BUILDING A TELECOMMAND WEB SYSTEM FOR SCIENTIFIC MICROSATELLITE CONTROL BASED ON OPEN SOURCE TECHNOLOGY

Romualdo Alves PEREIRA JÚNIOR [1]
Maria de Fátima Mattiello FRANCISCO [2]
Manoel Jozeane Mafra de CARVALHO [1]

[1] Instituto Nacional de Pesquisas Espaciais – Centro Regional de Natal (INPE/CRN)
[2] Instituto Nacional de Pesquisas Espaciais – Divisão de Sistemas em Solo (INPE/DSS)

ABSTRACT *- The scientific microsatellite platforms are moving towards operational independence among their scientific payloads. This paper highlights how SACI (Brazilian Scientific Application Satellite) project took on this challenge in terms of ground operation system. A scientific investigator support in controlling his/her own experiment was provided using Internet facilities. During satellite lifetime, the users are allowed to edit, organize and schedule remotely their Telecommands via WEB pages based on predict passages.*

Keywords: Flight Plan, autonomous operational routine, clent-server web-based system

1 - INTRODUCTION

The growing requirements to reduce costs in satellite missions have led to simplify ground segment as much as possible [1]. Following that approach, SACI Ground System was designed to aggregate the functionality of TT&C Ground Station, Control Center and Mission Center into a compact environment based on six Personal Computers, interconnected through Ethernet local area network [2]. That architecture is presented in figure 1 and the role of each PC software environment is described below:

- SERVER: Linux platform with a relational PostgreSQL database server and HTTP Apache Server responsible for hosting the Flight Plan Web System and telecommand database.
- ANTENNA: Windows 95 platform responsible for both Antenna Controlling during satellite passages over ground station and Orbit Determination Software execution after passages which provides satellite passages prediction to SERVER.
- TC/TM1: Windows NT platform dedicated to satellite communication during the passages. TC takes care telecommand transmission to satellite following CCSDS protocol while TM1 receives and store housekeeping data and payload telemetry packets.
- TM2: Windows NT platform responsible to receive, process and display in real time, telemetry packet samples in order to quicklook satellite health and status data. After passages the entire telemetry packets can be visualized in playback mode.
- TC OPERATION: Windows NT platform allows the operator to both follow the satellite communication protocol events during a passage and, eventually, send online telecommands as

well. This PC can remotely perform those activities, just being connected to the Ground Station via network.

