NOAO DPP Document PL011

FLIPPER: a FLexIble PiPeline framEwoRk

Francesco Pierfederici

National Optical Astronomy Observatory
Data Products Program

November 4, 2004
ASP Conf. Ser., Vol. 347 ADASS XIV, 2005

NOAO Data Products Program, P.O. Box 26732, Tucson, AZ 85732

Copyright © 2006 by the authors.

Table of Contents

List of Figures

Abstract

FLIPPER is a lightweight pipeline framework able to handle blackboard-based pipelines. Its ease of configuration and operation and the fact the it is extremely lightweight make FLIPPER the ideal solution for both small and large pipeline systems. FLIPPER is currently being used for the NOAO Mosaic Pipeline and has been chosen as pipeline framework for the NOAO NEWFIRM pipeline system. The present paper describes the architecture, features and design choices of FLIPPER in detail.

Keywords: Pipelines, Mosaic, Distributed Computing, High Performance Computing, Pipeline Frameworks, Python

1 Overview

FLIPPER is a lightweight pipeline framework able to handle blackboard-based pipelines. Some of its main features are:

- Only one polling process per pipeline.
- Very few software dependencies (just Python and the XML parser).
- All the major Operating Systems are supported (Linux, Mac OS X, Windows and most unices).
- Pipelines are fully described in XML files (one file per pipeline).
- Pipeline modules are activated when trigger conditions are met.
- The most popular trigger types are supported (time, file and event based).
- Pipeline modules can define pre and post-processing actions.
- Pipeline modules can be written in any language.

2 Architecture

FLIPPER's Data Model (fig. 1) is centered around the concepts of Pipeline and Pipeline System. A Pipeline System is a collection of Pipelines and dependency rules. Similarly, Pipelines are collections of Modules (atomic pieces of code) and dependency rules between them.

Each Pipeline System is handled by a Resource Manager (fig. 2). Its main role is to make sure that the dependencies between Pipelines are satisfied. Its other function is to act as both a load balancer between processing machines and as a registry of the resources available to the Pipeline System.

Each Pipeline has a Pipeline Manager associated to it. The Pipeline Manager keeps track of the instantaneous state (stored in two separate Blackboard objects) of the corresponding Pipeline.

Pipelines are sets of Modules. Each Module is controlled by a Module Manager, which, querying the parent Pipeline Manager, ensures that all the relevant dependency rules are satisfied. The interaction between Pipeline Managers and Module Managers also determines which dataset Modules need to operate on.

Modules are activated when Trigger conditions are met. FLIPPER defines three Trigger types: Event, File and Time. Event Triggers specify which dataset processing state activates any given Module (e.g. bias subtraction must occur before the flat field can be applied). File Triggers activate Modules when a file matching a given name pattern appears in a given directory. Time Triggers define Modules that need to execute at a given point in time and/or at a specified interval.

FLIPPER offers the possibility to define actions to be performed (i.e. executables to be run) just before each Module is activated and just after it has finished execution. It is possible to have different post-processing actions depending on the exit code of the corresponding Module.

