

## What's in the labs

The Sequencer interpreter is based on the Tcl/Tk core as explained in 3. The versions we use are 7.3 for Tcl and 3.6 for Tk. On top of these the Sequencer interpreter includes currently the following extensions:

1. *Extended Tcl/Tk*: this extension adds several commands useful in a Unix context; its version has to match the one of Tcl/Tk, as it patches this core.
2. *BLT*: the Bell Labs Toolkit adds a few commands to the Tk library. Presently at version 1.7.
3. *[incr Tcl]* or *iTcl*: this extension provides an object oriented programming environment in Tcl/Tk, which allows and promotes organisation of the scripting code in a much cleaner way than plain Tcl does. iTcl is at version 1.5.

A few months ago new versions of Tcl and Tk were released (7.4 resp. 4.0), which provide more functionality, better Motif compliance and tons of other modifications and improvements. Of course the Sequencer interpreter will be updated to these new releases of Tcl/Tk, but we also need matching versions of all the extensions included in our interpreter. In particular we are waiting for the 2.0 version of iTcl<sup>2</sup>, which should bring a major performance improvement as a bonus. That should manifest itself a.o. in all applications created with the Panel Editor.

Once we have the needed extensions, we will still have to address the compatibility issues: several VLT applications have been implemented as Sequencer scripts, and they may be affected by modifications in Tk's behaviour or syntax.

We expect to have this update for the Sequencer interpreter ready for internal use early 1996, and to release it externally with the next subsequent VLT Common Software Release.

## References

- Ousterhout J., 1994, *Tcl and the Tk toolkit*, ISBN 0-201-6337-X.  
Raffi G., 1995, *The Messenger* 81, 5.  
Welch B., 1995, *Practical Programming in Tcl and Tk*, ISBN 0-13-182007-9.

---

<sup>2</sup>[incr Tcl] version 2.0 should be available by the time you read this.

# The VLT CCD Detectors Control Software<sup>1</sup>

A. LONGINOTTI, C. CUMANI, P. DUHOUX, ESO

## Introduction

Charge Coupled Devices (CCD) are currently by far the most widely used type of detectors in Astronomy. It is foreseen to have in operation at the VLT about 40 technical CCD cameras for auto-guiding, field viewing and wavefront sensing, and probably more than 10 scientific CCD cameras for instruments working in the optical spectral range. Because of the critical role it plays, it has been decided that the CCD Software Package should be the very first VLT application to be built and tested on top of the VLT Common Software. Because of that, the CCD Software has provided, together with the NTT upgrade software, a considerable feedback to the VLT common software in terms of bug detection and suggestions for improvements.

It has also been decided to put in the development life-cycle a prototype stage, to allow a better evaluation of the validity of the whole camera concept, including, of course, its control software.

A field test has been successfully performed January 1995; from that time efforts have been concentrated in adding functionality and bringing the whole package to VLT standards, both in terms of software quality (quality of code, documentation, configuration control) and of general rules applicable to all VLT software applications (usage of standard commands, libraries, etc.).

The first release of the CCD Control Software, originally planned for August 1995, has been recently sent out (October 1995). It contains a sub-set of the foreseen functionality, and does not support technical CCD cameras yet (this will be done with the next release). On the other hand, it defines the complete interface towards external software.

## Baseline development criteria

The development of the CCD Control Software has been based on the following criteria:

- It must fit in the general VLT control architecture (distributed system consisting of Workstations, for high level operations and user interface, and Local Control Units, for real-time and sub-system hardware related operations). For more information, see the article of G. Raffi about the VLT control

software in the Messenger, issue September 1995.

- It must use, wherever possible and compatible with the performance requirements, components provided by the VLT common software.
- It must provide a programmatic interface to all packages (instruments and telescope control software) using CCD cameras, but also be able to work as a stand-alone simple instrument, mainly for test purposes.
- It must interface with the already defined and developed transputer-based CCD controller box, also called Array Control Electronics (ACE).
- CCDs being a technological area in very rapid progress (bigger and faster chips are continuously coming up on the market), it must have an interface to the hardware as flexible as possible, in order to accommodate possible upgrades of the ACE-based or totally new CCD controllers, keeping the bulk of this package independent from the particular controller used.
- In order to simplify its maintenance, the CCD software must be one package, providing the functionality required for both technical and scientific cameras. Differences in the control software between these two categories must reflect only the differences in

---

<sup>1</sup>This article is part of a series of regular reports on VLT Control Software, which started with the previous issue of the Messenger. (Raffi 1995)

