GAIA Spectroscopy, Science and Technology
ASP Conference Series, Vol. 298, 2003
U. Munari ed.

The GAIA data access and analysis study

Salim G. Ansari

Science Programme Coordination Office, European Space Agency, ESTEC, The Netherlands

Jordi Torra, Xavier Luri, Francesca Figueras, Carme Jordi

Departament d'Astronomia i Meteorologia, Universitat de Barcelona, Spain

Eduard Masana

Institut d'Estudis Espacials de Catalunya, Spain

Abstract. The GAIA Database Access and Analysis Study was initiated by ESA in July 2000 to investigate the feasibility of implementing a Data Reduction System for the mission. In its first phase, the study was limited to a few well-defined algorithms based around the astrometric problematics of data processing. Furthermore, the Study seamlessly integrated the GAIA Simulator, which has in the meantime grown to be a major effort of simulating all the data expected to be produced by GAIA. GDAAS has NOW begun its second phase, where a significant number of algorithms will be implemented and tested for performance and evaluation of the amount of data storage that will be required during the operational phase.

1. Introduction

During its 5-year mission GAIA (ESA-2000-SCI-4) is expected to produce some 150 terabytes of raw data resulting into an expected 1 petabyte archive. The aim of this ongoing study is not only to identify the individual steps and algorithms of data reduction, but to also estimate the processing power and data storage requirements that will need to be procured to handle the daily tasks of data processing and archiving. The aim will also be not to wait until the very end of the mission to produce results on which the GAIA community can work on, but to do this earlier on, thereby producing preliminary results for further analysis.

The initial study began in July 2000, based on a modular architecture that can follow an evolutionary path over a ten-year period prior to launch. The involvement of an industrial partner (GMV, Spain) and an academic partner (University of Barcelona) has allowed the consortium to bring together several experts with a wide range of backgrounds. Furthermore, the responsibility of running the hardware and provision of computational infrastructure has been

given to a *flexible* partner, CESCA (Spain), which has adapted very rapidly to the requirements of the study in its initial phase.

Phase I of the GDAAS Study was completed in June 2002 (González 2002) in which two major astrometric algorithms were implemented: Cross-Matching (Lattanzi et al. 2001) and the Global Iterative Solution (Lindegren 2001).In the next phase, which started in September 2002, a number of identified algorithms will be implemented by involving the GAIA community at large. The study will furthermore follow closely the evolutionary path of the mission's ongoing redesign efforts and adjust to them accordingly.

2. The GDAAS architecture

GDAAS has adopted a three-tier architecture (see Figure 1):

Database Management System: an object-oriented architecture was chosen for the reasons of scalability, efficiency and maintainability. At the time of writing this article, Objectivity is being used. The database choice, however, may change in the future, without a major effort in re-implementing the system. In fact, this was one of the major driving requirements of selecting a highly configurable architecture.

Data Manipulation Layer: this layer is composed of two parts: the data model, which describes the bulk of data required by all algorithms and is categorised by individual classes, while the data manipulation itself stores and retrieves the individual *data objects*.

Processing Framework Layer: this layer manages system resources and processes. It is the forefront layer upon which individual algorithms are implemented. This particular layer plays a crucial role in scheduling as well as simultaneously processing more than one task in the system. The number of algorithms may be arbitrarily selected. The limiting factor may be the total processing power at our disposal.

The driving factor of having a distributed architecture is concurrency. This is an advantage, since it allows for processing not only to be distributed across several processors, but also a in time. The system was originally designed having data and processors very close to one another. This meant that the individual computers with large data stores were setup to work on individual processes. This was somewhat limiting, since very large processes would have slowed down the performance. By changing the architecture and centralising the storage array, it became more flexible for a single processor to increase the required storage, whenever the need arose, without being limited by a local disk. The concept of a *node* in GDAAS is virtual, solely defined by an arbitrary processor and the required arbitrary storage. The main advantage of this setup is therefore the uniform distribution of nodes along all physical computers that make up the overall system.

