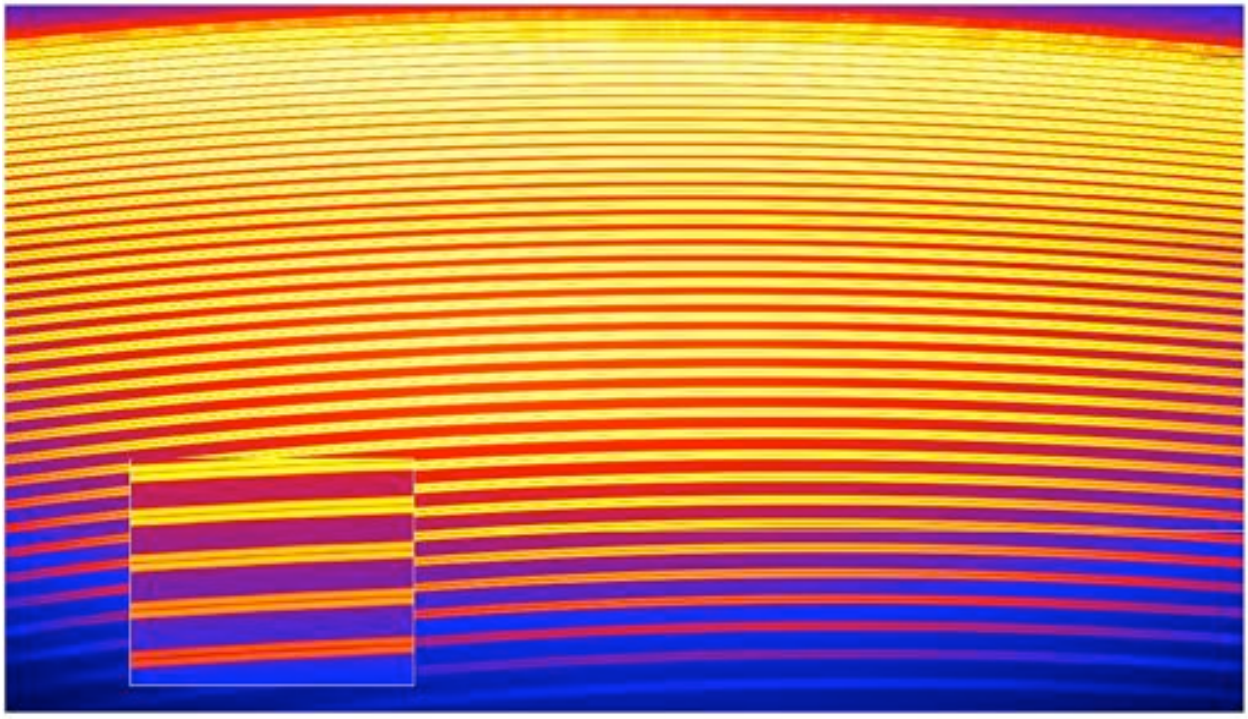
Canada-France-Hawaii Telescope Corporation
65-1238 Mamalahoa Hwy, Kamuela, Hawaii 96743 USA

Société du Télescope Canada-France-Hawaii
Telephone (808) 885-7944 Fax (808) 885 -7288

OPERA PIPELINE TECHNICAL AND OPERATIONAL REQUIREMENTS



DOUG TEEPLE
Canada France Hawaii Telescope
Waimea, HI January 2011

Contents

I. Introduction

OPERA (Open-source Pipeline for Espadons Reduction and Analysis) is an open-source collaborative software reduction pipeline for ESPaDOnS data. ESPaDOnS is a bench-mounted high-resolution echelle spectrograph and spectro-polarimeter which was designed to obtain a complete optical spectrum (from 370 to 1,050 nm) in a single exposure with a mode-dependent resolving power between 68,000 and 81,000. Each exposure contains 40 orders which are curved; the image produced by the slicer (pseudo-slit) has a shape which is tilted with respect to rows.

