Design and integration of the HARPS3 software system

John Young[a], Tim Naylor[b], Martyn Brake[b], Martin Fisher[a], Eugene Seneta[a], Peter Kunst[c], Francesco Pepe[d], and Danuta Sosnowska[d]

[a]University of Cambridge, United Kingdom
[b]University of Exeter, United Kingdom
[c]NOVA Optical Infrared Instrumentation Group at ASTRON, The Netherlands
[d]Geneva University, Switzerland

ABSTRACT

We present the design of the HARPS3 software system – a distributed, event-driven control system for robotic operation of the HARPS3 spectrograph at the Isaac Newton Telescope (INT). We also describe our approach to integrating the control software components incrementally at various stages of development, using a simulation framework. HARPS3 will be a high resolution ($R = 115,000$) echelle spectrograph operating at wavelengths from 380 nm to 690 nm, with a design based on the successful HARPS and HARPS-N instruments. It is being built as part of the Terra Hunting Experiment (THE) – a planned 10 year radial velocity measurement programme to discover Earth-like exoplanets around Sun-like stars.

Keywords: Robotic telescopes, Observatory control, Instrument control

1. INTRODUCTION

Although over 3700 exoplanets have been detected over the past 23 years (according to https://exoplanetarchive.ipac.caltech.edu), no true Earth-Sun analogue has yet been discovered. To address this gap in our knowledge, we are planning a "Terra Hunting Experiment" (THE; http://www.terrahunting.org/) to search for Earth-mass planets in Earth-like orbits around the nearest Sun-like stars. The exoplanet search will be conducted using the radial-velocity (RV) method, using time series of high resolution spectra of a star to search for varying Doppler shifts of spectral lines that indicate the presence of an orbiting companion. We are building a high resolution spectrograph, called HARPS3,[1] and plan to install it on an upgraded and roboticized Isaac Newton Telescope (INT) on La Palma, Canary Islands. The INT will be upgraded to extend its lifetime and to enable robotic operation, including procurement of new Telescope Control Software (TCS).

The RV signal induced in our Sun by the Earth has an amplitude of $9\,\mathrm{cm\,s^{-1}}$ and a 1-year period. The Terra Hunting Experiment aims to detect a signal at the $\sim 10\,\mathrm{cm\,s^{-1}}$ level at periods of $\gtrsim 100$ days. The Sun and similar stars produce RV signals with amplitudes an order of magnitude greater than this due to intrinsic activity (oscillations, granulation, active regions, magnetic cycles etc.) on their surface, however these activity signals have characteristics and timescales that should allow them to be separated from the planetary signals,[2] provided a sufficiently long-duration and intensively-sampled time series of measurements is obtained. Accordingly, the Terra Hunting Experiment (THE) will run for 10 years and during that time we will take regular, nightly RV measurements of a selection of our nearest and brightest solar-like stars (G and K-type dwarfs; $m_V \lesssim 8.5$). A fraction of every observing night will be allocated to THE observations, with the remaining telescope time offered to the community, including availability of HARPS3 for service-mode observations.

HARPS3 will be a fibre-fed, high resolution, high stability, echelle spectrograph. It is based on the designs of HARPS (High Accuracy Radial velocity Planet Searcher[3] and HARPS-N,[4] which are operational at the ESO La Silla 3.6 m telescope and the Telescopio Nazionale Galileo (TNG) respectively. The instrument comprises two main sub-systems: (1) the main body of the spectrograph housed in the large Coudé room of the INT and (2) the Cassegrain fibre adapter system installed at the Cassegrain focus of the telescope. The main body of the spectrograph is housed in a thermally stable room; increasing degrees of thermal control are provided by an arrangement of nested enclosures.

The Cassegrain fibre adapter is attached to the telescope at the Cassegrain focus. This unit contains the atmospheric dispersion corrector (ADC), a tip/tilt correction system, the calibration light projection optics, a polarimeter and the fibre selector mechanism. The calibration unit (providing a number of calibration light sources) will be housed in the large Coudé room of the INT; the light from here is fibre-fed to the Cassegrain fibre adapter.

The scientific goals for the Terra Hunting Experiment will only be achieved by efficient scheduling and reliable operations. The project partners have limited resources available to support operations, hence new control software is being designed and implemented with a goal of delivering robotic operation. In this paper, we present the key requirements (Section 2) and overall design for the robotic software system (Section 3).

The HARPS3 software is being co-developed by several different institutions. University of Cambridge are managing the HARPS3 project and are responsible for the observatory-level software and the instrument control software for the spectrograph and science detector. University of Exeter are developing the scheduling and data archiving software and will host the HARPS3 Archive database. NOVA are developing the instrument control software for the Cassegrain fibre adapter. The Data Reduction Software (DRS) will be based on that developed for ESPRESSO[5] by Geneva University, with suitable adaptations for processing data from HARPS, HARPS-N, and HARPS3.

