Status of ESO Very Large Telescope control software

G. RAFFI

European Southern Observatory (**ESO**), Karl-Schwarshild-Str. 2
D-85748 Garching bei Muenchen, Germany
Email: graffi@eso.org

Abstract

The Very Large Telescope (VLT) project consists of four 8 m diameter telescopes to be installed at a newESOobservatory in the Atacama desert in Chile. Extensive tests involving hardware and software will start at the Observatory and in Europe next year, but the whole commissioning will extend well after year 2000.

The control software architecture for the VLT project was introduced at the ICALEPCS '91 Conference. The size of the control software has been estimated to be equivalent to a 200 man-years development effort or about 1-1.2 million lines of code. This work is currently being done by ESO staff, partly in collaboration with industry and astronomical institutes.

The ESO team has created a central layer of software on top of a Standard architecture, based on Unix and VxWorks and on a commercial industrial control package, including a real-time database. This layer of software amounts by now to about 500,000 lines of code.

The introduction of an object-oriented event-driven architecture in the control software is one of the recent and most promising developments (reported in a separate paper at this Conference). Other interesting developments are based on the usage of Tcl/Tk in the implementation of a Sequencing tool and of a Panel editor, which allows the creation of homogeneous user interfaces for telescopes and instruments.

The experience and evolution of concepts in the VLT control software are reported, with emphasis on the goal of maintaining consistency while adapting to new technologies.

1 INTRODUCTION

The VLT project consists of four 8 m diameter telescopes, capable of working in parallel as an equivalent telescope of 16 m diameter, to be installed at a new ESO observatory in the Atacama desert in Chile.

The site of the new Observatory is in an advanced stage of construction, while the main mirrors, the telescope structures and all other components, including instrumentation, are being prepared in Europe.

The first components requiring control software will be the dome enclosure of the first telescope, which will be installed in Chile in spring next year. At the same time the first main telescope structure will be tested in Europe, together with its control software.

The first telescope will become operational in two years time, but the entire commissioning period for the four telescopes and the auxiliary telescopes used in the interferometric laboratory will extend well after year 2000. The instrumentation program, involving control software based on the same components and standards, will be developed in parallel and will obviously continue for a much longer time.

2 VLT CONTROL SOFTWARE

The VLT control software consists of all the software which will be used to directly control the VLT Observatory, telescopes and associated instrumentation.

The control software architecture for the VLT project was introduced at the ICALEPCS '91 Conference [1]. It consists of a fully distributed system based on a number of workstations and microprocessors, called LCUs (Local Control Units) (about 40 and 150 respectively in the complete VLT and instrumentation configuration).

This is now in the implementation phase, performed to a large extent directly by ESO staff in the VLT software group, but also by consortia of institutes responsible for some ESO instruments and contractors, who implement some of the telescope subsystems.

Some aspects of the control system architecture, notably the Local area network, the backbone network and the integration of a direct 2Mbit/s link to Europe, will require step-wise upgrades. Digital detectors for optical and infrared observations in particular are very demanding in terms of bandwidth and this is never wide enough. This means that while the global architecture is clear, the media used will have to evolve, and indeed an evolution path from Ethernet to FDDI and ATM is foreseen, although most of the testing and detailed engineering work has still to be done.

3 VLT COMMON SOFTWARE

The main foundation body for the VLT control software is called VLT common software. It consists of a layer of software over the Unix operating system, in the case of workstations, and on top of the VxWorks operating system for the LCU microprocessors. It provides mainly common services, like an architecture-independent message system, a real-time database for all telescope and instrument parameters, error and logging systems and a large number of utilities and tools.

The main Packages in the VLT common software are:

- CCS (Central Control Software) (see also [3]).
  It is a layer of software built on top of a commercial system (RTAP, Real Time Application Platform by Hewlett-Packard), which relys on a real-time database, runnable on several Unix platforms.
- LCC (LCU Common software) (see also [4]).
  It works on LCUs over VxWorks and is completed by a set of drivers for the ESO standard cards and a user interface (GUI) on the host workstation. It is the common platform for all the LCUs of the VLT and instruments. A motor library, dealing with the VLT standard control cards is also contained in this software. Test and debugging tools are also provided.
- HOS (High level Operation Software).
  It consists of a set of high level tools to provide support for operators and astronomers, also used in the preparation phase of Observing runs ahead of time.
- INStrumentation common software.
  Additionally there is a library, which forms what is called the Instrumentation common software and is specific to instrumentation applications.

4 VLT COMMON SOFTWARE RELEASES

It is used across all computers of the VLT observatory (telescopes and instruments) and is designed, implemented and maintained by the VLT software group. The group is also responsible for monitoring software developed outside ESO and for later integration into the VLT control software at the VLT Observatory.

