INAF-OAA Gruppo Strumentazione Infrarossa

Giano Project
Overview of Low Level Software

C.Baffa, E.Giani

Version 1.0, Firenze 15 October 2013
Arcetri Technical Report, 6/2013

Abstract

We briefly describe the structure and interconnections of Giano Low Level Software (LLS). We will also give all the references needed to get a deeper insight on the project.

1 General structure

Giano is an advanced hi-resolution ($\lambda/\Delta\lambda \sim 50000$) infrared spectrometer with the capacity to measure the spectra of an astronomical source in the wavelength interval of 0.9 to 2.4μm in a single exposure[3, 4].

To accomplish such a goal Giano needs a complex structure, a careful optimization and a continuous control[5]. An important part of this process is the *Low Level Software*, which initializes, programs and continuously monitors Giano system.

The *LLS* is structured as few interacting independent programs (in Unix jargon called *daemons*[8]), which interact using a *client–server* paradigm. The general structure is sketched in fig 1 (from [14]).

This means the control chain, as usual, is structured linearly: the astronomer interacts with a `Giano GUI`, which accept commands with *astronomical meaning*. The `Giano GUI` trasmits the high level commands to one central demon, `Balor`, which acts as *middle-ware*. `Balor` translates these orders in few *instrument level* commands and sends these to the other *LLS* daemons and receives back the status of the subsystems.

As far as possible, the communication protocol[28] was the same for all communications, with a general structure which mimic the lower level Internet protocols[6]

2 The Software Daemons

`Balor`, the middle-ware

`Balor` (previously named `Gbridge`) is the central node of *LLS*. This daemon is always