This division of labour is typical of contemporary projects in observational astronomy, but carries the risk of needing a lengthy integration phase to combine the various software components into a cohesive and reliable system. Our approach for mitigating the impact of this risk on the overall project schedule is to decouple the software integration activities from the hardware integration activies, and to integrate the software system in an incremental fashion as new functionality is implemented. Software integration will be performed within a simulation framework, developed specifically for this purpose. The software integration processes are described in further detail in Section 4.

2. REQUIREMENTS

2.1 Robotic operation

The INT/HARPS3 system must operate autonomously to carry out calibrations and science observations, managing adverse conditions and routine technical issues without human intervention. For most of the time during normal operations, no staff will be present at the INT. The operator at the nearby William Herschel Telescope (WHT) will be able to use a dedicated Graphical User Interface (GUI) from the WHT control room to suspend and resume robotic observing and to close the dome and abort the night's observing.

The robotic system must schedule and carry out the tasks that would be performed by a human operator in a conventional observatory. These tasks include:

- open and close the dome as necessary to ensure safe operations, manage the telescope thermal environment and enable both daytime calibrations and nighttime observations;
- automatically schedule daytime calibrations and nighttime observations;
- handle failed observations/calibrations;
- manage (re)booting computers and (re)starting software; and
- track the completion status of the set of observations linked to a scientific proposal.

The main use cases for the robotic system are summarised in Fig. 1 in the form of a Unified Modeling Language (UML) use case diagram. These use cases have been defined in some detail in the operations plan for HARPS3, but in the interests of brevity their definitions have been omitted from this paper.

Our approach to roboticizing the INT and HARPS3 is to design in all of the functionality that will be needed for fully robotic operations (i.e. no requirement for routine intervention by a local or remote human operator), but to recognise that the initial implementation may have hardware or software deficiencies that compromise the reliability of the system. We will accommodate schedule risk by allowing for the possibility of deferring some of the more sophisticated functionality, for example scheduling with complex constraints and automatic recovery from errors in executing observations.

It will also be possible to interact with the user interfaces for operating the INT and HARPS3 from a remote location, either to watch observing progress or to sequence an observation manually. For safety reasons, low-level control of the telescope from a remote location will be prohibited.

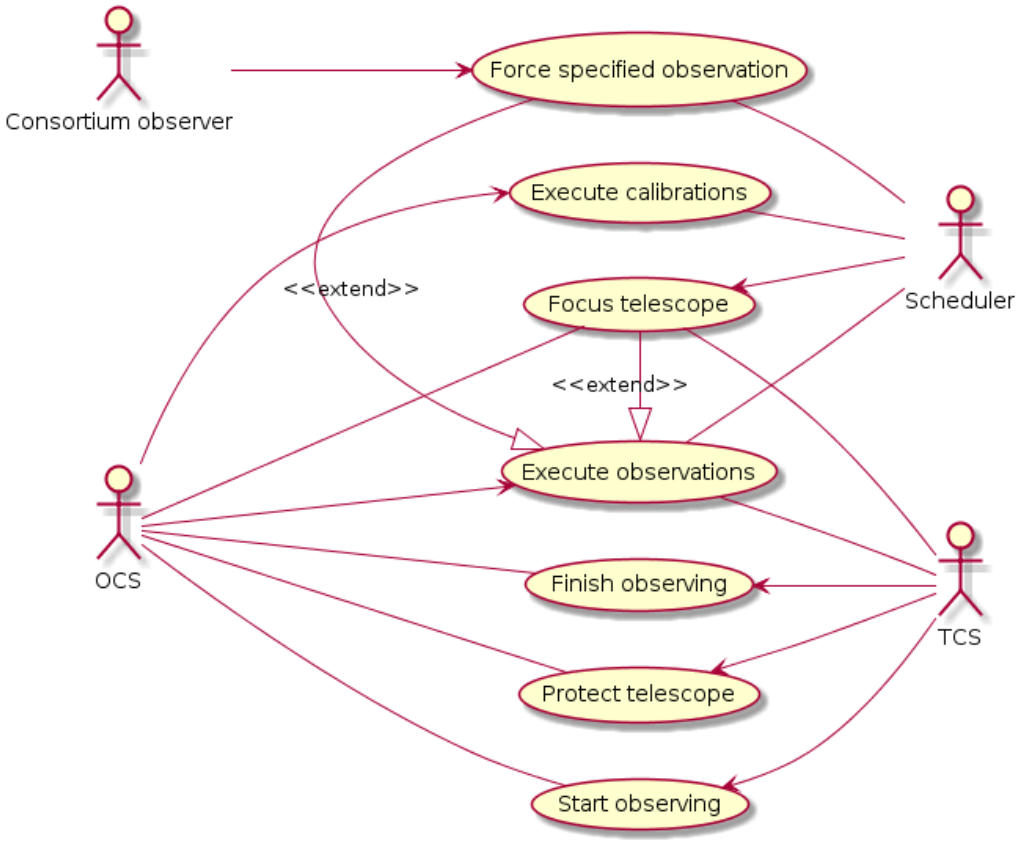

Figure 1. UML use case diagram showing the use cases for robotic observing. The arrow heads indicate the primary actor. The actors (shown as stick figures) are a human astronomer ("consortium observer") and three components of the control software: Automated Scheduler, Observing Control Software (OCS) and Telescope Control Software (TCS).

