

OPEN SOURCE TO SMALL SATELLITES GROUND STATION

Manceol Jozeane Maira de CARVALHO (manceol@cm.inpe.br) ¹

Romualdo Alves PEREIRA Junior (romualdo@cm.inpe.br) ¹

Maria de Fátima MATTEILO FRANCISCO (fatima@dss.inpe.br) ²

Jean Paul DUBUT (jean@cm.inpe.br) ¹

¹ Instituto Nacional de Pesquisas Espaciais - INPE
Av. Salgado Filho, 3000
59001-970 Natal - RN - Brazil
Fax: (55) (84) 231 4941

² Instituto Nacional de Pesquisas Espaciais - INPE
Av. dos Astronautas, 1758
12227-010 - São José dos Campos - SP - Brazil
Fax: (55) (12) 322 9285

ABSTRACT - After being treated for long time by non-academic institutions as vulnerable technology, in terms of reliability, Open Source has recently being pointed out by suppliers as a good alternative for software developers. This paper presents an example of using Open Source Software in INPE/CRN Ground Station in order to control small satellites. There is an emphasis on using Linux Operating System, a low cost environment to provide effectiveness and support efficient solutions for an user-friendly web application.

1. INTRODUCTION

Small satellite missions aim at high performance and low cost. At ground station context, some solutions have been failed to get these goals. Mainly in software concerns, where the reusability is frequent and the availability of the source code is a crucial factor since the use of existing ground station infrastructure is desirable or imperative.

A reasonable range of known Ground Station solutions for Microsatellite Missions are using even more widespread hardware open standards as the General-Purpose Interface Bus (GPIB), Ethernet, and PCI [Pres98] [Brow 98], permitting a real independence of equipment supplier. On the other hand, software solutions do not follow the same way. Considering just the machine, it is common to find references of Workstation based on Unix systems being substituted for PCs, because of both its low cost [Sore 98] and supplier independence [Gong98]. Nevertheless, this independence is not reached when the software solution cannot accomplish this idea.

Software, which in theory is the easiest part to modify, update and aggregate new modules, loses these features when only the supplier can make any change. Open Source alternative is an excellent option to address these needs.

2. OPEN SOURCE

In this paper, the terms Open Source and Free Software are considered to have opposite meaning to the terms Proprietary Software and Commercial Software. The conceptual differences among them are well reported on the following sites: <http://www.fsf.org/philosophy/categories.html> and <http://www.opensource.org/faq.html>. This paper does not approach these differences. For the paper purpose, it is sufficient to understand the Stallman¹ explanation about Free Software philosophy, in

¹ Richard M. Stallman is the President of Free Software Foundation (USA), creator of both GNU Project and Free Software concept.

the First International Forum Free Software 2000, held in Brazil. According to him, "This paradigm started to develop in 1970 decade and fill three requirements: it has open source; their copies are distributed with no restrictions; and their changes and improvements must be widespread and distributed with no restrictions" (<http://www9.terra.com.br/noticias/especial/>).

Traditionally, software has been delivered in binary distributions, where the developers are not able to modify the underlying source. In such case the developer must hope that it is flexible enough to be configured in binary form to meet local needs. Or, then, they have to pay to suppliers to make the needed modifications for them.

The basic idea behind Open Source is very simple: When programmers can read, redistribute, and modify the source for a piece of documented software, it evolves. Developers can improve it, adapt it, and fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.

The most commonly license used by Open Source Community is the General Public License (GPL). It grants users the freedom to modify the software and distribute this derived work only if, among other restrictions, the derived work is also licensed under the terms of the GPL. This license ensures that all users of a piece of software, even users of modified versions, have the freedom to see and modify the source code. The Free Software Foundation promotes the use of the GPL, producing a good contagious effect.

