Mem. S.A.It. Suppl. Vol. 3, 355
© SAIt 2003
Memorie della
Supplementi

WSSL: the Workstation Software System under Linux *

A. Caproni[1,2], M. Pucillo[1], R. Smareglia[1], A. Zacchei[1], E. Oliva[3,2]

[1] INAF/Osservatorio Astronomico di Trieste, Via G.B.Tiepolo 11, I-34131, Trieste, Italy
[2] INAF/Centro Galileo Galilei, Calle A. de Abreu 70, E-38700, Santa Cruz de La Palma, Spain
[3] INAF/Osservatorio Astronomico di Arcetri, Largo E. Fermi 5, I-50125, Firenze, Italy

Abstract. Workstation Software System is the control software of the Galileo Galilei Telescope. Developed more than 10 years ago for HP workstation, it needs now a general refurbishment to adopt the new hardware and software technologies. This paper will describe the development of this project from the linux operative system choices to the actual status of mixed control system passing throw a prototype realized under linux (Caproni et al. 2002).

Key words. Control system – Workstation Software System – Linux

1. Introduction

The Workstation Software System (WSS) is the middle layer control software of the Italian Galileo Galilei telescope (TNG) settled at the Roque de Los Muchachos mountain in the La Palma island. Developed about 10 years ago on HP-UX workstations, it basically offers all the facilities and tools to easily interconnect the low level VME and Windows GATEs, closely linked to the hardware and the real time tasks, with the high level tasks (usually external processes like Graphic User Interface applications), non real time data acquisition and run-time tools and facilities. WSS also presents a consistent interface to the Archive at Telescope (AaT), the database used to store telemetry and meteo parameters and to generate and store, on-the-fly, the FITS files homogenously for all compliant TNG instruments. The whole software and hardware system in the control room is composed of several HP workstations; where each of them may be connected to an arbitrary number of GATEs and support an arbitrary number of high level processes GATEs. High level processes communicate together via WSS: one copy of WSS runs on each workstation performing some automatic tasks that allow each process to read and modify parameters, send, receive commands and alarms to any other process or

Send offprint requests to: A. Caproni
* Poster at http://sait.oat.ts.astro.it/MSAIS/3/POST/Caproni_poster.jpg
Correspondence to: via G.B.Tiepolo 11, I-34131 - Trieste - Italy, caproni@ts.astro.it

task of the system independently of where it physically resides. It is achieved with an easy, highly reconfigurable and efficient addressing system. Each copy of WSS also automatically updates the AaT by sending commands, telemetry parameters and images produced by the low level GATEs and high level processes of the WSS instance itself.

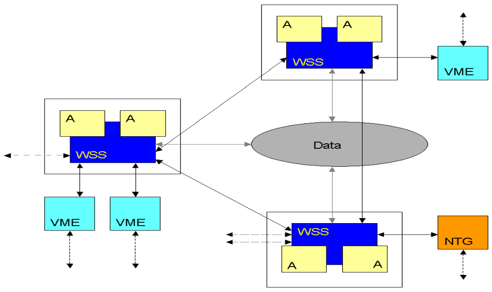

Fig. 1. WSS original structure

2. Refurbish vs remake

The choice to implement WSS under very expensive proprietary HP-UX workstations (at the time when the WSS was setup this choice was dictated by the fact that only HP guaranteed full functionality of its machines in a challenging environment) and the need to keep low maintenance costs forced to lock the hardware and software technologies adopted: ten years represent a great amount of time in computer world so we were obliged to decide how to upgrade our control system for both hardware and software.

Two main alternatives arose: remake the whole control software or update the existing one. The second choice was considered to be the best, because the first one required a lot of time and man power to studying and implementing it before taking any advantage from the new system. To efficiently face the upgrading and keep our system up to date at an affordable cost, the entire project has been realized keeping in mind several points:

- the market offers today very fast and inexpensive computers whose power and reliability are comparable to those of a workstation;
- the impressive growth of open source and free operating systems that now have reached a great degree of reliability and efficiency for free;
- the open source world offers new development languages, tools and technologies for free.

From these points, the natural choice was to upgrade the system to the UNIX open source world that means to port the Workstation Software System under Linux (WSSL).

3. Constraints in porting WSS under Linux

The main goal was to have immediate benefits from the porting i.e. to have as soon as possible a PC with WSSL on duty in control room. This means that a commercial PC with WSSL, very different from both hardware and software points of view from the HP workstations, is to be inserted early in the control room and must be able to cooperate with the existent WSS under HP. To achieve this point we need to

- minimize the impact on the current HP systems that now have a very good level of reliability obtained by
  - full transparency for the existing HP workstations that never distinguish between Linux and HP systems,
  - no changes on the internal WSS architecture,
  - no modifications in the WSS code under HP to avoid to reduce the reliability of WSS
- migrate the existing tools, i.e. we want to present at least the same interface for both high level processes like ancillary processes or IDL, and low level processes i.e. VME and Windows GATEs;

- present the same interface to the AaT that is not part of WSS and must not be affected by the porting;
- minimal impact for final users that, if possible, should not be able to distinguish the different behaviour of the two systems.