2.2 Data flow

The key requirements related to input and output of data to/from HARPS3 are as follows:

- HARPS3 must produce raw data files in a standard FITS[6] format, compatible with the HARPS/HARPS-N conventions.
- To conform to observatory policy, raw data files must be copied to the Isaac Newton Group (ING) raw data archive.
- The raw data must be processed automatically in order to assess data quality; the resulting quality metrics and flags will be reviewed periodically by an observer from the consortium.
- The automatic processing must generate scientific data products for the observer, to be accessed from the HARPS3 Archive in Exeter.
- The system must capture detailed diagnostic data for review off site, including but not limited to: environment sensor readings, scheduling decisions, and logs of command, status and error messages exchanged between sub-systems.
- Automatic alerts of failures in executing observations and problems with the processed data must be generated and sent to duty staff located off-site.

The raw data files must include a subset of the FITS header keywords used by HARPS/HARPS-N, to ensure that they can be processed by the DRS (for example the observation mode). The FITS data files must also contain sufficient engineering header keywords to summarise how the telescope and instrument are performing. These will mostly comprise the average and/or RMS of a subset of the collected engineering data.

An overview of the proposed data flow architecture is presented in Fig. 2. In the diagram, raw science data files are transferred immediately to the ING Archive, on the assumption that the ING Archive can handle the non-standard proprietary period (up to 10 years) for THE data.

To simplify the system design and reduce maintenance needs, pipeline processing of the raw data will be done in a single off-site location. It should be technically possible to transfer and process each file within a few minutes of the observation finishing. However, the results will not be used to modify the scheduling of the current night's observations, at least initially.

The science data products provided to observers will comprise a calibrated spectrum (with all instrumental signatures removed), a rebinned version of the spectrum, and a radial velocity. The consortium will revise the DRS code periodically, and reprocess archival data as necessary using the latest stable DRS. Previous reductions must be retained in the HARPS3 Archive. Observers must be notified when reprocessed data becomes available. Having the consortium take full responsibility for processing of open time data means that there is no need to make the DRS available for open time observers to use directly.

3. DESIGN

The HARPS3 software system has been designed as a fairly conventional distributed, event driven control system. Standard command-response and publish-subscribe messaging patterns are used to coordinate the actions of software components running on different computers and linked by an Ethernet network. The communications framework adopted for HARPS3 and the rationale for this choice are discussed in Section 3.2 below. Our software architecture is not a direct copy of any other robotic telescope, but follows similar design principles to many other projects.

Observations with HARPS3 are coordinated by the Observing Control Software (OCS), which commands the telescope and instrument control applications to acquire the celestial target, configure the instrument and obtain science exposures (Fig. 3). An Automated Scheduler application operates as a despatch scheduler, choosing the next observation or calibration from a pre-defined pool according to the prevailing conditions. The publish-subscribe messaging pattern is used to collect monitor data and engineering log messages originating from all of the control software components, in order to write them to the data archive. The roles of the components shown in Fig. 3 are elaborated in the following subsection.

3.1 Control software components

Independent software components are being developed for control of the telescope, Cassegrain fibre adapter, spectrograph and science detector. These will be able to operate in a standalone fashion for non-routine engineering tasks initiated by a human operator, and also under the supervision of the Observing Control Software for the default robotic operation mode. The roles of these software components are described in more detail below.

Cass Instrument Control Software (CICS) Controls the mechanisms in the Cassegrain fibre adapter and calibration unit, for selection of the spectrograph input sources and calibration light source, insertion of polarising optics, and correction of atmospheric dispersion. Provides target acquisition and fast guiding functions for observations with HARPS3.

Observing Control Software (OCS) Coordinates the actions of the telescope and instrument control software (TCS, CICS, SCS and DCS) to execute science observations or calibrations autonomously, and provide automatic responses to technical faults. The OCS incorporates an **Observation Sequencer** that sends commands to the telescope and instrument control software in order to execute an observation or calibration. The Sequencer publishes status information on the progress of the current observation and subsequently, its success or failure. The OCS includes the **FITS Builder** that automatically assembles science and calibration data files, including all of the required FITS header keywords.

Automated Scheduler Decides which science observation or calibration the OCS should execute next, from a pool of possible observations/calibrations. Scheduling decisions are informed by environment sensor data from the TCS and additional environmental data available from ING. Both THE and open time observations are scheduled by the Automated Scheduler.

Spectrograph Control Software (SCS) Monitors the spectrograph thermal environment and vacuum system.

