System Architecture

1 Introduction
   1.1 Purpose
   1.2 Scope
   1.3 Definitions, Acronyms and Abbreviations
   1.4 Reference
2 Architectural Representation
3 Architectural Goals and Constraints
4 Frameworks and Libraries
   4.1 Java Message Service (JMS)
   4.2 Rich Client Platform (RCP)
   4.3 XML serialization/de-serialization
   4.4 GWAVES Catalog Server (CAT)
   4.5 Lowell Observatory User Interface (LOUI)
   4.6 Joint Operations Executive (JOE)
   4.7 Lowell Observatory Instrumentation System (LOIS)
   4.8 Architecture Overview – Subsystem Layering
5 Data Flow
6 Deployment View
   6.1 Desktop Workstation
   6.2 JOE/ActiveMQ/Database Server
7 Performance
8 Appendices
   8.1 Partial List of JMS Topics

Introduction

Purpose

This document provides an architectural overview of the software system for the DCT RC Guider and Wavefront Sensor System (GWAVES). It is intended to capture and convey the significant architectural decisions which have been made on the system. Even though this document does not explicitly address the science instrument package, the main concepts apply to all the Lowell built instruments.

Scope

The DCT RC Guider and Wavefront Sensor System (GWAVES) was developed by the Lowell Instrument Group. This document will cover the software system responsible for operating GWAVES but will not cover the details of the interface to the TCS or AOS. The interface to external systems such as TCS and AOS will be covered in Internal Interface Control Document.

Definitions, Acronyms and Abbreviations

This section gathers together some of the relevant definitions and abbreviations used in this document

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>GWAVES</td>
<td>GWAVES Control Software</td>
</tr>
<tr>
<td>CAT</td>
<td>GWAVES Catalog Server</td>
</tr>
<tr>
<td>LOUI</td>
<td>Lowell Observatory User Interface or LOIS User Interface</td>
</tr>
<tr>
<td>JOE</td>
<td>Joint Operations Executive</td>
</tr>
<tr>
<td>LOIS</td>
<td>Lowell Observatory Instrumentation System</td>
</tr>
<tr>
<td>GDR</td>
<td>Guider Control System</td>
</tr>
<tr>
<td>JMS</td>
<td>Java Message Service</td>
</tr>
<tr>
<td>RCP</td>
<td>Rich Client Platform</td>
</tr>
<tr>
<td>AOS</td>
<td>Active Optics System</td>
</tr>
<tr>
<td>DCT</td>
<td>Discovery Channel Telescope</td>
</tr>
<tr>
<td>MCB</td>
<td>Message Conversion Bridge (LabVIEW interface to the broker)</td>
</tr>
<tr>
<td>TCS</td>
<td>Telescope Control System</td>
</tr>
<tr>
<td>WFS</td>
<td>Wave Front Sensor Control System</td>
</tr>
<tr>
<td>LIG</td>
<td>Lowell Instrument Group</td>
</tr>
</tbody>
</table>
Architectural Representation

This document presents a coarse view of the architecture for the GWAVES software system as a group of modules working together. We will also present a brief description of some of the underlying frameworks and libraries used and the reasons for choosing each one.

Architectural Goals and Constraints

There were some requirements and system constraints that had significant bearing on the architecture. They are:

1. The existing hardware control system (LOIS) will be part of the overall system.
2. System must be modular and provide the basis for the science instrument control systems.
3. LOIS communication should offer the possibility of some remote operations in the future.
4. GWAVES has to interface with the DCT telescope control system (TCS) and DCT active optics system (AOS).
5. All performance and loading requirements, as stipulated in the DCT system design documents, must be taken into consideration.

Frameworks and Libraries

GWAVES Control Software (GWAVES) is basically a client server application with an intelligent controller for routing messages. The communication between the clients and the servers is done through a message broker. GWAVES uses several frameworks and libraries which were chosen for robustness and ease of development process.

Java Message Service (JMS)

Using a message broker as the means of communication between different layers of the software allows distributed communication that is loosely coupled, reliable, and can be either synchronous or asynchronous. For GWAVES we have chosen publish/subscribe model within JMS as our messaging protocol. There are multiple JMS providers both open source and proprietary and we have chosen Apache ActiveMQ. It is important to note that it is rather easy to switch our implementation from ActiveMQ to another JMS provider since we code against the interface and not the implementation.

