

RTS2: a powerful robotic observatory manager

Petr Kubánek (1,2), Martin Jelínek (3), Stanislav Vitek (3), Antonio de Ugarte Postigo (3), Martin Nekola (1,4) and John French (5)

- (1) Astronomický ústav Akademie věd České republiky, Fričova, Ondřejov, Czech Republic
 (2) INTEGRAL Science Data Center, Chemin d'Ecogia 16, Versoix, Switzerland
 (3) Instituto de Astrofísica de Andalucía (IAA-CSIC), Granada, Spain
 (4) České vysoké učení technické, Fakulta elektrotechnická, Praha, Czech Republic
 (5) University College Dublin, Dublin, Ireland

RTS2, or Remote Telescope System, 2nd Version, is an integrated package for remote telescope control under the Linux operating system. It is designed to run in fully autonomous mode, picking targets from a database table, storing image meta data to the database, processing images and storing their WCS coordinates in the database and offering Virtual-Observatory enabled access to them. It is currently running on various telescope setups world-wide. For control of devices from various manufacturers we developed an abstract device layer, enabling control of all possible combinations of mounts, CCDs, photometers, roof and cupola controllers.

We describe the evolution of RTS2 from Python-based RTS to C and later C++ based RTS2, focusing on the problems we faced during development. The internal structure of RTS2, focusing on object layering, which is used to uniformly control various devices and provides uniform reporting layer, is also discussed.

The idea of a fully robotic telescope, which will be able to perform tasks at night when the observer is asleep, is not new. It emerged as soon as the first microprocessors which were capable of such tasks become available on the market.[7] The advanced scheduling makes a better use of telescope time. Fully autonomous operation of telescope enables real-time follow-ups, allows the unattended operations on remote sites and, in consequence, brings the costs savings.

Evolution of RTS2

RTS2, as the 2 in the name suggests, evolved from RTS.[6].

RTS2 was initially designed in pure C, e.g. without use of object oriented programming (OOP) techniques. After two years of development and active use of RTS2 on BART and BOOTES telescopes (and beta-testing version for FRAM telescope), we decided to abandon C-purism in favour of OOP design in the C++ language. That decision has enabled us to produce code which is easily maintainable, and after a year of active use, we are confident that it has paid off. It may have been more appropriate to rename the new code RTS3, since there is not much remaining from the original C design, but the name RTS2 was retained.

The evolution of both the late RTS and the whole RTS2 package can be tracked in Concurrent Versions System (CVS), which we use for version control.

