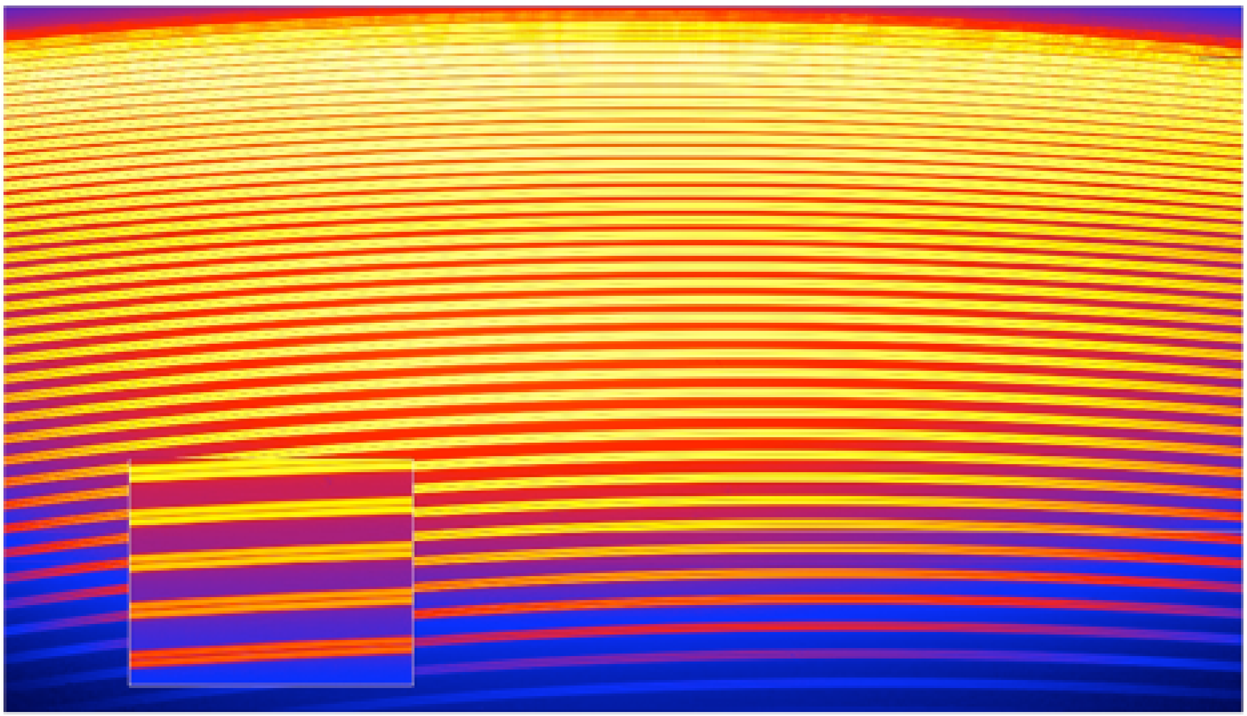
Canada-France-Hawaii Telescope Corporation
65-1238 Mamalahoa Hwy, Kamuela, Hawaii 96743 USA
CFH
Société du Télescope Canada-France-Hawaii
Telephone (808) 885-7944 Fax (808) 885 -7288

OPERA PRELIMINARY CONCEPTUAL DESIGN



CFHT OPERA TEAM
Canada France Hawaii Telescope
Waimea, HI February 2011

Contents

I. Introduction

OPERA (Open-source Pipeline for Espadons Reduction and Analysis) is an open-source software reduction pipeline for echelle spectro-polarimetric data which will be used for reducing the CFHT ESPaDOnS data. This document describes the conceptual design of the pipeline.

II. Rationale

CFHT is always looking forward to improve its resources to provide optimal data products aiming the best scientific utilization of these data. The internal development of instruments, softwares and operations has always been the keystone for making this happen, consequently turning CFHT a pioneer on all areas of research covered by our instruments.

After more than 5 years of operating ESPaDOnS, the CFHT team has achieved a considerably better understanding on how to improve the performance of this instrument. This knowledge has been reverted into the improvement of data products in several ways. As for the instrumentation, a couple of upgrades has been done recently, like the installation of a new ADC unit that minimizes the cross-talk between polarization modes and the development of a new CCD chip (Olapa) with better performance than the previous one (EEV1). As for the operational side, many advances has also been made, for instance the successful implementation of QSO (Queue Service Operation) mode, which has increased significantly the observing efficiency without losing quality.

The software to process the data is what makes the bridge between the knowledge obtained by experience with an instrument and the actual implementations that will improve the data products. There has been an inhouse development of a software to process ESPaDOnS data, the Upena pipeline. Upena is a high-level harness that controls and directs the execution of the lower level Libre-Esprit software. The latter contains the main modules to process all the calibrations and perform the extraction of the spectra from ESPaDOnS. For several times the CFHT team has been attempted to implement new ideas and improvements at the lower level data processing on ESPaDOnS. The necessity for this interaction has led us up to think of developing a new software.

ESPaDOnS will continue to be one of the main instruments at CFHT for an undetermined time. Therefore, CFHT decided to start the OPERA project, which is an open source software that will be managed by the CFHT team and will be developed as a collaborative work. Our main users, the scientists, are welcome to use OPERA and also to give their own input for further developments and improvements. The open-source nature of OPERA also permits other instruments and telescopes to be part of the project.