Many Principal Investigators have requested transparency in the reduction pipeline processes, which is not possible with the existing upena pipeline. In response CFHT has started an open source collaborative project OPERA, which will be developed in collaboration with contributors from the astronomical community and will be freely available as open source software.

This document outlines the operational and technical requirements of the OPERA project.

II. Terminology

It will be beneficial to define the meaning of certain terms that we will use consistently throughout this document.

1. **Detrending** - *Detrending* means removal of instrument signature from the raw data, It consists of the following steps:

- bias subtraction
- bad pixel masking
- cosmic ray rejection
- geometric calibrations: finding and tracking spectral orders, locating the "slit" and characterizing its shape
- flat-fielding

2. **OPERA Core Reduction** - *OPERA Core Reduction* extends *Detrending* steps to produce *reduction products* which will be the products of the CFHT-based pipeline for OLAPA. The core reduction is implemented currently for EEV1 and OLAPA-a with a CFHT Harness that calls Libre-Esprit. This core reduction pipeline is called Upena. The products will be distributed to an archiving center and as such must be FITS format files. *OPERA Core Reduction* adds the following steps to *Detrending*:

- extraction of (intensity) spectra, for all 3 Observing Modes (Star Only, Star+Sky, Polarization)
- use of a flat field response (solar spectrum) (optional)
- wavelength calibration based on comparison exposures
- correcting wavelengths based on telluric lines (optional)
- stitching orders together
- normalize the continuum (optional)

- calculation of polarized spectra for the Polarization mode with continuum polarization subtracted
- calculation of null polarization spectra
- sky subtraction for the Star+Sky mode
- calculation of error bars
- provide spectral resolution and S/N information for each order

2. **OPERA Analysis And Post Reduction** - *OPERA Analysis and Post Reduction* steps are optional and open ended. The pipeline must be flexible enough to add *OPERA Analysis and Post Reduction* steps albeit with an undefined amount of effort.

3. **OPERA Reduction Products** - *OPERA Reduction Products* are the output files of the *Core Reduction* steps. The files are in FITS format and will be distributed to an archiving center.

4. **OPERA Analysis Products** - *OPERA Analysis Products* are the output files of the optional *OPERA Analysis* steps. The files may be in formats other than FITS format and will be not distributed to an archiving center.

5. **Operational Requirement** - An operational requirement is one which concerns the running of the pipeline. Examples would be supported platforms or reduction times constraints required by a user.

6. **Technical Requirement** - A technical requirement is one which concerns the software development of the pipeline. Examples would be required programming languages or software development tools.

7. **mandatory/must [not]** - For the purposes of this document whatever is labelled as mandatory must be done for OPERA. If not specified mandatory is the default.

8. **should [not]** - For the purposes of this document should means we will make every effort within the time constraints to complete this item.
9. **may [not]** - For the purposes of this document we will only do this if there is clearly enough time to complete the *mandatory* and *should* items.
10. **recommended** - For the purposes of this document this means that the author would prefer to see this happen, based on prior experience
11. **suggested** - For the purposes of this document this means that the author would prefer to see this happen, based on prior experience, but this term is weaker than recommended
12. **must/should/may not preclude** - For the purposes of this document this means that this item or feature should be possible and not prevented from occurring. A simple example is parallelism which many users may not need, some some may. A module should not preclude multiple simultaneous calls.
13. **nice to have** - For the purposes of this document this item will only be done after the *must*, *should* and *may* items.
14. **Observing Mode** - Polarimetry, Star only, Star + Sky
15. **Readout Mode** - Readout mode refers to the speed at which pixels are read from the detector. The speeds are: Fast, Normal, Slow. The read mode has an impact on noise on the image.

III. OPERA Open Source Collaboration

OPERA is an open source and collaborative software project. It is a reduction pipeline for ESPaDOnS data. It consists of documentation, infrastructure for collaboration, a software harness, software calibration and reduction modules, software libraries and test data and harness. The software and documentation will be hosted on SourceForge. The project will be managed by CFHT personnel, with contributions from the scientific community. OPERA will check calibration consistency, produce calibration data and reduce and verify observational data.