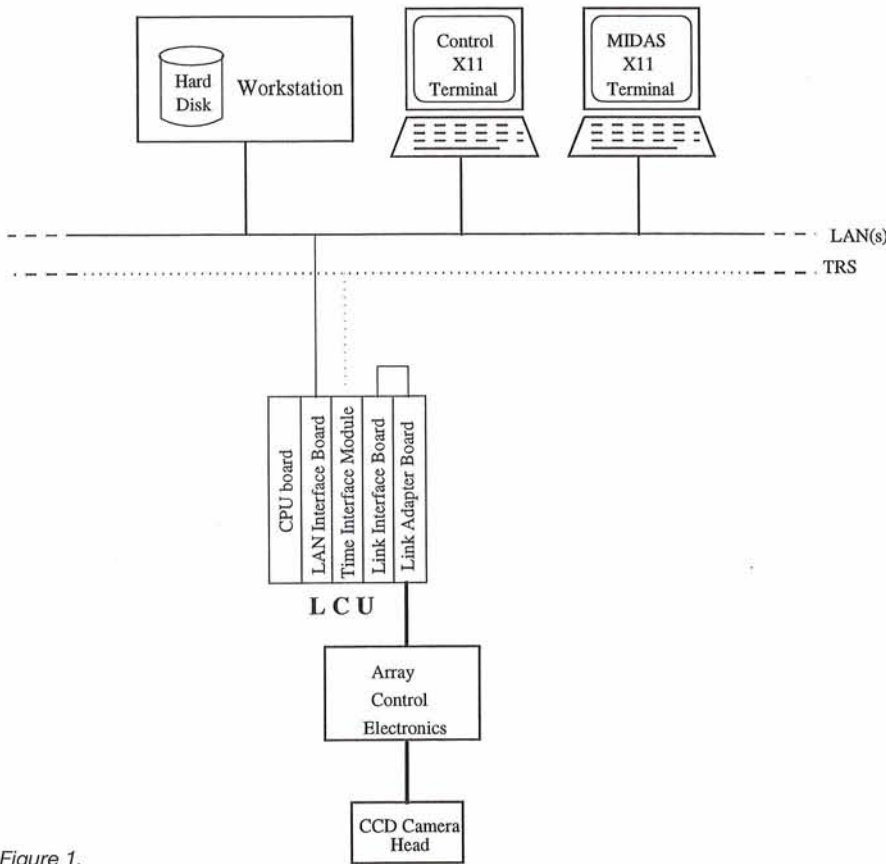


Figure 1.

the control electronics, and therefore be confined within ACE.

- The ACE transputer and DSP software for technical CCDs being developed by an external contractor, much attention and emphasis has to be put on the specification and implementation of the LCU-transputer interface.

## Architecture

The current version of the CCD software runs on three platforms:

- Workstation, where both programmatic and interactive user interface are located. The language used is C, whereby Tcl/Tk scripts are used within GUI panels.
- LCU, where the core of the control is implemented. It has been designed to simplify as much as possible the adaptation to other controllers. About 80% of the whole software is independent from the particular controller connected to the LCU VME crate. Basically one function library, called Device Control Library, provides the interface to ACE, and only this component must be replaced if a different type of controller is used. The language used is C.
- ACE, where the transputer and DSP based embedded software runs. It provides the direct interface to the camera electronics. The languages used are Occam (transputers) and C (DSP).

A schematic view of the architecture of a CCD stand-alone system is provided in figure 1.

## Functionality

The main functionality provided by the CCD software is:

- Single exposure execution (possibly repeated n times), with selectable readout parameters: binning, windowing, speed, gain, outputs used.
- Images are provided both as FITS files and as raw data, for quick look through the VLT Real-Time Display facility.
- A real-time image processing facility on LCU is foreseen, to fulfill the requirements for technical CCDs (centro-riding calculation, object recognition). This functionality is not implemented yet.
- A telemetry information system is foreseen, but not implemented yet.
- A temperature control system is foreseen, but not implemented yet.
- Besides the normal operational mode, simulation operational modes are provided, at all stages of the control architecture, in order to allow the integration and tests of this package in environments where the hardware needed is not fully available:
  - WS simulation. The whole LCU software is simulated on Workstation
  - LCU simulation. The ACE software is simulated on LCU.
  - ACE simulation. The hardware access is simulated in the transputer software