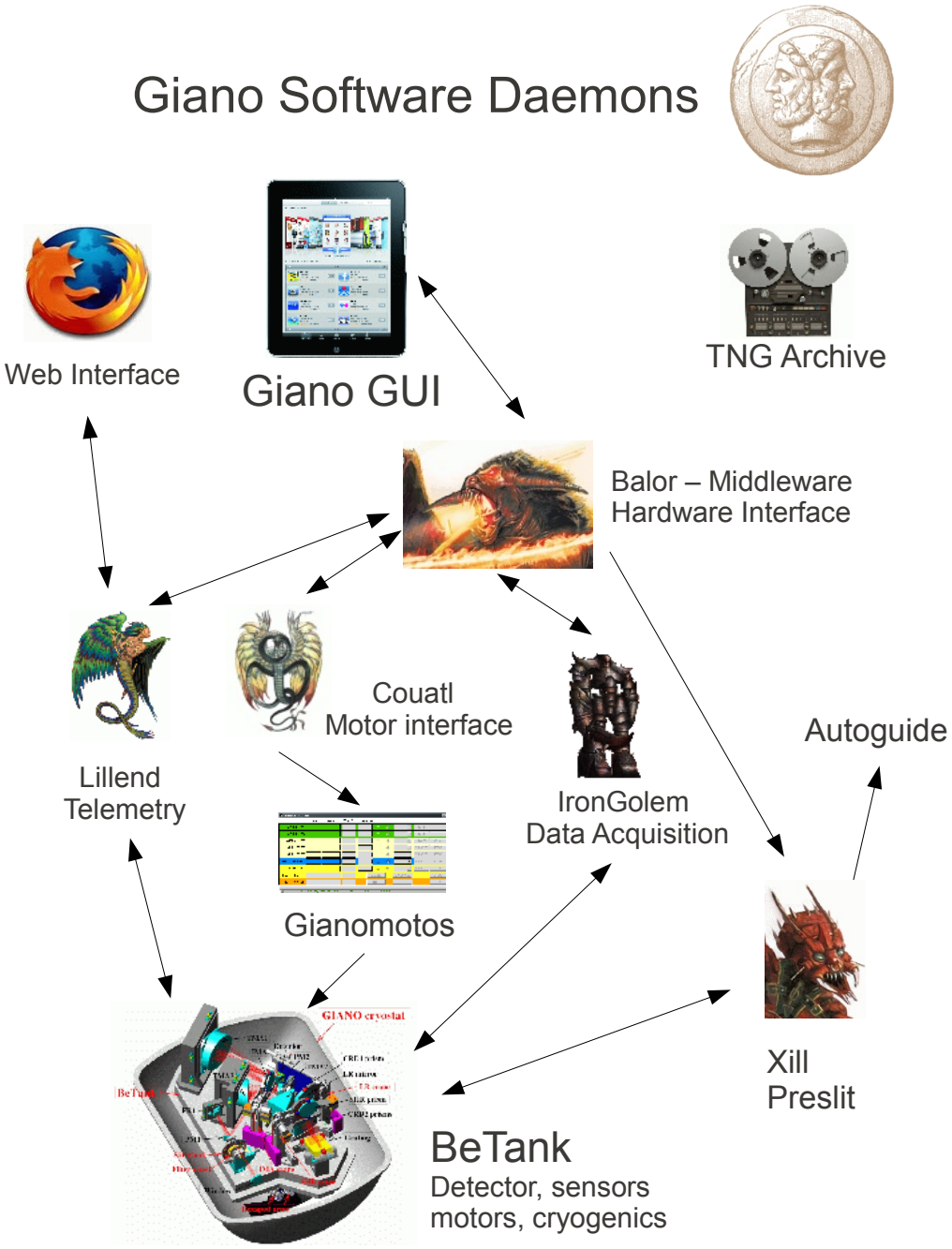

Figure 1: Giano *LLS* Daemons

running, and after start-up it waits for the `Giano GUI`. When the `Giano GUI` connects, `Balor` starts the initialization of all subsystems and, if needed, it launches missing daemons. When in observation mode, `Balor` gets command from the observer, by means of the `Giano GUI`, and translates them into smaller, hardware related, commands to send to lower level programs. In return `Balor` gets astronomical data, which are stored as `FITS` files. `Balor` also gets the status of all subsystems and generate a *system level* status available to the `Giano GUI` and to the TNG archive via `Lillend` daemon[17].

`IronGolem`, the detector controller

Giano is equipped with a complex and flexible data acquisition electronics[18, 19, 20, 21] to handle the different operational mode available with the Hawaii2 detector[7], matching the different measurements modes implemented now or foreseen as extensions. `IronGolem`[10] (previously known as `Server104`) handles all the necessary housekeeping and programming for the cold and room temperature parts of the acquisition system, gets the scientific data, and sends them to `Balor`, to be stored as FITS file. This daemon runs on a dedicated embedded system[25], strictly coupled with the hardware by means of a `PC104` bus and its status can be monitored by means of a 4 rows LED display on the electronics box.

`Couatl`, the motion handler

The motor control structure is somehow more complex. For historical reasons, the detailed motor handling is performed by a `tcl/tk` application, `GianoMotors`[13]. The interface with the general Giano *LLS* structure is guaranteed by the `Couatl` daemon. `Couatl`, being used during normal operations, can perform only a subset of available movements, while set-up and maintenance tasks can be performed using `GianoMotors`.

`Lillend`, the telemetry collector

The Giano phisical status (inside pressure, temperature, nitrogen level, ecc.) is continuously monitored and all data collected and stored in a database by the `Lillend`[27] daemon. This process also grabs the system global status variables collected by `Balor`, and stores the appropriate values in the TNG database. All physical data are continuously available by means of a web interface, with selectable temporal windows[16]. The web pages are accessibles inside the TNG network at the address `http://gianopc/lillend`, and are mirrored, from time to time, at `http://tirgo.arcetri.astro.it/lillend`.

`Xill`, the pre-slit daemon

The pre-slit optics comprises two main functions: the optical connection with the TNG telescope and the wavelength calibration. This portion is controlled by the `Xill`[15] daemon, which runs on a dedicated embedded processor, which shares the same name. This daemon gets its commands from `Balor` and responds with the subsystems status in real time. `Xill` interacts also with the autoguider GUI program. This daemon handles many devices, so, to avoid long execution delays, it is structured as a many threads task[26].

3 The comunication protocol

For the *LLS* of the recent Infrared instrument developed at Arcetri we use a robust communication protocol modeled on the low level Internet protocols[6]. For ease of implementation, we developed a C-language library, `Irsock`[28], used on all daemons.

From a logical point of view our protocol is composed by two distinct part: the protocol format[9], which defines the structure of the messages sent, and the packet contents, which is specific to each channel.

The use of our protocol is less straightforward with the fourth generation language used for the development of the Giano `Giano GUI`, so we developed a simplified *content* version for this specific use, detailed in [24, 23]

4 Conclusions

The Giano *Low Level Software* is a large project, composed by five main daemons, many support or ancillary programs and several active web pages. The code accounts for a total of more than 115000 lines of code and is documented in many memos and several technical reports, as of September 2013.

*LLS*was designed and developed having in mind robustness and ease of maintenance. The complexities of the instrument and of his history sometimes forced us to some compromises and so the final result still needs some refinement, but the overall result should be regarded as satisfactory.