4. Description of the work

Porting doesn't consist in a mere transfer of the code from HP to Linux systems and recompiling: it implies, at least

1. to find the differences between the two compilers and operating systems;
2. to create procedures and library functions to make the behaviour of WSSL identical to that of WSS;
3. to create procedures and library functions to present the same interface to the database;
4. to rewrite entire portion of the existing code to adhere the new technology;
5. to use a versioning tool, cvs in our case;
6. to create test applications to check all functionalities, libraries and interfaces of WSSL.

Commercial PCs and HP workstations are also different in their hardware architecture: this aspect must be taken into account while porting the software to make possible to use Linux PCs and HP workstations together. One must ensure:

- run-time consistency between WSS and WSSL for
  - communications implemented with alarms, messages, commands, parameters exchanged by using sockets;
  - binary configuration files.
- archive consistency;
- well defined and possibly easy start-up procedures to allow the interoperability between the two hardware architectures.

The porting of the original code "as is" will only give speed and price advantages, it is to be considered part of the porting

- the creation of tools and interfaces, to introduce new software technologies;
- the substitution of deprecated tools with new technologies.

5. Current situation

The project started about one year ago: a prototype was built to fully understand the possible technical solutions for the upgrade, to estimate the time and the man power needed and finally to isolate the portions of code that were to be rewritten or checked carefully. Once that phase was terminated, an operative version of WSSL has been built. The current situation is illustrated by the following points:

- a Linux PC with WSSL installed in control room and cooperating with the other HP workstations running WSS
- the first Java G.U.I. application;
- the first C++ ancillary;
- a running interface with the AaT;
- a working interface for IDL v5.6 ancillaries;
- the possibility to write ancillary processes with any language that supports standard UNIX sockets.

But not all the work is already done, actually WSSL panels, i.e. dialog boxes directly generated by WSS, are not yet ported (maybe they will never be ported because they will be substituted by new GUI processes)

- Table Editor has minimal functionality (it will not be used until at least one HP workstation is in control room)
- Graphic Editor not ported (but not used if WSSL does not implement panels)

The software enhancements already obtained are

- usage of last generation languages and compilers;
- multi-thread and multi-process programming;
- new libraries;
- modern graphic user interfaces.

6. Separate environments for HP and Linux

Separate environments are needed to safely run HP and Linux systems together. Hardware and software differences force to minimize the interaction with the two systems for both binary configuration files and run-time communications. Separate environments means:

1. creating a separate run-time environment for the Linux PC that must be at any given time consistent with the HP environment;
2. making WSSL able to recognize if a peer workstation is Linux or HP and to adopt the right interface for the two cases.

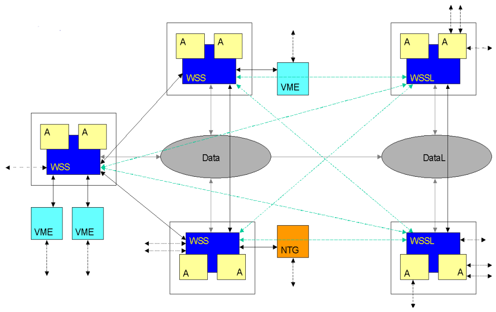

Fig. 2. Mixed environment

7. Developing projects

The following are the planned projects to be implemented in a short time using the new WSSL platform:

- plug and play: an ancillary, developed in C++, with a well defined socket interface to allow an arbitrary remote process to interact with WSSL (commands delivery and parameters reception);
- web display: use plug and play to see the current situation of the telescope directly on the web;
- connect GATEs and NTGATEs to one or more ancillaries instead of low level WSSL procedures.

Acknowledgements. TNG staff contribution in all phases, design and developing, was essential to test the new environment. We also wish to thank Claudio Vuerli for his openness in discussing design an implementation of the WSSL

References

Balestra A. et al., TNG Technical Report N.9, 1991.
Caproni A., Telescopio Nazionale Galileo, Technical report n.65; Pubblicazione Osservatorio Astronomico di Trieste n.2191; 1999.
Caproni A., Smareglia R., Zacchei A., Pucillo M., Astronomical Observatory of Trieste, pubblication n.2416, technical report n.116, 2002.
Marcucci P., Astronomical Observatory of Trieste, AOT pubblication n.1440.
Pasian F., TNG technical report n.51, 1995.
Vuerli C.et al., in *Proceedings of SPIE, Telescope Control Systems III*, Hilton Lewis Editor, Kona (Hawaii), March 1998.
Vuerli C., Pasian F., Pucillo M., Smareglia R., in: *Astronomical data acquisition, processing and storage: modern facilities*, O. B. Dluzhnevskaya, F. Pasian eds., Baltic Astronomy, 2001.