Astronomical Data Analysis Software and Systems IX
ASP Conference Series, Vol. 216, 2000
N. Manset, C. Veillet, and D. Crabtree, eds.

ESA Survey Missions and Global Processing

W. O'Mullane, A. Hazell, K. Bennett

Astrophysics Division, Space Science Department of ESA, ESTEC, 2200 AG Noordwijk, The Netherlands.

M. Bartelmann

MPI für Astrophysik, P.O. Box 1523, D-85740 Garching, Germany.

C. Vuerli

Osservatorio Astronomico di Trieste, Via G.B. Trepolo 11, Trieste 34131, Italy

Abstract. Two European survey missions are featured in the current ESA science program: Planck, a Cosmic Background Radiation mission, and GAIA, an astrometric mission. Both missions require global iterative processing over the spacecraft data in the spatial and time domains. The large data volumes and complex data structures involved demand novel analysis methods.

1. The Missions

Both Planck and GAIA are targeted for the earth-sun Lagrange point L2. Both satellites intend to adopt a continuous scanning strategy about L2 to give full sky coverage. Both missions aim to provide complete surveys to the limit of the instruments' sensitivities.

1.1. GAIA

GAIA is proposed for ESA's fifth cornerstone mission that has a prospective launch date of 2009. The objectives of GAIA are many-fold but the core objective is the discovery of the origin and formation of the Galaxy. To do this GAIA will combine information from astrometry, photometry, and radial velocity instruments using the proven principles of the Hipparcos mission. The astrometry will be complete to V=20 magnitude with accuracies of 4 micro-arcseconds at V=10, 10 micro-arcseconds at V=15 and 0.2 milli-arcseconds V=20. The radial velocity measurements will have accuracies of 1 km/s at V$\sim$10 and 10 km/s at V$\sim$17. There will be 4 broadband photometric filters and 11 medium spectral filters in the GAIA photometric system. GAIA is estimated to observe around 1.3 billion objects hundreds of times over its 5 year lifetime producing around 10TB of raw data. The current estimate of processing needed for GAIA is 10^{19} flops. For further information on GAIA see `http://astro.estec.esa.nl/GAIA`.

1.2. Planck

Planck is an accepted medium sized mission and is due for launch in 2007. Planck will provide a major source of information relevant to several cosmological and astrophysical issues, such as testing theories of the early universe and the origin of cosmic structure. The angular resolution of Planck will be 10 arcmin. Two instruments will be on board to give frequency coverage of 30-850 GHz. The temperature sensitivity of Planck will be $\frac{\Delta T}{T} \sim 2 \times 10^{-6}$ in the channels where the CMB is the dominant signal, and as close to this value as technically possible in all other channels. Planck will produce nine complete maps of the sky in different frequency ranges. The raw data produced by the instrument will amount to around 0.5 TB. For further information on Planck see `http://astro.estec.esa.nl/Planck`.

2. Split of Data Processing and Storage

All software design approaches and standards require a logical as opposed to a physical model of the system. This is incorporated in ESA's software engineering standards PSS-05-0 (ESA 1991) as part of the Software Requirements phase. As systems grow in complexity this logical approach becomes ever more important in order to insulate one from underlying libraries and databases which may be used to implement the system. It is however usually left up to the reader how best to bring the logical model into physical reality and frequently, at the physical design stage, radical changes are made to the logical model. A strict interface approach can allow for many different implementations while presenting a consistent interface to clients. The goal of this approach is similar to that of *The Bridge* framework as described in (Gamma et al. 1994). The Unified Modeling Language (UML) (Eriksson & Penker 1998) introduces the notion of an Interface. This is like a class but it defines only a set of abstract operation signatures. Other classes may opt to implement this interface, which means they must provide the concrete implementation of the operation. In Java, *Interface* exists as a programming construct while in C++ this could be seen as an abstract class with only virtual operations defined. Operations of other classes can be done purely in terms of these interfaces and thus remain completely independent of the actual implementation of the operations. A brief example is given in Figure 1. This example shows two implementations of the interface, although many more could be available. UML notation is used here: solid lines with triangles mean Inheritance while the dashed line means there is an *Implement* relationship. It should be noted that the interface definitions refer to other interfaces: e.g. getRGC() of Abscissa returns an RGC rather than some particular implementation of it; likewise getAbcissae() of the RGC interface returns an array of Abscissa. All software written in the system should then deal with the interfaces only.