III. The Data Reduction for ESPaDOnS

The ESPaDOnS instrument is an echelle spectro-polarimeter, which records the dispersed light of an astronomical source as a multi-order spectral pattern projected on a two-dimensional CCD array, which provides the raw image data.

1. Image Data

The data obtained by ESPaDOnS from a given exposure are condensed into a FITS image digital file. The FITS file is the standard format widely used by the astronomical community, which consists basically of two parts; the informative header units and the data units. Header units are purely text entries. The data units can be either a table or a multi-dimensional array. An ESPaDOnS exposure consists of a single FITS image file that contains one main header unit and one data unit array with 2080 x 4640 pixels. In the header unit one can find all sort of information regarding the instrument status, data format, physical and environmental conditions at the time of the exposure, time, etc. The data unit contains an array of pixels, on which each pixel is a digital record of the electrical charge generated by the photoelectric device. This record, also called “counts”, is a proportional[1] measurement of the total energy transferred by the incident light on that area of the detector during the exposure time.

2. Data Reduction

In order to perform a scientific analysis of ESPaDOnS data, one needs to convert the multi-order two-dimensional pattern into a one-dimensional spectrum, which is basically the energy flux versus the wavelength (or frequency) of the light. Therefore the main goal of a reduction pipeline software is to process these image data and provide the final 1-D wavelength/flux-calibrated spectrum. Since ESPaDOnS can also be used as a polarimeter, for the polarimetric mode the pipeline should also provide the spectrum of the degree of polarization for a given Stokes parameter.

The way to calibrate both flux and wavelength is by making use of reference exposures, which we call the calibration data (for short “cal”). These are basically spectral images obtained with known sources of light that will provide the reference for recognizing and removing instrumental or any other external effect. This will allow one to separate the information coming exclusively from the astronomical object. Another important step on this process is the extraction of the energy flux from the 2-D array data. There are many different approaches to perform the calibration and extraction. Here, as a preliminary concept for the pipeline, we will just describe some basic ideas and procedures to obtain the final calibrated spectrum.

3. The calibration data

The identification of instrumental effects on the data is possible through a set of calibration data, which should be taken at least once every night of observation and one set for each observing mode-speed. For ESPaDOnS, these data consist of four types of exposures; bias, flat-field, comparison (Th-Ar), and align (Fabry-Perot). Each of these types and their utilities will be briefly described on Section IV.1.

[1] The proportionality may not be linear for some ranges of counts and may also be color-dependent.

IV. The Components of the Pipeline

The pipeline can be divided into five components. (1) raw data, (2) library data, (3) configuration data, (4) products, and (5) software. Each of these components are described below.

1. Raw Data

Raw data is the set of exposures necessary to perform the reduction. The raw data consist of calibration exposures plus science exposures. Below we describe each of the raw data types.

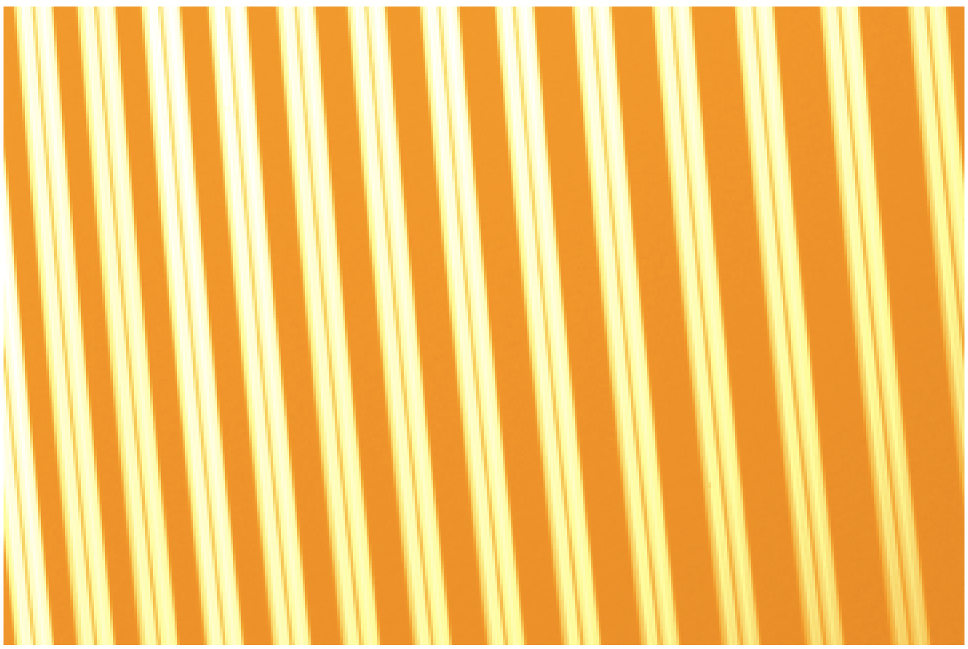
1.1. Bias

Bias is an exposure with zero integration time. The bias exposures allow the measurement of the intrinsic detector read-out noise due to electronics. This noise consists of a fluctuation of counts that pixels may contain on every exposure. Figure below shows a sample of a typical bias exposure for ESPaDOnS with the EEV1 detector.

The reason detectors have “bias” can be many, for example thermal generated electrons or static current caused by electronics. For most detectors the combination of all effects results as random noise, which can be best estimated by combining multiple exposures into a “master” bias frame.

1.2. Flat-field

Flat-field is an exposure taken with an incandescent source of light. The exposure time should be sufficient to obtain a significant flux overall the chip without saturating. Figure below shows a sample of a flat-field exposure taken with ESPaDOnS in polarimetric mode.