Target scheduling

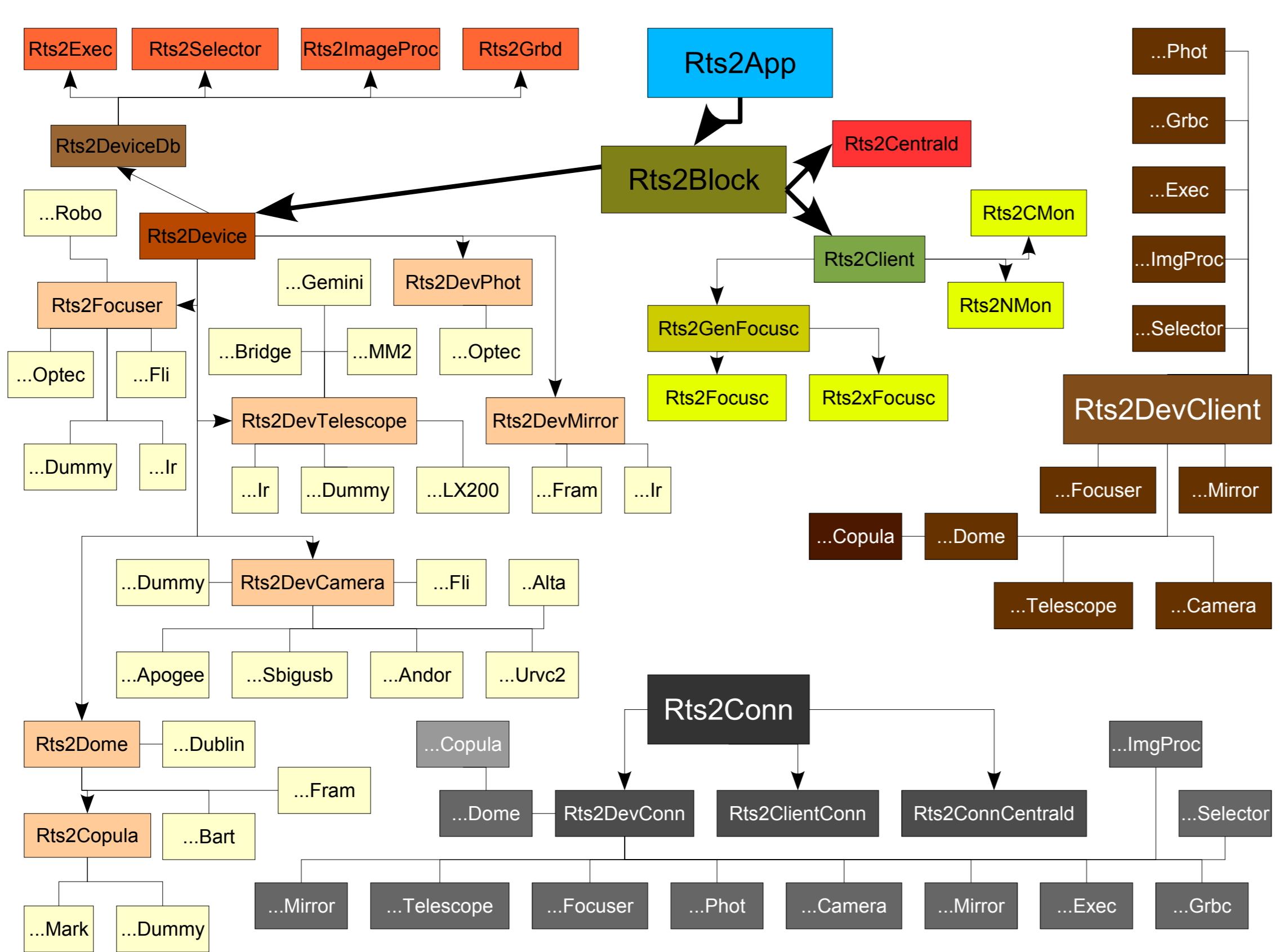
RTS2 enables two basic modes of operation – dispatch scheduling[4] and queue scheduling. Queue scheduling allows the observer to put a list of targets which he/she would like to observe to the database. It is convenient when the observer is at the telescope and can check local weather conditions. Dispatch scheduling is ideal for an autonomous system where there is no human intervention. We think that by facilitating both approaches users have a wider range of options at their disposal. Also, given the separation of the selection and execution components, it will be quite easy to implement other types of scheduling.

To enable observation of different targets, a target hierarchy was created. Its parent object is Target, which provides the interface to various target-related calculations.

Targets sometimes use “chaining”. This is when one master target holds a list of possible targets, from which it selects the actual observation to be performed.

C++ class hierarchy

The class hierarchy is depicted in following figures.



Current setups

RTS2 is controlling telescopes on three continents: four telescopes in Europe (BART[5], BOOTES 1A, BOOTES 1B [1]), and BOOTES-IR[2] one in South America (FRAM at the southern site of the Pierre Auger observatory), and one in Africa (Watcher[3]). In the following paragraphs, the current setups of all instruments is described.

BOOTES-IR is 60 cm alt-az mount, located on Observatorio de Sierra Nevada (IAA-CSIC), Spain.

Two other BOOTES setups – 1A and 1B – are small-class telescopes, located at El Arrenosillo (CEDEA-INTA) and La Mayora (EELM-CSIC) observing stations in southern Spain.

BART is a 25 cm telescope running on a Losmandy Titan mount (controlled by Gemini interface), located at ASÚ AV ČR, Ondřejov, Czech Republic.

FRAM is operating in a remote part of Argentina at the Pierre Auger south observatory. Its main target is the measuring of atmospheric conditions for Pierre Auger observatory.

Watcher is a Paramount ME based instrument, located at the University of Free State's Boyden Observatory, South Africa.

