Analysis Framework for GLORIA

Aleksander Filip Żarnecki[a], Lech Wiktor Piotrowski[a], Lech Mankiewicz[b], Sebastian Małek[c]

[a]Faculty of Physics, University of Warsaw, Hoża 69, 00-681 Warsaw, Poland;
[b]Center for Theoretical Physics, Al. Lotnikow 32/46, 02-668 Warsaw, Poland;
[c]National Centre for Nuclear Research, Hoża 69, 00-681 Warsaw, Poland

ABSTRACT

GLORIA stands for "GLObal Robotic-telescopes Intelligent Array". GLORIA will be the first free and open-access network of robotic telescopes of the world. It will be a Web 2.0 environment where users can do research in astronomy by observing with robotic telescopes, and/or analyzing data that other users have acquired with GLORIA, or from other free access databases, like the European Virtual Observatory. GLORIA project will define free standards, protocols and methodology for controlling Robotic Telescopes and related instrumentation, for conducting so called on-line experiments by scheduling observations in the telescope network, and for conducting so-called off-line experiments based on the analysis of astronomical meta-data produced by GLORIA or other databases.

Luiza analysis framework for GLORIA was based on the Marlin package developed for the International Linear Collider (ILC), data analysis. HEP experiments have to deal with enormous amounts of data and distributed data analysis is a must, so the Marlin framework concept seemed to be well suited for GLORIA needs. The idea (and large parts of code) taken from Marlin is that every computing task is implemented as a processor (module) that analyzes the data stored in an internal data structure and created additional output is also added to that collection. The advantage of such a modular approach is to keep things as simple as possible. Every single step of the full analysis chain that goes eg. from raw images to light curves can be processed separately and the output of each step is still self consistent and can be fed in to the next step without any manipulation.

Keywords: Telescope network, image processing, data analysis

1. INTRODUCTION

GLORIA[1] (GLobal Robotic-telescope Intelligent Array) is an innovative citizen-science network of robotic observatories, which will give free access to the professional telescopes for a virtual community via the Internet. GLORIA project will develop free standards and tools for doing research in astronomy, both by observing with robotic telescopes and by analyzing data that other users have acquired with GLORIA and/or from other free access databases, such as the European Virtual Observatory. Dedicated tools will be implemented for designing and running so called off-line experiments, based on analysis of available data. Many different types of experiments are considered, for example classification of variable stars, search for optical transient or searches for occultations of stars by solar system objects.

One of the challenges we have to face in designing environment for GLORIA off-line experiments is dealing with huge amounts of data and large variety of analysis tasks. We need an analysis framework which would be both very efficient and very flexible. These are requirements new to astronomy. However, High Energy Physics experiments deal with enormous amounts of data and complicated analysis tasks since many years. Experiments at CERN Large Hadron Collider read information from about 100 million electronic channels, which is equivalent to taking 100 MPixel image of the detector, every 50 ns (20 million times per second). Even after very strong (10^{-5}) on-line selection of events (using multi-level trigger systems) GBs of data are stored every second. Data analysis for LHC experiments has to be performed on the LHC Computing Grid (WLCG), which include currently about 170 000 TB of disk space and CPU power of about 1 800 000 HEP-SPEC06 units. Still, this analysis is only possible thanks to the custom designed, very efficient analysis software.

Send correspondence to A.F.Żarnecki, E-mail: zarnecki@fuw.edu.pl

Detectors at the future International Linear Collider (ILC), which is next generation e^+e^- collider under study, will deal with even larger "images". Although the project will not be realized before 2020, detailed studies of physics and detector concepts, as well as detector prototype tests are under way since many years. Large samples of Monte Carlo data have been already generated to test detector performance and analysis methods. Dedicated Marlin[2] framework has been developed for efficient data reconstruction (corresponding to image reduction in astronomy) and analysis. We decided to adopt this framework for the needs of data analysis in GLORIA.

2. BASIC CONCEPT

Marlin (Modular Analysis and Reconstruction for the LINear collider) is a simple modular application framework for development of reconstruction and analysis code for ILC. Data reconstruction and analysis should be divided into small, well defined steps, implemented as so called processors. Processors are grouped in modules, dedicated to particular tasks or types of analysis. As a lot of different groups worldwide are involved in the ILC project, the assumption was that the framework should allow distributed development of modules and to combine existing modules in a larger application according to the users needs. The crucial requirement is such an approach is that each step of the analysis have a well defined input and output data structure. In case of Marlin, all possible data classes, which can be exchanged between processors are defined in LCIO (Linear Collider I/O) data model. LCIO is used by almost all groups involved in linear collider detector studies and thus has become a de facto standard in software development. By defining universal data structures we make sure that different processors can be connected in a single analysis chain, i.e. exchange data and analysis results.