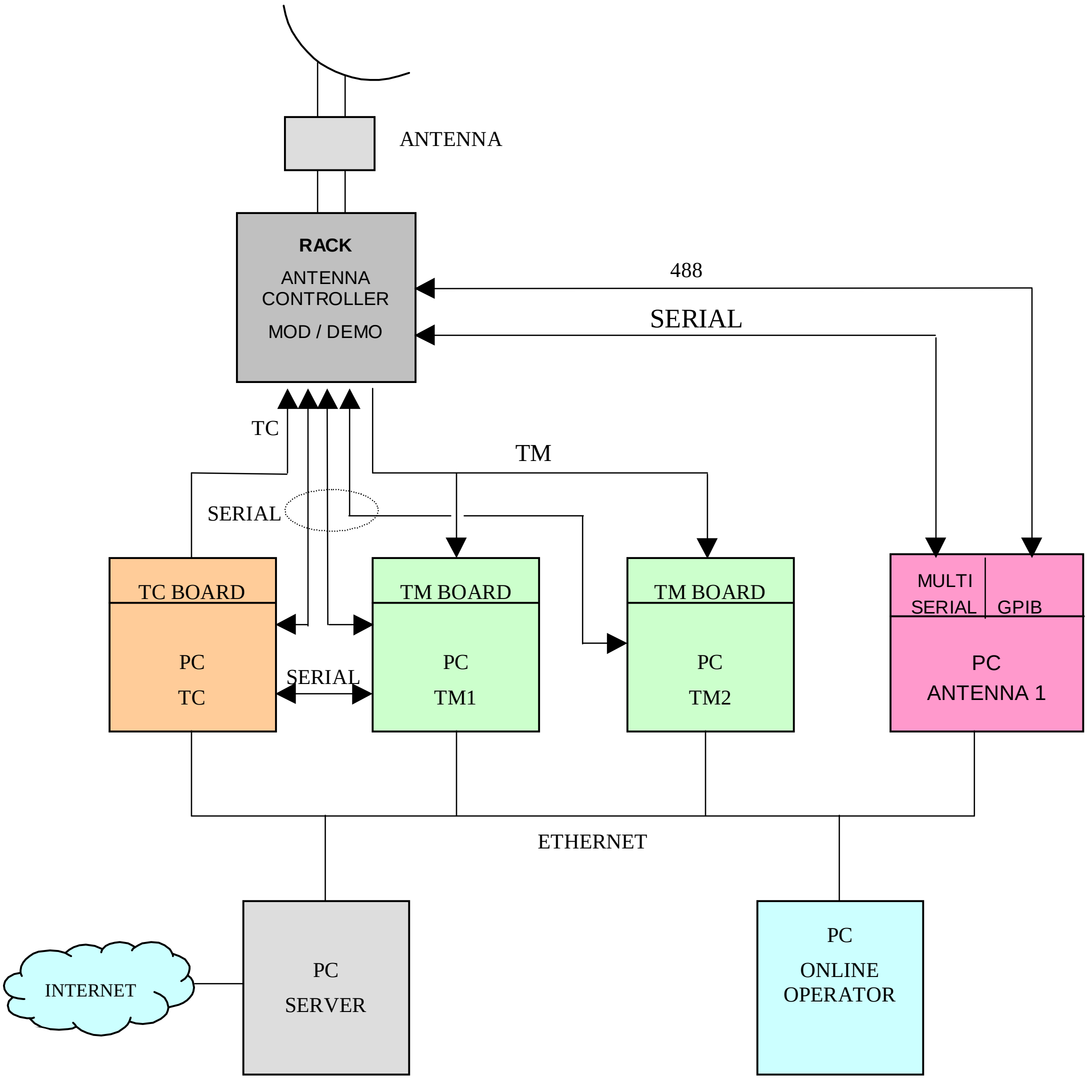

Figure 1: SACI Ground System Architecture

2 - AUTONOMOUS OPERATIONAL ROUTINE

The architecture introduced above allowed the definition of an autonomous and simple operational routine. It is based on 6 files which filenames follow the predict passage data/time: **YYYY-MM-DD-HH-MM-SS-saci-x.AAA** where **YYYY**: year; **MM**: month; **DD**: day; **HH**: hour; **MM**: minute; **SS**: second; **saci-x**: Saci-1 or Saci-2; and **.AAA**: the extensions, meaning:

1. **.PVA** – Authorized Flight Plan. It contains all TC programmed and authorized by the ground station operation manager. It is the final Flight Plan ready to start transmission to satellite on time YYYY-MM-DD-HH-MM-SS according to the predicted passage.
2. **.PVD** – Wished Flight Plan. Every satellite subsystem user (including the scientific investigator) is allowed to select via WEB pages, per satellite passage, a set of telecommands to be sent to his/her subsystem/experiment during that passage. The wished TCs are stored in PC SERVER and can be updated up to 2 weeks before the passage. On that time, the satellite operation coordinator evaluates the consistence of all wished TCs for that passage in order to generate a unique .PVD file. Therefore each YYYY-MM-DD-HH-MM-SS-saci-x.PVD can be remotely planned through

INTERNET and stored in SERVER Database until being authorized by the ground station operation manager.

3. **.PVR** – Flight Plan Report – It contains the transmission status of the TCs sent on YYYY-MM-DD-HH-MM-SS passage following the YYYY-MM-DD-HH-MM-SS-saci-x.PVA.
4. **.DAT** – Telemetry Data Packet – It contains all raw telemetry data packets acquired during passage YYYY-MM-DD-HH-MM-SS**.**
5. **.DBT -** Distributed Data Packet – It contains the raw telemetry data packets acquired during passage YYYY-MM-DD-HH-MM-SS previously separated by type.