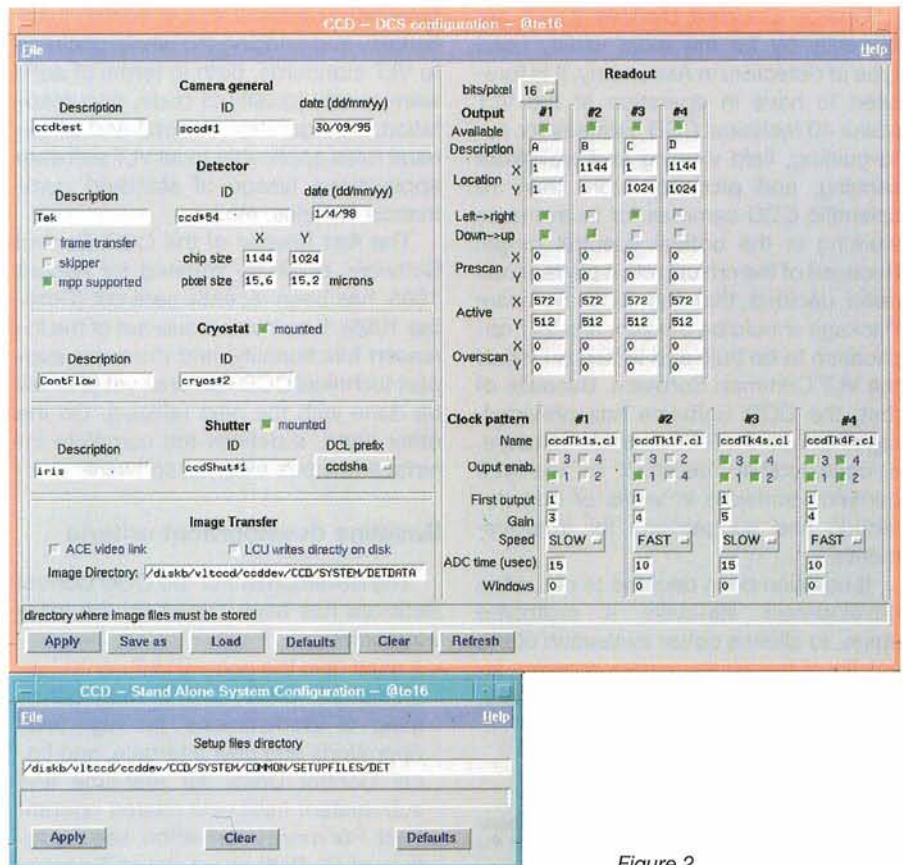


Figure 2.



Figure 3.

## Usage by external software packages

A general software development criterion, which we tried to apply also to the CCD Software, is to make the interface to external software as simple as possible, using as much as possible standard components. The interface to external software consists of:

- Command/Replies, based on the VLT Software Message System, to/from one CCD Command Interface process, running on Workstation. All actions associated with an exposure, including the definition of the exposure setup, are performed through commands.
- Data. They are basically of four types:
  - Camera configuration data. They are stored in the on-line database and can be retrieved through a public function provided by the CCD software. In order to simplify the operation of setting the configuration values

for a specific camera (this operation is done normally by people familiar with CCD cameras, but NOT with software), one configuration GUI panel is provided (see Figure 2).

- Exposure setup parameters. They are stored in ASCII files, called setup files, as specified by the Instrumentation Common Software.
- Image Data. They are delivered by the CCD software as FITS files, to be used later for further analysis by any image analysis package, and/or, according to the setup definition, as raw data in shared memory for quick look through the VLT Real-Time Display facility (see also article by T.Herlin et al. in the September 1995 issue of the Messenger).
- Real-time image processing results. They are used mainly by applications using technical CCDs, such as auto-guiding software. For performance reasons, they are stored directly in the LCU on-line database.

## Usage as stand-alone instrument

As already mentioned before, the CCD Software must be able to work as simple stand-alone instrument. It must therefore implement on Workstation a simplified version of those modules foreseen for any VLT Instrument: Observation Software (OS), Observer Support Software (OSS), Maintenance Software (MS). The current release provides a set of panels to allow the following (see also Figure 3):

- startup/shutdown the system
  - define a single exposure setup and save it in a setup file
  - start an exposure, monitor its status, possibly pause or abort it.
  - display the results (image) with the VLT Real-time Image Display facility.
- Foreseen, but not implemented yet, are:
- interface to on-line Midas.
  - definition of sequence of exposures and automatic execution (useful for

standard calibration operations, such as linearity tests).  
– interface to VLT Archive.