Also defined in the Marlin framework is the base class for a Marlin processor. It defines a set of standard callbacks that the user can implement in their subclasses. These callbacks are used to initialize the analysis chain, process subsequent sets of data and to conclude the analysis. A steering file mechanism allows to activate the needed processors and set their parameters at run time. The dedicated processor manager loads selected processors and calls their corresponding methods for subsequent steps of data analysis. An example of Marlin analysis chain for silicon pixel detectors, developed within the EUDET[3] project, is shown in Fig. 1. Processing of data from pixel detectors in High Energy Physics is in fact like CCD image analysis in astronomy. Charged particle tracks are measured instead of photons, but analysis steps are similar: raw data are read from file, pixel cluster are found (object finding), their position and charge are reconstructed (photometry) and used to fit the particle track (astrometry). As the Marlin framework turned out to be very efficient and flexible, and is widely used by the ILC community, we decided to use the same concept in development of the Luiza framework for GLORIA.

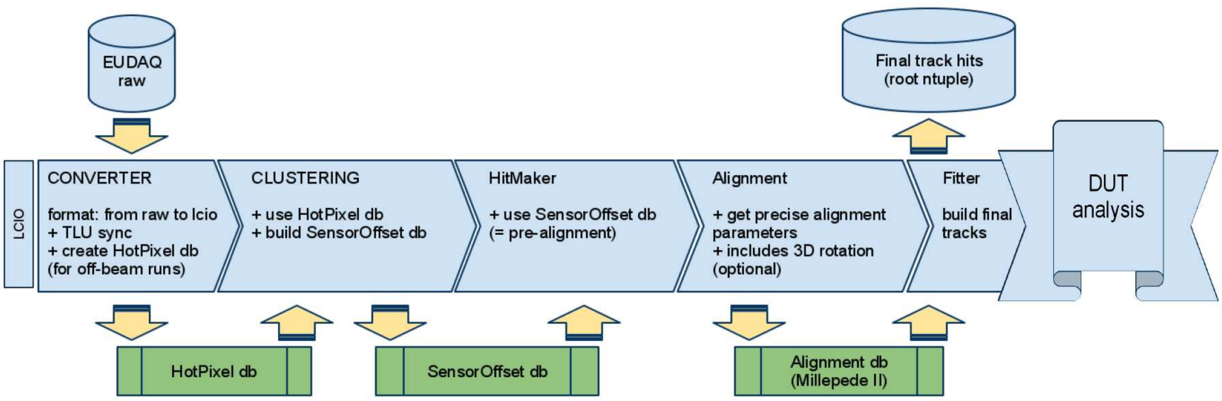

Figure 1. Example of Marlin analysis chain for MIMOSA silicon pixel detectors, developed within the EUDET project (EUTelescope package).

3. LUIZA FRAMEWORK

3.1 Data structures

FITS[4] (Flexible Image Transport System) is the standard astronomical data format endorsed by both NASA and the IAU. FITS is much more than an image format (such as JPG or GIF) and is primarily designed to store scientific data sets consisting of multi-dimensional arrays (1-D spectra, 2-D images or 3-D data cubes) and 2-dimensional tables containing rows and columns of data. When developing Luiza we decided to use CFITSIO[5] library for reading and writing data files in FITS format. Following basic data classes are defined:

GloriaFitsImage class for storing 2-dimensional FITS images, with either integer or floating point pixels. It includes also basic methods for image manipulation (addition, subtraction, multiplication and division);

GloriaFitsTable class for storing data tables. Different column types are allowed (integers, floats, strings, but also vectors of integers and floats);

GloriaFitsHeader class for storing FITS header data, includes basic methods for accessing and modifying header information. Both GloriaFitsImage and GloriaFitsTable inherit from this class.

Additional classes can be implemented on this basis, as for example **GloriaObjectList** class. This class defines GloriaFitsTable with predefined columns for storing object position on CCD ("CCD_X" and "CCD_Y") and object brightness ("Signal"). This makes sure that object lists will be exchangeable between processors. User can add additional columns if needed.

For internal storage of all data being processed a dedicated class **GloriaDataContainer** was implemented. It stores vectors, so called "collections", of images or tables. Each collection has a unique name (string), which can be used to access its elements. Multiple collections can be stored in memory, each with multiple images or tables (although in many cases collections will just contain single image or table). Pointer to GloriaDataContainer is passed to each Luiza processor in the data processing loop. Processors can analyse data already stored in memory, but can also add new collections (eg. when reading data from storage or saving analysis results).