IV. Operational Requirements

1. Platforms

From the point of view of CFHT the *OPERA Core Reduction* modules OPERA **must** execute on existing CFHT server hardware. As such it **must** execute on 32 bit hardware running a Linux kernel 2,4 or greater and **must** have Network File System support. Project completion time constraints (3) below mandate that the software **should** execute on as few platforms as possible and that those platforms **should** be as similar as possible. However, since OPERA **should** also needs to execute on collaborators hardware, we **may** extend support to MacOS. The pipeline **should not** require the use of particular directory hierarchies.

Discussion

While execution on many platforms would enhance widespread adoption, it also imposes a cost. Implementing for widely differing platforms requires supporting libraries, filepaths, file formats, or integer size to name a few examples. A platform with a directory structure similar to UNIX or Linux is not a burden. Supporting both 32 and 64 bit architectures may be more of an issue as it requires considerably more testing and requires the use of 32 and 64 bit libraries, which may be hard to locate. As such at this point and given the project completion time constraints it may be wiser to only support 32 bit architectures and given time also support 64 bit architectures.

The author believes, based on development of two other reduction pipelines at CFHT, that daily reduction runs at CFHT will require the use of mul-

tiple machines. As such, the base operating system must support a network file system. Most, if not all, modern Operating Systems support this feature.

2. Reduction Speed

OPERA *Core Reduction modules* **should** process a worst case set of 200 calibration images and 250 observational images (one night in the case of CFHT reductions) within 5 hours in order to meet CFHT commitments We currently advertise that Espadons PIs can get reduced data by noon HST the day following a night of observations.

Discussion

If the OPERA *Core Reduction* pipeline is significantly slower than upena, there would be little incentive to use it. OPERA is required to support OLAPA-ab mode, but if that is the only benefit and it is much slower than upena, it would be much less resource intensive to simply add OLAPA-ab mode to upena rather than expend the effort in writing a whole new pipeline. There are two main impacts of the need for speed on technical decisions that must be made. The first is the use of compiled versus interpreted computer language in OPERA and the second is the capability to reduce data in parallel. This discussion will be taken up in further detail in the Technical Requirements section.

3. Project Completion Schedule

All software and testing **must** be complete and operational by the timeframe listed in the scope document - i.e. by 2013.

Discussion

One implication is that OPERA **should** be made available on as few platforms as possible, and **should** have as few dependencies as possible. At the same time, be available on as many platforms as commonly required by the astronomical community.

Another implication is that features **must** be implemented in order of priority.

4. Instruments

OPERA Core Reduction **must** reduce data taken from the ESPaDOnS EEV1 device and the OLAPA device in either one or two amplifier instrument device modes. OPERA **must** handle instrument device modes being mixed in a given night. OPERA **must** associate calibrations with data taken in the same instrument device mode. OPERA **must** reduce spectrographic **and** polarimetric data.

OPERA provides an open source framework that **should not preclude** use by other instruments.

Discussion

All parameterization **should** be localized to header files or definition files. Interfaces to data **should** be done through a *parameter access layer* for reduction parameters and a *data access layer* for image data which **may** resolve to database or other data sources. The harness **should** be configurable so that reduction modules can be added or inserted reasonably eas-

ily. All products and calibrations **must** be qualified by at least: instrument-detector-mode-speed.

5. Observational Data and Calibrations

The *OPERA Core Reduction* unit is a set of related calibration and observational data ordered by time and odometer number. Calibration data **must** be available in order for the pipeline to operate and the calibration and observational data must correlate within a time span and a given mode, readout speed and detector.

Discussion

The simplest method of grouping calibration and observational data is to place the images in a single directory. From a practical point of view that is all that CFHT requires on a daily reduction basis. Given this fact is it worth putting any effort into any other grouping or data access method? A particular PI may in fact want to group data into a single reduction unit that spans many days of observation containing multiple reductions units. Gathering up and putting data into reduction units is the duty of the Harness. In any case, modules should access data through the Data Access Layer.