As the software becomes freely available under GPL, more developers will look at, examine, and critique the code. It is expected that additional people will add features and correct bugs, too. This will result in a stable software product and more efficient code. Also, software will be produced more quickly. As a consequence, the Open Source movement will have a major impact on the quality of code being produced. The Open Source community has learned that this rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see source and everybody else must blindly use an opaque block of bits. Some years ago, Open Source was a restricted subject to hackers, but today the software industry is upholding this movement. For instance, IBM is giving perhaps the more emblematic and meaningful support to Linux, as announced in the IBM site at <http://www.ibm.com/linux>.

There are two basic ways to produce software in the Open Source Movement. It is possible to make changes, updates or improvements in an existent open source code; or develop a completely new source code creating a new community around. In these two cases the programmers can be sure that their contributions will be available to the community.

Actually, the running gears of the Internet (its mail transports, web servers, and FTP servers) are almost all open source. The Netcraft Web Server Survey (<http://www.netcraft.com/survey/>) tallies, which web servers, are used on the Internet. It consistently shows the open source Apache web server to have over 50% and steadily increasing market share, beating out better-hyped proprietary products like Netscape's and Microsoft's server suites.

3. LINUX

Linux, often known as GNU/Linux, is the leader, the principal success history and the greatest popularity in the Open Source Movement. According to the Linux Journal magazine [Rich 00], "Linux is a free, UNIX-like operating system, developed originally for home PCs, but which now runs on a variety of platforms including Alpha, SPARC, PowerPC, Crux, ARM, M68000 and many others. Linux is a multi-user, multitasking operating system whose technological heritage hails from the mainframes and supercomputers for which the UNIX model was developed over three decades ago. Even today, Linux aims for POSIX compliance to maintain maximum compatibility with other UNIX-like systems".

Linux Journal magazine additionally published that: "Linux is also the birthplace of the open source methodology" which, derived from the free software philosophy and GUIs. Today, with a firm base proliferation of free software, from compilers to word processors and GUIs. Today, with a firm base or at least 12 million users world-wide, Linux continues to grow exponentially as programmers, enthusiasts and end users exchange thoughts, implement ideas, contribute code and cooperate in the phenomenon known as open source to produce the operating system known as Linux".

Linux is very popular in education, Internet service applications, software development shops, and (increasingly) in small businesses. It is typically installed via a Linux distribution - a bundle containing the Linux kernel, applications, and utilities to form a complete working system. As a benefit to the source code for the Linux kernel being freely distributed, a number of companies have developed their own distribution of Linux. These distributions have their own feature set, and some are geared towards specific types of computer systems. In some cases, they are available at no charge via FTP and in other cases they may be purchased on CD for a relatively low cost, or as a companion to a Linux book. Several Linux distributions are available, some such as RedHat (www.redhat.com), Caldera (www.calderasystems.com), TurboLinux (www.turbolinux.com) and SuSE (www.suse.com) also provide commercial support. This several successful companies sell Linux and other free software. Why to buy a software that could be free? One benefit for purchasing a commercial Linux distribution is that a limited support contract is generally included in the purchase price. These same companies have long-term, unlimited support contracts available for purchase by businesses that are new to Linux and apprehensive about making changes in their choice of operating systems.

The Linux community has known for some time that for many applications, Linux is a stable and robust product. Unfortunately, there are still many people, including key decision-makers, which are not aware of the existence of Linux and its capabilities. The reason that products of many companies have done well in the marketplace is not so much due to the product's technical superiority but the company's marketing abilities.

In much the same way that the Linux kernel has been written by software developers world-wide, users around the globe have also written much of the Linux documentation. A large segment of the written documentation is contained in the various HOWTOs and miniHOWTOs that make up the Linux Documentation Project. Other sources of free support may be found via IRC, mailing lists, and other website.

Linux is just the kernel. Without the efforts of people involved with the GNU project, MIT, Berkeley and others too numerous to mention, the Linux kernel would not be very useful to most applications. Thus, Linux is not the whole success story of open source. There are many other open source operating systems and applications available, including Programming Software, Database Tools and Utilities, Financial Programs, Text Editors, Spreadsheets, Networking Tools and Utilities, etc.

It has long been agreed that for Linux to succeed in a business environment, it needed to have a user-friendly desktop and included all the usual applications for the office. Today, much effort is directed to desktop development. There are two desktops being developed for Linux: KDE and GNOME.