2.1. Configuration control

Due to the expected longevity of the code being currently implemented and the changes that the system will undergo over the next few years, it is highly desir-

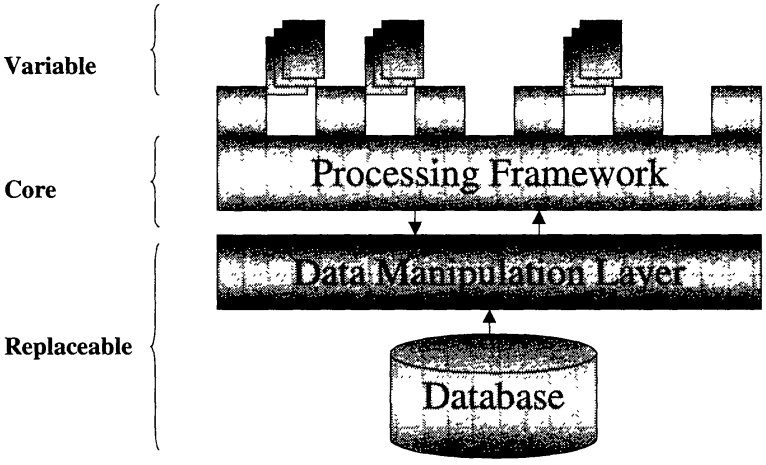

Figure 1. GDAAS System Architecture. A three-tier architecture was adopted to ensure the highest possible flexibility during system design.

able to keep a tight configuration control over various aspects of the algorithm implementation. Not only should the core system be properly and highly detailed and documented, but all changes that are carried out at various stages of system definition need to be explained and noted.

2.2. Algorithms

GDAAS algorithms are classified in three categories:

Critical: algorithms that have a direct impact on the data. These may modify the original data, the result being critical to the overall data reduction. Such algorithms must also be evaluated in terms of optimisation, as they will play a crucial role in the overall data processing performance. Typically cross-matching or the global iterative solution would fall under this category. This code may have to be translated into the internal language of the system for performance and maintenance reasons.

Non-critical: code, which may not have a direct impact on the overall data processing would typically fall into this category. In this case, the algorithm may be run independently of the ongoing processing, and use calibrated data to extract meaningful physical parameters. Photometric measurements or spectra may fall under this category. Such code may be written in any language and a wrapper would then be provided to allow it to communicate with the GDAAS main engine.

Plugins: these algorithms make use of end products and can be considered user-specific. Anyone wishing to make use of the GDAAS database to calculate e.g. the dynamics and evolution of the galaxy, or to extract a set of classified objects, based on given criteria may use this feature. The goal of such an approach at this stage is to ensure that the system can sustain

the management of these types of algorithms. This code is physically independent of the GAIA data processing, but access its resulting data.

2.3. Types of users

There are three types of users foreseen for GDAAS:

Administrator: the Administrator has all access privileges to the system and can also allocate disk space and processing power to individual tasks.

Power User: users who require to work on a substantial part of the database and/or require a large amount of processing power can request for an account to be setup on the central node.

Casual User: all other users wishing to extract data to work on individually can get access to the data through a special client being developed.

3. The GDAAS Phase I prototype

During the GDAAS Phase I we developed a model for the satellite, instruments and their operation allowing a description of the raw telemetry data (including needs and rates) and a model of the whole relationship among data, access and processing needs. At present, a user interface provides all the necessary functionality to operate and test the system.

3.1. Satellite model

Several simplifications of the GAIA payload and of the actual observations were considered. Only data coming from the astrometric instruments (two focal planes constituted by 10 x 26 TDI operated CCDs) was considered (the spectro instrument was not taken into account). We assumed that while crossing the focal plane the star follows a single CCD row, with fixed acquisition window size and constant PSF.

The datation of the observations is a critical issue because it determines the astrometric precision. A simplified yet effective model of datation was used where only the time of the ASM detection was transmitted to ground and the datation of the individual observations was inferred from the same algorithm used on-board. This schema allowed a precise datation of the observations while, at the same time, saving bandwidth.