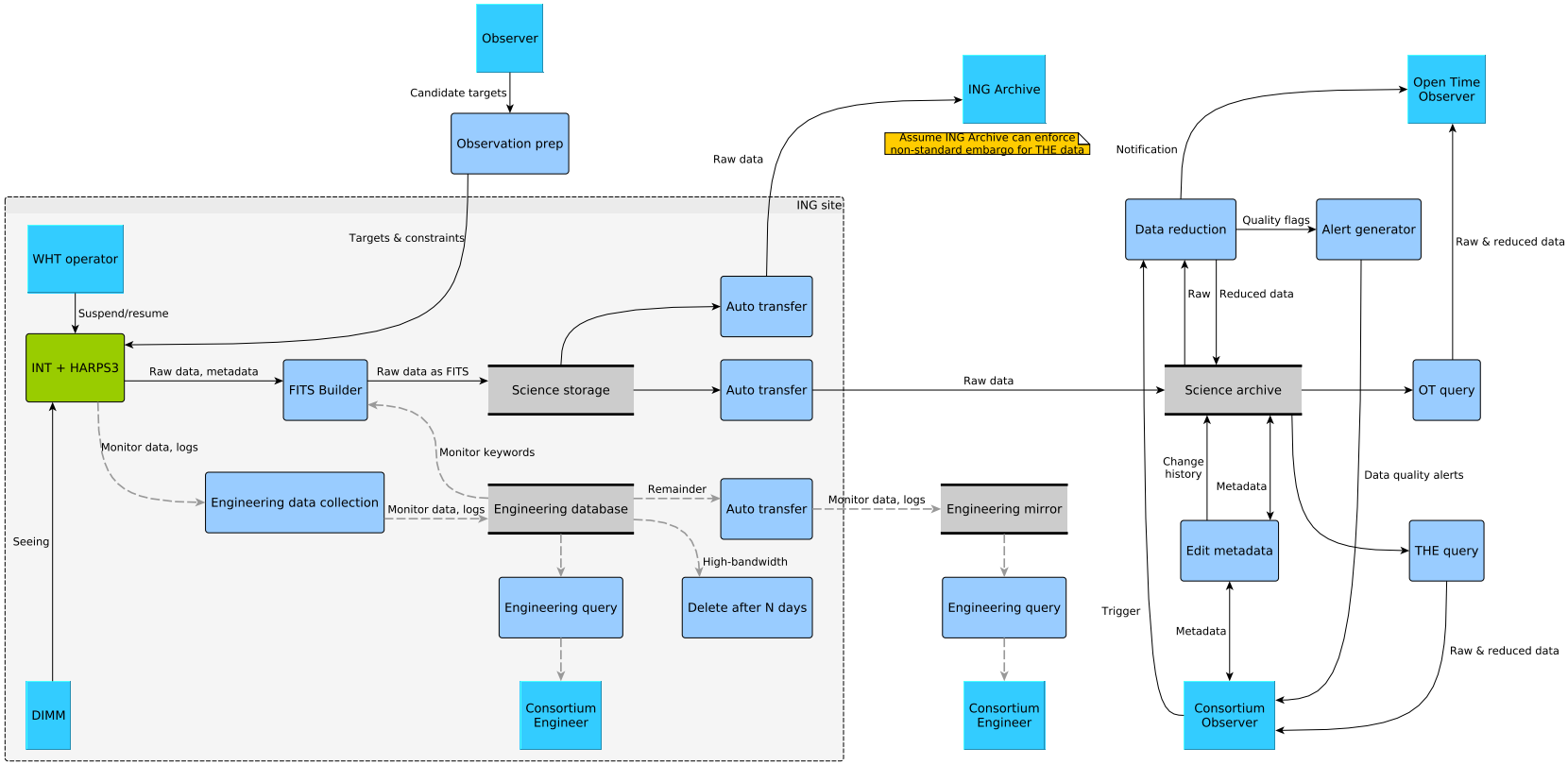

Figure 2. Contextual data flow diagram showing the data paths into and out of the user-facing HARPS3 software components.