KDE is a mature desktop suite providing a solid basis to an ever-growing number of applications for Unix workstations. KDE has developed a high quality development framework for Unix, which allows for rapid and efficient application development. Applications developed with this framework include KOffice, a full-featured Office Suite, KDevelop, a C/C++ IDE (Integrated Development Environment), and many others.

GNOME is the GNU Network Object Model Environment. The GNOME project intends to build both a complete, easy-to-use desktop environment for the user, and a powerful application

framework for the software developer. GNOME is part of the GNU project (www.gnu.org) and is free software compliant with the Open Source(tm) definition.

4. EXAMPLE

A software system was developed at CRN/INPE using Open Source tools to support SACL1 Flight Plan Preparation through the Web. The system's main purpose is to allow the experiment scientific coordinator and ground station operators to have the use of all its resources independently of the place they are, since they have an Internet connection with any browser. Basically, it is a client-server web-based system arranged in some interface pages with dynamic fill-out forms. The command data (telecommands) selected by the users are stored and maintained in a relational PostgreSQL database server.

The software system is organised in seven modules, succinctly explained below:

- **Authentication** - The authentication process occurs when users type their identification: username and password. 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 than one satellite. In case of SACL1 mission, the user must choose the satellite (SACL1 or SACL2) for preparing the wished flight plan (PVD).
- **Passage Updating** - A dedicated ephemeric software system produces a file with the previewed passages. The database server keeps this file and it is automatically updated. The main function of this module is to make this file information available for remote user since the Flight Plan is oriented by satellite data/time passages.
- **Passage Choosing** - Depending on the chosen satellite (SACL1 or SACL2), the system shows to the user the list of its passages previously calculated. The user selects a previewed satellite passage based on the operations needs and the interval of passage.
- **Planning and Visualisation** - From both the user identification and selected passage, the system recognises the user permissions and presents only the authorised 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. There are operations for inserting, deleting and updating data; creating mnemonics; and generating the final flight plan file. 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 recognised by satellite is associated to each TC only when generating the .PVD file. Moreover, the user is able to see only the authorised destinations and subsystems related to him/her. Thus, each users group of the system has options for performing actions in a restricted universe ruled as shown in Table 1.

User Groups	Permission to VIEW TCs from	Permission to INSERT, UPDATE and DELETE TCs from
Coordinator	<ul style="list-style-type: none"> • All Satellites • All Experiments 	<ul style="list-style-type: none"> • All Satellites • All Experiments
Technician	<ul style="list-style-type: none"> • All Satellites • All Experiments 	<ul style="list-style-type: none"> • Restricted Satellites Subsystems
Investigator	<ul style="list-style-type: none"> • Own Experiments 	<ul style="list-style-type: none"> • Own Experiments

Table 1 - User Groups and Operating Permissions

- **Additional Data** - This module deals with the telecommand additional 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.

- **Operation Result Visualisation** - This module carries out the page that presents to the user the final result of the current operation.

- **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 an autonomous operational routine [Perc00].

The adopted HTTP server, scripting language and database management system are respectively described below, everything running in a Linux platform:

- **HTTP server** - Apache Server (<http://www.apache.org>), 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.
- **Scripting language** - PHP 3.0 (<http://www.php.net>), a server-side HTML-embedded scripting language. 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.

- **DBMS: PostgreSQL** (<http://www.postgresql.org>), a sophisticated open source Object-Relational Database Management System, supporting almost all SQL constructs, including subselects, transactions, and user-defined types and functions. It uses a client/server model of communication.

The basic process occurred in this environment for preparing the flight plan is presented in Fig. 1 where the user is connected to the Internet through a WWW Browser, and both the HTTP Server (Apache) and CGI Application (PHP 3.0) in a Linux platform.