3.2 Data processing

We assume that every computing task can be implemented as a processor (module) that analyzes the data stored in a GloriaDataContainer structure and created additional output is also added to that collection. User defines the analysis chain at run time, by specifying list of active processors in an XML steering file (see Fig. 2). The idea is to develope a large number of processors in GLORIA, doing many different tasks, so user is always able to find a set which matches his/her needs.

```
<execute>
    <processor name="DarkImageReader"/>
    <processor name="FlatImageReader"/>
    <processor name="RawImageReader"/>
    <processor name="CalibrateImage"/>
    <processor name="FindObjects"/>
    <processor name="DoAstrometry"/>
    <processor name="StoreFinalImage"/>
</execute>
```

Figure 2. Example steering file header, containing information on the modules selected for analysis chain.

The main "working horse" of Luiza is processor manager (ProcessorMgr class). It is used by Luiza to create a list of active processors (after parsing XML file), and setting values to parameters required by this processors (given in the same XML file). Same processor type (eg. processor reading fits images - FitsImageReader) can be used many times: the instances are distinguished by a unique names given by user. Each instance has its own set of parameters, so one instance of image reader can be used to read dark frames used for calibration and another instance to read actual images; example of parameter section for one processor is shown in Fig. 3.

```
<processor name="DarkImageReader" type="FitsImageReader">
  <!--Processor for reading input FITS images. Reads images from given files-->
  <!--List of FITS files to be read-->
  <parameter name="FitsFileList" type="StringVec"> dark.fit  </parameter>
  <!--Name of the image collection to which images from file should be stored-->
  <parameter name="ImageCollectionName" type="string">DarkImages </parameter>
  <!--Number of images to be read per processing loop (0 for all))-->
  <parameter name="ImagesPerLoop" type="int">0 </parameter>
  <!--Name of file containing FITS file list (one per line)-->
  <parameter name="ListFileName" type="string"> </parameter>
  <!--Flag for collections, which should not be deleted after loop is finished-->
  <parameter name="PermanentCollection" type="bool">true </parameter>
  <!--verbosity level for processor ("DEBUG,MESSAGE,WARNING,ERROR,SILENT")-->
  <parameter name="Verbosity" type="string"> MESSAGE </parameter>
</processor>
```

Figure 3. Example section of the steering file for Luiza, with parameters of the processor reading dark frames.

3.3 Analysis tools

So far we mainly focused on the development of the general structure and functionality of Luiza: data classes based on FITS standard were designed steer file parsing and processing management were adopted from Marlin, processors for input and output of FITS image files were implemented based on CFITSIO library. Nevertheless, current version of Luiza includes already some basic tools for image processing:

- image viewer based on CERN ROOT[6] package (see Fig. 4)
- image normalization processor, allowing for dark/bias subtraction and flat correction
- processor for image stacking or averaging
- processor for simple geometry operations: image cropping and rotations
- two simple object finding algorithms: one based on the particle identification algorithm developed for silicon pixel detectors and another one based on Python library Mahotas
- astrometry algorithm based on Astrometry.net (still under tests)

Moreover, a dedicated user interface, LuizaGUI, for creating and editing of XML steering files was prepared.

3.4 Development plans

We plan to continue development of basic tools for image manipulation and analysis (astrometry, photometry, light curve reconstruction). General purpose algorithms, which should be flexible enough to cope with data from all GLORIA telescopes, are not supposed to be the most precise ones. They can be used as examples and starting points for future improvements and development of more advanced tools, dedicated to particular studies. Dedicated processors will also be development as a part of GLORIA off-line experiments. Our current plans include development of:

- interface to star catalogues and external databases
- interface to Virtual Observatory resources
- processor for smart image stacking (correcting for image shifts and rotations)
- frame quality analysis

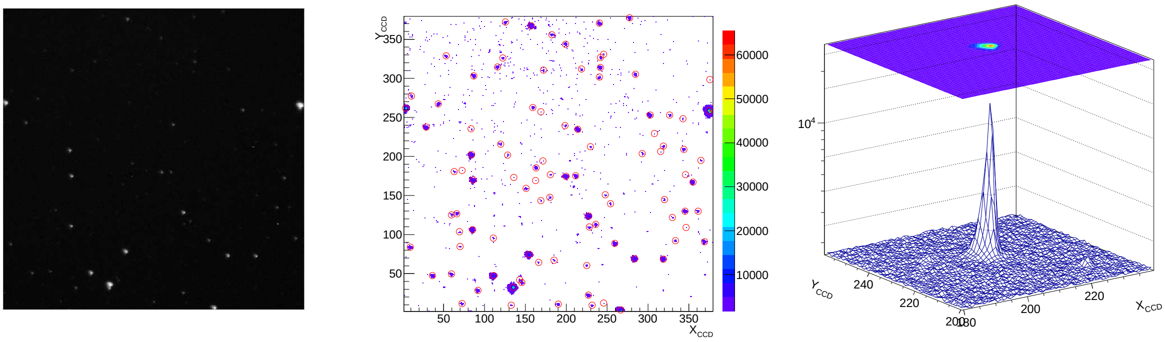

Figure 4. Different graphics options implemented in CERN ROOT, available for viewing FITS images in Luiza: as an image (left), as a histogram (middle), as a histogram in 3D projection (right). Red circles in the middle plot indicate objects reconstructed on an image by PixelClusterFinder processor. In the right plot, only the small section of the image is shown for clarity, presenting the PSF of the flare star RXJ0413.4-0139 at the outburst maximum.

