RevMexAA (Serie de Conferencias), **45**, 45–46 (2014)

III Workshop on Robotic Autonomous Observatories (Torremolinos, Malaga, Spain, 7-11 October 2013)
Editors: J. C. Tello, A. Riva, D. Hiriart & A. J. Castro-Tirado

NEW DEVELOPMENTS IN JASTROCAM - SOFTWARE FOR ASTRONOMICAL DATA GATHERING

S. Zola,[1,2] and M. Budyn[1]

RESUMEN

Presentamos una aplicación de software llamada *JAstroCam*. Es un paquete de software basado en Java que está siendo desarrollado en cooperación con el Observatorio Astronómico de la Universidad de Jagiellonia y el Observatorio de Mt. Suhora de la Universidad Pedagógica, y cubre todos los aspectos de adquisición de imágenes, incluyendo el control del hardware, monitoreo de ambiente y ayuda a los observadores a adquirir datos científicos de alta calidad. El motivo principal para hacer aún otro software más es la carencia de herramientas avanzadas para la adquisición de datos con CCDs en Linux. La mayoría de los vendedores de hardware proveen drivers y aplicaciones de apoyo principalmente en el sistema operativo Windows, dejando a los usuarios de Linux con un número limitado de soluciones código abierto, usualmente no optimizadas para astrónomos profesionales. En cooperación con investigadores académicos, hemos logrado implementar una aplicación con una interfaz de usuario intuitiva además de herramientas poderosas que realizan adquisición de imágenes programada y análisis de datos preliminar online, incluyendo medidas de parámetros estadísticos de cuadros individuales al igual que fotometría de filtros individuales y múltiples. Se presentan las características actuales, las mejoras más recientes en la versión 3.X al igual que los planes de desarrollo futuro para la versión 4 de *JAstroCam*.

ABSTRACT

We present a software application called *JAstroCam*. It is a Java-based software package being developed in cooperation of the Astronomical Observatory of the Jagiellonian University and the Mt. Suhora Observatory of the Pedagogical University, and covers all aspects of image acquisition, including hardware control, environment monitoring and provides support for an observer in gathering high quality scientific data. The main motivation for making yet another software application is lack of advanced tools for data acquisition with CCDs available on the Linux platform. Most of the hardware vendors provide software drivers and supporting applications mainly for the Windows operating system, leaving Linux users with a limited number of open source solutions, usually not optimized for professional astronomers. With cooperation of academic researchers, we have managed to implement an application with an intuitive user interface and powerful tools that perform scheduled image acquisition and crude online data analysis, including measurements of a single frame statistical parameters as well as single and multifilter photometry. We present existing features, latest enhancements built in version 3.X as well future development plans for version 4 of *JAstroCam*.

Key Words: methods: data analysis

1. *JASTROCAM* VER. 3.X FEATURES

Version 3 of *JAstroCam* is implemented as a modular desktop application, which features can be extended by number of plugins loaded once at application startup. *JAstroCam* itself provides a user interface, core services for handling exposures, while devices, analysis tools and other extensions are written as modules (plugins). Such an approach provides flexible deployment and configuration for various hardware setups.

[1]Astronomical Observatory, Jagiellonian University, ul. Orla 171, 300-244 Krakow, Poland (szola@oa.uj.edu.pl).

[2]Mt. Suhora Observatory, Pedagogical University, ul. Podchorazych 2, 30-084 Krakow, Poland.

Below, a number of selected features is presented, which make *JAstroCam* an unique among other Linux-based image acquisition applications.

1. User interface (UI) is highly optimized for quick execution of exposures and reflects filter wheel and a camera features like filters, gains, chip size, binning etc. All UI controls change dynamically when a device parameters change.