Basically, user requests the forms for preparing the SACL1 Flight Plan, starting with URL: <http://saci-server.crn.inpe.br/cgi-bin/php/planodevo0/autenticacao0.html>. Then, HTTP Server retrieves related forms sending them to the client. After filling out the forms, user submits them to a CGI Application through HTTP Server. Then, PHP application processes the requests based on the filled forms and sends back to the user the obtained result, via HTTP Server.

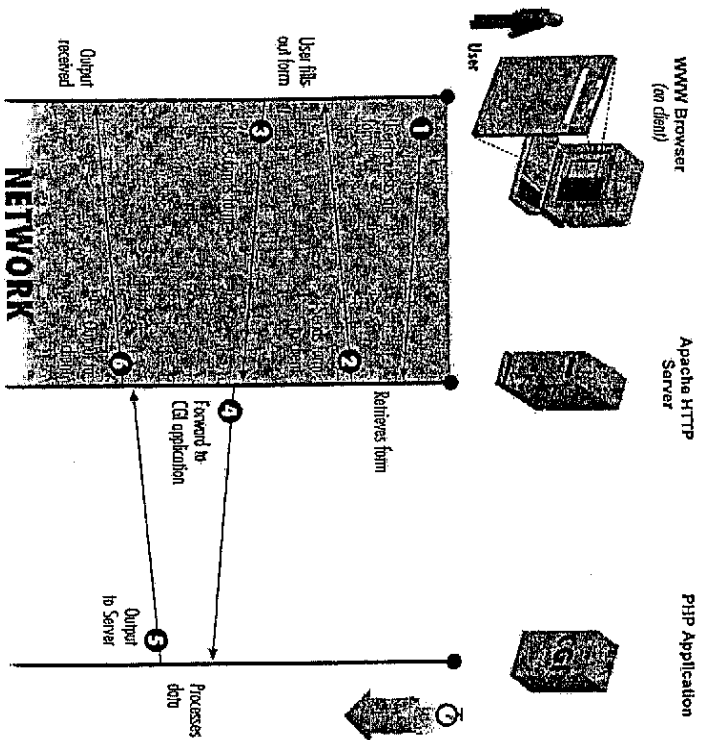


Figure 1 - CGI Process for Preparing the SACI Fight Plan via WWW

5. SOURCES OF INFORMATION:

For being a software development model based on the collaboration among numerous groups in different and possibly distant places, the information exchanging among the involved ones happens almost in the totality through Internet, although the number of conferences, symposium and meetings dedicated to this subject has increased every year.

As the mainly source of information and data repository related to this paradigm, Internet has excellent sites for a more detailed research, as:

- <http://www.linuxdoc.org> (Linux Documentation Project)
- <http://www.linux.org> (Linux Online)
- <http://www.li.org> (Linux International web site)
- <http://www.lpi.org> (Linux Professional Institute)
- <http://www.gnome.org> (GNOME project)
- <http://www.kde.org> (KDE project)
- <http://www.gnu.org> (GNU Project - Free Software Foundation (FSF))
- <http://www.opensource.org> (The Open Source Page)

6. CONCLUSION

Nowadays, there is a large variety of software available as Open Source. In some cases, they are considered better than proprietary similar ones that are commonly very expensive. As presented in this paper, there are several reasons for Open Source recommendation, such as: the quick production and software efficiency as a consequence of the increased number of people working with. Also, the participants' contribution in testing and correction of bugs improving quality and maintainability of Open Source software.

