RTS2 – Open Source Observatory Manager

Petr Kubánek
The University of Arizona
309 PUBLICATIONS **1,332** CITATIONS

RTS2 – Open Source Observatory Manager

Petr Kubánek
IPL Universidad de València, poligono La Coma s/n, E-48008 València
IAA CSIC Granada, Cm. Bajo de Huétor 50, E-18008 Granada, petr@iaa.es

Martin Jelínek
IAA CSIC Granada, Cm. Bajo de Huétor 50, E-18008 Granada, petr@iaa.es

Resumen

RTS2, the Remote Telescope System, 2nd version, is an ambitious project, which aim is to develop a complete open source replacement of observatories night observers. This complex package is build using own supporting library. The logic which governs its development is plug-and-pry – the system must be able to dynamically reconfigure to react on new instrument added to it and device failures. It includes drivers for various instruments present at observatories, TCP7IP network messaging interface, and logic for target database, observation scheduling, execution, and image processing. RTS2 is currently used on more then 10 sites on 4 continents.

Palabras Clave: observatory manager, open-source, fully autonomous observatory

1 INTRODUCTION

As mentioned in the abstract, the aim of RTS2 project is to develop a set of programmes, which will replace night observers. Having a fully autonomous system should provide various benefits. Not only it might be cheaper to run the observatory due to removal of the observer night pay, but also the observatory should be more productive and should enable a new science. Quick reaction to transient events, notably gamma ray bursts, is most probably the best known example of the new science, enabled by fully autonomous observatories.

To attack this problem, the system design splits the package to three layers. The lowest is a hardware access layer, which provides access to various devices, forming part of the observatory. Next is a central server, which manages a list of devices and services forming the system. Uppermost is a service layer, which provides various autonomous and interactive services for the observer. More information about the code structure can be found in [1] and [2].

This article focuses on ideas which should be implement in the near future. Further information about the project, about the code and its structure can be found on http://www.rts2.org/ and in [1] and [2].

2 RTS2 HISTORY

RTS2 greatly benefits from experience gained during development of Remote Telescope System - RTS, system developed as student software project at Prague Charles University. Its development started at 2002, with a first night run on an observatory occurring at the end of year 2002. During following years, a lot of new features was added, system stability was vastly improved, and a new installations come to live. Full history of development until 2008 is described in [1] and [2]. As can be seen from publication history at RTS2 web site, observatories running RTS2 already performed hundreds of interesting and world competitive observations.

3 CURRENT DEVELOPMENTS

Currently three major issues are being solved by new development: lack of sufficiently robust and versatile image processing pipeline, graphical user interface and network management.

3.1 IMAGE PROCESSING

Current image processing is a complex process, which uses a combination of custom developed C programmes, bash scripts and IRAF[3] calls. Although this approach works and delivers results, it has a following major bottlenecks:

- it is not easy to install it and keep it up to date
- it is non trivial for people not familiar with it to do image processing, and learning curve is pretty steep
- it is not yet fully integrated into RTS2 so it provides results from the first image obtained with the observatory and it will keep running and informing users about observatory progress.

We would like to develop new system, based on approaches used by HEASoft[4] for (not only X and Gamma ray) data processing. We hope to provide well documented software, built with the extensive help of already available libraries. It should fully process all obtained images, calculate various data quality issues and produce final, trustable results. The system should provide robust, well portable set of routines, which would be fully integrated into RTS2 data acquisition strategy.

3.3 GRAPHICAL USER INTERFACE

Various approaches were tried for a robust, easy to install graphical interface. RTS2 now provides Stellarium[5] interface, so at least navigating telescope to holes in clouds on cloudy night for focusing is not as laborious as it was before. After some experiments, the following architecture was selected for the GUI:

- XML-RPC interface for communication with RTS2. XML-RPC was picked up for its easy extensibility, good platform and language independent support, and network transparency
- the interface will be written in GTK toolkit, either using Python or C++ (GTKmm) interfaces

Beside this interface, we already have PHP backed Web interface. And we performed some design attempts, using Google Web Toolkit, for a Web 2.0 interface. Although this looks promising, RTS2 now has own web-server, based on service already serving XML-RPC requests. This provides a quick preview of the images and access to target database. One possibility currently under consideration is that this component would grow to a full Web GUI, including some Web 2.0 capabilities. This would make our lives easier installing and upgrading it on multiple sites.