There is also a file named **PREVIS.RES** with the Selected Passages. It contains a list of 2 weeks satellite selected passages to be tracked by ground station. The data/time of beginning and ending of each selected passage are described in this file.

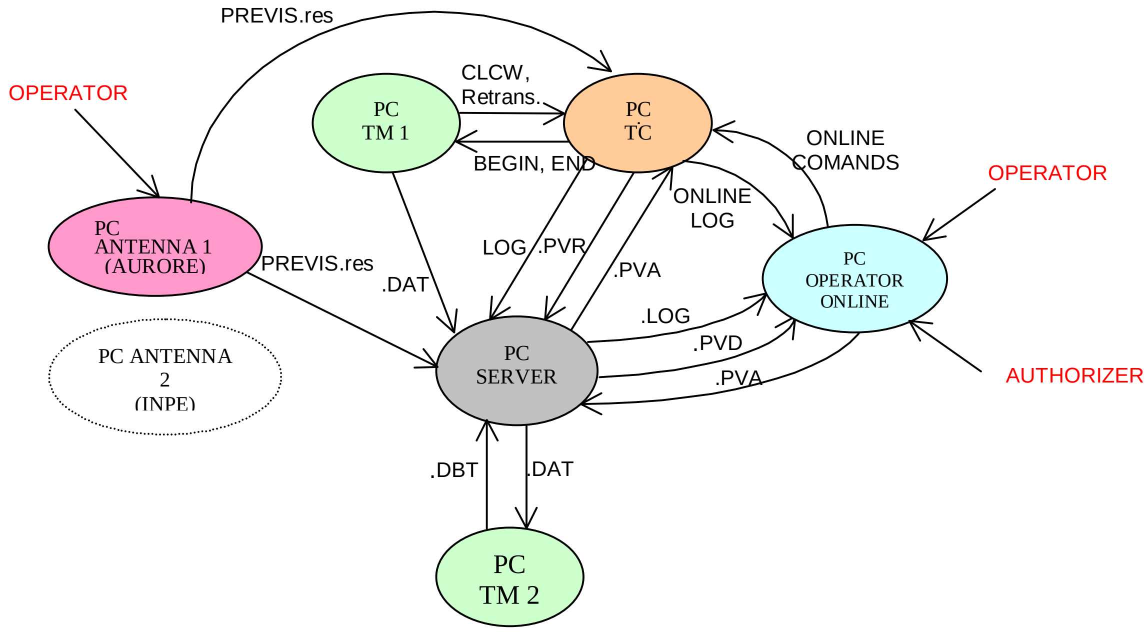

Figure 2: Automatic Ground Station Operation Diagram

The dynamic of producing and using such files follows the data flow presented in figure 2. It consists of the execution of an autonomous operational sequence of activities that intents to support both unattended Ground Station operations during the passages and payload remote control by the experiment investigators, via INTERNET.

The operational sequence of activities is divided in three phases: Pre-passage, In-passage and Out-passage.

2.1 – Pre-passage Activities

Ten minutes early each passage TC PC software checks on PC SERVER the existence of a compatible YYYY-MM-DD-HH-SS-saci-x**.PVA** filename with the data/time of next passage. If the searched file is founded, it is read and will be followed by the TC application software in order to transmit the authorized TCs to satellite on time. In case of none file, the entire passage will be followed just to acquire telemetry data.

The communication between TC and TM1 PCs is used in order to prepare TM1 about the beginning and end of next passage.

2.2 – In-passage Activities

The ground-satellite protocol runs on both TM1 and TC PCs during the entire passage over station. The communication between TM1 and TC PCs is also used during the whole passage in order to support the implementation of ground-satellite protocol execution (CLCW data according to CCSDS and Retransmission Request). At application level, on TC PC, the TCs successfully transmitted to satellite are ticked in the YYYY-MM-DD-HH-MM-SS-saci-x**.PVR** file. This file is made available to PC SERVER database as well as the ONLINE LOG file which reports the occurrences in the communication with satellite during that passage. The ground station operation manager would use these files in order to authorize the .PVD for next passage. Eventually, some problem that occurred on last TC transmission would result its retransmission on next passage.