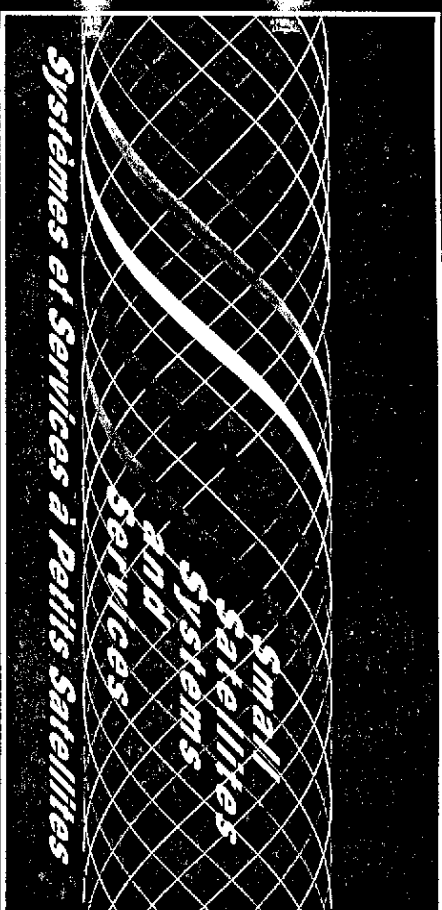
In critical software systems like satellite ground station, reliability is a very expensive need to be reached. It requires so many tests. Still in ground station context, it is often that operation requirement being incorporated to the software system during the life cycle of a satellite mission, which demands software modifications. Beside common needs in maintainability, it is also desirable that ground station software system be used for new satellite missions.

In order to attend those requirements with low cost, software development based on the Open Source Technology is suggested. It grants important advantages for maintaining and updating ground station systems.

REFERENCES:

- [Brow 98] A. J. B. Brown, A. K. Ward, R. M. Myat : "High Performance Ground Stations for Low Cost Missions", *4th International Symposium Small Satellites Systems and Services*, Antibes Juan les Pins, France, September 1998.
- [Gonc 98] L. Gonçalves, P. Cardoso, A. Ambrósio : "Satellite Control Center: Solution for Adaptable System", *4th International Symposium Small Satellites Systems and Services*, Antibes Juan les Pins, France, September 1998.
- [Pere 00] Romualdo A. Pereira Jr: "Building a Telecommand Web System for Scientific Microsatellite Control Base on Open source Technology", *Symposium SPACE OPS 2000*, Toulouse, France - 19-23 juin 2000.
- [Pres 98] P. Pressecq, B. Roger : "La création d'une ligne de produit récurrente et économique", *4th International Symposium Small Satellites Systems and Services*, Antibes Juan les Pins, France, September 1998.
- [Rich 00] M. Richardson : "What Is Linux?", *Linux Journal - Issue 72*, Seattle - WA - USA, April 2000.
- [Sore 98] T. Sorensen, D. Bakens, T. Probert : "DataLink: A Commercial Ground Solution for end-to-end Satellite Operations", *4th International Symposium Small Satellites Systems and Services*, Antibes Juan les Pins, France, September 1998.

RECUEIL DES ACTES PROCEEDINGS



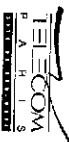
2^e Symposium International
19-23 Juin 2000
La Baule France

5^e International
Symposium
19-23 June 2000
La Baule France

Manoel Jozeane M. Carvalho
IAS - ch 3304 - BP 4013
31028 - TOULOUSE - Cedex 4
☎ (33) (0) 5 62 17 3304



CENTRE NATIONAL D'ETUDES SPATIALES



CENTRE NATIONAL D'ETUDES SPATIALES

CNES - Délégation à la Communication
Département Expositions-Colloques
18, avenue Edouard Belin
31401 Toulouse cedex 4 - France
<http://www.cnes.fr>

