

## PESCA instrument control system

M. Prieto<sup>1</sup>, MD. R-Moreno<sup>1</sup>, S. Seiseddos<sup>1</sup>, C. Martín<sup>1</sup>, J. E. Roza<sup>1</sup>, S. Sanchez<sup>1</sup>, Meziat<sup>1</sup>, J. Medina<sup>2</sup>, E. Bronchalo<sup>2</sup>, C. Olalla<sup>2</sup>, and E. Campo<sup>1</sup>

<sup>1</sup>Departamento de Automática. Universidad de Alcalá. Ctra. Madrid-Barcelona km. 31,600. 28871 Alcalá de Henares. Spain

<sup>1</sup>Departamento de Física. Universidad de Alcalá. Ctra. Madrid-Barcelona km. 33,600. 28871 Alcalá de Henares. Spain

**Abstract.** The description of the PESCA instrument control system is presented. The PESCA instrument has been designed and built with the purpose of studying the Solar Energetic Particles and the Anomalous Cosmic Rays from helium to iron in the energy range 1.5-50 MeV/uma and will be part of the Russian PHOTON satellite payload. The instrument comprises two different blocks: the PESCA Instrument Amplification and Shaping Electronics (PIASE), for the amplification and analog to digital conversion, and the PESCA Instrument Control and Acquisition System (PICAS), for the control of the whole instrument. A system has been mounted to control and test the PESCA instrument. It allows the complete control over the instrument, verification and validation of the PESCA instrument (unit tests, integration tests and system tests) and PESCA calibration, from a single PC under Linux operating system.

and, finally, a PC computer with the Electrical Ground Support Equipment (EGSE) software.

The PESCA instrument (del Peral et al., 1995; del Peral et al., 1997; Prieto et al., 1999) is composed of a telescope, made up of four silicon ion implanted detectors, which register the particle crossing through the detectors, the PIASE for signal amplification and analog to digital conversion, and the PICAS, for the control of the whole instrument. PICAS is composed of a Data Processing Unit based on the MAS281 microprocessor and a set of interfaces with the rest of the instrument and the satellite. PICAS admits telecommands sent from ground for data acquisition and control parameter setting. The communication between ground and PICAS is performed through the OBDH.

The OBDH emulator acts as the satellite on board computer and it fulfills the OBDH ESA standard. The communication between OBDH emulator and EGSE is performed through two serial RS232 ports, one for telecommands and one for telemetry.

In the other hand, the EGSE software, developed for Linux operating system, allows the complete control over the instrument, being the end point of the system. It basically allows the telecommand sending and data reception, making possible the verification, validation and calibration of the PESCA instrument.

### 1 Introduction

The control system that has been mounted in the laboratory is depicted in figure 1. It is composed of the PESCA instrument, an On Board Data Handling (OBDH) Emulator

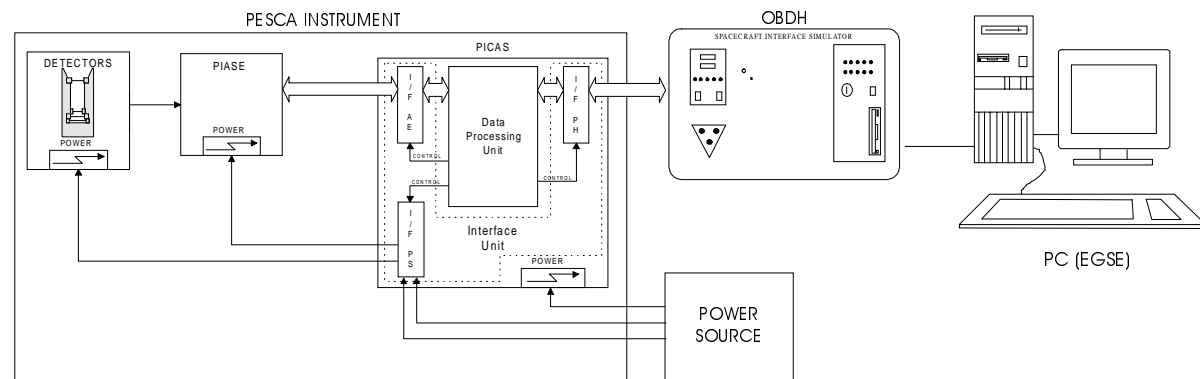


Fig. 1. Control system diagram

Correspondence to: M.Prieto (mpm@aut.alcala.es)

