Towards a new Data Analysis System for the age of the Virtual Observatory

A White Paper describing the ESO /Finnish Collaboration

Richard Hook, European Southern Observatory, October 2004

DRAFT

Introduction and Scope

The data analysis systems in use by astronomers in Europe and elsewhere were originally developed in the 1980s or even earlier. Although they have evolved well to cope with changing hardware, new operating systems and huge increases in data volume, the fundamental underlying infrastructure and many of the basic applications are unchanged. As a result the technology is old and the major packages have only minimal support and are not an adequate framework for the development of future major applications. However, these systems still enshrine many, if not most, of the basic applications currently in use. The user community is largely caught between being unsatisfied with the current software (although largely dependent on it) and cynical about any grand new scheme that is proposed. A pragmatic culture of foraging for what is needed and building from bits and pieces has evolved and there has been a move to seeing software as small pieces rather than grand, interlocking systems.

The areas of deficiency of the major current systems are well known. They are reviewed in the second part of this document. Some new "analysis systems" have been developed in the last few years (Eclipse, AIPS++, CIAO) but more emphasis has been put into pipelines to deliver higher quality science products by many observatories, certainly ESO and also STScI. ESO made a clear decision in the mid-1990s to follow this route. As a result there is a perceived lack of some analysis software for ESO instruments, although this is probably seen as a lack of specific applications rather than a lack of underlying infrastructure.

In addition, the rapidly evolving Virtual Observatory efforts are defining standards and new data access methods while making greater demands on data products. Any new system will need to facilitate convenient access to the VO, both as a data producer and consumer. The requirements of this new world of data are the primary drivers for future developments.

Background to this Project

In Summer 2004 Finland joined ESO. As part of its joining fee there was a contribution in kind of software effort, amounting to about 18 person-years over three years starting in early 2005. It was decided to devote this resource to addressing the question of data analysis software of the future, specifically for ESO, but also with wider applications. Although there is astronomical involvement in Finland with this project the skills are mostly in the areas of computer science and mathematics. ESO will provide a project manager and project scientist to manage this project. The scientific input to the project will come from a Finnish Astronomical Advisory Group, the ESO Faculty and the ESO community. It will be funneled through the project scientist.

In a parallel, and earlier effort, an Opticon Network (3.6), with significant ESO involvement, is also looking at the question of future analysis systems and will also supply input to the project.

Following meetings between ESO and ESA senior management in mid-August 2004 it was felt that much of the project management expertise might come from the ST-ECF, which has much relevant background in this field.

Overall Project Goals

The project will run for three years (2005-2007). The primary goals are:

- To have developed a clear idea of the science requirements of a data analysis system to satisfy the needs of the ESO community for the decade 2010-20. These will come from extensive consultation with ESO users as well as the views of the international community and appropriate experts.

- To have assessed the technology required and tested it on realistic astronomical data sets.

- To have executed several pilot studies that illustrate critical steps along the road to a new system, validate the concepts and also to produce significant tools of immediate value for the ESO community.

Proposed Organisation Structure

It is proposed that the project is divided into three organisational components:

- The scientific oversight for the project is vitally important. It will come from the Finnish Astronomical Advisory Group set up for this purpose, the ESO Faculty, the Opticon Network devoted to this task and other experts in the community. The gathering and collating of scientific requirements of the project are the responsibility of the Project Scientist.

- A Project Manager and Project Scientist will manage the day-to-day work of the project and report on progress at regular meetings with the Head of the ESO DMD. The Project Scientist will work closely with the Project Manager to convert the views of the community into clear requirements and the Project Manager will oversee the implementation.

- The team from Finland will agree goals and targets with the Project Manager and Scientist and work closely with them on a day to day basis. This group will have a coordinator who will be the primary point of contact with the ESO-based team. All actual software development will be done by the Finnish Team, but they are also expected to bring along considerable expertise in the area of computer science.

Modus Operandi

The work will progress as a series of projects, each typically taking 6 months to a year, in which specific questions or problems are addressed. Initially, as the Finnish team is not familiar with astronomical software, a short initial project should be selected that will help them to learn about astronomical software.

The deliverables from these projects will be software tools that are pilot implementations addressing specific problems posed by scientific oversight bodies listed above, complete with detailed documentation, or reports describing studies. As well as reporting to the ESO DMD these projects will be written up and presented at the ADASS or other conferences and, where appropriate, presented as demonstrations. The model of annual cycles with demonstrations and meetings at the same times over a period of several years, as adopted by the Virtual Observatory, should be considered.

Where possible projects should be chosen so that the resulting software tools are of direct value to the ESO community and not merely "proof of concept". At the end of the three years a more detailed report will be presented to the ESO Head of DMD setting out conclusions from the projects and making detailed recommendations for the form that further work should take.