2.1. Distribution

Although storage and processing of the data would be separated by a software layer both may be distributed. When the data is distributed spatially then certain algorithms will benefit by running close to the data they are working on

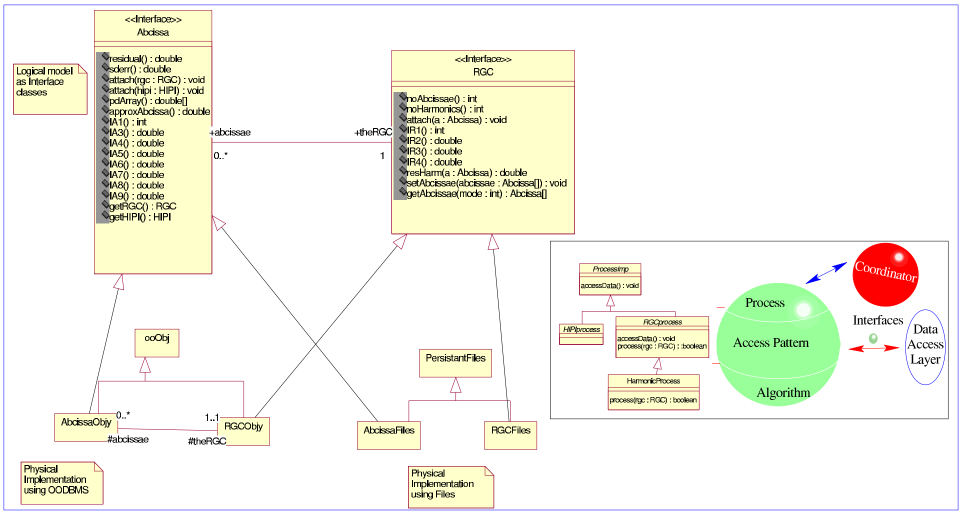
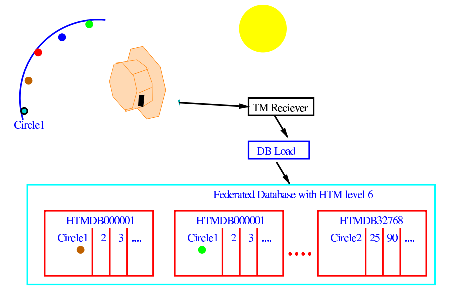


Figure 1. Interfaces for Data. OO design for Data Driven Algorithms (inset).

- this is equally true for Planck or GAIA. So although the processing software should not be aware of the storage form of the physical data it should be aware of the topology of storage since the processes will run on the same machines holding the data.

3. Processing

Processing for survey missions requires data to be accessed readily in both the time and spatial domains. Typically a set of global iterative processes will run over the different types of raw data to produce calibrated values using calibration information which will include values from other processes. An successful example (O'Mullane & Lindegren 1999) of this type of processing was implemented using Hipparcos Intermediate data, Java and the Objectivity OODBMS. The algorithms produce a mission chromaticity matrix, reference great circle harmonics, and astrometry updates. Each process has an effect on the other two and therefore has a global effect on the result similar to the type of dependencies which will occur in the GAIA data. Each process can easily be run in a parallel fashion on multiple machines under the supervision of a coordinator. To achieve insulation of the algorithm from the storage of the data, a data driven approach for the processing is adopted. Processes accept data and process them in the order they are given, this will allow for a process on a machine to be given the data on that machine first, furthermore it can be given data in the order it is on disk which will lead to more efficient processing. The access pattern for a set of algorithms, then, can be encapsulated in a class from which specific algorithms inherit (see Figure 1 inset). Another advantage of this approach is that algorithm writers are not burdened with needing intimate knowledge of the storage and distribution of data. Another layer of coordination is required on top of



Test	Objy C++	O2 C++	Objy Java	O2 Java
Creation of ~400K objects	52	242	341	317
Creation of ~4 million	571	Crashed	3,447	3,409
Search the big DB	194	-	380	91
Search Small DB SubObj	19	59	13	109
Create 4 BigObj (~96Mb)	194	8160	380	9,300
Create 18 BigObj	854	-	2135	-
Load BigObj (~96Mb)	38	< 1	60	8

TestObj contains 200 SubObjs, 5 doubles an int and a String attribute. A SubObj has 5 doubles and one int attribute. BigObj contains an array of doubles. All times in seconds.

Figure 2. Spatial Time Data Partitioning (Left). DB performance (Right).

the coordinator - a Planck pipeline prototype in Java was developed at MPA Garching and allows scripted sequential running of arbitrary processes.

4. Storage

Driven by the processing requirements, access to a large amount of data in both spatial and temporal domains needs to be provided. Furthermore the data is constantly being updated by the processing. A multi-dimensional index system should suit these purposes and indeed two such systems already exist: the Hierarchical Triangular Mesh (Kunszt et al. 2000) and the Hierarchical Equal Area isoLatitude PIXelisation(`http://www.tac.dk:80/~healpix`). Both schemes allow spatial splitting of the sky. A second split may then be performed in the temporal domain. Figure 2 shows how observation(dots) data from given time frames(circle) might stripe across separate spatial databases. Some performance tests on Objectivity and O2 in both Java and C++ (see Figure 2 Right) have been run. Although much slower for small objects Java speed is improving and development time is fast.

5. Conclusion

Processing scan survey data correctly is a non-trivial task and requires careful thought to be successful. Technologies and schemes exist to help however they require integration and adoption of them can change the way a system is built entirely.

References

Eriksson, H. & Penker, M. 1998, UML Toolkit, Wiley

ESA Software Engineering Standards PSS-05-0; Issue 2

Gamma, E. et al. 1994, Design Patterns, Addison Wesley

Kunszt, P., et al. 2000, this volume, 141

O'Mullane, W. & Lindegren, L. 1999, An OO Framework for GAIA Data Processing. Baltic Astronomy, v.8, 57