On ONLINE OPERATOR PC, the application software allows the operator to follow up in real-time each event occurred in satellite-ground communication during a passage. Following the YYYY-MM-DD-HH-MM-SS-saci-x**.PVA**, each authorized TC is delivered automatically on time. The same way, the ONLINE COMMANDs are sent to TC PC from where it is transmitted to satellite. The system events and actions related to the ONLINE COMMANDS effectively transmitted to satellite and their acknowledge are reported to ONLINE OPERATOR PC as ONLINE LOG. That LOG information are stored and displayed in real time on ONLINE OPERATOR PC.

Particular care related to TC emergency operation has been provided by the application software running on ONLINE OPERATOR PC. It supports the manual interference of an authorized ground operator in transmission of non planned TC using the Urgent Service offered by the telecommand stack at protocol transfer layer implementation.

2.3 – Out-passage Activities

Following the data flow presented in figure 2, immediately after each passage, the ANTENNA PC updates PREVIS.RES file on PC SERVER making it readable to TC PC. Actually, PC SERVER keeps passage preview information in database for further user accessing since that file guides the ground station operational routine.

In parallel, all tasks related to the processing and visualization of telemetry packets in playback mode can proceed on PC TM2.

The YYYY-MM-DD-HH-MM-SS-saci-x.DAT file, acquired in real time from satellite during the passage, is transferred to PC SERVER where it will be recorded. Under ground station operator request, .DAT files are processed by TM2 PC application software in order to separate the telemetry packets, per type, in .DBT files. In case of SACI, 8 telemetry packet types can be identified and stored in specific files as raw data.

The TM2 PC application software also provides facilities to process .DBT files giving to ground station operator a quick look of each satellite subsystem behavior. Files with DBT extensions are kept available in PC SERVER for further access.

Additionally, on PC named ONLINE OPERATOR, the ground station operation manager shall evaluate the YYYY-MM-DD-HH-MM-SS-saci-x**.PVD** planned to the next passage in order to authorize it, generating the correspondent YYYY-MM-DD-HH-MM-SS-saci-x**.PVA** file. There are no constraints in authorizing more than one .PVD file, for instance in case of early preparation of next three passages.

3. FLIGHT PLAN SOFTWARE

In order to deal with the different faces of the Flight Plan as presented in section 2, a software system was developed to support SACI Flight Plan Preparation through the Web. That software system, resident on PC SERVER, was organized in seven modules, known as:

1. Authentication
2. Passage Updating
3. Passage Choosing
4. Planning and Visualization
5. Complementary Data
6. Operation Result Visualization
7. File Generation for the Wished Flight Plan (PVD)

The environment for the development and operation of this system is based on the open source technology [3]. All software used is available for free use under GNU license [4]. It refers to the users' freedom to run, copy, distribute, study, change and improve the software. The chosen operating system, HTTP server, scripting language and database management system are respectively described below:

- The operating system: Linux – is a complete operating system that is similar but not identical to UNIX[1]. Linux runs on a wide variety of hardware. Probably the most unique characteristic of Linux is that it is freely distributed, which means that the source code for the kernel and most software cannot be withheld[2].
- The HTTP server: Apache Server [5] – is the most popular web server on the Internet. It is a robust, secure, efficient and extensible server which provides HTTP services in synchronisation with the current HTTP standards for various modern desktop and server operating systems, such as UNIX and Windows NT.
- The scripting language: *PHP 3.0* [6] – a server-side HTML-embedded scripting language [7]. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly. A significant feature in PHP is its database integration layer. The source code is available for free use.
- The database management system: *PostgreSQL* [7] – a sophisticated open source Object-Relational DBMS, supporting almost all SQL constructs, including subselects, transactions, and user-defined types and functions. It uses a client/server model of communication[3].

The basic process [8] occurred in this environment for preparing the flight plan is presented in Fig. 3 where the user is connected to the Internet through a WWW Browser; and both the HTTP Server (Apache) and CGI Application (PHP 3.0) under Linux platform.