The VLT common software is used in the whole VLT programme, telescope and instruments, by ESO staff, contractors and consortia.

Therefore the VLT software group started a system of releases of this software one and a half years ago, which is by now distributed to about 15 sites, both internal and external to ESO and runs on both HP (HP-UX) and SUN (Solaris 2) platforms. The last release was distributed externally at the end of August 1995.

The VLT common software has a size of about 500.000 lines of code (including code, comments and test procedures), mostly written in C, but the use of C++ is on the increase and Tcl/Tk procedures are also present.

The level of confidence in the capabilities and quality of the VLT common software is quite high at this stage.This does not only result from the application of rules, programming standards, configuration control and test procedures, but mainly from feedback based on field tests. In particular the ESO telescope NTT (new Technology Telescope) is being upgraded to the VLT standards in parallel by an independent but closely linked ESO team.

5 VLT CONTROL SOFTWARE FOR TELESCOPES AND INSTRUMENTS

This is the main part of the VLT control software and emphasis is shifting towards it as more and more design work gets completed. Implementation has started within the VLT software group in several areas, such as telescope subsystems, telescope coordination, CCDs and instrument software prototypes.

The first internal milestone of the Telescope Control System software (TCS) was reached in September, with integration of coordination software and subsystems. This is actually a joint venture between the VLT and the NTT upgrade teams, so that most of the software will be the same. The next major milestone for TCS is coming early next year when TCS has to be ready for tests with the structure in Europe, scheduled for a period of about six months.

Concerning instrumentation and detector software, the detector CCD (Charge Coupled Device) software, after the January 1995 prototype tests, will reach its main milestone in early 1996. This software will be used not only to control and acquire data from scientific CCDs, but also to get images from the “technical” CCDs used for example to drive automatic guiding of telescopes. Real-time display software with a number of interactive features has been developed for this purpose and as part of the VLT common software also.

External teams are well advanced in the development of certain VLT instruments, while ESO staff is also working onothers in parallel. Counting the number of telescope foci available for instruments, one comes to a first generation of instruments close to ten, even without taking possible multiple copies of an instrument into account.

6 VLT CONTROL SOFTWARE TRENDS

A set of new tools in the CCS software (Database Loader, Extended CCS, Event Handler, Class Browser) support an object-oriented design and implementation of the VLT software (workstation part).

The object-oriented Event Handler in particular is one of the recent and most promising developments and is presented in a separate paper at this Conference [2]. It is used as a key element of the Telescope and Instrumentation control software developments, which are on-going at the moment, and enforces an object-oriented design of the surrounding software.

A vendor-independent platform of VLT common software on Unix workstations has also been developed and is called CCS-lite. This does not support access to a real-time workstation database, as the Rtap product is not used in this version of CCS. However it provides access to the LCU database and support at the workstation level for passing messages, errors and logs and allows easy implementation of user interfaces to LCU software

Other interesting developments are based on the use of Tcl/Tk in the implementation of the Sequencer, a tool to support the preparation of an observing program ahead of time, introducing some logic between the observations to be done. This should play an important role in the way to operate the VLT telescopes, where astronomers will not normally be present and will have to describe accurately their observations to a service observer. Tcl, complemented with a number of commands to access message and database services, is then the internal control language of the VLT [6].

Another tool based on Tcl/Tk is the Panel editor (see Figure 1), which allows users to create homogeneous interfaces for telescopes and instruments on the VLT. The basic concept is that it should be easy to develop and change a control panel, but that this should be done according to a given set of conventions. After writing down these conventions, it was then clear that the only way to enforce them would have been to have an editor capable of producing control panels according to them, without requiring programming. The Panel editor, when used within the VLT environment, contains all the hooks to the message system and the database, which are needed.