## 2 EGSE features

As stated in the introduction, EGSE is a program that runs under Linux. This operating system provides relevant advantages such as stability, multiuser, real time operation, open source and GNU LGPL. EGSE has been tested under Red Hat v6.1 and it has been developed using GTK+, GLib and GTK+ Extra ([www.gtk.org](http://www.gtk.org); [gtkextra.sourceforge.net](http://gtkextra.sourceforge.net)). Figure 2 shows the EGSE v0.1.0 main window.

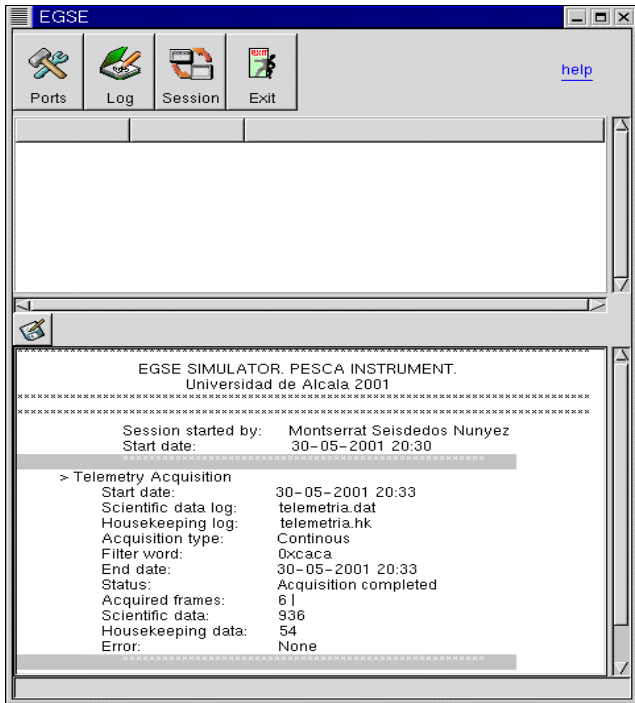


Fig. 2. EGSE main window

EGSE main features are:

- Telecommand sending. EGSE allows telecommand sending to both, OBDH and the PESCA instrument. By means of these commands, operator can act over the PESCA instrument, changing the data acquisition parameters and other PESCA operation states. These commands are commands such as detectors power on/off, coincidence/anticoincidence mode setting, threshold setting, etc. Through telecommand sending PESCA operation is tested.
- Telemetry reception. Telemetry consists of two kind of data: status data (housekeeping data) and scientific data. Through the housekeeping data, user may know the PESCA instrument status at any time (detector status, for instance) and therefore may act in consequence, for example in case of failure detection. EGSE scheduling, described in point 2.4, is based on housekeeping data. In the other hand, scientific data contain the interest data, purpose of the instrument. Scientific data contain the energy lost in the detectors.
- Scientific Data processing. By means of EGSE, operator can make some processing over the received

scientific data. This processing includes, among others, real time data visualization, data representation in different modes and data export to different formats.

- Scheduling. EGSE has an autonomous operation mode that avoids human supervision. In this mode, EGSE continuously checks housekeeping data and, by means of a planner, takes the specific actions to react against possible on flight failures.
- Network operation. Thanks to Linux Operating System, any authorized user can execute EGSE from any other computer connected to the Internet, that is, it is not necessary to be physically in the laboratory to operate PESCA.