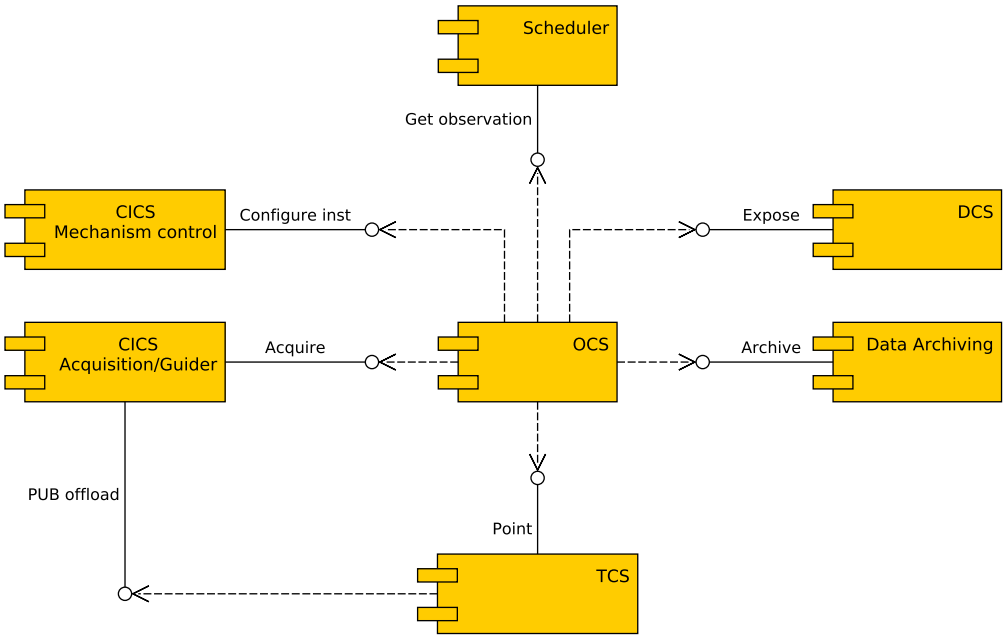

Figure 3. UML component diagram showing how the OCS uses the high-level interfaces provided by the Automated Scheduler, telescope and instrument control software and Data Archiving sub-system to execute observations and calibrations. Each "ball and stick" in the diagram represents a function implemented by the software component (such as the "expose" function provided by the DCS) that can be triggered by the OCS.

Detector Control Software (DCS) Reads the exposure meter and activates the shutter to control science exposures. Captures raw frames from the science detector and configures the CCD readout.

Data Reduction Software (DRS) The DRS is integrated into the robotic control system to provide automatic pipeline reduction of HARPS3 science data and calibration frames. A subset of the quality parameters calculated by the DRS will be used to automatically alert consortium staff to serious data quality issues. The consortium will provide reduced data products to open time observers, via the HARPS3 Archive, and re-process the data to provide updated products as necessary.

Data archiving Captures and stores HARPS3 science data and engineering data from all of the HARPS3 sub-systems. The data archiving system takes science data files generated by the OCS FITS Builder, and transfers them off site, for automatic archiving and pipeline processing. Independent copies of the engineering data are stored at the observatory and at the HARPS3 Archive in Exeter – both locations providing query facilities for data access.

Telescope Control Software (TCS) Controls the telescope mount and dome and reads associated environment sensors. Provides an interface to the OCS for high-level control of the mount and dome as an integrated system, and manages opening and closing of the dome according to the time of day or night and the prevailing conditions.

In addition, observation preparation tools will be needed to enable an observer to determine the optimum instrument and scheduling parameters for potential science observations, apply for observing time ("phase 1"), and submit observation blocks ("phase 2").

3.2 Communications framework

We have adopted a conventional distributed architecture for the HARPS3 software system. The system comprises software components running on separate computers linked by an Ethernet network. This architecture relies on messaging software to coordinate the actions of the constituent parts. Messaging software frameworks are usually referred to as "middleware".

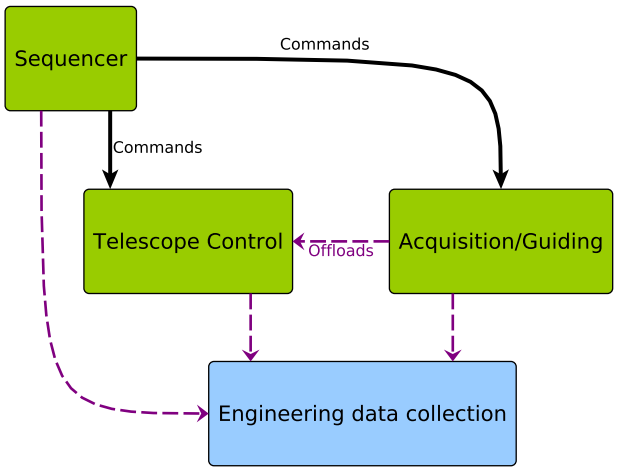

Figure 4. Data flow diagram showing the use of RPC (solid black arrows) and publish-subscribe (dashed purple arrows) messaging patterns for communication between the HARPS3 Sequencer, TCS, and Acquisition/Guiding system

We decided not to use the middleware libraries that were employed for HARPS and HARPS-N because they are closely coupled to ESO software frameworks and the TNG telescope control software respectively. Given the limited resources available for the project, writing a new messaging framework from scratch was also ruled out.

Hence several options for middleware to use in the HARPS3 control system were evaluated during 2016. We looked for existing solutions that provided both Remote Procedure Call (RPC; i.e. command-response) and publish-subscribe messaging patterns over TCP, with a documented wire protocol and support for the C/C++, Java and Python programming languages. Our performance requirements are modest in all respects (throughput, latency, and number of connections), hence performance issues were not considered.

The most promising candidate solutions were Crossbar.io/Autobahn (https://crossbar.io/autobahn/), YAMI4 (http://www.inspirel.com/yami4/), and NATS (https://nats.io/). We finally decided to adopt the Crossbar.io router software and Autobahn client libraries for HARPS3. These implement the Web Application Messaging Protocol[7] (WAMP). Crossbar/Autobahn was preferred over NATS because the latter would require additional effort to integrate a serialisation layer into the communications protocol. Crossbar/Autobahn was preferred over YAMI4 because of deficiencies in YAMI4's support for the Python programming language, which we expect to use extensively.

