THE@INT Software Systems

John Young

April 26, 2019
OUTLINE

Introduction

Software architecture

Data flow
Overview

Introduction

Software architecture

Data flow
OVERALL STATUS OF SOFTWARE SYSTEMS

- Preliminary Design Review in May 2017
  - Updated documents needed
- Prototyping partially done
- Final Design Review planned for December 2019
  - Telescope refurb tender on critical path
Mission

- Design and implement a fully robotic control system for the INT and HARPS3
  - To carry out the Terra Hunting Experiment and open time observations
  - Must accommodate a possible companion instrument
  - Must accommodate a possible solar telescope
- Science requirements – see HARPS3-EXE-0004
- Operations model and use cases – see HARPS3-CAM-0007
OVERVIEW

Introduction

Software architecture

Data flow
**Control System Architecture**

- **Software Architecture**
  - Automated scheduler
  - Observing control
  - Engineering data collector
  - Engineering database

- **Data Flow**
  - Metadata
  - Science frame

- **System Components**
  - Cass
  - Tip-tilt
  - AG cameras
  - Motor controllers
  - TT actuators
  - Calibration sources
  - Telescope
  - Mirror sensors
  - Dome
  - Mount
  - Spectrograph
  - Housekeeping
  - T control
  - Pump control
  - Shutter
  - Science CCD

- **Architectural Diagram**
  - WAMP router
  - FITS builder
  - Science data archiver
**Instruments control software**

- **Telescope Control System (TCS)**
  - Integrates mount, dome and environment sensing
- **Cass Instrument Control Software (CICS)**
  - Cass Unit
    - ADC
    - Cassegrain Fibre Adapter
    - Polarimeter
  - Acquisition and guiding (2 fields)
  - Calibration Unit
- **Spectrograph Control Software (SCS)**
- **Detector Control Software (DCS)**
SUPERVISORY SOFTWARE

- Automated scheduler
  - Schedules daytime calibrations, nighttime observations, focus, pointing test
- Observing Control Software (OCS)
  - Observation Sequencer, Fault Manager, FITS Builder
- Data handling
  - Engineering data collector and database
    - Local database and automatic transfer to Exeter mirror
  - Science data archiver
    - Local storage, automatic transfer to Exeter, load into database
- Data Reduction Software
  - Based on ESPRESSO DRS
OVERVIEW

Introduction

Software architecture

Data flow
**Actors**

**Open time observer**  A scientist who wishes to carry out observations with the INT, in service mode, and make use of the resulting data

**Consortium observer**  A scientist from the HARPS3 consortium working on the Terra-Hunting Experiment (THE)

**Consortium engineer**  An engineer from the HARPS3 consortium, responsible for ensuring that both open time and THE observations are carried out successfully

**WHT operator**  Responsible for ensuring safe startup and shutdown of the INT. MoU states that the ING will provide safety override for robotic operation
SERVICE MODE OBSERVING AND DATA REDUCTION
Data handling overview
OBSERVING DASHBOARD

### Status

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduler</td>
<td>OK</td>
</tr>
<tr>
<td>Sequencer</td>
<td>OK</td>
</tr>
<tr>
<td>Telescope</td>
<td>OK</td>
</tr>
<tr>
<td>Dome</td>
<td>OPEN</td>
</tr>
<tr>
<td>Cass</td>
<td>OK</td>
</tr>
<tr>
<td>A&amp;G</td>
<td>OK</td>
</tr>
<tr>
<td>Spectrograph</td>
<td>OK</td>
</tr>
<tr>
<td>Detector</td>
<td>OK</td>
</tr>
<tr>
<td>Humidity</td>
<td>12%</td>
</tr>
<tr>
<td>Temperature</td>
<td>8.7 °C</td>
</tr>
<tr>
<td>Dew point</td>
<td>-19.6 °C</td>
</tr>
<tr>
<td>Wind</td>
<td>2.3 m/s</td>
</tr>
<tr>
<td>Cloud</td>
<td>CLEAR</td>
</tr>
<tr>
<td>Seeing</td>
<td>1.1&quot;</td>
</tr>
<tr>
<td>Sidereal</td>
<td>01:18:22</td>
</tr>
<tr>
<td>Local</td>
<td>18:42:01</td>
</tr>
<tr>
<td>UTC</td>
<td>18:42:01</td>
</tr>
<tr>
<td>Observation Block</td>
<td></td>
</tr>
<tr>
<td>Target</td>
<td>51 Peg</td>
</tr>
<tr>
<td>Duration</td>
<td>11 / 20 min</td>
</tr>
<tr>
<td>State</td>
<td>EXPOSE</td>
</tr>
</tbody>
</table>

### Webcams

- [Camera 1](image1.png)
- [Camera 2](image2.png)
- [Camera 3](image3.png)