Section II – Data Analysis Software – some notes

Current Views and Perceived Needs

In the early days of major data analysis systems in astronomy the system itself provided all the components needed by a programmer – the data access was specific to the system, as was the scripting language, the parameter interface and the graphics package. In the last decade this monopoly position has eroded. Now the almost ubiquitous choice for data structures is FITS (often using the CFITSIO library) and general purpose scripting languages are generally preferred (Python being a current favourite) to the less powerful and system-specific originals. There is a wide choice of graphics packages and parameter interfaces can be handled using simple control files or UNIX shell facilities. As a result many new and very useful tools are not written "within" a specific astronomical software system.

MIDAS and IRAF have not been formally adopted as the system of choice for any major new project in the recent past, with the exception of Gemini and NOAO where there are well established links with the original teams. Despite these reservations, if there is a de facto standard system at present, in optical astronomy, used by a majority for standard processing, it must be IRAF.

So, are data analysis environments now redundant? No. There are many reasons:

Firstly the scenario of stringing together bits and pieces within a Python (or other) wrapper may work but it is very inelegant, clumsy and does not allow access to the full power of the scripting language. This approach often results in a majority of the code being devoted to getting around minor hassles: reformatting tables, modifying headers etc. There is no flow but instead a stop-start sequence of writing and reading intermediate products. As a result the scripts are neither quick to develop nor quick to run.

Secondly this approach completely ignores the new ways of accessing and processing data which are becoming available as the virtual observatory becomes a working reality. Soon data will no longer be regarded as files sat on a local disk but instead much more distributed resources on the net. In addition the ability to run tasks in a distributed and parallel way, possibly in a grid environment but also, more modestly, on a local mini-grid of multiple machines at one institute, will become fundamental as it is not handled by current systems.

So, do we need a "new IRAF" or a "new MIDAS"? No, but we do need a new "software environment" in the broader sense of a variety of software elements (libraries, tools, standards, applications etc) which work together, or separately, to allow efficient data handling in the VO world and we do need some way of ensuring access to legacy algorithms and experience.

How do Current Software Systems fall short?

For analysis of optical and IR data sets the most popular software systems are IRAF, MIDAS and IDL. IDL is a special case as it is a relatively expensive commercial product that is not specific to astronomy. It is robust and well-supported and has kept up to date with technology. Many people find it a very convenient way to interact with simple data objects, such as arrays, write effective scripts and visualize data. There is a large body of procedures for astronomical use in many collections, notably the "Astron" one at Goddard. However, it is less suitable for large-scale projects and its memory (rather than disk-file) based design can be limiting. We will not discuss it further here.

IRAF and MIDAS were developed originally in the 1980s and share many features:

- A command language has been written that is specific to the environment. The user either enters commands interactively or runs scripts written in this custom language.

- Applications are separate programs that run in a separate process, often as monoliths that are persistent to increase efficiency.

- Applications read and write data in a variety of formats and are controlled by parameters.

- Most user applications are expected to be scripts, although there are APIs to allow compiled applications, typically in C or Fortran.

- Applications are grouped in packages with related users – eg, tasks for handling data from a specific instrument.

- The environment, including the applications and scripts, are portable between operating systems although such ports need specialist knowledge and are difficult for the system itself.

Deficiencies include:

- The command languages are primitive and don't compare to the powers of modern scripting languages. There is no compatibility between many aspects of the systems – a MIDAS script would need to be completely re-written to run within IRAF.

- Because tasks typically read data from disk and write it back afterwards there is a very heavy i/o overhead for many scripts.

- There are no Windows ports, this may or may not be considered a deficiency.

- These systems are cumbersome and difficult to learn.

- Writing either a script or a compiled application involves significant learning which is of little value except within a particular system.

- Many tasks can now be carried much more conveniently within common commercial software. A good example is the ease of manipulating modest sized tables within Excel.

- The underlying technology is old which makes support and enhancements more and more difficult as time goes on. Much of IRAF is based on SPP, a language invented for, and only used by, IRAF. The number of people who are familiar with the internals of these systems, or interested in acquiring these skills, is very small.

- For people familiar with modern quality applications (Photoshop, Excel etc) IRAF and MIDAS feel cumbersome and lacking in any intuitive character or elegance.

- The future will mean that data analysis is more closely bound to evolving distributed systems such as the virtual observatory and Grid concepts.

- Data is getting more copious and complex. Although current systems do not have many hard-wired limitations they can become very inefficient when dealing with massive data sets.

Positive aspects:

- Both MIDAS and IRAF contain a huge quantity of useful applications, which embody wisdom and experience regarding the processing needs of many instruments. No one is seriously considering re-writing all this.

- Many people are familiar with these systems, they use them all the time and can use them efficiently. They trust applications they know and are suspicious of new systems and wary of having to learn something new.

- Although many aspects of these systems are specific and inhibit interoperability (different scripting languages, different APIs, different package structure, different parameter mechanisms) the use of FITS makes sharing data quite effective. Many people accept that the easiest way to get things done is to use a bit of this and a bit of that.

- Applications within these systems have been guaranteed long lives – the porting of the infrastructure by the support organizations to new hardware and operating systems has freed the authors from worrying about these problems and allowed then to concentrate on fundamentals.