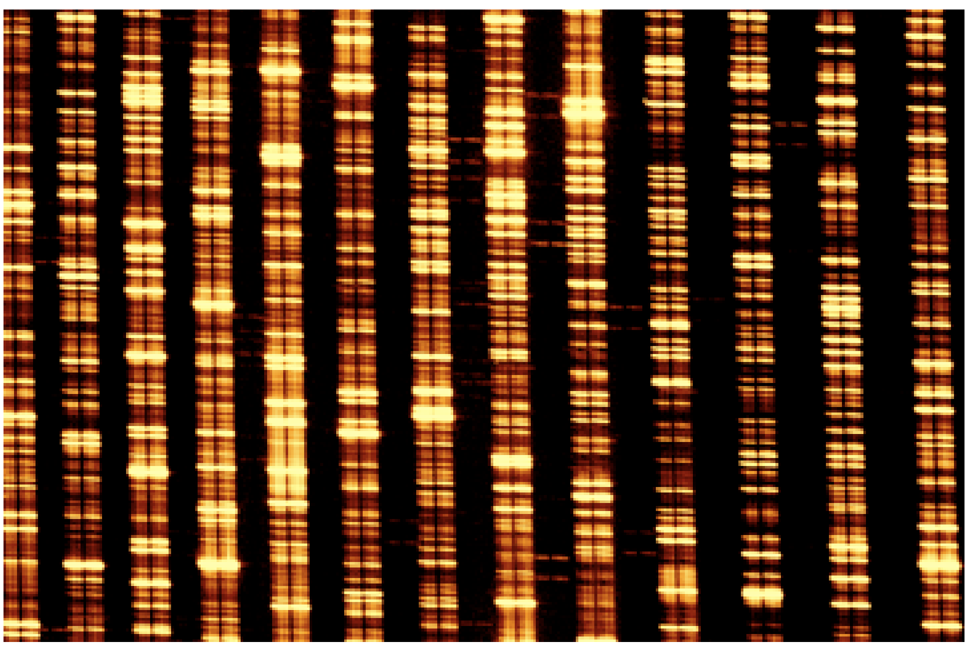
The flux of an incandescent light is ideally smooth over the spectrum, which provides a way to identify pixel-by-pixel sensitivity variations. However, the identification of a flat-field response for echelle spectrographs is not trivial. Some regions of the CCD in between orders do not receive enough light to allow the estimation of the flat-field response. Therefore it is only possible to perform the flat-field correction over the illuminated region.

Flat-field exposures has another important role for echelle spectroscopy. They can be used to detect the regions of the CCD where light is allowed to go in. In other words one can use flat-fields to detect the position of spectral orders.

Another utility of the flat-field exposure is for measuring the blaze function, which is the large scale illumination pattern over the entire CCD. This variation is mainly caused by vigneting and also by the variation of intensity between spectral orders.

1.3. Comparison (Th-Ar)

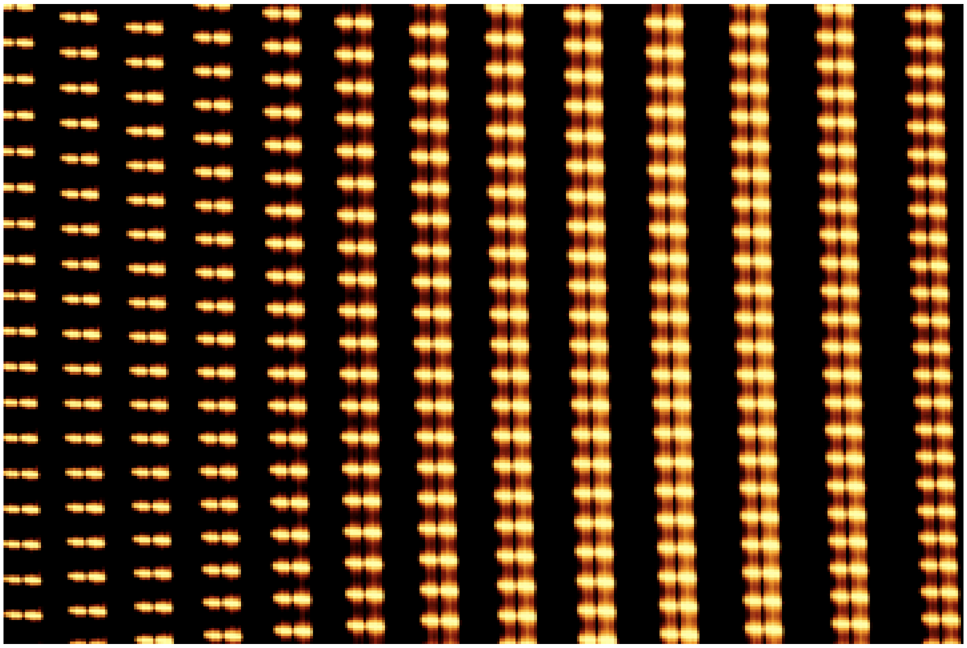
Comparison is an exposure of a reference lamp for which the source of light is dominated by quantum transitions between orbitals. At CFHT we use a lamp of Thorium and Argon (Th-Ar) gas as reference. It produces a spectrum that contains many emission lines along the whole spectral range that is covered by the spectrograph. These lines are known to emit in very specific energies, therefore the wavelength of the light detected for those lines are known. This permit us to build a pixel-to-wavelength transformation function for the entire CCD, which provides a way to obtain the desired calibration of the spectrum. Figure below shows a sample of Th-Ar comparison exposure.