A model of the on-board data flow was developed and, according to it, science data (17 *astrometric patches* plus 5 broad band *photometric patches*) was generated by the GDAAS simulator developed at the UB (Masana et al. 2001). About 1 million stars were used for GDAAS testing and, with the simplifications described above, the prototype managed about 250 GB of telemetry data.

3.2. Definition of the data model

Before any system design can be carried out, a data model describing the structure and relationships of the data has to be defined. This data model essentially contains the definition of the Java classes modelling the different sets of data that enter into the system. In addition to raw science data and source data

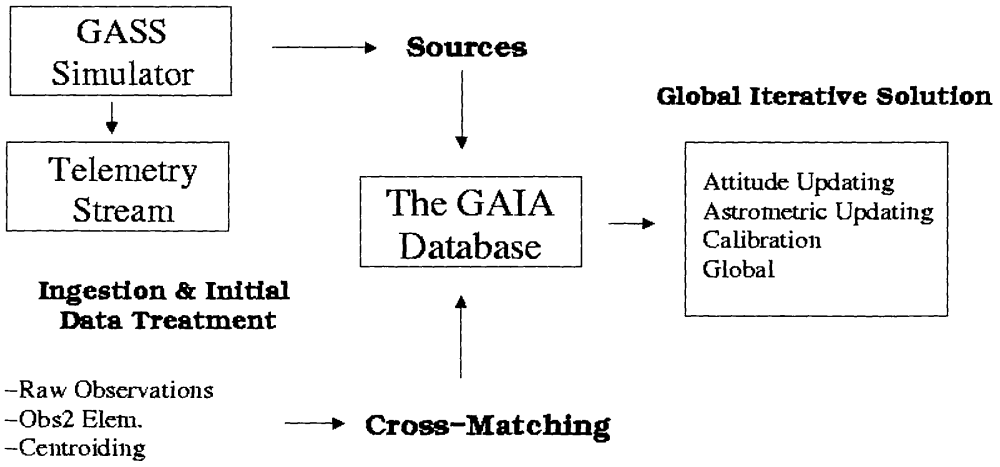

Figure 2. GDAAS Phase I prototype.

(basically astrometry and photometry), several sets of data have been designed to contain attitude, calibration, satellite data as well as auxiliary data. The data model has been designed and is maintained using UML tools.

3.3. Processing prototype

Figure 2 describes the prototype of the processing structure. From the processed telemetry stream the raw observations are stored and enter the initial data treatment, where, *elementary observations*, the ones needed on the GAIA data processing, are derived. Different observations of the same source are linked together during the cross-matching process where the DB *source objects* are created as necessary.

The needs of access in temporal, spatial and instrumental domains were identified, as well as the complex interdependencies between the different types of objects. The DB itself was designed as a scalable system, allowing remote and distributed processing and access. From the early stages of the GAIA project an object oriented DB system has been considered the best suited to handle the complexity of the GAIA structure.

4. The core processing

It is an iterative process requiring access to the DB in the temporal, spatial and CCD domain as well as in the source domain, and it is the most complex process to be executed in the GAIA system. The general concept for the astrometric reduction, is called the Global Iterative Solution (GIS). The GIS aims to compare and minimize the differences between the observed and calculated positions of the stars, provided that a model exists for the satellite attitude, the objects, and the instruments (see Figure 3). Astrometric relativistic treatment is required to reach the μas accuracy.

The practical implementation of GIS, fully described by Lindegren (2001), can be understood as an iterative sequence of four steps, that, through minimiza-