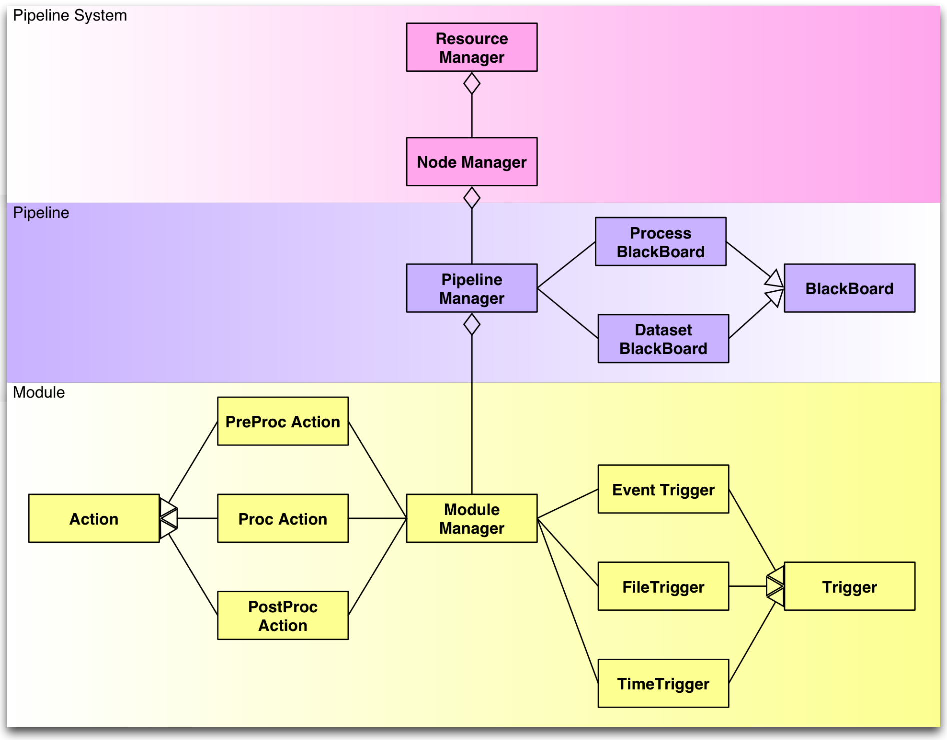

Figure 1: Class diagram for FLIPPER.

Each processing machine is controlled by a Node Manager (fig. 2 and fig. 3). The Node Manager monitors the machine load, available disk space and, communicating with the Resource Manager, makes its machine (and Pipelines) available to the Pipeline System.

Operators can monitor and interact with the System by using optional GUIs (that communicate directly with the Resource Manager). Alternatively, it is possible to log into any machine and use host-callable commands to query the status of the different components.

3 Configuration

Each Pipeline is fully described in a single XML file. This pipeline description uses the objects defined in the Data Model (fig. 1). A separate XML file is used to describe the full system (and how different Pipelines interact).

This approach not only simplifies the overall setup and configuration of the whole system, but also allows for Pipelines to be build up dynamically at runtime. It then becomes easy to envision a system in which, say, archive users are able to build customized Pipelines on the fly.

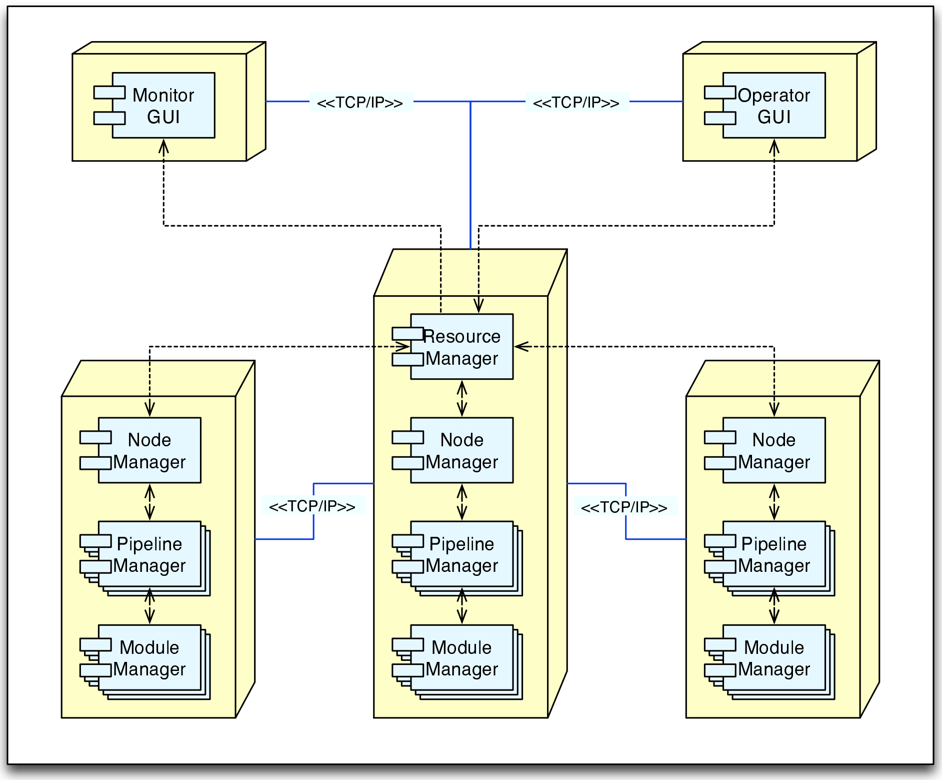

Figure 2: Multi-node deployment diagram. Three processing modes are shown. Optional GUIs can be used to monitor and control the whole processing network.