WAMP was designed as communications infrastructure for the Internet of Things, a vision of network-connected devices containing software, control electronics, actuators and sensors. WAMP is built on the lower-level WebSocket protocol. WebSocket provides bi-directional communication, possibly encrypted, using streams of messages over TCP. Most modern web browsers implement WebSocket. WAMP adds RPC and publish-subscribe messaging patterns and data serialization using JSON or MsgPack. Design and maintenance of WAMP is being driven by Tavendo GmbH, the developers of Crossbar and Autobahn, but there is involvement from a wider community.

We expect to use WAMP for all network communications involved in controlling the telescope and instrument and collecting engineering data. The vendor-supplied TCS may require an adapter layer to enable communication via WAMP.

Both the RPC and publish-subscribe messaging pattern can be used for control. In most cases, the RPC messaging pattern will be used to implement a "command" that invokes an endpoint (method) on the remote system. In a few other cases, there is a pseudo-continuous flow of information, such as the offloads from the Acquisition/Guiding system to the telescope mount. This is more naturally implemented using the publish-subscribe pattern. Figure 4 illustrates the use of both messaging patterns for communication between the HARPS3 Observation Sequencer, TCS, and Acquisition/Guiding system.

3.3 Concurrency

In order to minimise overheads in executing an observation, we require certain operations to be performed in parallel. In particular, the telescope must begin slewing to a new target as soon as the shutter on the science detector has closed at the

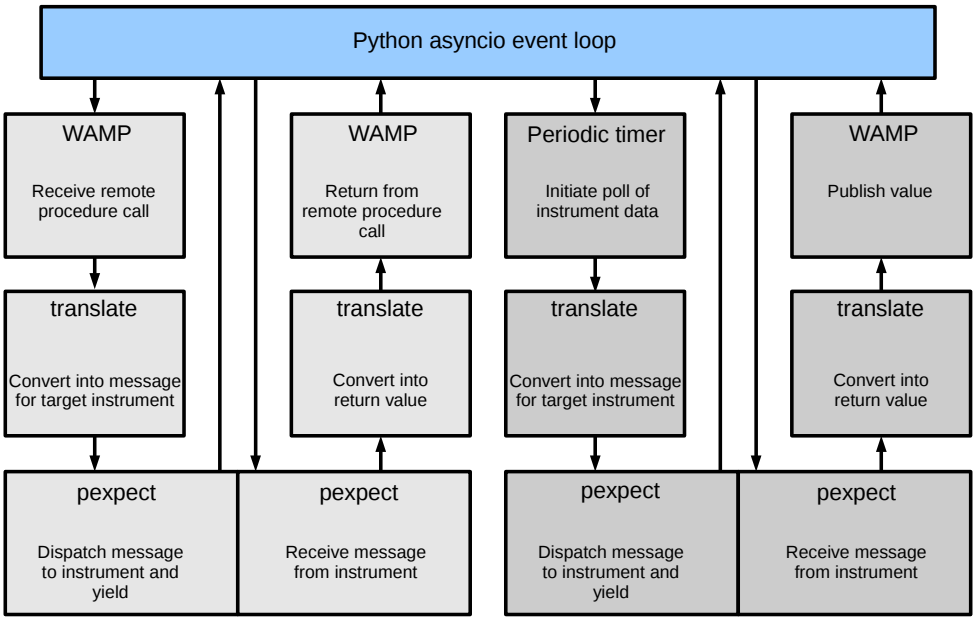

Figure 5. Proposed design of the detector software, showing how remote procedure calls (left, light grey) and publishing (right, dark grey) are handled. We plan to use the same architecture for the spectrograph control software.

end of an exposure. Readout of the science detector and reconfiguration of the Cassegrain fibre adapter must take place during the slew.

The WAMP RPC messaging pattern is fully asynchronous. Requests can take an arbitrary amount of time to execute on the remote system. During this time the caller can send further requests to the same or other callees, without opening a new connection.

The client library Application Programming Interfaces (APIs) are typically based on event loops (the "reactor" pattern) and "futures" (sometimes called "promises"). A future is an abstraction of a result (which may be an error) that might be delivered at some unknown future time. The APIs typically implement waiting on a future or set of futures, chaining futures, or assigning a callback function to a future. This approach is more scalable than operating system threads, and avoids many of the common pitfalls associated with threads programming.

Figure 5 shows an example of how a particular event loop implementation, asyncio in Python 3, will be used in the HARPS3 detector software. This involves using the pexpect Python library to exchange messages with external devices. The devices controlled by the DCS are shown in Fig. 6.

4. INTEGRATION AND TESTING

We are following an agile approach to developing the HARPS3 software system. The control software components are being developed – *and critically, integrated* – in a step-by-step fashion. This approach mitigates the risk of discovering major incompatibilities between components developed by different institutes at a late stage of the project.