6. Reliability and Availability

The *OPERA Core Reduction* **should** exhibit availability of 95% for the needs of daily reductions at CFHT, in order for CFHT to meet its commitments to the astronomy community.

7. Autonomous Operation

The *OPERA Core Reduction* **must** operate autonomously, with no human intervention to complete calibrations and reductions.

8. Verification

The *OPERA Core Reduction pipeline* **must** be capable of reducing data with exactly the same results as Libre-Esprit.

Discussion

This may be the only way that we can verify the proper operation of OPERA. OPERA may be called upon to reduce historical data. In that case fewer bias or flat field calibration images may have been taken that a different algorithm requires. If OPERA is not capable of reducing historical data sets then CFHT would have to revert back to upena to service re-reduction requests.

The implication is that the Libre-Esprit methods will have to be exactly duplicate M. Donati's algorithms and methods, or at least the calibration data requirements of the algorithms. This represents a significant time burden on OPERA.

9. Pipeline Scope

For the needs of CFHT, the *OPERA Core Reduction* pipeline **must** have these phases.

1) It **must** create the associations between calibrations and object images
2) It **must** support the creation of master calibrations from calibration images.
3) It **must** support the reduction of observational data to spectra.

4) It **must** also have a post verification phase which will test the completeness and correctness of the products.

5) It **must** produce a report log of the reduction.

The *OPERA Analysis Reduction* phase **may** have:

1) A simulation phase.

2) An open ended Analysis modules such as Radial Velocity calculations, LSD support,and others as yet unknown.

3) Excepting, at CFHT, a distribution phase **must** be created.

10.*OPERA Core Reduction Products*

The *OPERA Core Reduction* pipeline **must** produce all the products available now from the upena pipeline. All final products **must** be in FITS format suitable for archiving by an archiving center. As such OPERA **must** produce at minimum :

1) intensity spectra
2) polarimetry spectra
3) normalized and un-normalized data
4) with and without autowave correction
5) Error bars
6) Check spectra
7) Continuum polynomial removal.

11. License

The prevailing license of a software contribution is the license of the institution providing the software. OPERA will have a blanket disclaimer that ensures that a license of any single module may not infringe on any other module that is part of OPERA.

V. Technical Requirements

1. Programming and System Requirements

At CFHT, the Operational Requirement is that OPERA *Core Reduction* pipeline execution should complete "a typical night of reduction data" before noon of the next morning after observation. A typical night would consist of 200+ object images and associated calibration data.

Modules **must** execute on Linux and **may** compile and execute on MacOSX. All software libraries used **must** be open source libraries and freely available. OPERA modules **should not** require that users purchase licensed software in order to use OPERA. Some small helper functions **may** be written in a self-identifying high level scripting language. Tools used to build and execute the pipeline **must** be freely available on Linux platforms. The source code and documentation **must** be maintained on SourceForge under the name "opera pipeline".

Our assumption at this point is the the implementation language will be C/C++ and that contributions from collaborators will be accepted in other languages (with the associated algorithm) and these algorithms will be rewritten in C/C++ with reference to the algorithm and source code.

OPERA *Analysis and Post Reduction modules* may be written in any convenient language as processing speed is not an issue. These modeule are expected to conform to the overall structure of the Harness so that the Harness available needed for the Core Reductions will not be impacted. However, since adding languages also implies adding the support infrastructure (astronomy libraries, etc) these modules will not be a part of *Core Reduction* testing and will be tested by the individual contributors.

Discussion

From previous experience the author feels that, in order to meet CFHT's *Core Reduction* pipeline speed requirements, all major modules comprising the OPERA *Core Reduction* pipeline **should** be written in a compiled computer language.

Libre Esprit is written in C, a compiled computer language known for efficiency. Upena uses Libre Esprit and some small helper scripts written in bash and uses a Makefile as the harness. The author's experience indicates that interpreted languages such as IDL (Interactive Data Language) and python are about 50 times slower than the equivalent program in C. This experience is based on the author's experience in rewriting certain modules from IDL to C and witnessing the difference in execution speed. However, we must face the fact that many modules from contributors may be already written in IDL. In that case we must weigh accepting the modules and have slower modules versus taking the time to rewrite the IDL modules in a compiled efficient computer language. The author's experience is that the rewriting process is not extremely time consuming, given that helper software libraries are available for use.