### 2.1 Telecommand section

Figure 3 shows the telecommand window. From this window the user can send telecommands to the OBDH emulator or to the instrument itself. The communication with the OBDH emulator is performed through two RS232 serial ports completely configurable from the EGSE main window. Telecommands can be sent through a command line, provided by the graphical interface, or through a list control that displays the telecommands listed in a user defined file. This ASCII file, easily editable from a common text editor, contains the telecommand identifiers and the telecommand data bytes (header, body and terminator) to be sent through the serial port. The default file is the one for the PESCA instrument. The telecommand list can be modified, as well, through the control buttons in the left side of the window. This gives versatility to EGSE because, in this way, it is not restricted to a fixed set of telecommands of a given telecommand format, and therefore, it is not restricted to a specific instrument.

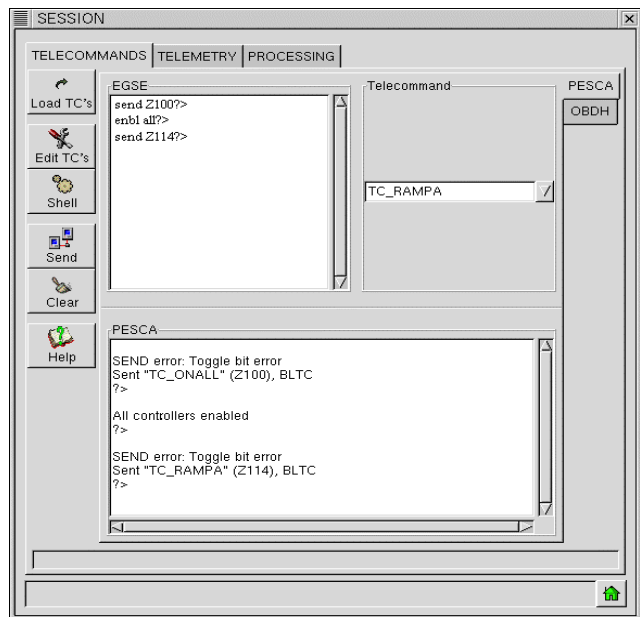


Fig. 3. EGSE telecommand window

For every telecommand sent, the OBDH response is shown in order to know if the telecommand has been successfully delivered. EGSE allows to program the telecommand sending in time (Time Tag Telecommands). From a menu option, the user can program a list of telecommands to be sent in a specific time and date. This feature allows to program a telecommand sequence for delivery.

All telecommands issued and OBDH responses are registered in a log file for later inspection. Log files are stored in ASCII format, legible from any text editor.

## 2.2 Telemetry section

Figure 4 depicts the window for telemetry data acquisition. From the options provided, the user can select the number of data packets to collect. The telemetry data are received according to the loaded telemetry profile and are usually divided into housekeeping data and scientific data. Both data types are stored in different files in raw binary format. Telemetry profiles are user defined, allowing the EGSE configuration for other instruments. During the data acquisition, an integrity check is performed and bad data are discarded. EGSE also permits continuous acquisition, storing the corresponding data files.

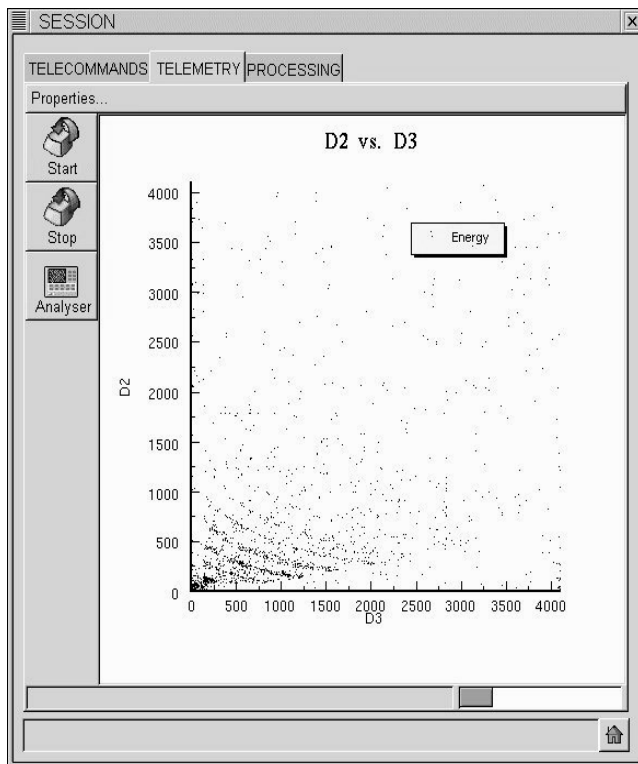


Fig. 4. Telemetry window

All the acquisition process is saved in a log file for later inspection, including the starting time and date of the acquisition, the data packets received, errors detected, etc.