[1] Most UNIX software runs on Linux with no changes at all and there are many commercial applications that have been ported over to Linux. Additionally there are commercial programs that allows the users to run many Windows applications under Linux, if they so wish.

[2] It does not mean that companies cannot charge for it.

[3] That means that the PostgreSQL Server continually runs, waiting for client requests. The server processes the requests and returns the result to the client.

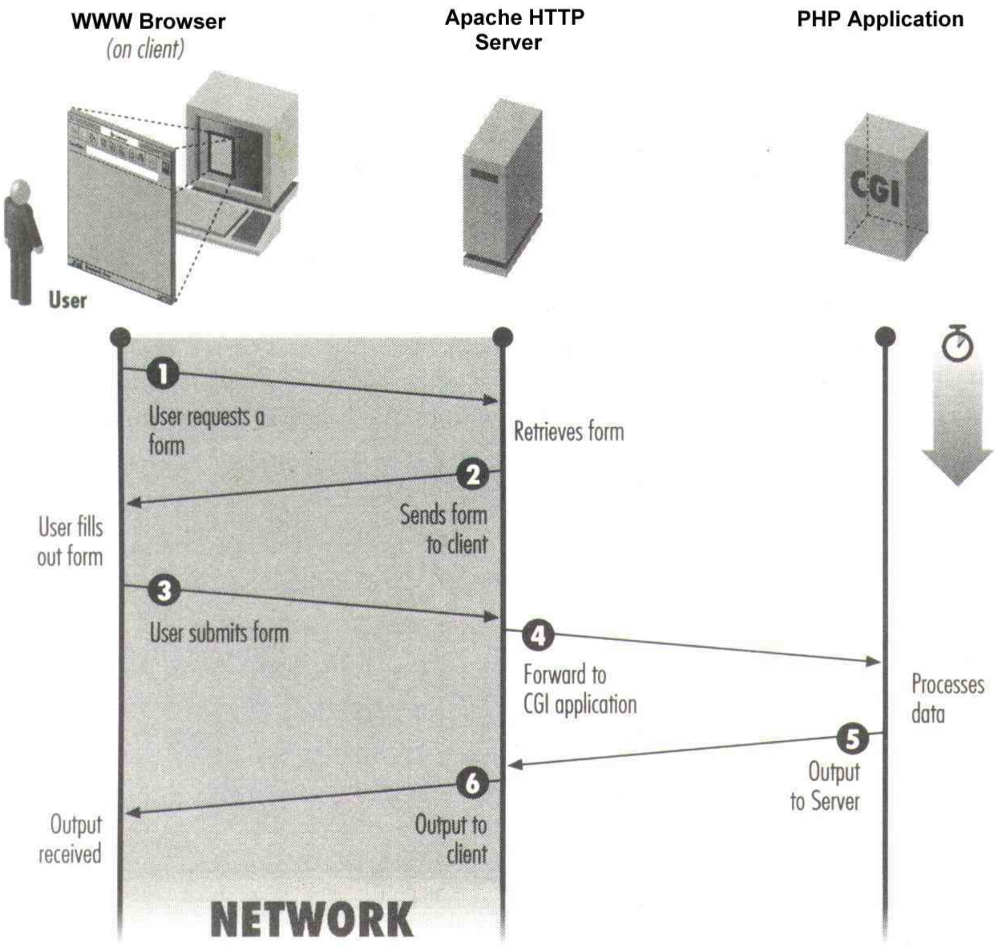

Figure 3 – CGI Process for Preparing the SACI Fight Plan via WWW

At first, the user requests the starting form (authentication page) for preparing the SACI flight plan via URL: http://saci-server.crn.inpe.br/cgi-bin/php/planodevoo/autenticacao.phtml. Then, the HTTP Server retrieves this form on PC SERVER, sending it to the client. After filling out the form, the user submits it to the CGI Application through the HTTP Server. Then, the PHP application processes the requests based on the filled form and sends back to the user the obtained result, via HTTP Server.

Each flight plan module developed in this environment is described in the following subsections.

3.1 – Authentication

The authentication process occurs when users type their identification: username and password (Fig. 4). Based on this information, the system is able to identify the available subsystems for a specific user, not permitting his/her entry in the system if absent in the cryptographer system permissions table.