1.4. Align (Fabry-Perot)

Align is an exposure of an incandescent source of light passing through a Fabry-Perot etalon device. The light is reflected multiple times inside the etalon causing interference of light, which produces a pattern of multiple peaks (lines) on the spectra. These peaks are regularly spaced over the entire spectra. This feature makes the align exposure an excellent tool to identify some instrumental features, like the “slit” geometry and position. The slit is the focused image that defines the “point-like” response of the spectrograph. As a matter of fact ESPaDOnS does not have a real slit, rather it has a slicer unit that splits up the rounded fiber output into aligned pieces to form a “pseudo-slit”. This makes the slit to have an unusual shape as shown in the figure below.

The align exposure allows one to identify the slit geometry and defines the spectral bin where one can extract the flux with the maximum possible spectral resolution. Figure below shows a sample of a Fabry-Perot align exposure.

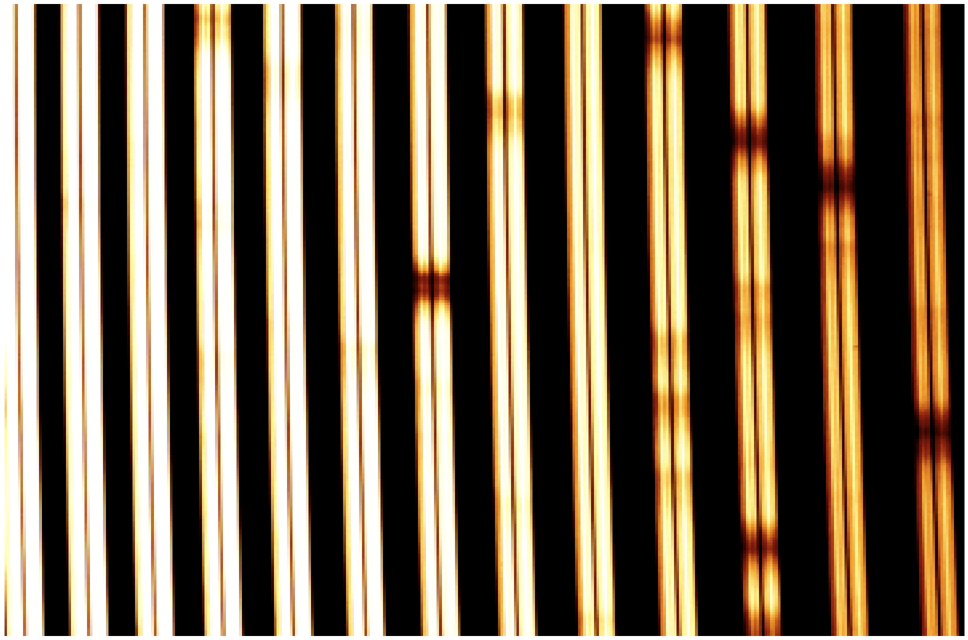


1.5. Dark

Dark is an exposure taken with the same integration time as the science exposure but with the shutter closed, so no light can reach the detector. This is usually obtained in order to estimate the noise generated by the cumulative electronic noise on the detector. For ESPaDOnS instrument we do not make use of dark exposures because the detector noise for the maximum practical exposure time can still be neglected.

1.4. Science (object exposure)

The science (or object) data is an exposure of the astronomical object. The nature of the source of light may vary a lot depending on which object is being observed. Some spectra are almost featureless (mostly continuum with a few lines), some have absorption lines, others have emission lines, all of various widths and depths. Therefore, the science data must not be used to obtain any information for calibration. Figure below shows a sample of a science exposure, where the source is a bright star.



2. Library data

The reduction at some level may require some additional information to calibrate the data with respect to an absolute reference. This information is stored as the “library data”. All library data are assumed to be constant over time. These are, for example, the Th-Ar lines, the quantum efficiency of the CCD, the bad pixel mask, etc. Although they are assumed to be constant, they should always be checked and may be updated by newer better data as needed.

3. Configuration Data

The data reduction may also depend on parameters that one may want to change according to the necessity. Some of these parameters can be instrument related, choice of statistical approaches, threshold acceptance levels, etc. In order to set-up those parameters the pipeline will have to access a Configuration Data Layer (CDL).

The CDL is accessed by a software library (Configuration Access Layer) that the Harness calls to abstract out the actual storage mechanism for the Configuration Data. A particular installation may choose to store this data in a text file, and that file may be plain text or perhaps XML, or the Configuration Data may be store in a database. No matter what the actual storage, the Harness does not change, it calls the CDL library routine to request the configuration parameter by name, the library uses the particular installation method access and returns the value. Note that the Harness accesses the CDL, not the module. The Harness passes the parameter values to the modules as arguments.

Configuration for ESPaDOnS has to be qualified by <detector>-<mode>-<speed>, where:

<detector> is one of EEV1e, EEV1, OLAPA-a, OLAPA-ab

<mode> is one of POL, SP1, SP2

<speed> is one of FAST, SLOW, NORMAL

4. Products

Products are files generated by the pipeline. These are FITS or text format files containing tables of reduced spectral data. The exact content of OPERA products will at least match the contents of Upena-1.1 products. Below we discuss the Upena products.