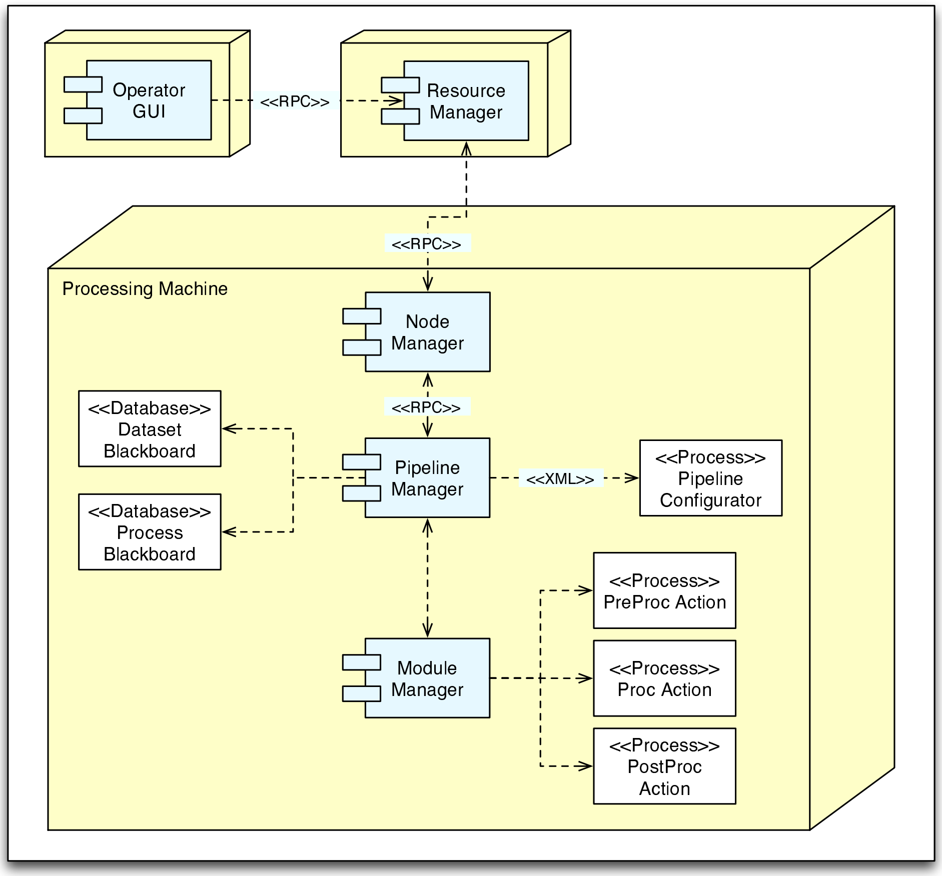
Operator GUI
<<RPC>>
Resource Manager
<<RPC>>
Processing Machine
Node Manager
<<RPC>>
<<Database>> Dataset Blackboard
<<Database>> Process Blackboard
Pipeline Manager
<<XML>>
<<Process>> Pipeline Configurator
<<Process>> PreProc Action
Module Manager
<<Process>> Proc Action
<<Process>> PostProc Action

Figure 3: Deployment diagram for a single processing node. Resource Manager and Operator GUIs are shown for completeness.

4 Status and Availability

FLIPPER is still under development. In its current state it is stable enough to be used in operational environments (e.g. both the NOAO Mosaic Pipeline System and the NEWFIRM Pipeline System).

Two components of the architecture need more development: the Resource Manager (in particular, the interaction between pipelines) and Operator/Monitoring GUIs. For this reason, FLIPPER is not yet available to the public.

References

[1] Pierfederici, F., Valdes, F., Smith, C., Hiriart, R., Miller, M. 2004, in ASP Conf. Ser., Vol. 314, ADASS XIII

[2] Hiriart, R., Valdes, F., Pierfederici, F., Smith, C. & Miller, M. 2004, in ASP Conf. Ser., Vol. 314, ADASS XIII

[3] Miller, M., Valdes, F., Smith, C., Hiriart, R. & Pierfederici, F. 2004, in ASP Conf. Ser., Vol. 314, ADASS XIII

[4] Valdes, F., Smith, C., Hiriart, R. Pierfederici, F. & Miller, M. 2004, in ASP Conf. Ser., Vol. 314, ADASS XIII