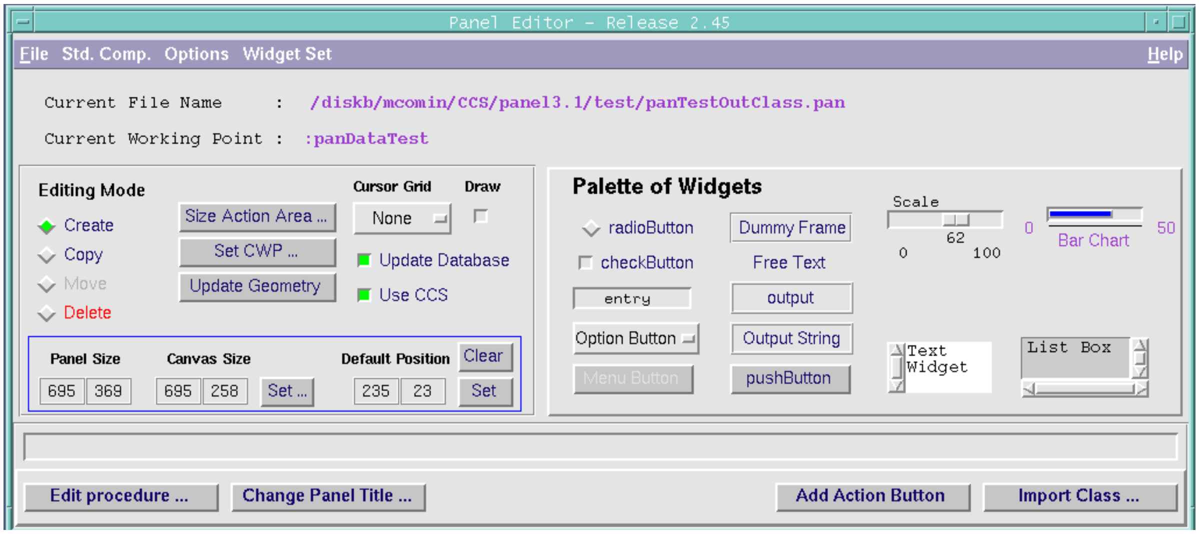

Figure 1 The Panel editor: a tool to develop control panels enforcing a set of conventions.
A stand-alone demo version of the Panel editor can be obtained by the author M. Comin, **mcomin@eso.org**.

7 SOFTWARE ENGINEERING

Software engineering plays an important role in the VLT control software developments.

At first this seemed to be necessarily linked to the introduction of appropriate CASE tools, but after an initial period it was clear that people did not find it an advantage to write functional specifications or design documents according to a given tool. It was a difficult job also for reviewers to get to terms with these tools.

What did remain from this initial phase is rather the correct approach to software development, whereby every software package must follow a given cycle, even if based more on paper and English than on CASE tools, apart from drawings. This means that the functional specifications and design documents must be reviewed and that there must be test procedures and user manuals before a software product is accepted. Measures such as written comment procedures preliminary to reviews, enforcement of standards and lately the introduction of automatic test procedures are all important steps to obtain a coherent and reliable result.

Developers by now realise that all this is right and necessary if they work in a large team, and it has been applied to the VLT common software. We would not have been in a position to distribute that amount of code at the pace of a release every 6 months, with plenty of new functionality each time, without moving more and more towards a more rigorous way to develop software.

The maintenance activity for the common software is taking quite some time, while we have still to develop most of the application software. We use for this a system of SPRs (Software Problem reports or Change requests) and we have already more than 500 this year.

A complete set of specifications and user manuals exists for the VLT software and for the VLT common software in particular. The user manuals by themselves already fill three thick folders.

A first assessment of the early phase of these activities was given at ICALEPCS 93 [5] and it will be interesting to follow the progress in the life of the VLT software.

8 CONCLUSION

The evolution of concepts in the VLT control software has been explained. The experience gained so far has normally been put immediately to use in the next phase of the project. Although a number of developments have already been based on the VLT common software, most of the feedback has still to come, with most of the commissioning activities to be done.

The goal of maintaining consistency in a very distributed development where different groups are involved, has been achieved so far, but integration will show how successful this was. This has not been based on a set of unchangeable rules and standards, but on a pragmatic approach of following innovative techniques when affordable, but avoiding on purpose to be really the first ones to try them out. This way of keeping in touch with evolution, while being conscious of the priorities and deadlines of our project, is also necessary when long commissioning times spanning several years are involved.

For people who want to know more about the VLT project, access to ESO VLT software documentation can be gained via the Web (under **https://www.eso.org**).

9 ACKNOWLEDGMENTS

It is a pleasure for me to acknowledge the contributions of my colleagues in the VLT software group who have been the authors of most of the software that I have mentioned in this paper.

10 REFERENCES

[1] G.Raffi - Control Software for the ESO VLT - Proc. of ICALEPCS, Tsukuba, 1991, KEK Proc. 92-15

[2] G. Chiozzi - An object-oriented event-driven architecture for the VLT Telescope Control Software - These Proceedings

[3] B.Gilli - Workstation environment for the VLT - Proc. of SPIE, vol.2199, pp.1026-1033, 1994

[4] B.Gustafsson - VLT Local Control Unit Real Time Environment - Proc. of SPIE, vol.2199, pp.1014-1025, 1994

[5] G.Filippi - Software engineering for ESO's VLT project - Proc. of ICALEPCS, pp.386-389, Berlin, 1993

[6] E.Allaert, G.Raffi - The VLT control software - Local and remote support for operations by astronomers and operators - ESO Technical preprint No. 69, 1995