4.1 Upena 1.1 Products

Upena processed data is presented in 2 formats: a FITS image table format, and the original Libre-ESpRIT ASCII table format (*.s files).

a) **FITS format processed files (*i.fits *p.fits)**

i spectroscopic

p polarimetric

All object files (*o.fits) are individually processed to extract the spectrum (polarimetric and "Star Only" modes) or spectra ("Star + Sky" mode), whether they were taken in spectroscopic or polarimetric mode. These spectra are saved in *i.fits files. The reduction is done once with the continuum normalized to 1, and again with the continuum not normalized. In the FITS processed files, the reduction is also done the standard way (without the 'w' option), and with the 'w' option (which cancels automatic wavelength correction from telluric lines, and is used mostly by spectroscopists). Therefore, all *i.fits files contain 4 spectra (or 12 in the case of "Star + Sky" mode). The spectral and polarimetric data is stored as FITS image data in 32 bit binary floating point tabular format.

For the polarimetric case, sets of 4 exposures produce one *p.fits file, which again presents data reduced with the continuum normalized to 1 and not normalized, and also with and without the 'w' option. Whether the continuum is normalized or not, the Stokes parameters are always unnormalized (i.e., the column gives for example V/I * I where I is the intensity).

Appendix 1 presents the FITS keywords that identify the type of reduction and content of each file.

b) **Libre-ESpRIT formatted processed data**

The ASCII output from Libre-ESpRIT is also provided to PIs in *.s files, using the odometer number of the raw file (or the first raw file of a polarimetric sequence) as a base, and then adding one or more letters to identify the content:

i	intensity spectra
p	polarimetric spectra
n	continuum normalized to 1
u	continuum not normalized
w	data reduction done with the “w” option

All spectra are provided along with respective uncertainties (error bars). For the polarimetric spectra, it is also provided two null polarization spectra (or also called check spectra). Each *.s file also has a *.out file which gives details of the reduction. In particular, the end of this file presents SNR values for each order, radial velocity correction from telluric lines (if applicable), and estimates of the V magnitude and effective temperature (in the normalized case only).

4.2. By-products

By-products are those files produced by a module that are required by another module later in processing, but are not delivered as a OPERA pipeline product. The byproducts are deleted by the harness with a deletion command.

4.3. Temporary Products

Temporary products are files that are only used within a module and may be deleted on module exit. The module itself should remove these files on successful exit.

5. Software

OPERA software will be divided into three main components:

1) **Harness**

   The harness is the high level layer software that manages all operations of the pipeline. The task of the Harness is to accept user commands and realize these commands by executing modules in processing order. The Harness accesses the Configuration Data Layer (CDL) to gather installation configuration data and the Parameter Access Layer to retrieve processing parameters and passes these to the modules. The Harness also access the Data Access Layer to locate the source data (calibration images, object images) and pass the locations of the data to the modules. The Harness manages cleanup of partly created pipeline Products on abort. The Harness manages parallel module execution and distribution of processing on multiple computers if that is required on a particular installation.

2) **Modules**

   Modules are execution units. Conceptually a module takes multiple inputs from the Harness and products a single output product. A module executes a single step in the pipeline process. A module does not make any decisions about file or directory locations or location of processing parameters - all these are the domain of the Harness and will be passed to the module on the command line. A module should not preclude multiple instances executing simultaneously on different inputs. For example, it could be a module that combines many exposures into a master frame, or it could be a module that performs some calibration. The structure and functionality of modules will be discussed in more detail later.

3) **Software Libraries**

Three software libraries are planned. These are:

a. Parameter Access Layer - a software library that retrieves OPERA pipeline execution parameters by name. Called by the Harness.

b. Configuration Access Layer - a software library that retrieves installation configuration data by name.

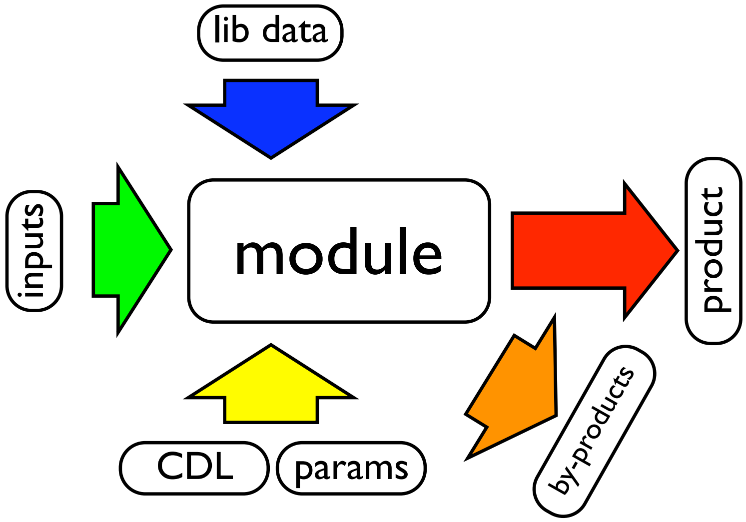
c. Image Access Layer - a software library that offers image manipulation routines, such subtracting a number from a whole image, accessing a pixel efficiently, etc.

V. The Pipeline Concept for OPERA

1. Modularization

As an open source and collaborative work, OPERA should be a software that is portable and most importantly it should have the ability to add and/or remove parts to the main frame of the pipeline. This feature will make the external collaboration easier since contributors will be able to develop small pieces of program that can be incorporated into the main pipeline. Additionally, this feature also allows us to offer different set up modes for the reduction, which may be changed to be optimal for each specific scientific goal.