- high quality aperture and profile photometries
- light curve determination and variability analysis
- search for optical bursts, flares etc.

Thanks to simple and modular structure of Luiza, individual GLORIA users will also be able to contribute to software development. New packages can be compiled as independent libraries and loaded dynamically at run time without the need to change anything in Luiza or other modules. It is therefor possible for users to develop a "private" Luiza modules and libraries, deducted to for their particular analysis, which can be later included in Luiza as a separate package after proper testing.

3.5 Documentation

We decided to use Doxygen[7] package to manage framework documentation. Web page and/or LaTeX documentation is created automatically from class header files, based on simple tags used in the comments included in the code. Additional work needed to keep the documentation up to date is minimal, assuming developers do put comments in the code anyway. This solution make it also straightforward to add the code submitted by users to the documentation. Documentation for the public Luiza release is available on the dedicated web page.[9]

4. RESULTS

On November 24, 2011, just before midnight, Pi of the Sky telescope located at INTA near Huleva, Spain, automatically recognized a new object on the sky. Unfortunately it was visible to our detector only for about one minute, fading fast below our limiting magnitudo. We asked Bootes group operating bigger telescope at the same site to make a follow up observation. Figure 5 shows the light curve of the object, later identified as a flare star RXJ0413.4-0139, as observed by Bootes-1 telescope.[8] We clearly see a secondary outburst, more than one magnitudo brighter than the first one (11.8^m at maximum) and much, much longer. Data analysis resulting in this light curve has been performed with Luiza.

5. CONCLUSIONS

An efficient and flexible analysis framework for GLORIA has beed developed based on the concept taken from the High Energy Physics. Basic data classes, framework structure and data processing functionality are implemented, as well as selected data processing algorithms. The framework will be further developed as a part of work on GLORIA off-line experiments. First public version of the framework has been released via GLORIA project SVN.[10]

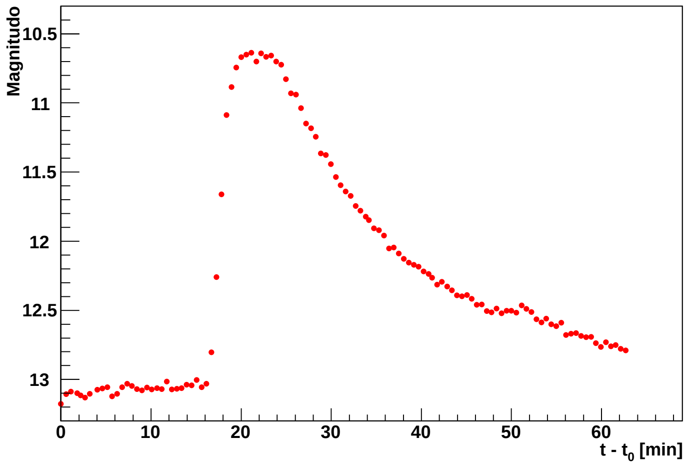

Figure 5. Lightcurve of flare star RXJ0413.4-0139 reconstructed with Luiza. Data from followup observation by Bootes-1 of the outburst observed by Pi of the on Nov. 24, 2011. Secondary outburst of the star was measured.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 283783. The research work is also financed by the Polish Ministry of Science and Higher Education in 2011-2014, as a co-funded international project.

REFERENCES

[1] `http://www.gloria-project.eu/`

[2] F.Gaede, “Marlin and LCCD: Software tools for the ILC”, *Nucl.Instrum.Meth.* **A559**, 177-180, 2006. `http://ilcsoft.desy.de/portal/software_packages/marlin/`

[3] I.Rubinskiy, “EUTelescope final status”, presented at *EUDET Annual Meeting 2010*, `http://ilcagenda.linearcollider.org/conferenceDisplay.py?confId=4649`

[4] `http://fits.gsfc.nasa.gov/`

[5] `http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html`

[6] `http://root.cern.ch/`

[7] `http://www.doxygen.org/`

[8] Martin Jelinek and Petr Kubanek, private communication.

[9] `http://hep.fuw.edu.pl/u/zarnecki/gloria/luiza/doc/html/index.html`

[10] `http://sourceforge.net/projects/gloriaproject/`