Acknowledgments

PK and MN would like to thank to GA AV ČR for the support via grant A3003206 and ESA PECS Project 98023. MJ would like to thank to the Spanish Ministry of Education and Science for the support via FPU grant AP2003-1407. We would like to acknowledge INTA for the hospitality provided at the INTA-CEDEA BOOTES-1 observing station.

Observation scripting

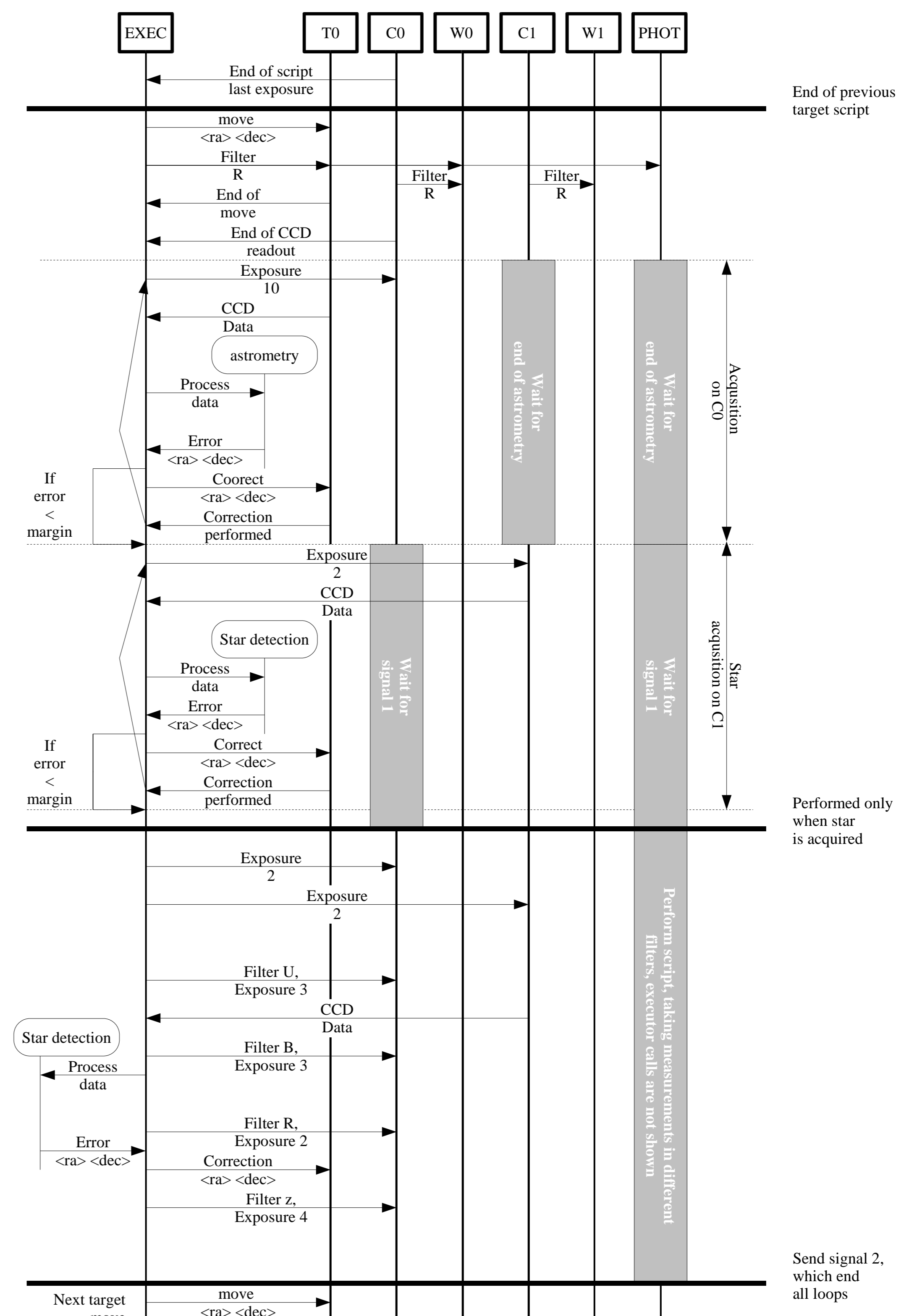
In order to fulfil the requirements of the various setups, the observation system must enable customisable scripting of observations, so that images using various filters can be obtained.