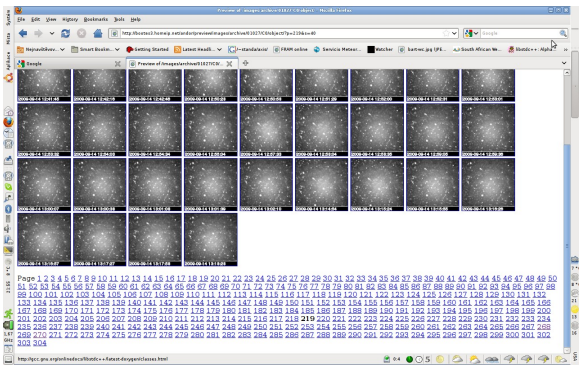

Figure 1: WWW image preview

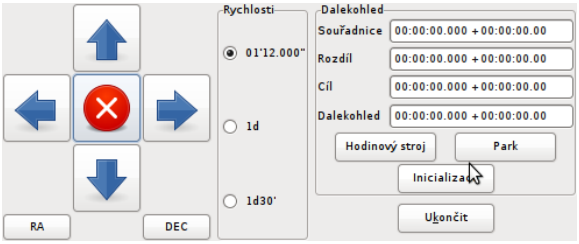

Figure 2: Base of telescope GUI

3.3 NETWORK MANAGEMENT

With multiple instruments working, our team quickly realized that constructing them was one part of the story, yet operating them continuously for a year or more is another. Various software and hardware pieces failed on various network nodes. Moreover, solving those issues eats our time, which we can otherwise use for system development. Of course most of the instruments can run for a few nights without any problem, but even if the instrument is failing once per 7 night, with 7 telescopes you have to keep care of, on average you are fighting with one failure per day.

We foreseen the development of a network management tool, which would inform us about both hardware and software problems, send reports to various local people which can fix them, and allow us to devote more time to science. It will use XML-RPC protocol for communicating over the Internet, and most probably GTK based GUI for user interfaces.

Other piece in the mosaic for functional and effective network is an intelligent network scheduling. It will be based on search for Pareto optimal front. As scheduling is beyond scope of this short article, please see [6] and [7] for more information.

4 CONCLUSIONS

This article gives a short review of RTS2 system and its uses. It then focuses to major three bottlenecks, which are making our lives hard. Solutions for those problems are outlined. RTS2 is a fully open package, with a wide list of supported devices, and with a record of successful operations. Everybody is welcomed to either try it and use it, or even provide us with some help for new developments. Experiences gained by our group suggest that development of a fully autonomous observatory belongs to category “it is an interesting idea, which is quite hard to implement” - together with such goals as a fully autonomous vehicle, driving on its own electronic brain. We hope that real progress in this area can be achieved only by cooperating rather then competing on the development of the software

infrastructure. And we are ready to take part in collaborative efforts.

Agradecimientos

PK would like to acknowledge generous financial support provided by Spanish *Programa de Ayudas FPI del Ministerio de Ciencia e Innovación (Subprograma FPI-MICINN)* and European *Fondo Social Europeo*. Work on RTS2 was supported, influenced and encouraged by numerous people, whose list would be too large for this article. Persons from this list which according to authors deserves to be mention explicitly are Alberto Castro-Tirado, Antonio de Ugarte Postigo, Michael Prouza, René Hudec, John French, Lorraine Hanlon and Victor Reglero.

Referencias

[1] Kubánek, P. et al. (2006) RTS2: a powerful robotic observatory manager, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Volume 6274*.

[2] Kubánek, P. et al. (2008) The RTS2 protocol, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Volume 7019*

[3] Tody, D. (1993), "IRAF in the Nineties", *Astronomical Data Analysis Software and Systems II, A.S.P. Conference Ser., Vol 52, eds. R.J. Hanisch, R.J.V. Brissenden, & J. Barnes, 173*

[4] HEASoft - http://heasarc.gsfc.nasa.gov/

[5] Stellarium – http://www.stellarium.org/

[6] Kubánek, P. (2008), Genetic Algorihm for Robotic Telescope Scheduling, *Máster en Soft Communication y Sistemas Inteligentes, Universidad de Granada*

[7] Förster, F. et al. (2009) Scheduling in targeted transient surveys and a new telescope for CHASE, *Advances in Astronomy, Hindawi Publishing*