

The Future of Data Reduction at UKIRT

F. Economou

*Joint Astronomy Centre, 660 N. A'ohōkū Place, University Park, Hilo,
HI 96720, USA*

A. Bridger and G. S. Wright

*Royal Observatory Edinburgh, Blackford Hill, Edinburgh EH9 3HJ,
United Kingdom*

N. P. Rees and T. Jenness

*Joint Astronomy Centre, 660 N. A'ohōkū Place, University Park, Hilo,
HI 96720, USA*

Abstract. The Observatory Reduction and Acquisition Control (ORAC) project is a comprehensive re-implementation of all existing instrument user interfaces and data handling software involved at the United Kingdom Infrared Telescope (UKIRT). This paper addresses the design of the data reduction part of the system. Our main aim is to provide data reduction facilities for the new generation of UKIRT instruments of a similar standard to our current software packages, which have enjoyed success because of their science-driven approach. Additionally we wish to use modern software techniques in order to produce a system that is portable, flexible and extensible so as to have modest maintenance requirements, both in the medium and the longer term.

1. Background

UKIRT¹ has been using automated data reduction for one of its main instruments, CGS4, for some years. The benefits of data reduction in near-real time are many, including more efficient use of telescope time and a higher publication rate for data. However, the program CGS4DR (Daly 1995 & 1997) that was used for this purpose, despite its successes proved to also have its drawbacks. As part of the ORAC project² (Bridger et al. 1998) and the preparation for the arrival of two new instruments, the future data reduction at UKIRT is being reassessed in the light of our experiences with CGS4DR. In particular, while we wish to continue to provide near publication quality data at the telescope, we also want to:

¹<http://www.jach.hawaii.edu/UKIRT/>

²<http://www.roe.ac.uk/abwww/orac/>

- Ensure that operational staff are not left with a large body of code to support
- Be able to make changes easily to data reduction sequences to reflect a change in the performance or our understanding of an instrument
- Have a system that can be speedily ported to other platforms
- Be well placed to take advantage of new astronomical software and techniques

The above requirements are not, of course, unique to UKIRT; and it seems that the astronomical software community in recent years has shifted from large instrument-specific programs to data reduction pipelines.

2. The ORAC Data Reduction Pipeline

Our proposed data reduction pipeline consists of five major parts:

- *An algorithm engine*: One or more generally available packages containing suitable algorithms for data reduction
- *A pipeline manager*: A program written in a major scripting language that initiates and oversees control of the pipeline
- *A messaging system*: An interface between the pipeline manager and the algorithm engine
- *A recipe bank*: A catalogue of data reduction recipes that are associated with the data (by the data file header) and the observation sequence used to obtain it (by the observation preparation system)
- *A data reduction dictionary*: A translation of in-house data reduction recipes to actual scripts in the language of the algorithm engine

Of these, it is our intention that only the pipeline manager, the recipe bank and the dictionary would be supported by the local staff, whereas the other two components would already be supported by another organization.

The aim is that in the future any one of these components could be upgraded or changed without affecting all other parts of the system; for example we could chose a different algorithm engine without needing to change the code of the pipeline manager.

2.1. The Algorithm Engine

At least at first, our algorithm engines will be some major packages maintained by the Starlink³ organization (KAPPA, FIGARO, CCDPACK etc), who support astronomical computing in the UK. These come in the form of monoliths that

³<http://www-star.rl.ac.uk/>

can be loaded in memory once and then triggered to execute commands sent to them via the ADAM messaging system (see §2.3.) without the start-up overheads imposed by their (more usual) Unix shell invocation.

Starlink packages use the hierarchical extensible N-dimensional Data Format (NDF) as their native data format which is already in use at UKIRT. Its main attraction to this project is its HISTORY component, which contains a list of operations that were performed on the dataset and its output. Similar components can be used to record processing instructions, so that data carries its own data reduction recipe with it.

Moreover, the NDF data format has quality and error as well as data arrays which are correctly propagated by the majority of Starlink packages.

2.2. The Pipeline Manager

The pipeline manager's tasks are:

- To detect the arrival of new data
- To understand the nature of the data (object, calibration frame etc) and whether it fits in the expected data reduction sequence
- To parse the recipe(s) appropriate for the data
- To send the necessary messages to the algorithm engine to process the data

The pipeline manager is also expected to be robust, have good error recovery, propagate meaningful error messages to the observer and under no circumstances interfere with the data acquisition.

2.3. The Messaging System

Using a messaging system rather than, for example, generating scripts to be executed by the Unix shell, has the following advantages:

- Fast execution because the data reduction tasks are memory resident
- Keep track and parse the status returned by the individual data reduction tasks
- Control the flow of informational and error messages to the user

The choice of the messaging system is dependent on the algorithm engine (for example to use IRAF as the algorithm engine one would chose the IRAF message bus); however we hope to introduce a messaging layer that would enable alternative messaging systems (and their algorithm engines) to be used with relative ease.

The ADAM messaging system used by Starlink packages has been long used at UKIRT as part of the instrument control system and has proved very reliable. We have interfaces to it from both major scripting languages (perl and tcl).

2.4. The Recipe Bank

It is envisaged that each standard observation sequence will be associated by one or more data reduction recipes. These contain high level commands (eg SUBTRACT_BIAS, DIVIDE_BY_FLAT) that represent conceptual steps in the reduction of the data. The instrument scientist would be expected to specify what these steps should do in the particular context of an observation, and the software engineer would then make appropriate entries in a data reduction dictionary.

2.5. The Data Reduction Dictionary

As an important part of divorcing the specifics of the data reduction implementation (algorithm engines and messaging systems) from the pipeline manager as well as the end user, we are introducing a data reduction vocabulary that maps into specific commands. For example the data reduction command DIVIDE_BY_FLAT could map into one or more actual commands (eg the KAPPA command `div` or the IRAF command `imarith /`).

This method has the further advantage of reducing user documentation load – even if the implementation of DIVIDE_BY_FLAT were to change, the user documentation, that does not delve to this lower level, remains unaffected.

Furthermore, the actual scripts can be shipped with the data for the observer's future use.

3. Delivery

UKIRT is expecting the arrival of two new instruments – the UKIRT Fast Track Imager (UFTI) in early 1998 and the Mid-infrared Echelle (MICHELLE) at the end of the same year. The ORAC project is required to deliver a functional system for UFTI and the full system by the arrival of Michelle.

References

- Bridger, A., Economou, F., & Wright, G. S., 1998, this volume
- Daly, P. N., 1995, in ASP Conf. Ser., Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne & J. J. E. Hayes (San Francisco: ASP), 375
- Daly, P. N., 1997, in ASP Conf. Ser., Vol. 125, Astronomical Data Analysis Software and Systems VI, ed. Gareth Hunt & H. E. Payne (San Francisco: ASP), 136