## 2.3 Data processing

After the data acquisition, both housekeeping data and scientific data may be processed through the data processing option. Housekeeping data contains the status data provided by the instrument according to the telemetry profile specified. These status data are useful to know if an error has occurred during the flight and are specific of the instrument.

Scientific data may be displayed in an histogram form, transforming EGSE in a multichannel analyser, or representing the data in two axes, for example like the one shown in figure 4, where the collected data from detector 2 is represented versus detector 3 data.

As stated in previous section, all the data received are stored in binary format. In order to allow the data export to other data processing programs, EGSE allows the data conversion into ASCII format. Figure 5 shows the scientific data displayed in columns. Housekeeping data can also be transformed to ASCII.

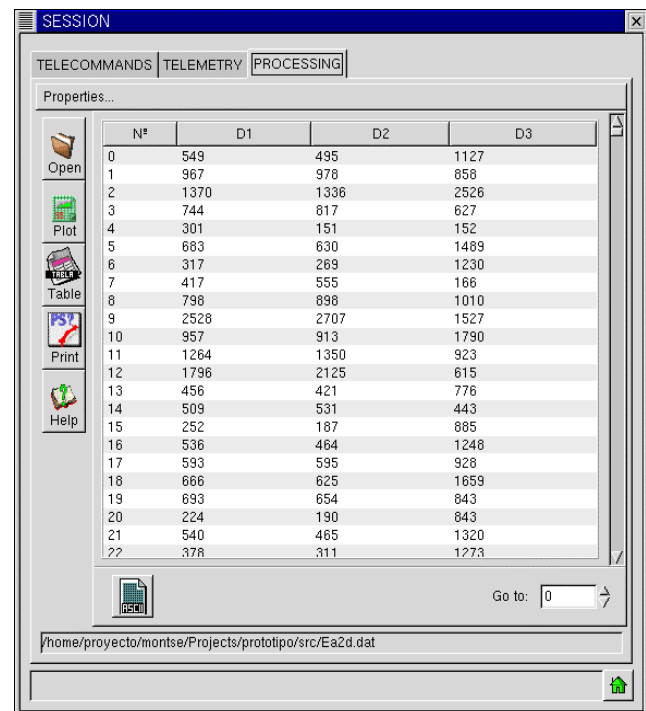


Fig. 5. Data processing window

## 2.4 Planning

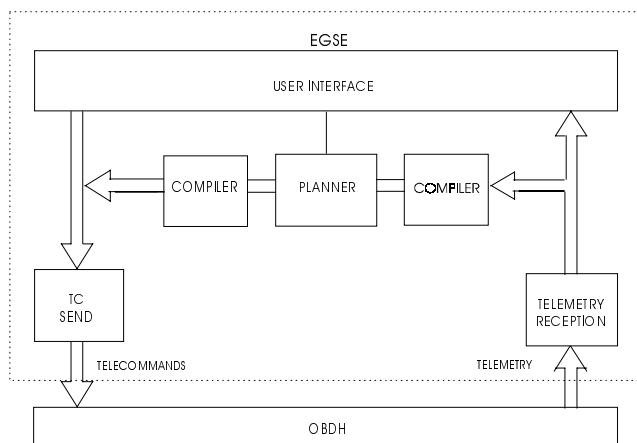
The EGSE program also has the capability of working in an autonomous mode thanks to the integration of AI planning techniques. We have used the deliberative independent planner PRODIGY (Velooso et al,1995) for this purpose. PRODIGY is an integrated intelligent architecture that has been used in a wide variety of domains: from artificial domains to real applications as logistics or robotics. The problem solver is a non-linear planner that uses a backward chaining search procedure with full sub-goal interleaving,

that is, the planning process starts from the goals and adds operators to the plan until all the goals are satisfied.