In order that software development can proceed quasi-independently of hardware development, the integration will be performed within a simulation framework, created specifically for this purpose. Development of the simulation framework will also proceed incrementally, with new functionality being added as required.

In the spirit of test-driven development, we will write a number of new integration tests at each point where new functionality is incorporated into the simulation. It must be possible to automate both the running of the simulation and verification of the test outputs. However, this does not preclude implementation of interactive modes for testing user

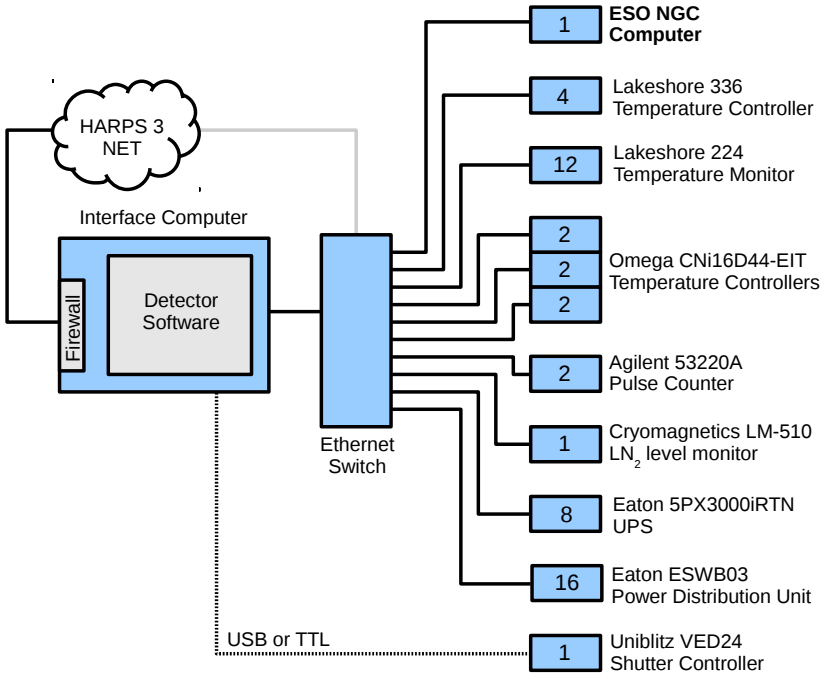

Figure 6. Hardware context for the detector software. The software runs on a computer in the detector rack that appears to the HARPS3 middleware as a WAMP-enabled detector instrument. The devices shown in the diagram are replacements for instruments used at HARPS-N and the numbers inside their boxes are the number of devices each can monitor or control.

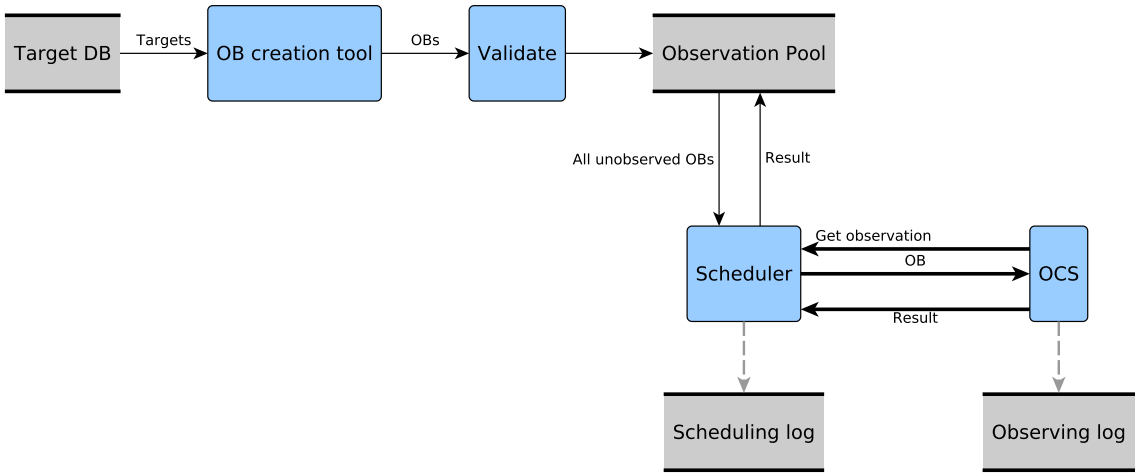

Figure 7. Data flow diagram for stage 1 of the simulation. The WAMP router is not shown.

interface components. Automatic verification will be performed by comparing the test outputs (typically log files and database content) with expectations.

The initial phase of the simulation, illustrated in Fig. 7, aims to demonstrate the core functionality of the Automated Scheduler. Preliminary versions of the Automated Scheduler and OCS are required, as well as the WAMP router and tools for creating Observation Blocks and estimating the time to execute them.

Another important activity will be to develop proof-of-principle software for the communications interfaces between WAMP-enabled software and hardware devices. Many devices will be connected using Ethernet (e.g. Fig. 6) and are

not expected to pose any particular problems. The Cassegrain fibre adapter will use motion controllers from Galil (http://www.galilmc.com/), for which Linux drivers are available. This activity will take place in parallel with developing the simulation.