The entire pipeline is a sequential or parallel execution of modules. Each module is an independent program, which can be called from the command line. The module should also have all input files, parameters, options, and product declared explicitly as arguments on the command line. The diagram below illustrates the schematic model for a given module of OPERA.




```
./module [options] par1 par2 ... input1 input2 ... product
```

As shown above, all options, parameters, input files and product/by-products are listed on the command line. A module may take multiple inputs, but should produce only a single main product. The By-products should be produced with care because it will not be identified and managed by the harness. Byproducts should never be tested for existence and skipped if the Byproduct exists. Rather the module should always create the Byproduct from scratch. The reason for this is that if a Byproduct is only partially completed when an abort is received, then the next execution could fail with incomplete input. Parameters are retrieved by the Harness and passed as arguments on the command line. The minimum standard options for any module should be the following.

`--verbose`: activate verification messages.

`--debug`: activate debug messages.

`--trace`: activate messages that trace modular calls with inputs and output.

`--help`: print out a short explanation about the module functionality as well as an instance of usage, and abort.

- Byproduct directory
- Temp directory

In order to facilitate the collaboration, CFHT will provide a template of a module written in C language.

2. The Processing Pipeline

Although each module is completely independent they will certainly have dependency on some files that are eventually generated as a product from another module. As a practical point of view, the inputs of any module must either exist from the raw data or from by-products of modules that have been called previously. The execution order of modules will be completely managed by the Harness. However in order to recovery from aborts possible the harness must have available the product of each module.

All modules must send a log report as would be useful for determining proper operation of the module to standard output (stdout or cout) and errors to standard error (stderr or cerr). No module should ever expect input from stdin.

2. Adding/Removing Modules

Modules should be able to be added and removed from the pipeline in the OPERA Pipeline Analysis and Post Reduction phase. However, the pipeline can only run through if all modules to be removed and/or replaced do not compromise the functionality of the whole system. Therefore, all expected inputs from all modules must exist and should be in the accepted format for the next modules to start running, otherwise the pipeline will break. The Harness guarantees the existence of complete input data for the modules. This OPERA version distributed to the community may have modules that contributors want to replace by either a testing new module or by an additional module with different approaches. In this case the replacement/ additional module must have exactly the same inputs and the product should be the same format as the one produced by the current module. Modules may be attached to the end of the pipeline in what we term the

OPERA Pipeline Analysis and Post Reduction phase. Products may be input to additional modules, and so on. This would create a new branch of the pipeline that is open-ended, which illustrates the flexibility and extensibility of OPERA.

3. Abort-ability

Any module should be able to abort at any time of execution without causing damage to any of the input or library data files. Temporary files and By-products must be created from scratch to ensure that they are not partial. The Harness manages cleanup of Products on abort but not Temporaries or Byproducts. A corrupted Product is not a problem since it is explicitly declared on the command line and therefore can be handled by the Harness.