References

[1] "**Progetto Giano**. Protocollo di comunicazione tra `Gbridge` e `Giano GUI`", Vers 1.08, 2007, C.Baffa, E.Giani, Memo.

[2] "**Progetto Giano**. Protocollo di comunicazione tra tra `Gbridge` e `Giano IDL-GUI`", E.Giani, C.Baffa, 2012, Rapporto Interno di Arcetri N° 2-2012,

[3] "GIANO: an ultrastable IR echelle spectrometer optimized for high-precision radial velocity measurements and for high-throughput low-resolution spectroscopy", Oliva, E., et al., 2004, SPIE conference "Ground-based Instrumentation for Astronomy", eds. A.F.M. Moorwood & M. Iye, vol. 5492, pp. 1274-1279.

[4] "The GIANO-TNG spectrometer", Oliva, E., et al., 2006, SPIE conference "Ground-based and Airborne Instrumentation for Astronomy", eds. McLean, Ian S. & M. Iye, vol. 6269, pp. 626919

[5] "The versatile acquisition system of Giano", C. Baffa, et al., 2006, SPIE conference "Advanced Software and Control for Astronomy", eds. Lewis, Hilton & Bridger, Alan, Volume 6274, pp. 62740T

[6] "Internet Core Protocols: The Definitive Guide" E. Hall, Sebastopol, CA (USA), 2000.

[7] "User Guide for the HAWAII-2 2048x2048 Pixel Focal Plane Array", A. Haas, Rockwell Scientific Company, LLC, 2002.

[8] "Dungeons & Dragons - Monster Manual 3.5", 2003, J. Tweet, M. Cook e S. Williams, TSR.

[9] "Definizione del protocollo interno del software di basso livello", versione 1.9, 2006, C.Baffa, E.Giani, Memo

[10] "**Progetto Giano**. Il programma server104 ed il sistema embedded", 2010, C.Baffa, V. Biliotti, E. Giani, Arcetri Technical Report, 2/2010.

[11] "**Progetto Giano**. Interfacciamento con il programma gianoMotors: il server `Couatl`", E.Giani, C.Baffa, 2010, Arcetri Technical Report 6/2010

[12] "Gbridge: il software middle-ware di Giano", 2010, E.Giani, C.Baffa, Arcetri Technical Report 7/2010.

[13] "Progetto Giano. Inventario e memorandum del software di controllo", E.Rossetti, 2010, Memo.

[14] "Nuova struttura del software di basso livello", Giano Memo, 1/2011, C.Baffa, E.Giani

[15] "Xill: il software di controllo del sistema della pre-slit di Giano", 2011, E.Giani, C.Baffa, Arcetri Technical Report, 8-2011

[16] “Il programma di telemetria lillend - L’interfaccia WEB”, 2011, C.Baffa, E.Giani, Arcetri Technical Report 7/2011,

[17] “I file ausiliari di Giano”, E.Giani, C.Baffa, Memo, Novembre 2011

[18] “**Elettronica di Acquisizione di GIANO**, Descrizione d’insieme, 2012, V. Biliotti, Arcetri Technical Report 6/2012.

[19] “**Elettronica di Acquisizione di GIANO**, Parte analogica a temperatura criogenica, 2012, V. Biliotti, Arcetri Technical Report 7/2012.

[20] “**Elettronica di Acquisizione di GIANO**, Interfaccia digitale, 2012, V. Biliotti, C.Baffa, E.Giani, Arcetri Technical Report 8/2012.

[21] “**Elettronica di Acquisizione di GIANO**, Parte analogica a temperatura ambiente, 2012, V. Biliotti, Arcetri Technical Report 9/2012.

[22] “Comandi sintetici per la GUI IDL di Giano”, E. Giani, C.Baffa, Memo Marzo 2012

[23] “Comandi sintetici di stato per le applicazioni server di Giano”, E.Giani, C.Baffa, Memo, Febbraio 2012

[24] “Protocollo per la comunicazione tra Balor/Gbridge e Giano-IDL”, E.Giani, C.Baffa, Memo, Marzo 2012

[25] “I sistemi embedded Porteus per Giano”, 2012, E.Giani, C.Baffa, Arcetri Technical Report 5/2012,

[26] “Xill: aggiornamenti al software di controllo del sistema della pre-slit di Giano”, 2013, E.Giani, C.Baffa, in preparation

[27] “Il programma di telemetria lillend - La struttura interna”, 2013, C.Baffa, E.Giani, Arcetri Technical Report, in preparation

[28] “La libreria Irsock”, 2013, E.Giani, in preparation

Contents