5. CONCLUSIONS AND CURRENT STATUS

We have presented the design of the HARPS3 software system. The system will schedule and execute observations robotically, coordinated by Observing Control Software that communicates with the telescope and instrument control software using the WAMP messaging protocol over Ethernet. An Automated Scheduler application operates as a despatch scheduler, choosing the next observation or calibration from a pre-defined pool according to the prevailing conditions. The WAMP publish-subscribe messaging pattern is used to collect monitor data and engineering log messages originating from all of the control software components, in order to write them to the data archive.

A simulation framework is being developed to facilitate staged integration of the control software components as they are developed. Current work is focused on implementing the first stage of the simulation. An existing simulation of the Automated Scheduler is being updated to the latest scheduling algorithm and integrated with a PostgreSQL database of target stars. A schema defining the structure and content of Observation Blocks is being constructed using JSON Schema, and will be used to validate blocks exchanged between the Scheduler and OCS. The draft schema is inspired by RTML,[8] an XML-based standard for describing astronomical observation requests.

REFERENCES

[1] Thompson, S. J., Queloz, D., Baraffe, I., Brake, M., Dolgopolov, A., Fisher, M., Fleury, M., Geelhoed, J., Hall, R., Hernández, J. I. G., ter Horst, R., Kragt, J., Navarro, R., Naylor, T., Pepe, F., Piskunov, N., Rebolo, R., Sander, L., Ségransan, D., Seneta, E., Sing, D., Snellen, I., Snik, F., Spronck, J., Stempels, E., Sun, X., Tschudi, S. S., and Young, J., "HARPS3 for a roboticized Isaac Newton Telescope," in [*Ground-Based and Airborne Instrumentation for Astronomy VI*], *Proc. SPIE* **9908**, 99086F (2016).

[2] Hall, R., Thompson, S., Handley, W., and Queloz, D., "On the Feasibility of Intense Radial Velocity Surveys for Earth-twin Discoveries," *Monthly Notices of the Royal Astronomical Society* (2018). Submitted.

[3] Rupprecht, G., Pepe, F., Mayor, M., Queloz, D., Bouchy, F., Avila, G., Benz, W., Bertaux, J.-L., Bonfils, X., Dall, T., Delabre, B., Dekker, H., Eckert, W., Fleury, M., Gilliotte, A., Gojak, D., Guzman, J. C., Kohler, D., Lizon, J.-L., Curto, G. L., Longinotti, A., Lovis, C., Megevand, D., Pasquini, L., Reyes, J., Sivan, J.-P., Sosnowska, D., Soto, R., Udry, S., Kesteren, A. V., Weber, L., and Weilenmann, U., "The exoplanet hunter HARPS: Performance and first results," in [*Ground-Based Instrumentation for Astronomy*], *Proc. SPIE* **5492**, 148 (2004).

[4] Cosentino, R., Lovis, C., Pepe, F., Cameron, A. C., Latham, D. W., Molinari, E., Udry, S., Bezawada, N., Black, M., Born, A., Buchschacher, N., Charbonneau, D., Figueira, P., Fleury, M., Galli, A., Gallie, A., Gao, X., Ghedina, A., Gonzalez, C., Gonzalez, M., Guerra, J., Henry, D., Horne, K., Hughes, I., Kelly, D., Lodi, M., Lunney, D., Maire, C., Mayor, M., Micela, G., Ordway, M. P., Peacock, J., Phillips, D., Piotto, G., Pollacco, D., Queloz, D., Rice, K., Riverol, C., Riverol, L., Juan, J. S., Sasselov, D., Segransan, D., Sozzetti, A., Sosnowska, D., Stobie, B., Szentgyorgyi, A., Vick, A., and Weber, L., "Harps-N: The new planet hunter at TNG," in [*Ground-Based and Airborne Instrumentation for Astronomy IV*], *Proc. SPIE* **8446**, 84461V (2012).

[5] Sosnowska, D., Lovis, C., Figueira, P., Modigliani, A., Marcantonio, P. D., Megevand, D., and Pepe, F., "A Flexible and Modular Data Reduction Library for Fiber-fed Echelle Spectrographs," in [*Astronomical Data Analysis Software and Systems XXIV*], *ASP Conference Series* **495**, 285 (2015).

[6] Hanisch, R. J., Farris, A., Greisen, E. W., Pence, W. D., Schlesinger, B. M., Teuben, P. J., Thompson, R. W., and Warnock, A., "Definition of the Flexible Image Transport System (FITS)," *Astronomy and Astrophysics* **376**(1), 359 (2001).

[7] Oberstein, T. and Goedde, A., "The Web Application Messaging Protocol." https://wamp-proto.org/static/rfc/draft-oberstet-hybi-crossbar-wamp.html (Apr. 2018).

[8] Hessman, F. V., "Remote Telescope Markup Language (RTML)," *Astronomische Nachrichten* **327**(8), 751 (2006).