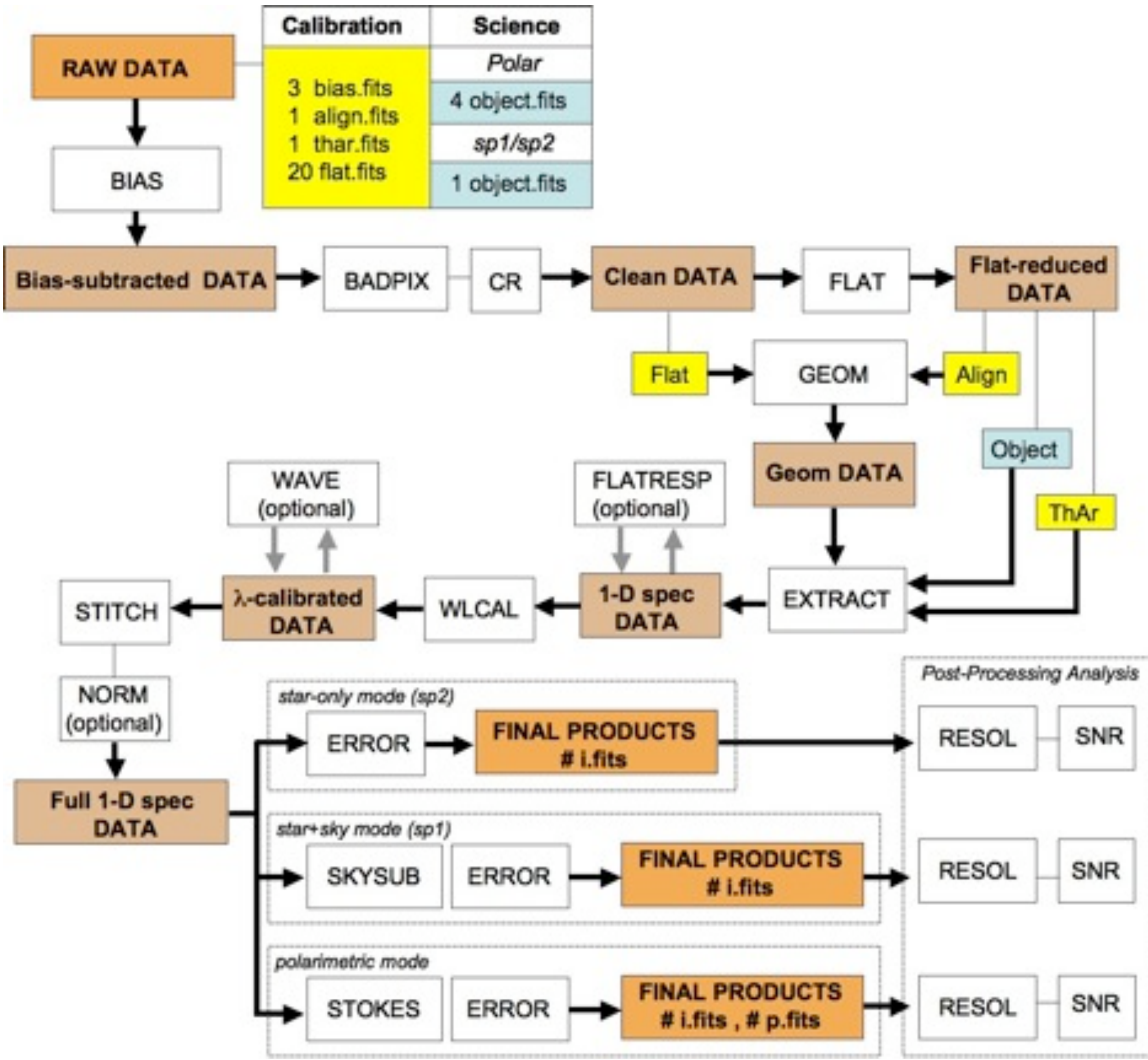
4. Restart-ability

The pipeline should be able to restart from an intermediate step or steps where it has been aborted. The ability to restart is related to the ability of recognizing where it has been aborted the last time. We propose the following model to build a pipeline with this characteristic. The pipeline runs backwards. It is forced to start the processing from the last module and if it fails, then it moves back to the previous module (or to a branch of modules if they are assembled in parallel). Since all inputs are declared on the command line, the harness will check if the Products exist for that module, then it will only restart to processing when the check does not fail.

VI. The Structure of OPERA Core

1. OPERA Core Reduction

The *OPERA Core* (OC) is the pipeline version that CFHT will provide and support for the standard reduction to distribute reduced data to the PIs on a daily basis. The OC is the most basic version of OPERA, where it has the basic modules to perform the minimum standard reduction of the data. The OC will be the first release of the pipeline and will establish the baseline for further developments. The preliminary design of the OC is based on Upena. The flow-chart below presents the current structure of Upena modules and data flow.



On the diagram above the yellow boxes represent the calibration data, blue boxes are science data, brown boxes are byproducts, orange boxes are both input raw data and output final products, and white boxes are the modules, which are defined on table below.

MODULE	SHORT DESCRIPTION
BIAS	bias subtraction
BADPIX	bad pixel masking
CR	cosmic ray identification and rejection
GEOM	geometric calibrations: finding and tracking specral orders, locating the “slit” and characterizing its shape
FLAT	pixel-by-pixel sensitivity flat-fielding
EXTRACT	optimal extraction of intensity spectra
FLATRESP	normalize by flat-field response obtained from solar spectrum (optional)
WLCAL	wavelength calibration using comparison Th-Ar lamp
WAVE	additional wavelength calibration using telluric lines
STITCH	stitching orders together
NORM	normalize the continuum (optional)
STOKES	calculation of polarized spectra (for polarization mode only)
SKYSUB	sky subtraction (for star+sky mode only)
ERROR	calculation of error bars
RESOL	calculation of spectral resolution for each order

2. OPERA Analysis and Post Reduction

The *OPERA Analysis and Post Reduction* modules are optional per installation. These modules are added by adding new Product targets to the Harness and changing the requested target. Adding a new module will require knowledge of the workings of the Harness.

3. Observing modes

ESPaDOnS at CFHT has the following three modes of operation. Star-only, star+sky, and polarimetric. Each mode requires specific procedures for processing the data.

The star+sky mode has the additional step of subtracting the sky from the stellar spectra. Both spectra are obtained simultaneously through two separate fibers, and the image slicer is set with a different configuration so that the two pseudo-slits are aligned leaving a little gap between them, and therefore the two spectra are imprinted at the same time. All the steps that follow the flat-fielding will have to identify and extract the two spectra separately and then subtract one from another at the end.

For the polarimetric mode the spectrograph is also fed by two fibers. However, the light comes from a single source. Each fiber contains the light at a certain polarization. The final products should contain the total flux and also the degree of polarization for a given Stokes parameter. In order to obtain the polarization, one needs to include additional steps in the reduction. The calibration for obtaining the degree of polarization also requires four exposures, so the input data for this mode must be at least four science raw images.

The detailed structure of OPERA Core for each observing mode will be better described in a further document with the pipeline design.

APPENDIX 1 - FITS keywords of Upena-1.1 Products

The following FITS keywords identify the type of reduction and content of the Upena product FITS file.

```
REDUCTIO= 'Intensity' / Type of reduction
FILENAME / Base filename at reduction
```

```
DATE / UTC Date of reduction
UTIME / UTC time of reduction
HSTTIME / Local time in Hawaii of reduction
FILENAMO / Base filename at acquisition
DATE / UTC Date of reduction
UTIME / UTC time of reduction
HSTTIME / Local time in Hawaii of reduction
```

For the "Star Only" and polarimetric data, the i.fits files have:

```
COL1 = 'Wavelength' / Normalized
COL2 = 'Intensity' / Normalized
COL3 = 'ErrorBar' / Normalized
COL4 = 'Wavelength' / UnNormalized
COL5 = 'Intensity' / UnNormalized
COL6 = 'ErrorBar' / UnNormalized
COL7 = 'Wavelength' / Normalized, no autowave correction
COL8 = 'Intensity' / Normalized, no autowave correction
COL9 = 'ErrorBar' / Normalized, no autowave correction
COL10 = 'Wavelength' / UnNormalized, no autowave correction
COL11 = 'Intensity' / UnNormalized, no autowave correction
COL12 = 'ErrorBar' / UnNormalized, no autowave correction
```