## Acknowledgments

The authors would like to take this opportunity to thank all people who helped in the development of the VLT CCD Control Software up to the current stage:

- Norbert Fiebig, consultant, who implemented the LCU part of the CCD software for the prototype version.
- all members of the VLT Software Group, for their collaboration in assisting us in their area of competence.
- all members of the newly formed Optical Detector Team, for their valuable contribution in terms of comments to the prototype version.

---

For more information or questions on the VLT CCD Control Software, please contact, preferably by e-mail, A. Longinotti (alongino@eso.org). Additional information can also be found in the anonymous ftp area of the VLT Software Group, which is accessible via WWW (Eso Home page → VLTProject → VLT Software and Documentation → doc → ccd\*)

# Restructuring La Silla

*D. HOFSTADT, ESO*

During the past months the La Silla staff which is closely involved with the telescope activities has been asked to reflect on ways to re-organise the Observatory operations.

La Silla has a long tradition of extensive and user-friendly support and enjoys a large record of praise from its visitors.

## If it works, why change it?

There are essentially three main factors which call for a change:

1. The VLT operation era is nearing. We have still an opportunity to try out different and more ambitious operation modes at La Silla and let the VLT benefit from the experience.
2. The budgetary pressure has become stronger and stronger. Cost control has been emphatically requested by our Governing Bodies. Cost efficient priorities have to be worked out and implemented.
3. The number of facilities and services, under ESO's direct supervision at La Silla, has decreased.  
This calls for a re-assignment of the staffing resources at the Observatory.

It is tempting to breakdown the La Silla operations into Telescope Teams. Undoubtedly, great benefit is to be obtained if the teams and their telescope merge into a common identification.

This has been achieved at SEST since it was brought into operations. Also at the NTT, D. Baade and his colleagues managed to build-up a team well integrated and identified with its telescope.

## The Change

It was decided to create two additional teams for the support of the remaining telescopes:

- one for the 3.6m/CAT

- one for the 2.2m/1.54m/1.52m.

The Schmidt, the 1m, the 50 cm Danish and the Bochum telescope already have separate operational arrangements.

For the time being the Dutch and 50cm ESO telescopes would receive technical support from the 2.2 m/1.5 m team.

J. Melnick took on the task of crystallising and elaborating a plan for the restructuring, which he called the "Team Theme".

Much input was provided by the scientific and technical staff at La Silla.

The Team Theme was developed in a number of working groups. All of them focused on the self managed team concept organised around a functional process.

Integrated by astronomers, night assistants, operation and electronic staff, the telescope teams are to be self-reliant for the daily support. This includes scientific support, instrumentation setup and control, data management and the first aid technical assistance.

In general the welfare of a particular facility will be exclusively in the hands of a team.

The teams will also handle their internal administration for tasks and personnel scheduling, planning, reporting and training. They will ensure the operational interface with their visitors.

The teams are also to be linked with Garching for extended technical support and commitments.

In view of the large range of work and the limited manpower it was not feasible to provide for complete technical autonomy in each team.

Six technical sections are maintained and regrouped as support teams in the areas of software, general electronics, mechanics, optics, IR detectors and optical detectors.

They will provide the specialised skills, maintain units which are common

to all the telescopes (e.g. CCD cameras) and execute long term upgrade projects.

Both the telescope and support teams have similar structures.

The intent is to have each team with extensive freedom of self-management. Each team has a manager assisted by a deputy. Their role is to plan and manage the resources, schedule operational and technical tasks, monitor and report the performance of the team. The managers, however, are an integral part of the support and not limited to supervisory functions. They report to the Observatory management composed of a Director and his deputy.

The Observatory management is essentially responsible for the application of the scientific objectives.

It defines the goals and objectives, harmonizes the structures, policies and operational procedures for the teams and ensures a smooth interface and integration with the other divisions, both in Europe and in Chile.

In parallel with the merging of the Astronomy and TRS Departments into a common project, the Maintenance and Construction Department has undergone changes.

Here the objective was extensively focused on cost efficiency. As the host country Chile has considerably developed during the last few years, many services, which ESO operated internally, became available at much lower costs. Maintenance for the buildings, roads and power lines, has been outsourced. The plan is to keep a limited in-house support for emergencies and the maintenance of electro-mechanical equipment, which is highly diversified.

## Will it work?

The re-engineering experts would propose a strength-weakness-opportunity-threat analysis.