Pan-STARRS PS1 Observatory, Telescope, and Instrument Control

E.A. Pier and K.C. Chambers and the Pan-STARRS Team
University of Hawai'i Institute for Astronomy

Abstract

This paper discusses the architecture and design of the Pan-STARRS Observatory, Telescope, Instrument, Software system or "OTIS". PS1 is a remotely operated observatory where we will gain the experience necessary to implement the fully robotic PS4.

1 Overview

"OTIS" is an acronym for "Observatory, Telescope, and Instrumentation Software". OTIS is the Pan-STARRS subsystem responsible for control of the observatory, in particular, hardware control, communication between components, metadata logging, atmospheric monitoring, automated operation, user interfaces, and data processing. See Figure 1.

Below we describe each of these aspects of Otis and the design choices we have made for them as summarized in Figure 2.

2 Communication

The PS1 observatory is intended to be operated remotely, and eventually robotically. This dictates a client-server architecture with a middleware layer defining the intervening communication protocol. Several middleware options exist. We chose CORBA because it is an open standard with open source implementations, object-oriented, language neutral, and uses an interface definition language to clearly specify allowed method calls and their forms.

3 Hardware Control

For each piece of observatory hardware (e.g. Telescope, Dome, Camera, Filter Changer, etc.), we create a CORBA server. This server is responsible for translating CORBA commands into the "native language" of the hardware. In some cases the CORBA server is talking directly to a microcontroller using a special purpose text-based protocol over a TCP socket. In other cases (such as the telescope and dome) there is a vendor-supplied server with a proprietary command protocol. Thus we use CORBA to provide a uniform mechanism for commanding and querying metadata from all our components.

4 Metadata Logging

Otis keeps a detailed and comprehensive record of the state of the observatory. We plan to use this information for diagnosing faults, and for performance and trend analysis.

We have chosen a MySQL database for logging metadata from the OTIS servers. This gives us the power of the SQL query meta-language with the flexibility of an open-source solution. Each hardware server is responsible for logging its metadata by writing time-tagged rows to database tables.

5 Atmospheric Monitoring

OTIS needs to monitor the current atmospheric conditions in order to calculate the corrections to the telescope pointing due to atmospheric refraction, and to keep the equipment safe from hazardous conditions. The OTIS weather station includes the following equipment:

- Paroscientific MET3A Meteorological System - Measures Atmospheric pressure with an accuracy 0.008 kPa from 62 to 110 kPa, Temperature with an accuracy 0.1 deg C from -50 to +60 deg C, and Relative Humidity with an accuracy 2 Percent from 0 to 100 %RH at 25 deg C.
- Vaisala WS425 Ultrasonic Wind Sensor - Measures wind speed with an accuracy of 0.135 m/s in the range 0...65 m/s, and wind direction with an accuracy of 2 degrees for wind speeds over 1 m/s.
- Vaisala CARBOCAP GMP343 Carbon Dioxide Probe - Measures the atmospheric abundance of carbon dioxide with an accuracy of 2.5% after factory calibration in an operating environment with temperature between 0 and 60 C, and pressure between 70 and 130 kPa.
- Vaisala FD12 Visibility Meter - Measures visibility from 10 - 50,000 m in an operating environment with temperature between -40 and 55 C, and relative humidity between 0 and 100%.

We will also have a Vaisala WXT510 Weather Transmitter mounted on the telescope, primarily to measure wind loads on our large secondary mirror. We will also use this instrument to measure the equalization with the outside air, and detect precipitation.

Finally, OTIS has an imaging sky probe to measure atmospheric transparency along the telescope's line of sight. This instrument is described in detail elsewhere in this volume.

6 Summit Data Processing

OTIS will need to do some image data processing in order to get immediate feedback on the atmospheric conditions, and the telescope focus. This subsystem will handle a vastly smaller volume of data than the Image Processing Pipeline described elsewhere in this volume. In particular OTIS's pipeline will determine transparency from skyprobe images, analyze wavefront sensing data from the Gigapixel Camera, and assess image quality from a small sample of the science data.

7 User Interfaces

OTIS includes a graphical user interface (GUI) capable of monitoring and controlling all aspects of the observatory. The design of this tool is heavily influenced by our requirement to operate the observatory remotely.