2. Exposures to be performed are packed as exposure sets with parameters for every filter defined (gain, time and individual counter). Such an exposure set is scheduled for execution and put into waiting queue. When hardware is ready for handling new exposures, the sets are executed in order of subscrip-

III Workshop on Robotic Autonomous Observatories (Torremolinos, Malaga, Spain, 7-11 October 2013)
Editors: J. C. Tello, A. Riva, D. Hiriart & A. J. Castro-Tirado

tion. Even though *JAstroCam* is designed to execute large sets of exposures, it is possible to perform also a single exposure and perform basic image analysis. The most commonly used tools are: histogram, 1D and 2D cross-sections of an image or a part of it and statistical calculations (min, max, avg, stdev). Applied analytical tools are updated whenever scheduled exposure has been performed, thus analysis is updated during exposure set execution as well.

3. Automatic Reduction and Analysis Subsystem (ARES) is the most demanded and important feature of *JAstroCam*, giving a possibility to do an online reduction and analysis of performed exposures. An observer chooses a target object (V), a comparison star (C) and (optionally) a check star (K). For each exposure, *ARES* performs aperture photometry and displays charts of objects intensities (including background) or magnitude differences. Separate charts are drawn for each filter.

4. The tracking control extension of *JAstroCam* provides an auto-guider functionality, but works in different way than in other astronomical tools. When doing short exposures a dedicated auto-guiding system is usually not required. For this reason, *JAstroCam* tracking control system uses images acquired from the primary camera. The successive images are analyzed, position of a selected star is computed, and correction of a telescope position is performed just before the next exposure starts. This simplifies further analysis as stars are well-aligned, while no additional hardware and effort is needed to have aligned images.

A number of other features are implemented in current version of *JAstroCam*, but only most important ones (from an observer's perspective) were presented in this paper.

2. FURTHER DEVELOPMENT

The current development of *JAstroCam* v3 is mainly focused on maintenance and reliability. As the demand for new functions and more advanced plugins grows, it became necessary to review the architecture and concepts that were the basis of *JAstroCam* v1.X to v3.X. The new *JAstroCam* architecture is a result of analysis of bottlenecks, architecture limitations as well as foreseen requirements in the future. Our primary goal was to increase flexibility of the application and ease of enhancements. As the result, version 4.X is based on the OSGi framework, which is an component framework for JAVA applications, providing fully dynamic runtime and service discovery mechanisms. It is a key enhancement of the new architecture and a 'glue' that tights small, independent components that can be installed, removed or updated in the runtime. Additionally, OSGi dependency mechanisms enforces components, with clear responsibility and separation of functions.

The core benefits of application of OSGi framework are:

1. *JAstroCam* is now divided into smaller, but well defined components called 'bundles'. Every bundle provides either definition of service or its implementation, thus other bundles depend on service definition, while OSGi links it with a particular implementation. This way, bundles describe what they need and provide, while OSGi links them together making application work.

2. OSGi framework is able to start or stop bundles at any point of time, allowing bundles to be fixed and re-deployed without restarting the whole application. Additionally, by selecting a particular set of bundles, *JAstroCam* can be easily adopted to particular needs.

3. In statically-linked applications, all packages are available to any other package at any point of time. From the other side, there is no standardized way of querying a particular function or service. OSGi introduces a service model and service discovery mechanism which simplifies and standardizes the way of querying a particular functionality.

Introduction of the new architecture requires reimplementation of most of the existing JAstroCam code and re-designing core concepts. In return, resulting prototype shows enormous flexibility end enhance ability. The dynamic, component and service-based architecture allows a much higher level of customization. By selecting particular bundles, *JAstroCam* can act very differently and can be customized and fit according to needs. For example, the storage settings can be implemented as filesystem storage or database storage; star-matching algorithm can be filter-based or function fitting based. Finally, image rendering can be implemented with or without hardware acceleration. Almost every component of JAstroCam v4.X can be easily replaced with another implementation.

REFERENCES

Budyn, M., Zola, S. & Wojcik, K. 2010, ASPC, 435, 87
Budyn, M. & Zola, S. 2010, AIPC, 1273, 558 MNRAS, 297, 1163