SOFTWARE FOR THE ESO NEW GENERAL DETECTOR CONTROLLER

Claudio Cumani, Andrea Balestra, Joerg Stegmeier
European Southern Observatory, Karl-Schwarzschild-Str. 2, 85748 Garching, Germany

Abstract: An introductory overview is given of the control software for NGC, the ESO New General detector Controller, which will handle the detectors of both optical and infrared new instruments at the ESO telescopes.

Key words: CCDs, optical and infrared detectors, detector controllers, software

1. INTRODUCTION

NGC is the ESO New General detector Controller, designed to handle the detectors of both optical and infrared instruments, for scientific imaging as well as advanced signal sensing applications. It is being developed on the basis of the experience with the actual ESO standard controllers - FIERA for the optical domain (Beletic, 1998 and Cumani, 1998) and IRACE for the infrared (Meyer, 1998) - and the requirements coming from the new generation of detectors.

2. NGC HARDWARE

NGC is a modular system consisting of back-end and front-end modules based on the XILINX Virtex Pro FPGA XC2VP7 FF 672. The back-end modules are connected to the acquisition computer (the so called LCU - Local Control Unit) via a 64 bit PCI bus interface

The front-end modules – which create and receive the detector signals – are connected to the back-end modules via fiber optic links with transmission rates of 2.5 Gbit/s. Clock signals for the detectors are generated by the sequencers which are completely contained within the FPGA of the front-end modules. The communication between all system modules is based on packet transmission over high speed serial links (see Meyer et al. and Reyes et al., these proceedings).

3. NGC SOFTWARE REQUIREMENTS

3.1 High-level requirements

The VLTSW (Very Large Telescope control SoftWare) environment defines the software standards to be used or fulfilled, in terms of programs, utilities, protocols and interfaces. At high level, the NCG software must be able to replace the FIERA and IRACE software in a completely transparent way: all the interactions with the instrument and telescope software will be performed in the same way as for FIERA and IRACE (VLT message, error and logging system, SETUP files and image files in FITS format, and so on).

The experience has shown that it is important that the same code which is used for operation at the telescope is also used in the laboratory for rapid prototyping, testing and calibration. This requires a detector control software which is flexible enough to let the engineers test a system deeply and extensively, but at the same time defines well and carefully different levels of operational freedom, to ensure safety during the normal operation.

As defined by the VLT paradigm, operations through Observation Blocks will drive the NGC software at the telescope, adding one more level of protection from operational errors.

3.2 Intermediate-level requirements

Particularly at the LCU level the merging of optical and infrared detector controllers is an interesting challenge also under the software point of view. The NGC software must consider the different requirements coming from the different domains:

- huge on-the-fly processing of the acquired data (infrared detectors)
- shutter handling and cryostat vacuum and temperature control (optical detectors - see C.Geimer et al., these proceedings)
- strict timing requirements (interferometry or high time resolution imaging or spectroscopy - see Cumani, 2000)

- synchronization with external signals (multiple cameras, va-et-vien, nodding, chopping)
- interface to real-time-computers (adaptive optics)
- fast image assembly and storage in FITS files (big detector mosaics)
- continuous readout and storage in FITS files (drift-scanning).

All these functionalities must be achieved along with stiff performance requirements.

3.3 Low-level requirements

The interface with the NGC modules defines the requirements for the low-level part of the NGC software.

Signals to be sent to the detectors must be loaded into the sequencer RAM of the FPGA of each so called NGC front-end "Basic Module", following a well defined structure (see Meyer et al., these proceedings).

4. NGC SOFTWARE DEVELOPMENT ENVIRONMENT

The EUP (Entreprise Unified Process) iterative approach is being used, following the standard development phases of requirements, analysis&design, development, testing, commissioning.

Extensive usage of “Use cases” has been the basis for the software design.

In order to guarantee reusability, the use cases and design patterns which are being defined for NGC should become the future "building blocks" in the ESO detector control software area.

DOORS is being used during the life-cycle of the NGC software. DOORS is a widely used requirement management tool, which allows an easy management of all the documents, with mutual references and easy traceability (requirements vs. design vs. implementation vs. test).

The possibility of using XML language is currently being explored. XML may be used for the configuration description of systems. In a later stage it may also be conceivable to use XML files for code generation.

Version control is enforced by using standard ESO tools (e.g., CMM).

5. NGC SOFTWARE DEVELOPMENT AND STATUS

The core of the NGC software is developed with C/C++, while Java and Tcl/tk are used for graphical user interfaces or standalone applications.

The NGC LCU will run Linux. At the moment, a device driver for Red Hat Linux 2.4.20 has been developed for the NGC prototype, together with part of the NGC Base Software, which contains functionalities such as a transparent threads interface, priority control, etc. Porting to Scientific Linux and kernel 2.6.x is under way.

The pre-processing framework for the multi-threaded Acquisition Process - which is required only for IR applications - has been taken over from IRACE.

The Sequencer is the code used to load elementary detector timings (stored in patterns), voltages (stored in configuration files) and programs into the sequencer of the NCG. Patterns can be executed in series and/or loops, at different speeds (i.e., clock frequencies), with different intervals. Changes in pattern execution can be performed on the fly.

A graphic tool for the pattern creation is under development.

A Controller Interface provides modular objects to control the Sequencer and the ADCs on the front-end modules, for interfacing to the Acquisition Process and for the Asynchronous Data Reception. The objects can be assembled in an arbitrary way to reflect all functionalities of any NGC hardware configuration.

Test code is developed in parallel as integrated part of the NGC software and is integrated in the ESO modular test procedures.

Concerning the code on the LCU, analysis has shown that the differences between optical and infrared operations suggest different designs for infrared and visual in order to achieve higher efficiency and to avoid unneeded complexity.

The goal is a control software which is robust, modular (for fast prototyping and easy maintenance), safe (the same software must run in the laboratory and at the telescope, with different levels of operational freedom) and flexible (to handle any future development in hardware and software).

References

Beletic, J.W., Gerdes, R., DuVarney, R.C., *FIERA: ESO's New Generation CCD Controller*, Kluwer, ASSL, vol. 228, pp. 103-114

Cumani, C., Donaldson, R., 1998, *The Architecture for two Generations of ESO VLT CCD Controllers*, Kluwer, ASSL, vol. 228, pp. 115-122

Cumani, C., Mantel, C. H., 2000, *Phase resolved High Speed Photometry and Spectroscopy of Pulsars*, Kluwer, ASSL, vol. 252, pp. 311-317

Meyer, M., Mehrgan, H., Nicolini, G., Stegmeier, J., 1998, *The ESO Infrared Detector High Speed Array Control and Processing Electronic IRACE*, SPIE, vol. 3354, pp. 134-138

ESO VLT Software Project: http://www.eso.org/projects/vlt/sw-dev/

Telelogic DOORS: http://www.telelogic.com/products/doorsers/index.cfm