As such, the author highly recommends that new core reduction modules be written in a compiled language such as such as C or C++ and that IDL

modules be translated to C. Software libraries may best be written in C so that they may be used by both C and C++ modules.

Execution on Linux with optional execution on MacOS is driven by two factors. First, the platforms are quite similar, both being related to UNIX, and thus the overhead to support both is not high. Second, while CFHT pipeline execution will be on Linux, many researchers may use MacOS. And given that the development and maintenance cost is not excessively high, the pipeline may experience wider acceptance running on these platforms.

While we need to be flexible in terms of accepting scripting languages for small “one-liner” helper functions, we also need to be careful not to draw in the requirement to install special libraries, as it takes time, becomes another dependency and is a maintenance headache. The best scenario is that the only scripting languages used are only those already installed on a standard Linux distribution.

2. OPERA Harness

The Harness is a separate piece of software which controls and directs the execution of reduction modules. The Harness and Core Reduction modules **must not preclude** parallel module execution and **may** support module execution on multiple machines. The harness **should** be command-line driven. It **should** be abort-able, but when restarted, **may** begin reduction from the beginning.

Discussion

It is a fact that a parallel execution harness is more effort and therefor has an impact on schedule. However the author feels, from past experience,

that a linear pipeline will not meet the daily reduction time requirements of CFHT. Many, if not most, external users will not require parallel operation and as such it should not be mandatory.

The reasoning behind the weak **may** condition for restart-ability is that testing if a product exists and then assuming it is complete (i.e. the abort didn't occur part way through generation of the product) is dangerous and requires specialized programming techniques. Since the number of images reduced at a time is relatively small, it was deemed to be simpler to deal only with the basic (start from the beginning) case. If there is sufficient time then recovery based on final products **may** be possible.

That said, the author feels that, based on past experience, it is important to support restart-ability without starting reduction from the beginning for CFHT daily reductions. This feature does have implications for decisions that will be made in the overall architecture. For example, the architect would have to decide whether the Harness or each Module handles maintenance of partially created pipeline products when an abort is received. It has implications on choice of language to use, since, for example, versions of IDL do not have facilities to receive abort traps, so the responsibility **must** fall on the Harness. The choice also have impact on how products, vs byproducts and temporaries are handled.

In general, the software modules should be concerned with science processes and not be concerned with file management or efficiency. The harness is responsible for optimal scheduling and marshaling parameters, including processing parameters, filepaths and filenames, etc, to be sent to the modules.

3. Parameterization

OPERA pipeline execution **should** be parameterized, based on instrument, detector and other characteristics. The parameters **may** be stored in a data table or database, but use of a database is **not mandatory**. Access to parameters **should** be through a *parameter access layer* provided as a software library.

4. Software Modules and Software Libraries

Software modules **must** take all inputs and the names of all outputs as command line arguments. In addition, the harness **should** pass, as standard arguments, a location to store temporary byproducts and modules **must not** arbitrarily store temporaries in a hard-coded directory. Nothing in any module **should** preclude multiple instances of a module executing simultaneously. Common software libraries **should** be constructed and **should** be used by modules where possible. Do due time constraints the software libraries **should** be written in only one computer language. At minimum the following software libraries **should** be created for everyone's benefit, with more to be defined as required:

- data access library - the data access library is intended to provide a common interface for accessing calibration and image files.
- parameter access library - the parameter access library is intended to provide a common interface for retrieving configuration parameters,
- image access library - the image access library is intended to provide helper function that do such things as access pixels by row and column, multiply, divide, add, subtract a whole image by a value, find the mean and median, variance values of an image, median two or n im-

ages, and the like as to be defined in a more detailed design document.

5. Configuration and Parameterization Access Module

An interface module will be written to both query the configuration/ parameterization access data and update the configuration/ parameterization data. It is a simple module that allows command line access to these libraries.

