analyzed in short time (1hour) from download to monitor the status of the experiment and to assist in mission planning and acquisition configuration. A prototype for the visualization of the data will run on a TOMCAT web application server, allowing an off-line analysis tool using just a browser and taking advantage of part of already written code for the maintenance system. Development of data retrieving is in production phase, while a GUI interface for human monitoring, is on preliminary phase as well as a JSP/JSF web application facility. On a longer timescale (1-3 hours from download) scientific data are analyzed, indexed and stored in ROOT files for further calibration and processing. YODA++ is currently being used in the integration and testing on ground of Pamela data.

1 Introduction

The main objectives of PAMELA are the accurate measurements of the antiproton and positron fluxes, with a sensitivity and statistics out of the reach of balloon-borne experiments. The energy range goes from below 100 MeV to above 100 GeV and the search of antihelium with a sensitivity better than 10^{-7} in antihelium to helium ratio[1]. The PAMELA telescope will be installed on board of the Russian Resurs DK-1 satellite and will be launched in the year 2005. The expected data flow of the experiment will be 20 GByte/day, divided in four different downlink sessions over two stations. Data quality and housekeeping information will need to be analyzed in short time for the mission planning of the experiment. The software running on the on-board CPU, has been developed using a realtime operative system software [2, 3]. The acquisition of housekeeping data was designed in order to have an efficient control system on ground too, to recognize and correct anomalies. After the quicklook phase, another requirement to be met is to share data through the collaboration as soon as are available in a file/data format widely used and as much flexible as possible. In this spirit we used the CERN's ROOT framework[4] either to develop a custom analysis tool and to store data in a worldwide used file format.

2 Yoda evironment: General scheme

YODA in an acronym for Your Own Data Analysis. It is a system designed to store and retrieve data allowing users to perform analysis at different levels with different tools. At the first step it handles data processing for the Ground Segment, namely:

- Statistics on raw data to determine quality of transmission;
- Statistics on inner instrument packet data to determine instrument status;
- High flexibility to minimize reprocessing caused by unforeseen situations;
- Storage of preanalyzed data in such a way to allow an easy successive access either to telemetry and physical data;
- Ground infrastructure as simple as possible to minimize maintenance costs;
- Data have to be available worldwide through internet with the smallest delay possible;

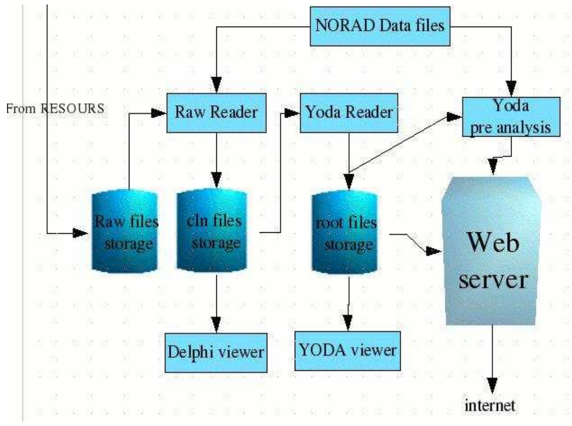

Figure 1: Left: The data flow from downlink to distribution. It is possible to see the three main processing steps and their dependencies.

The data have to pass a three-step process (see Figure 1) before the physics data can be considered qualified for analysis. Each of these steps is governed by a specific tool; in the first step the RawReader (RR) collects the data from the receiving station and process it in order to estimate the quality of transmission and to remove the headers introduced by the inner satellite transmission protocol. The first step produces intermediate files used the second step by the YodaReader (YR) which has to check if the stream satisfies the Pamela protocol in order to extract the several types of packets defined in the Pamela transmission protocol. The third step represents the physics data analysis.
The first two steps of data processing are used to recognize peculiar situations not managed by PA onboard software or to check response to earth previously transmitted commands.

3 Satellite data colection: from satellite to ground

The mean expected amount of particle data information from Pamela is about 2 Gbyte per a day; the expected background from false triggers coming from secondary particles produced in the main body of the satellite is about 18 Gbyte. The information is stored in memory device of the satellite Resurs-DK1 and can be transmitted to ground by portions in several downlink sessions.
The receiving antenna system TNA-7D has a parabolic reflector of 7 m in diameter and azimuth-elevation fulcrum-rotating mechanism and has two frequency diverged radio channels. Pamela data reception is performed at

Digital Processing Data System (DPDS), which is the component of Moscow Ground station at NTSOMZ institute with high-rate disk recording and transmission system. The information from DPDS enters to operational data set archive server. This server provides security connection with the ground segment of Pamela where first analysis takes place.