This module also deals with flight plans for more then one satellite. In case of SACI mission, the user must choose the satellite (SACI-1 or SACI-2) for preparing the wished flight plan. (PVD).

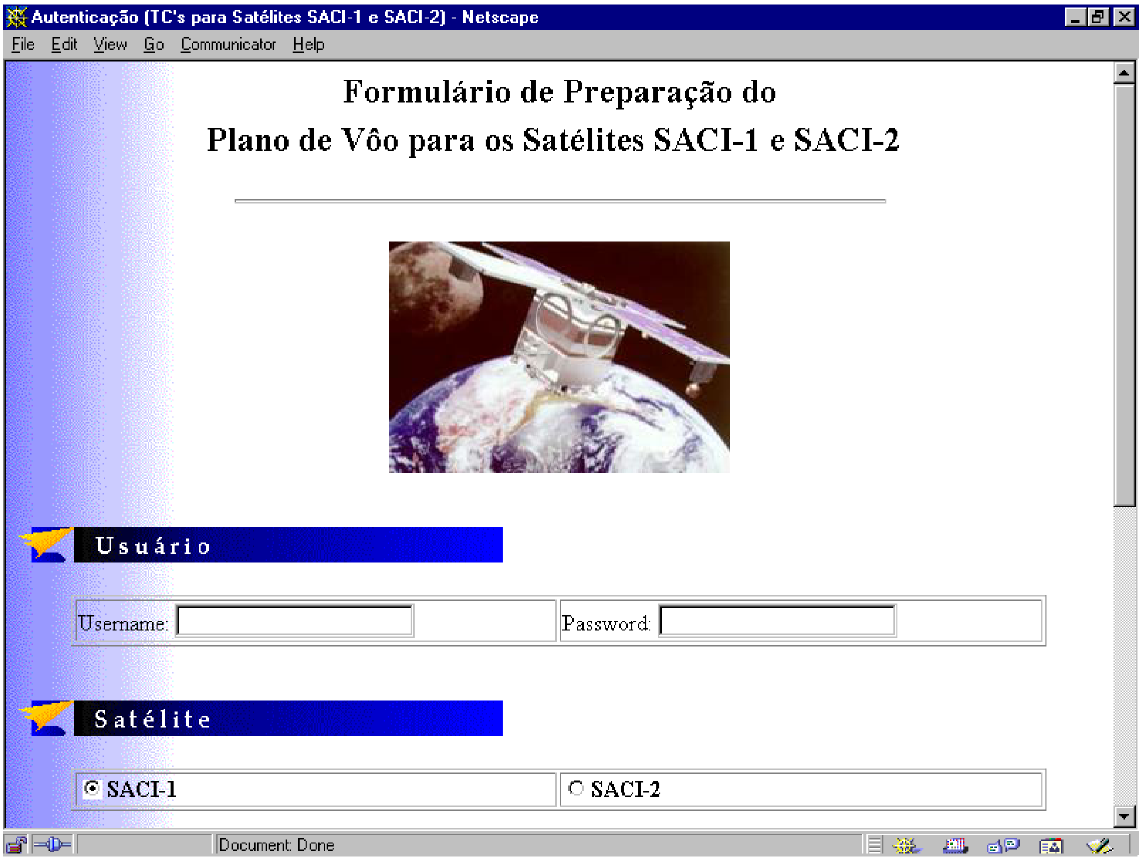

Figure 4 – Authentication Page

3.2 – Passage Updating

The dedicated software system resident in Antenna PC produces PREVIS.RES file with the previewed passages. This file is kept by the database SERVER and is automatically updated as described in 2.3 section.

The main function of this module is make PREVIS.RES information available for remote user since the Flight Plans are oriented by satellite data/time passages.

3.3 – Passage Choosing

Depending on the chosen satellite (SACI-1 and SACI-2), the system shows to the user the list of its passages previously calculated and available on SERVER.

The user selects the satellite previewed passage based on the operations needs and the interval of passage.