### Actions

- [Suspend](button1.png)
- [Resume](button2.png)
- [Shutdown](button3.png)

Log messages:

- 18:31:01 OB start 51 Peg
- 18:33:17 Acquisition done
- 18:33:18 Expose start
Detail of telescope/instrument control
**Automated Scheduler**

**THE Scheduler**

- **Phase 1** - New observation group requested
  - Accepted?
    - Yes
    - **Phase 2** - Add observation group
      - Observation blocks added to pool
        - Scheduler gets observation block for 'best' target from pool
          - Scheduler
            - WEB APPLICATION MESSAGING PROTOCOL (WAMP)
              - Request for target sent across network
                - Observation block returned back across network to sequencer
                  - Sequencer
**Detail of Post-collection Data Handling**

**Diagram:**
- **Observer:** Candidate targets → Observation prep.
  - **Science storage:** Raw data as FITS → Auto transfer.
  - **Science archive:** Monitor keywords, remainder, monitor data, logs → Auto transfer.
- **ING Archive:** Raw data → Notification, assume ING Archive can enforce non-standard embargo for TIE data.
- **Data reduction:** Quality flags → Alert generator.
- **Alert generator:** Raw & reduced data → Open Time Observer.
- **OT query:** Data quality alerts, metadata → THE query, raw & reduced data.
- **Edit metadata:** Trigger, metadata → Consortium Observer, raw & reduced data.
ENGINEERING DATA COLLECTION

- INT/HARPS3 sub-systems will publish engineering data continuously
- Published data will automatically be collected and stored in the on-site engineering database
  - Data collection must be robust to sub-systems going offline/online
  - Query facility used by OCS FITS Builder
- Database mirrored in Exeter, updated nightly (TBD)
- High-bandwidth data only kept for N days, available for manual transfer
  - Guide camera video
  - Very low-level debug messages
SCIENCE/CALIBRATION DATA COLLECTION

- FITS files will be assembled from a raw detector frame and ancillary data/metadata provided by the robotic control system
  - Science metadata: required by DRS (timestamp, exposure, target and telescope coordinates, …)
  - Engineering metadata: summarise instrument status during exposure
  - Exposure meter time series
  - Acquisition and integrated guide images
- Temporary storage on site
- Transfer to main Archive in Exeter
  - Automatic pipeline reduction on import
  - Retain multiple, traceable reductions of each observation
    - Edit metadata (e.g. spectral type) and re-reduce
    - Upgrade DRS (occasionally) and re-reduce
Backup slides
Data Reduction Software

- ESPRESSO DRS status
  - Validated and operational on Paranal
  - Able to read HARPS data
  - New web-based "trigger" front-end for CPL recipes

- New work for HARPS3
  - Adapt ESPRESSO DRS for HARPS-X data and test (Geneva)
  - FITS Builder (Cambridge)
  - Integrate/adapt Trigger (Cambridge)
    - Details to be discussed with Geneva
  - Interfaces with HARPS3 Archive for pipeline input/output files (Exeter)
  - Science QC tools (includes alerts) for consortium observer (Exeter)
Alerts

- Alerts can be generated from
  - Engineering data
  - DRS-processed data

- Flexibility
  - Use any monitored variables
  - Complex alert criteria?
    - Combinations of variables
    - Linked to events
Operations UIs (consortium use)
  - Observing dashboard
  - Scheduler “queue” UI
  - Engineering archive access
User interfaces (2)

- Engineering GUIs (view only during robotic operation)
  - Telescope system
  - Cass instrument
  - Spectrograph
  - Detector

- Observer tools (community/consortium use)
  - Exposure Time Calculator
  - Scheduler simulation tool
    - For repeated observation or survey
  - Science archive access
Communications framework

- Robotic control of telescope + instrument requires a distributed event-driven software system
- Components communicate over network using WAMP protocol
  - Open standard WebSocket subprotocol
  - Data serialization using JSON or MsgPack
- Use open-source Crossbar.io router and Autobahn client libs (many languages)
WAMP

WAMP implements two communications patterns:
1. Remote Procedure Call (command/response)
2. Publish Subscribe
Realistic WAMP example

Python asyncio event loop

WAMP
- Receive remote procedure call
  - Translate
  - Convert into message for target instrument
  - Pexpect
  - Dispatch message to instrument and yield

WAMP
- Return from remote procedure call
  - Translate
  - Convert into return value
  - Pexpect
  - Receive message from instrument

Periodic timer
- Initiate poll of instrument data
  - Translate
  - Convert into message for target instrument
  - Pexpect
  - Dispatch message to instrument and yield

WAMP
- Publish value
  - Translate
  - Convert into return value
  - Pexpect
  - Receive message from instrument