The following inputs have to be provided to the planner:

- **Actions:** are represented by telecommands, also called operators in planning. An operator consists of preconditions (conditions that must be true to allow the action execution), and post-conditions or effects (usually constituted of an add list and a delete list). The add list specifies the set of logical formulae that are true in the resulting state while the delete list specifies the set of logical formulae that are no longer true and must be deleted from the description of the state. Each EGSE telecommand (such as *turning on* the PICAS) is translated into a PRODIGY operator.
- **Initial state:** in planning, one has to specify the starting situation of the posed problem. In the case of this domain, the initial state would be all knowledge that EGSE has about PESCA. Examples of initial states would be the state of each of the PESCA components before the telemetry acquisition starts.
- **Goals:** are often viewed as specifications for a plan. They describe the successful behaviours that execution of the plan should produce: specify what one would like to be true at the end of the solution of the problem. For instance, our goal would be to receive scientific data without interruption.

As outputs, AI planners generate a plan or set of plans if a solution exists. A plan can be seen as a sequence of operator applications that can lead from the initial state to a state in which the goals are reached.



**Fig. 6. EGSE-planner architecture**

When EGSE is in “auto” mode, is able to send telecommands without human supervision as a consequence of the status data received during the telemetry reception process. Every time a telecommand is sent following the *plan* generated by PRODIGY, an exhaustive check of the operator effects is performed in order to detect differences between the operator effects and the status data received

after the telecommand is sent. If any difference is found, a re-planning process starts (that is, the generation of a new plan). If a solution cannot be obtained, the program will abort and a log file will be generated with all the decision made by the planner. Figure 6 shows the architecture that integrates EGSE and the PRODIGY planner.

For example, one of the possible actions in the plan if one wants to receive scientific data will be to turn on the detectors. If in this process, the program realise that the detector is still off, the planner will generate a new plan. In this new plan, one of the actions will be to send the same telecommand three more times (sometimes the problem is not due to a detector damage but a bad transmission/reception). If this fails, a new plan is generated to find a different mode of acquisition but if no valid solution can be obtained, the program will abort and a log file will be generated with dates and decisions made by the planner. It is also possible, when EGSE is running in this mode, that the user interacts or inhibits the planner decision.

### 3 Conclusions

A programme for the calibration, verification and validation of the PESCA instrument has been presented. The software has been successfully proved in the laboratory, and it has become a key part in PESCA testing. These tests are essential for the integration with PHOTON satellite. EGSE also has a great versatility, and it may be used with other instruments, modifying several parameters.

*Acknowledgements.* This work has been supported by Spanish CICYT (grant ESP99-1066-C02), by the European Community-Access to Research Infrastructure action of the Improving Human Potential Programme, No HPRI-CT1999-00019 and by the NATO grant PST.CLG.977003.

### References

- del Peral, L., Medina, J., Sánchez, S., Bronchalo, E., Rodríguez Pacheco, J., Sequeiros, J., and Meziat, D. *Nuclear Instruments and Methods in Physics Research*, A354, 539-546, 1995.
- del Peral, L., Bronchalo, E., Medina, J., Rodríguez Frías, M.D., Sánchez, S., and Meziat, D. *IEEE Transaction on Nuclear Science*, 44, 1442-1447, 1997.
- Prieto, M., Martín, C., Quesada, M., Meziat, D., Medina, J., Sánchez, S., and Rodríguez-Frías, M.D. *Nuclear Instruments and Methods in Physics Research*, A443, 264-276, 1999.
- Veloso M., Carbonell J., Perez A., Borrajo D., Fink E., and Blythe J. *Integrating planning and learning: The PRODIGY architecture*. *Journal of Experimental and Theoretical AI*, 7: 81-120, 1995.