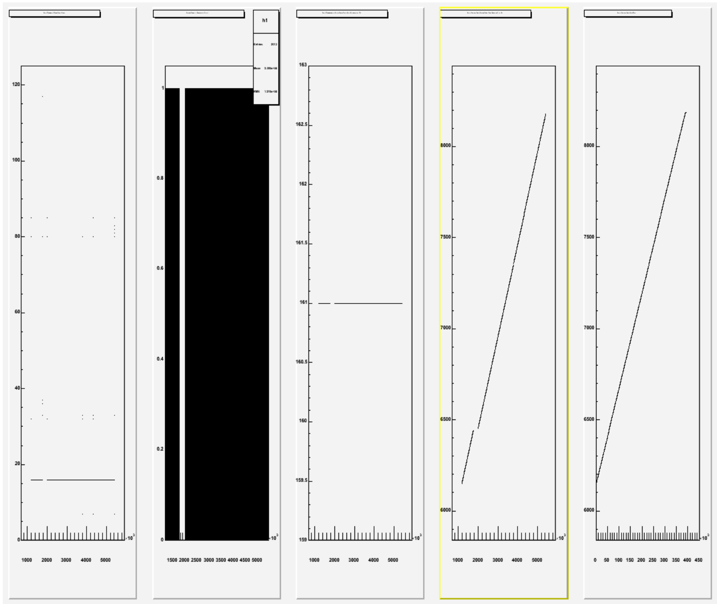

Figure 2: From left to right: 1) A typical time distribution for packet type vs. time (ms). Note the line along the Y=16 representing Physics Data Packet and the pause around t=2000ms due to end of data taking. 2) PhysicalEvent packet rate (Hz). In this case the acquisition was fixed at one event/second with an forced trigger 3) PhysicalEvent packet lenght (bytes). In this readout only some detectors are read, giving a total packet length of 161 bytes. 4) PhysicalEvent packet counter. Note the pause in the counter when the acquisition is stopped. 5) Packet Counter vs position in the downlinked file.

4 The YODA reader

For our specific needs the objects composing the YR tool have been modelled over the ROOT libraries to generate a corresponding file structure. C++ was used as the main programming language.
In general we have four types of objects:

- **packet types objects**: these objects wrap both, the specific data produced by the single devices (TrackerPacket, CalorimeterPacket, etc...) or by the CPU controlled procedures (several telemetry packets);
- **algorithms type objects**: meanwhile the packets could be considered as a static structure, that is just like wrappers, these classes (TrackerReader, CalorimeterReader, several telemetries reader) are the dynamical conterpart which knows the rules about how to read and check the data of each specific packet from the raw data (that is the files generated from the Raw Reader);
- **a class containing utility methods**: this class (PamelaRun) and the derived ones (SpecificModelPamelaRun) represents an interface between YODA and the ROOT objects and structures (TFiles and TTrees);
- **an additional facade class**: inside it, is defined an user interface command line and in the end is where the main reside.

The packet types are structured around a protocol wrapper, called *Header* which is meant to define general characteristic of the specific data inside it.
Each packet uses its own specific reader; each reader is a derivation of a common parent for all the various readers: the *Algorithm* object. We want to stress how the flexibility of this approach have made possible to insert in the reader classes some FORTRAN routines; at the same time specific interfaces are in a developing phase to allow the same group to be able to read ROOT files inside their FORTRAN programs.
The facade class is nothing more than an interface needed to organize the interaction with the user command line or the graphical user interface (GUI) whatever it should be available.
In conclusion our code can be represented as a the action of the EventReader searching in the stream of data for the Header packet signature. The stream is then passed to the HeaderReader which will check if the various parameter defined inside the actual stream are coherent with the header parameters (PacketLength, Counter, OBT, CRC,etc....); if several constrains are verified, the EventReader will call the specific Reader according to the PacketId in the header as mapped previously and in the end, through the PamelaRun class, the EventReader will store the read packet and the relative header.
Once the data are extracted the files can be managed using the ROOT framework either by direct interaction with the root files or using specific scripts taking advantage of the CINTerpreter (CINT). The collection of all these scripts constitutes the so called *YODA Viewer*, which can be either a stand-alone GUI (a prototype has already been implemented using the Signal-Slot features of the ROOT GUI libraries on a Model-View-Controller architecture) but it can be a web application too, extracting the graphs from the same application serving

the internet users; the main reason for this is because to minimize the code to write; the YODA viewer shows the most meaningful information about the downlinked data and in the end about anomalies; this can be done using some ROOT batch scripts activated by files modification; also if in a preliminary phase is being developed an on-demand web application (WA) running on a TOMCAT server (wrote using JSP/JSF specifications).

References

[1] Simon, M., XXVIII ICRC, OG 1.5, 2117 Tsukuba, Japan, 2003.

[2] Straumann, T. Open Source Real-Time Operating Systems Overview, "8th International Conference on Accelerator and Large Experimental Physics Control Systems", San Jose, USA, 2001.

[3] F. Altamura, M. Casolino, et al., "The Pamela CPU", these proceedings.

[4] ROOT - An Object Oriented Data Analysis Framework, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. and Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.