3.4 – Planning and Visualization

As presented in figure 5, from both: the user identification and selected passage, the system recognizes the user permissions and presents only the authorized destinations which correspond to the satellite subsystems, preventing impertinent requests. So, user may choose the sequence of telecommand to be transmitted, its order position and transmission time.

Concerning to information safety through INTERNET, the TC planning looks after the traffic of non-final TC code. In the ground station operational routine, the TC code recognized by satellite is associated to each TC only when generating the .PVD file. Moreover, the user is able to see only the authorized destinations and subsystems related to him/her. Thus, each users group of the system has options for performing actions in a restricted universe. The rules are the following:

- Investigator: is allowed to visualize and perform actions in specific experiments.
- Technician: is allowed to visualize all flight plans (including investigator ones), but can perform actions only in specific satellite subsystems.
- Coordinator: is a kind of super-user, who is allowed to visualize and perform actions in all flight plans with no restrictions.

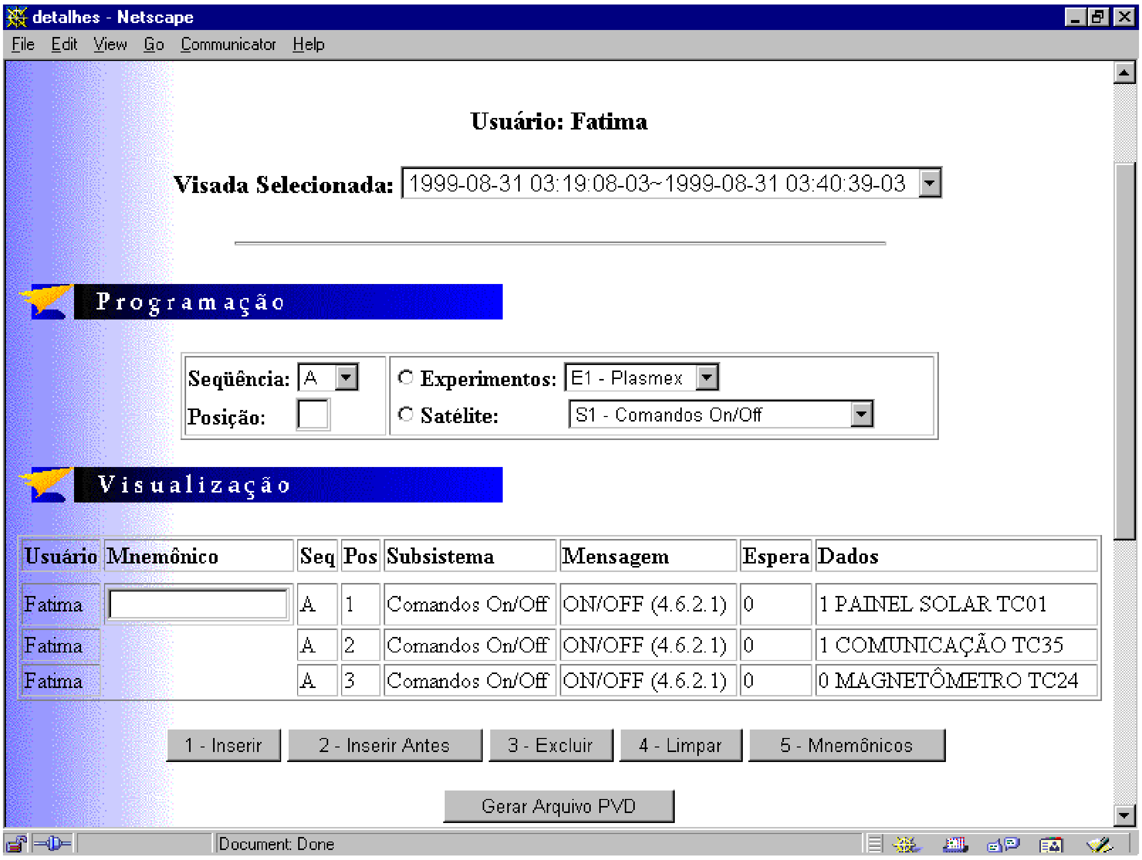

Figure 5 – Planning and Visualization Page

The operation buttons are the following:

1. **INSERIR** - 'Insert' Button – Used to add a new telecommand in the end of a sequence. By the way, a sequence can have one or more elements in it. For example, if the user create a new sequence 'A', then the element 'A-1' will be automatically created. The next element will be 'A-2' and so on.
2. **INSERIR Antes** - 'Insert Before' Button – Used to insert a telecommand before another one in the same sequence. For example, in a sequence 'A' with three elements, to include a new 'A-1' element, at first 'A-3' goes to a new position, 'A-4', 'A-2' goes to 'A-3', the current 'A-1' goes to

‘A-2’ position and, finally, the place ‘A-1’ is booked for the new telecommand which is being prepared.

3. **EXCLUIR -** ‘Delete’ Button – Used to delete existing telecommands. The confirmation for deleting elements is made by typing the associated subsystem name. Besides, it is necessary to indicate the associated subsystem in order to avoid misunderstandings. In case of not realizing the correspondence telecommand-subsystem, the system will not execute the deletion request and will show a warning message to the user.
4. **LIMPAR -** 'Reset' Button – Used to reset all fields of the current form, turning it back to the default values.
5. **MNEMÔNICOS -** 'Mnemonic Updating' Button – Used to create/update the mnemonic name associated to any telecommand sequence.
6. **GERAR ARQUIVO PVD** – ‘Generation of PVD File’ – Used to create the wished flight plan file, exclusively by the coordinator user.

Additional facilities include navigation buttons such as:

- ‘Back to Authentication’ Button - Option for going back to the homepage, starting a new session of authentication.
- ‘Help’ Button – Option for obtaining help tips in the process for preparing the flight plan. When pressed, this button opens a small window with context sensitive help.

3.5 – Complementary Data

This module deals with the telecommand complementary data, which must be provided by the user to the system for particular chosen telecommands. Such telecommands, for instance configuration table, need the specification of a set of parameters value. In order to support the edition of that kind of telecommand, the Flight Plan Software keeps hierarchical pages per subsystem.

3.6 – Operation Result Visualization

This module carries out the page that presents to the user the final result of the current operation.

3.7 – File Generation for the Wished Flight Plan (PVD)

This module generates a unique file containing the totality of records stored in the flight plan user database for a specific satellite passage. This file is generated following the autonomous operational routine described in section 2.

4. CONCLUSION

Aiming to create unattended operational environment, the Ground Segment architecture, a cost effective networked PC-based solution, allowed the implementation of a client-server web-based system. Interface pages with dynamic fill-out forms were designed in order to support remotely command data (telecommands) selection by the users, since they have an Internet connection with any browser.

This developed system proves the possibility of taking effectiveness and efficient solutions for user-friendly web applications including fill-out forms, file uploading and database facilities, putting in evidence the power of Open Source information technology.

4. REFERENCES

[1] Neri, José A. da C. F.. SACI-1 - A Cost-Effective Microsatellite Bus for Multiple Mission Payloads. International Conference on Small Satellites: Missions and Technology. Madri, Espanha, 9-13 de setembro de 1996.

[2] Carvalho, Manoel J. M.. Development of a Communication Protocol for a Scientific Applications Satellite. IASTED International Conference: Computer Systems and Applications. Irbid, Jordan, March 30 - April 2, 1998.

[3] ISO/IEC TR 14252:1996 Information technology - Guide to the POSIX Open System Environment (OSE).

[4] GNU Organization. Philosophical Issues that Pertain to the GNU Project. [online] Available in Internet via WWW. URL: http://www.gnu.org/philosophy/philosophy.html

[5] Apache Digital Corporation. Apache HTTP Server Project. [online] Available in Internet via WWW. URL: http://www.apache.org.

[6] PHP. PHP3 Hypertext Preprocessor. [online] Available in Internet via WWW. URL: http://www.php.net.

[7] Momjiam, Bruce. PostgreSQL: Introduction and Concepts. Addison-Wesley, Reading, MA: 2000.

[8] Schwartz R. L.; Christiansen T. Learning Perl. O'Reilly, Cambridge: 1997.