**Overall Software Design**

The overall design is shown in Figure 1. It shows the high-level components which make up the overall design, each of which are discussed briefly below, and further decomposed in their own Subsections. The software developed for this project is intended for extensible use, potentially accepting source data from multiple catalogues.

The Application Programming Interface (API) is designed to take the original source data (in this case, SDSS data), examine it to locate the required data for the main software processes, and extract only that data which is needed into a “tidy data” format as described by Wickham, 2014. [1] This format is known as the Local Catalogue, and is structured in a manner that can be used by the main data pipeline.

It is planned that the project will be extended to take in input data from another source. This will require that a corresponding API will be generated to suit, using existing SDSS API as a template.

The main data pipeline is a flexible process that uses parameter files and command line arguments to identify input data from the local catalogue and specify the particular analyses to be performed. By modifying the input parameters, the software can be run in two different modes (Catalogue Traversal and Target List.) This project incorporates two primary analyses which used these modes: an analysis of quasars contained in SDSS and a full analysis of all stars in SDSS for their suitability for exoplanet search by the transit method.

The Parameterisation program, generates Parameter files for both API and Pipeline. Parameter files contain lists of the paths to input files needed for the API, and both files and targets for the Pipeline. This program allows for flexible use of the two main components.

Grid computing allows each of these components (API, Pipeline and Parameterisation) to be run many times in parallel to deal with the volume of data that has to be processed. Each has their own suite of grid management software to generate, submit and manage grid jobs. As the various sets of grid management software have similar requirements, and thus are designed in a similar way.

Specific to SDSS is the requirement to identify fields by means of SQL queries to the CAS. These queries, accept user parameters and generate lists of targets and field identifiers used as input to the parameterisation software. These queries are used in relation to both the Source and Local Catalogues because they use the same directory structure.

Within each of the C programs, a robust system of Error Checking is used to ensure reliable operation, and the identification of both fatal errors and warnings within the program. This system was developed as a single portable module and is used universally throughout the project. Note that while error checking is not explicitly referenced in the definition of the programs below for clarity, all function calls and any other operations which could potentially raise an exception are subject to error checking.

For each catalogue

Local Catalogue

Local Index

SDSS catalogue

Source catalogue

SDSS API

CAS output

Output catalogue

Source catalogue

Other API

Catalogue Index

Main Data Pipeline

Parameters

Parameterisation

See 7.2.3

SQL Queries

Other Index Query

Arguments

Figure 1: Top-Down Software Design: User Parameters define SQL queries to the CAS and are used by the Parameterisation system to define API jobs which process the SDSS catalogue to generate the Local Catalogue. The Local Catalogue is then processed with further user arguments in the Main Data Pipeline to produce the Output Catalogues. Each component is discussed in its respective subsection. Not shown are Grid Management and Error Checking systems which are integrated within other software elements.

# References

|  |  |
| --- | --- |
| [1] | H. Wickham, "Tidy Data," *Journal of Statistical Software,* vol. 59, no. 10, 2014. |