Rich Client Platform (RCP)

Providing a rich and flexible user interface generally means the use of a standard framework such as wxWidgets, Qt, or XUL. The Eclipse Foundation's RCP was chosen as the GUI framework due to its portability, stability, and high quality development tools. RCP is a java framework which provides a consistent and native look and feel across applications and operating systems.

The RCP framework also provides a standardized component model. In other words, RCP applications are composed of components that plug into the platform. This component model allows very modular design which creates a powerful foundation for control software for other instruments built by LIG.

XML serialization/de-serialization

GWAVES leverages XML as its message transport protocol across the different software layers. We chose the Simple framework for the Java layer. The C/C++ layer in LOIS and the LabVIEW layer both use custom code written in house.

Logical View

The logical view of the GWAVES software system is comprised of 4 main modules:

GWAVES Catalog Server (CAT)

CAT is an RCP application responsible for searching and identifying suitable candidates for guide and WFS stars. Current database includes FK6, Tyco, and UCAC4 implemented on MySQL. CAT searches the database around a point of interest, usually the current science target, and will present an area template for the chosen instrument where the a suitable candidate is accessible. The potential candidates are represented both graphically and textually in a table. For more information on CAT and its operation please refer to the online documentation on Confluence.

Lowell Observatory User Interface (LOUI)
LOUI an RCP application responsible for presenting the primary camera control interface to the operator. There will be 2 main applications; one for the guider (GDR) and one for the wave front sensor (WFS). Each application is composed of multiple views with the primary views being the camera control, facility summary, and the imager.

The Shack-Hartmann analysis for WFS will be done externally via existing IDL procedures. The results are displayed and relayed to AOS via the message broker. The WFS LOUI is not directly involved with the analysis.

**Joint Operations Executive (JOE)**

JOE is a controller module responsible for intelligent routing of messages. JOE is basically an abstraction layer between the LabVIEW systems and GWAVES.

In addition to the camera controllers, GWAVES includes the hardware for control of:

- fold mirrors for instrument operations
- X & Y Stage of probes
- Focus Stage
- Filter Wheels

The above operations are performed in JOE by relaying the commands to the OMS cards via standard TCP sockets.

**Lowell Observatory Instrumentation System (LOIS)**

LOIS is a C/C++/TCL application responsible for direct camera/CCD control. LOIS will also perform the centroid operation on the guide star.

**Architecture Overview – Subsystem Layering**

- Gliffy Macro Error
  
  You do not have permission to view this diagram.

**Data Flow**
Instrument
Fold Mirror
Data Flow

start

Select Instrument

JOE

Move Fold Mirror

end
Deployment View
Desktop Workstation

Currently a dual monitor iMac is used by the operator for main control. This workstation will handle GWAVES operations but not those of the LabVIEW subsystems (TCS, AOS, ...). The LabVIEW systems run under MS Windows on a separate workstation.

The specifications for the iMac:
- Retina 5K, 27-inch, Late 2015
- 3.2 GHz Intel Core i5
- 32 GB 1867 MHz DDR3
- Dell P2715Q external display
- 2 TB Fusion Drive

JOE/ActiveMQ/Database Server

One linux server (joe.lowell.edu) is used to run JOE, the message broker, and the database server. There is a backup server which needs to be configured so it can replace joe in case of a hardware failure.

The specifications for joe:
- IBM System x3650
- QuadCore Intel Xeon CPU (E5420 @ 2.50GHz)
- 18GB DDR2 RAM
- 280GB SAS driver

Camera Controllers

Each camera will need one linux box with a PCIx card to accommodate the GEN-III Leach PCI card. We have chosen Dell PowerEdge R200 for this purpose.

Performance

We do not foresee any performance issues regarding the GWAVES software under the current architecture as long as the guide rate remains at ~1 Hz. Much faster guide rates may require some modifications to the architecture. The latency requirements and constraints for the messaging between GWAVES and TCS/AOS are documented in GWAVES Requirements Document.