6. Sample Module

A single sample module **should** be created by CFHT staff and available to all contributors. The sample **should** be written in only one computer language due to project time constraints.

7. Development Model

OPERA software **should** be based on the “sandbox” model. Each contributor **should** have their own full installation and multiple users at one site **must** be able use their own instance of the pipeline without interfering with the master and without interfering with one another. Support tools **should** be available and developed for the project as necessary for integrating independently developed modules. These tools are intended to ease integration issues that arise in collaborative development.

8. Realtime Observatory Support

CFHT has a requirement that OPERA *Core Reduction* **should** support file-by-file single image at a time reduction for use by observatories to calculate SNR values and spectra for feedback during observation.

9. Documentation

OPERA documentation **should** be in a format readable on multiple platforms. Either plain text or PDF is recommended.

10. Testing and Bug Fixing

OPERA software **should** be tested after integrations by an independent "test" user. A fixed test data set **should** be retained at CFHT and **should** be available to contributors. The results **should** be compared with Upena and previous releases of OPERA. CFHT will provide hardware servers for testing and CFHT staff will run the test cases as part of the integration process. Bug tracking **should** be online using "tracker". Individual contributors members **must** have module ownership and responsibility to fix bugs and maintain and integrate their module in a timely fashion.

There **should** also be developed tests and verifications of the quality of the data produced.

11. Installation and Releases

OPERA software **should** be under source code control using CVS with Internet access using pserver on Sourceforge and a CFHT-hosted computer (opera.cfht.hawaii.edu). CFHT personnel **should** set up Sourceforge and the CFHT opera computer. Builds **should** be tagged as releases. An installer **should** be constructed to ease the addition of new team members.

12. OPERA Core Reduction Steps

The instrumental signature removal includes bias subtraction, bad-pixel masking, flat-fielding, order tracking and fitting, pseudo-slit fitting, and wavelength calibration. At minimum OPERA **must** remove the instrumental signature (what is called "detrending" at CFHT) and further reduction steps so as to provide similar products provided by the current Upena/Libre-ESpRIT pipeline:

- bias subtraction
- bad pixel masking
- cosmic ray rejection
- geometric calibrations: finding and tracking spectral orders, locating the "slit" and characterizing its shape
- flat-fielding
- extraction of (intensity) spectra, for all 3 Observing Modes (Star Only, Star+Sky, Polarization)
- use of a flat field response (solar spectrum) (optional)
- wavelength calibration based on comparison exposures
- correcting wavelengths based on telluric lines (optional)
- stitching orders together
- normalize the continuum (optional)
- calculation of polarized spectra for the Polarization mode with continuum polarization subtracted
- calculation of null polarization spectra
- sky subtraction for the Star+Sky mode
- calculation of error bars
- provide spectral resolution and S/N information for each order

13. External Dependencies

External dependencies **should** be kept to a minimum. The OPERA *Core Reduction* modules **should not** require software that is not open source and must be purchased. The *OPERA Core Reduction* modules **should no**t link to proprietary software libraries or data.

Discussion

The Open Source model does not actually require that the modules be written in a computer language that is interpreted by a commercial interpreter for which any user of the OPERA pipeline is required to purchase a license. A case in point is IDL, whose source may be released as open source, but that requires the user to purchase an IDL interpreter. Many pre-existing modules may in fact be written in IDL.

External dependencies such as software libraries bring with them maintenance and versioning issues, which, for the sake of time, should be avoided where possible.

14. OPERA *Analysis and Post Reduction* Steps

OPERA Analysis Reduction steps are open ended and **may** be created, integrated and tested. These steps will not run as part of standard reduction at CFHT and will not create products that will be archived by an archiving center. As such, the OPERA Analysis and Post Reduction steps, excepting those developed by CFHT, will not be tested by CFHT contributors.

CFHT will create a post reduction step for internal use which distributes OPERA pipeline products to PI's. This module is not part of the OPERA

Core Reduction pipeline. For the purposes of CFHT, and since web page creation is outside the scope of OPERA, the step of metadata creation will not be part of OPERA.