Scripting attempts to make the best use of the available observing time – all commands that can be executed during mount moves or camera readouts are executed at that time. RTS2 is designed to allow early observations of possible GRB OTs. It supports a continuous mode of observing (useful for mounts with multiple detectors). When a target remains unchanged, scripts are executed in a loop on all detectors until a new target is selected.

Scripting is quite a complex issue, so we give an example in the following subsection.

Example observation script

```
C0 FRA 0.01 20 SW 1 ifacq {loop 2 {ER 2 EU 3 ER 2 EB 3 ER 2 E z 4 } else { E 10 } }
C1 FRA star 10 2 SS 1 ifacq {guide 2 2}
PHOT FRA Aw SW 1 ifacq {P 10 2 R P 10 2 V P 10 2 1 P 10 2 2 P 10 2 3 P 10 2 4 P 10 2 5 P 20 2 R F d} SS 2
```



RTS2 supported hardware

Hardware, which is currently supported by RTS2, is described in following table:

Devices	Manufacturer	Notes
Mounts	Software Bisque	Paramount ME with proprietary serial port driver
	Gemini	L3 and L4 of firmware, Losmandy mounts, Mountain Instrument mounts and custom mounts
	Meade	all LX 200 protocol models
	Astelco	BOOTES-IR mount, OpenTPL interface
Cameras	Dynostar	LX 200 protocol based mount, no longer in use
	Apogee	all models, including Alta series (with patched drivers)
	FLI	all models
	SBIG	all models, including new USB models
Photometers	Starlight Xpress	all models
	Optec	ISA I/O card based photometer, with own kernel driver
Filter wheels	FLI	all models
	Optec	Intelligent Filter Wheel (IFW)
	SBIG	filter head integrated to SBIG cameras
Focusers	Robofocuser	all models
	FLI focuser	all models
	Astelco	OpenTPL controlled focuser
Mirrors	Optec	TCF-S and TCF-S3
	Astelco	OpenTPL controlled mirror
Meteo stations	own design	serial controlled mirrors
	Davis	all models supported by modified Meteo (http://meteo.othello.ch) package
	own design	stations communicating thought TI I/O cards
Roof and cupola	own controllers	own design serial I/O board (FRAM, BART, Watcher), TI I/O card controlled roofs
	cupola	Profibus-based serial interface

References

- [1] A. J. Castro-Tirado et al. BOOTES: A Stereoscopic and Robotic Ground-Support Facility for the INTEGRAL Era. In V. Schoenfelder, G. Lichti, and C. Winkler, editors, *ESA SP-552: 5th INTEGRAL Workshop on the INTEGRAL Universe*, pages 637–644, October 2004.
 [2] A. J. Castro-Tirado et al. BOOTES-IR: near IR observations by a robotic system in Southern Spain. volume this volume. SPIE, 2006.
 [3] J. French, L. Hanlon, B. McBreen, S. McBreen, L. Moran, N. Smith, A. Giltinan, P. Meintjes, and M. Hoffman. Watcher: A Telescope for Rapid Gamma-Ray Burst Follow-Up Observations. In E. Fenimore and M. Galassi, editors, *AIP Conf. Proc. 727: Gamma-Ray Bursts: 30 Years of Discovery*, pages 741–744, September 2004.
 [4] T. Granzner. What makes an automated telescope robotic? *Astronomische Nachrichten*, 325:513–518, October 2004.
 [5] M. Jelínek, P. Kubánek, M. Nekola, and R. Hudec. BART: an intelligent GRB and sky monitoring telescope (2000–2004). *Astronomische Nachrichten*, 325:678–678, October 2004.
 [6] T. Jilek, P. Kubánek, F. Krollauer, and F. Kvařil. *Detekce astronomických objektů s proměnnou intenzitou za pomoci robotického teleskopu*. Katedra softwarového inženýrství MFF UK, May 2000.
 [7] M. Truhsblood and R. M. Genet. *Telescope control – 2nd ed.* Willmann-Bell, 1997.