For the "Star + Sky" data, the i.fits files have:

```
COL1 = 'Wavelength' / Normalized
COL2 = 'Star ' / Normalized
COL3 = 'Star+sky' / Normalized
COL4 = 'Sky ' / Normalized
```

```
COL5 = 'ErrorBar1' / Normalized
COL6 = 'ErrorBar2' / Normalized
COL7 = 'ErrorBar3' / Normalized
COL8 = 'Wavelength' / UnNormalized
COL9 = 'Star ' / UnNormalized
COL10 = 'Star+sky' / UnNormalized
COL11 = 'Sky ' / UnNormalized
COL12 = 'ErrorBar1' / UnNormalized
COL13 = 'ErrorBar2' / UnNormalized
COL14 = 'ErrorBar3' / UnNormalized
COL15 = 'Wavelength' / Normalized, no autowave correction
COL16 = 'Star ' / Normalized, no autowave correction
COL17 = 'Star+sky' / Normalized, no autowave correction
COL18 = 'Sky ' / Normalized, no autowave correction
COL19 = 'ErrorBar1' / Normalized, no autowave correction
COL20 = 'ErrorBar2' / Normalized, no autowave correction
COL21 = 'ErrorBar3' / Normalized, no autowave correction
COL22 = 'Wavelength' / UnNormalized, no autowave correction
COL23 = 'Star ' / UnNormalized, no autowave correction
COL23 = 'Star ' / UnNormalized, no autowave correction
COL24 = 'Star+sky' / UnNormalized, no autowave correction
COL25 = 'Sky ' / UnNormalized, no autowave correction
COL26 = 'ErrorBar1' / UnNormalized, no autowave correction
COL27 = 'ErrorBar2' / UnNormalized, no autowave correction
COL28 = 'ErrorBar3' / UnNormalized, no autowave correction
```

A FITS keyword identifies the reduction:

```
REDUCTIO= 'Polar ' / Type of reduction
```

In general, reduction is done with 4 exposures, as indicated by those keywords:

```
POLARSEQ= 4 / Number of images in polar sequence
STARTSEQ= 1 / Start image of polar sequence
```

If only 2 exposures were used, the above keywords will have different values.

The p.fits files have:

```
COL1 = 'Wavelength' / Normalized
COL2 = 'Intensity' / Normalized
COL3 = 'Stokes ' / Normalized
COL4 = 'CheckN1 ' / Normalized
COL5 = 'CheckN2 ' / Normalized
COL6 = 'ErrorBar' / Normalized
COL7 = 'Wavelength' / UnNormalized
COL8 = 'Intensity' / UnNormalized
COL9 = 'Stokes ' / UnNormalized
COL10 = 'CheckN1 ' / UnNormalized
COL11 = 'CheckN2 ' / UnNormalized
COL12 = 'ErrorBar' / UnNormalized
COL13 = 'Wavelength' / Normalized, no autowave correction
COL14 = 'Intensity' / Normalized, no autowave correction
COL15 = 'Stokes ' / Normalized, no autowave correction
COL16 = 'CheckN1 ' / Normalized, no autowave correction
COL17 = 'CheckN2 ' / Normalized, no autowave correction
COL18 = 'ErrorBar' / Normalized, no autowave correction
```

```
COL19 = 'Wavelength' / UnNormalized, no autowave correction
COL20 = 'Intensity' / UnNormalized, no autowave correction
COL21 = 'Stokes ' / UnNormalized, no autowave correction
COL22 = 'CheckN1 ' / UnNormalized, no autowave correction
COL23 = 'CheckN2 ' / UnNormalized, no autowave correction
COL24 = 'ErrorBar' / UnNormalized, no autowave correction
```

Please note the following comments:

```
COMMENT For Stokes Q, V, and W, keep the Stokes parameter sign as is
COMMENT For Stokes U, invert the sign of the Stokes parameter
```

FITS keywords from the 4 raw exposures are sometimes repeated 4 times (with new names) to indicate values that change (or should not change) from one exposure to the next. For example:

```
FILENAME= '977289p ' / Base filename at reduction
FILENAM1= '977289o ' / Base filename at acquisition
FILENAM2= '977290o ' / Base filename at acquisition
FILENAM3= '977291o ' / Base filename at acquisition
FILENAM4= '977292o ' / Base filename at acquisition
EXPTIME1= '356.000 ' / Integration time (seconds)
EXPTIME2= '356.000 ' / Integration time (seconds)
EXPTIME3= '356.000 ' / Integration time (seconds)
EXPTIME4= '356.000 ' / Integration time (seconds)
MJD-OBS1= '54544.362' / Modified Julian Date at start of obs.
MJD-OBS2= '54544.367' / Modified Julian Date at start of obs.
MJD-OBS3= '54544.371' / Modified Julian Date at start of obs.
MJD-OBS4= '54544.376' / Modified Julian Date at start of obs.
```

```
MOONALT1= '81.0000 ' / Moon altitude at start in degrees
MOONALT2= '80.9000 ' / Moon altitude at start in degrees
MOONALT3= '80.6000 ' / Moon altitude at start in degrees
MOONALT4= '80.0000 ' / Moon altitude at start in degrees
```