The OTIS GUI is a CORBA client written in Java. Java allows us to run the same tool on different platforms to accommodate user's preferences on their home computers. It also has built in SOCKS proxying support. We use this in conjunction with SSH port-forwarding to provide a secure means for monitoring and controlling the observatory from anywhere.

The GUI consists of several small "pages" which can be displayed one at a time in a tabbed pane, or can be "torn off" and displayed simultaneously on a user's desktop. The user can save a layout of pages and restore it later. This design makes the GUI versatile enough to use on a small laptop screen or on a large, multi-monitor desktop system.

Some additional features of the GUI include spoken audible alerts using the Java Speech API, a built in IRC (chat) client so that users in different physical locations can communicate, and a high-accuracy rendering of the sky which the user can pan and zoom to show the entire sky or just the instrument field of view.

OTIS also has several means for scriptable control of the observatory. The most basic of these is to make direct CORBA calls using Perl or Python. We also have a TCL-based shell-like command interface which can be scripted or used interactively.

8 Automation

As times goes on we plan to implement increasing levels of automation. The first and lowest level of automation is health and safety monitoring to e.g. close the dome if the weather gets bad. The next level of automation is queued observing, where a user enters a list of targets and OTIS observes them in sequence. The highest level of automation is full robotic operation, requiring no human actions during the night or even over several days.