Appendices

Partial List of JMS Topics

The number of topics used in the different LOUIs and JOE is very large (> 150). Some of the topics are specialized and are used for very specific communication within the apps. Below we have list of the more relevant topics which are used for communication between LOUI, LOIS, JOE and TCS.

<table>
<thead>
<tr>
<th>Name</th>
<th>Producer</th>
<th>Consumer</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOUI.RC1.loisAnalysis</td>
<td>LOIS on RC1 [2]</td>
<td>JOE</td>
<td></td>
</tr>
<tr>
<td>LOUI.RC1[2].loisCommand</td>
<td>JOE</td>
<td>LOIS on RC1 [2]</td>
<td>Send command to LOIS</td>
</tr>
<tr>
<td>LOUI.RC1[2].loisImage</td>
<td>LOIS on RC1 [2]</td>
<td>JOE</td>
<td>The last image readout from the CCD</td>
</tr>
<tr>
<td>TCSTcsStatusSV</td>
<td>MCB</td>
<td>JOE &amp; LOUIs</td>
<td>Telemetry broadcast by the MCB (TCSPublicStatus)</td>
</tr>
<tr>
<td>Functionality</td>
<td>Developer(s)</td>
<td>Funder/sponsors</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TCSTcsCommandSV</td>
<td>JOE &amp; LOUs</td>
<td>MCB</td>
<td>TCS commands: scienceTargetConfiguration, guideTargetConfiguration, wfsTargetConfiguration, etc....</td>
</tr>
<tr>
<td>TCSCommandResponseSV</td>
<td>MCB</td>
<td>JOE &amp; LOUs</td>
<td>Response to the TCS command (an echo of the command with an error code)</td>
</tr>
<tr>
<td>TCSGuiderCommandSV</td>
<td>JOE</td>
<td>MCB</td>
<td>Guiding correction offsets in ideal TCS (X,Y) focal plane co-ordinates</td>
</tr>
<tr>
<td>TCSGuiderCommandResponseSV</td>
<td>MCB</td>
<td>JOE</td>
<td>Response to Guiding correction offsets</td>
</tr>
<tr>
<td>TCSGuiderStatusSV</td>
<td>MCB</td>
<td>JOE</td>
<td>Expected Guide Target/Probe position in ideal TCS (X,Y) focal plane co-ordinates</td>
</tr>
<tr>
<td>NewScienceTargetSV</td>
<td>MCB</td>
<td>JOE</td>
<td>Science Target</td>
</tr>
<tr>
<td>TCSWavefrontSensorStatusSV</td>
<td>MCB</td>
<td>JOE</td>
<td>Expected WFS Target/Probe position in ideal TCS (X,Y) focal plane co-ordinates</td>
</tr>
<tr>
<td>aos.loisTelemetry</td>
<td>JOE</td>
<td>LOIS</td>
<td>AOS telemetry (e.g. secondary focus position) needed for inclusion in FITS headers</td>
</tr>
<tr>
<td>deveny.loisTelemetry</td>
<td>JOE</td>
<td>LOIS</td>
<td>DeVeny telemetry (e.g. lamp conditions) needed for inclusion in FITS headers</td>
</tr>
<tr>
<td>instrumentCube.loisTelemetry</td>
<td>JOE</td>
<td>LOIS</td>
<td>Cube telemetry (e.g. probe stage positions) needed for inclusion in FITS headers</td>
</tr>
<tr>
<td>lmi.loisTelemetry</td>
<td>JOE</td>
<td>LOIS</td>
<td>LMI telemetry (e.g. filter positions) needed for inclusion in FITS headers</td>
</tr>
<tr>
<td>tcs.loisTelemetry</td>
<td>JOE</td>
<td>LOIS</td>
<td>TCS telemetry (e.g. telescope pointing) needed for inclusion in FITS headers</td>
</tr>
<tr>
<td>wrs.loisTelemetry</td>
<td>JOE</td>
<td>LOIS</td>
<td>WRS telemetry (e.g. weather data) needed for inclusion in FITS headers</td>
</tr>
</tbody>
</table>