POSTERS

- (P-1) **RESUME project**
Roggero E., Cecozzoli M., COMAE, Satory J., Bakkeli D., UTM-FTH, Buenos Aires, Argentine
- (P-2) **Satellite Tracking from the Top of the World;**
Operation of a Multi-Purpose Ground Station at 80°N
Skatteboe R., Norwegian Space Centre, Oslo, Norvège
- (P-3) **NOVEL Range Measurement System**
Carayon J.L., Chatain C., CNES, Toulouse, France
- (P-4) **The Agile Mission**
Musso C., Augelli M., Cricconio M., Favella M.C., Giommi P., Pallizzoni A., Valentini G., Bignami G.F. Agenzia Spaziale Italiana, Rome, Italie
- (P-6) **Suivi de changements sensibles ou dangereux par système satellite et solution proposée**
Dutuit J.P., de Carvalho M.J.M., dos Santos M.A.F. INPE/CRN, Natal, Brésil
- (P-8) **Xenon Feed Systems for Electric Propulsion**
Smith P., Polytech Aerospace Ltd, Cheltenham, Royaume-Uni
- (P-9) **The Design and Implementation of an Autonomous Ground Station for PANSAT**
Hurling J.A. Space Systems Academic Group, Monterey, Etats-Unis
- (P-10) **Near Equatorial Orbit (NEqO) Satellite Constellation for Remote Sensing and Digital Communications**
Othman M., Ashhad A. S., United Nations Office for Outer Space Affairs, Vienne, Autriche
- (P-11) **Space Systems Using GEO and LEO Satellites. Mission: Long Distance Activities Support**
de Miguel J., Alario N., Garrido B. Instituto Nacional de Técnica Aeroespacial (INTA), Madrid, Espagne
- (P-12) **JAVA à l'épreuve des petits satellites / JAVA face to small satellites**
Apyville L., Senar P., Richard F., Diez M. Alcatel Space Industries, Toulouse, France
- (P-13) **A Gamma Ray Observatory on the MITA Platform Agile a Forerunner for ASI Small Satellite Scientific Programme**
Crocco L. Carlo Gavazzi Space, Milan, Italie
- (P-14) **Open Source to Small Satellites Ground Station**
de Carvalho M.J.M., Alves Pereira R. Jr, Mattheilo Francisco Made F., Dutuit J.P.
Instituto Nacional de Pesquisas Espaciais, Natal, Brésil
- (P-15) **Fifteen Years of Small Satellite Communications: A Lesson Learned**
Peyrel B., Rondineili G., Telespazio spa, Rome, Italie
- (P-17) **MINISAT-01: Tools for scheduling, data analysis and payload monitoring**
Izavera A., Garcia J., Quintana C. Lab. de Astronáutica Espacial y Física Fundamental, Madrid, Espagne
- (P-18) **A Method for Predicting on Board Computers Single-Event Effects Induced Failure Rates, Based on Markov Chains**
Ferreira P.A., Marques C.A., Gaspar J.P. Facultad de Matematica Astronomia y Fisica, Cordoba, Argentine
- (P-20) **Cryogenic Transition-Edge Microbolometers and Calorimeters with on-chip coolers for X-ray and Far-Infrared Detection**
Kinnunen K., University of Jyväskylä, Jyväskylä, Finlande
- (P-21) **Design and development of on-board data handling (OBDH) software for a university micro-satellite**
Fonte S., Department of Aerospace Engineering, Aversa, Italie
- (P-22) **Design of the Smart Microsatellite Electrical Power Subsystem**
D'Ercole M., Pastena M. Dipartimento di Ingegneria Aerospaziale, Aversa, Italie
- (P-23) **SMART Microsatellite structure, Mechanisms and Thermal Design.**
Fragitto M. Aerospace Engineering Department, Aversa, Italie
- (P-24) **The European Spacecraft Platform Database**
Linnel P., Toussaint M., Debruyne E. Eurospace, Paris, France
- (P-25) **Smallsats: from infancy to adulthood?**
Lonnert P., Toussaint M., Debruyne E. Eurospace, Paris, France
- (P-26) **Space School for Teenagers**
Sausen M.I., Avila J. Ministerio da Ciencia e Tecnologia, Sao Jose dos Campos, SP, Brasil
- (P-27) **Station Falbe Côté pour Micro-Satellites**
Lemaigner F., Thomas C., CNES, Toulouse, France
- (P-28) **Swedish Small Satellites**
Lundvall K., Swedish Space Corporation, Sohra, Suede
- (P-29) **Let it run. Automation of Ground Mission Operations and Satellite Testing Activities**
Aked R., Izkovitz M., Wymnicki F., Lobrecht J., Space Applications Services (SAS), Zaventem, Belgique
- (P-30) **Dial-A-Sat: Access Links for Small Satellites via Commercial MSS**
Cohl E., De Caunes H., Ribes A., Winton A., ESA/ESTEC, Noordwijk, Pays Bas