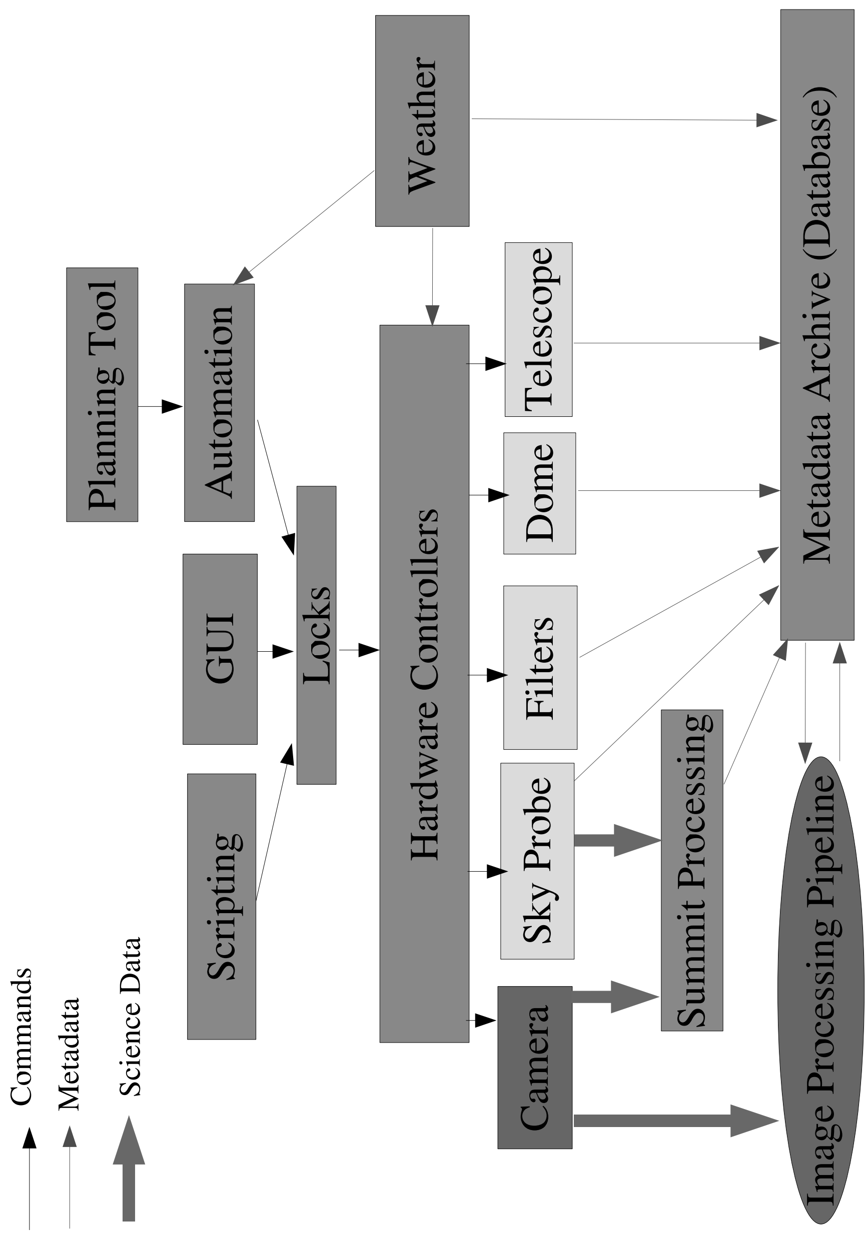

Figure 1: An Overview of the Pan-STARRS OTIS subsystem. OTIS is responsible for controlling hardware, logging metadata, monitoring the weather, and processing the small subset of data as needed for immediate response at the summit.

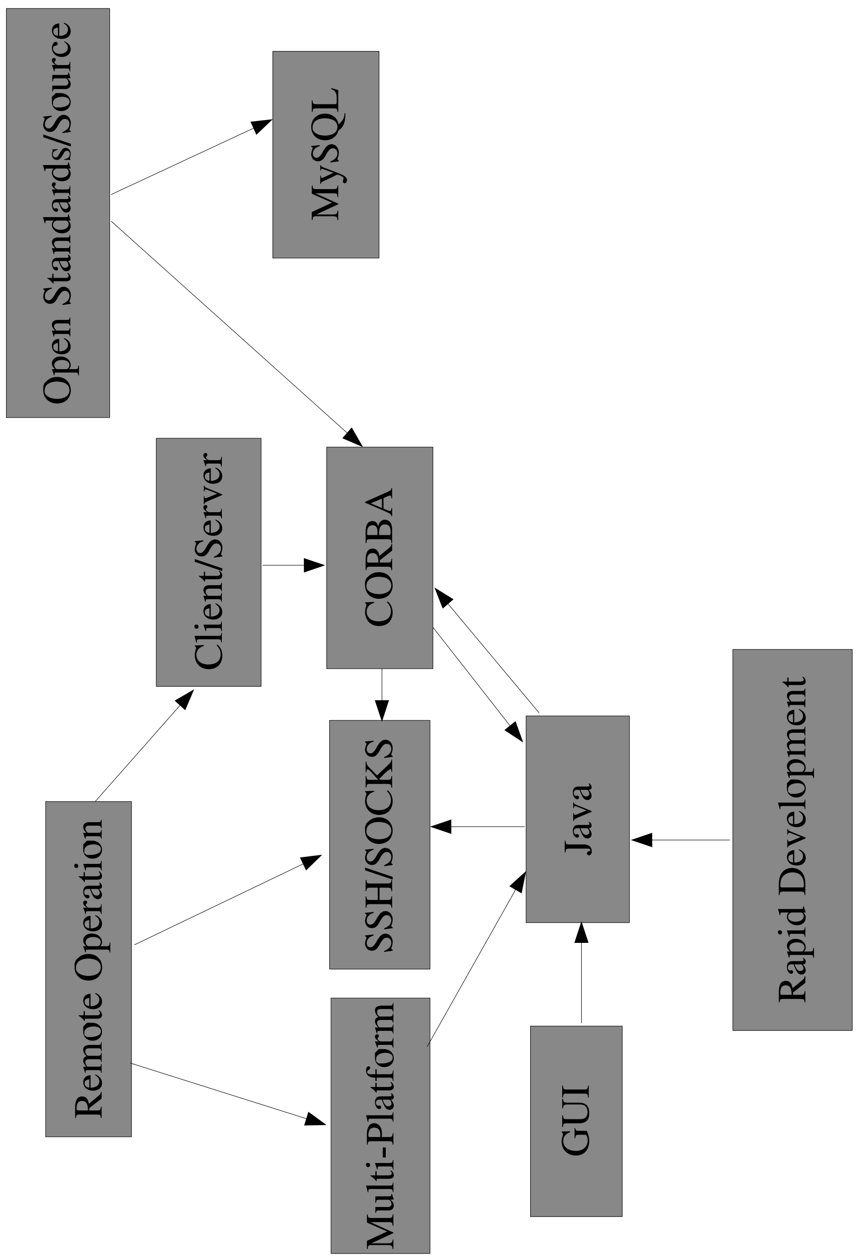

Figure 2: Design choices for OTIS.