The Legacy Challenge

Millions of lines of code have been written as applications within these systems. Although much of this is redundant there is still a huge body of knowledge enshrined within. How can it be re-used? Although the algorithms may be highly tuned the wrappings may be inconvenient and the use obscure. In addition it is very likely that such software does not either read or write such a comprehensive set of metadata as is required by VO standards.

To move forward there are two extreme positions – providing facilities for re-using legacy software as it is, or re-writing vital applications from scratch. The former approach, although convenient and much more modest in its demands, tends to entrench the faults and deficiencies of the original code as well as its virtues. The latter requires much more work and can often result in new applications that are either inferior, despite the best efforts of the authors, and less trusted. A new version of anything in the software world has to fight for acceptance.

As the standard systems have relatively simple APIs a further option is to retain the original source code but rebuild the applications to allow use of more capable libraries and interfaces. An example of this is the way in which the IRAF FITS kernel, when introduced, allowed all IRAF applications to access FITS files as well as older, IRAF-specific formats, simply by relinking. This sort of "future-proofing" is very desirable.

A good example of an approach closer to the first position is that of Pyraf which has been developed in the last two years as a partial replacement for IRAF at STScI. In this case it was felt very important to retain access to IRAF applications exactly as they are, but also important to open a route to the future by replacing the IRAF CL with Python and developing ways in which new applications can be pure Python tasks whilst retaining the access to IRAF. This has proved very successful and required modest efforts, but has not been widely adopted beyond the HST world.

At some stage it is necessary to "retire" much of the application base and carry forward what are seen to be the core tasks. Finding the best way to do this is a significant part of the proposed project.

The Virtual Observatory and GRID

During the lifetime of this project the many Virtual Observatory initiatives around the world will have matured into a working system. It will be possible to search multiple, distributed archives in a flexible and uniform way and there will be powerful tools for browsing and data-mining. A clear indication of the sorts of facilities that will be available is given by the recent AVO demonstrations. Much of the work so far has been in the development of standards and protocols for interoperability between archives, software tools and users. A major challenge is the establishment of useful standards for the "metadata" which define the scientific context and meaning of data.

The VO is at present primarily a "data grid". Much work is in progress, within astronomy and outside, towards the development of a computational grid that will allow major processing to performed where the resources are located rather than on the desktop or with local dedicated processing power.

The data analysis system interacts with VO in many places. Firstly software is used to prepare data for inclusion in the VO. This software ideally needs to automate the inclusion of metadata as much as possible so that the results are largely "VO compliant" without extensive and demanding header editing. To a large extent pipelines are built from, rather than distinct from, offline data processing tools.

Secondly, and most obviously, analysis tools need to be applied to the data within the VO. Some may be applied directly on the remote data and some may be applied on local copies of data downloaded from remote sites to the local machine. In both cases the metadata and its interpretation are important. An example is the measurement of an object's magnitude which has been found on a image in the VO – either the software to do this is run within the tools used to view the image, or perhaps remotely on a server or, more conventionally, the data are downloaded and a tool applied locally. In all cases metadata such as zeropoints, units etc, are vital. Within software it may be important to allow direct access to remote data as an alternative to access to local data and hence blur the distinction between the two.

Thirdly some tools are suitable to be made available as applications for GRID use. For example a mosaic-generating piece of software may be run in parallel on many remote CPUs to build large mosaics from multiple images that may be located elsewhere within the VO. Similar tools, without the burden of security worries, may be used on local clusters of machines.

Finally some tools may be made available as web services. This is related to the concept of on-the-fly recalibration – software is run on a remote machine using data that may, or may not, be also remotely located.

In all these cases there is a clear need for software which may be used as components, without excessively burdensome infrastructure.

Essential Characteristics of a New System

One goal of this project is to define the characteristics of a new system. Some of these will only become apparent at a later stage but we can safely suggest some of them already as they follow directly from the perceived successes of current software. This is a partial list:

- The core system must have an underlying design that is conservative, robust, portable and have long life. This is a higher priority than being "state of the art".
- The software must be open source and follow normal industry standards.
- Installation must be robust, fast and involve minimal user interaction.
- Where possible standard libraries should be used and there should be a minimal amount of re-invention of wheels.
- It should be well integrated with one (or more) industry standard scripting languages.
- It should be easy to port the system to another operating system, certainly all flavours of UNIX and possibly Windows also.

- There should be a simple API to allow the development of applications in C, Fortran and probably Java.

- There should be minimal coupling between elements of the system so that separate parts may be used independently where appropriate.

- It should be possible to re-use legacy applications in some way.

- The design should consider from the outset the possibility of the components being used in a parallel, grid-style manner or possibly as web services.

Conclusions

It is clear that a new software infrastructure will be needed to replace the ageing systems in current use if the ESO community is to be ready for the age of the Virtual Observatory and the new generation of instrumentation. The project described will allow a detailed characterization of the requirements and produce many useful components from pilot projects along the way. Some aspects of such a system are already apparent and we expect that a much clearer and complete view will emerge over the three years of the project.