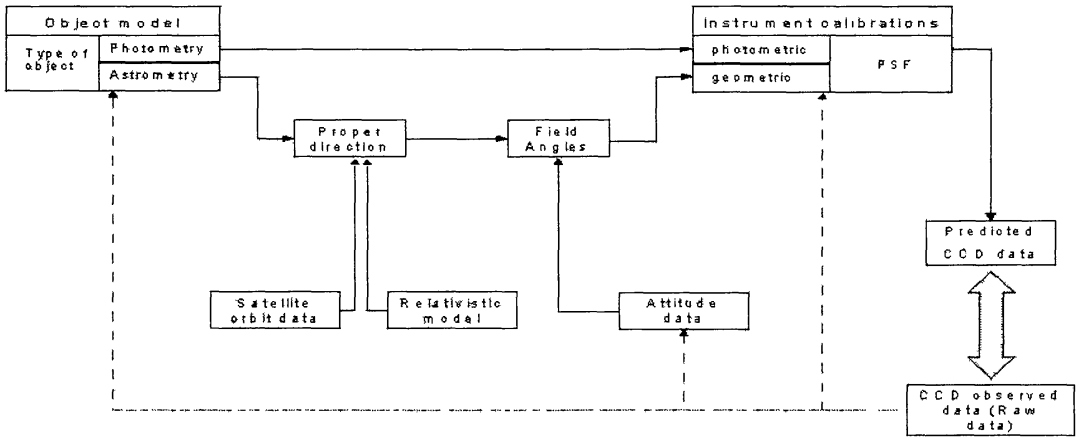

Figure 3. Core processing

tion, improve individual parts of the data entering in the model: the attitude of the satellite, the source astrometry, the calibration of the instruments and the global parameters.

5. The simulator

The development of a mission simulator is an independent project under the responsibility of the GAIA Simulation Working Group (SWG). However, the provision of simulated data for the GDAAS project is one of the key functions of this group and therefore the development of the GAIA simulator is closely linked to the GDAAS project.

More specifically, a specialised module of the overall simulator, named the GAIA System Simulator (GASS, Masana et al., 2001) was developed to cater for the GDAAS needs. This module generates realistic GAIA telemetry that can be ingested into the GDAAS system to fill its database. During Phase I the GDAAS simulator was adapted to the data model and simplifications assumed in this early stage. GASS is now being updated to take into account the revised GAIA design and to cover the needs of the second phase of GDAAS.

6. Testing the prototype

The tests discussed in this section cover Phase I of GDAAS. This phase included cross matching routines (Lattanzi et al. 2001) and the global iterative solution (Lindegren, 2001). It covers a full range of simulated data acquisition over a five-year period.

The performance of the system has been fully evaluated and the results from several tests have been used to devise possible optimisation techniques. Special attention was paid to the ingestion, cross-matching and GIS processes, which have extensive CPU and DB I/O requirements.

6.1. Daily processing

A basic requirement for the GDAAS system comes from the daily operations of the mission (downloading data from the satellite, storage, processing and DB ingestion). A test was carried out to evaluate the time consumption of the ingestion, initial data treatment and cross-matching of a full day of mission using simulations up to 20^{th} magnitude (realistic conditions). Time consumption for a single processor was about 280 CPU hours using the present design and hardware.

A complementary test for the ingestion, initial data treatment and cross-matching of the full mission telemetry was run by generating and processing a scaled-down simulation up to 13^{th} magnitude. Assuming a scaling factor of 380 from 13^{th} to 20^{th} magnitude (Torra et al. 1999 galaxy model), the final DB size would be around 460 TB. The average ingestion and cross-matching time consumption per single processor is of about 1.5 CPU hours per day of observation at 13^{th} magnitude. The total processing time can be easily reduced using distributed processing.

Therefore, taking into account the increase in hardware performance up to the mission start, the possibility of running a distributed ingestion process and the possible tuning and optimisation of the system, one can safely assume that the daily processing of telemetry is a feasible task.

Finally, the performance of the cross-matching algorithm in crowded regions up to faint magnitudes was also tested. First results indicate that the system is able to cope with high-density areas, although it should be adapted to cope with processing power peaks when certain regions in the galactic plane are observed by the satellite.

6.2. GIS processing

As expected, all tasks of GIS are complex and time consuming. The four GIS steps ran successfully in terms of data retrieval and storage. Because too few observations were processed during the initial GDAAS phase, some tests did not reach convergence. At present, GIS testing of 6 months of mission data considering 1 million of processing sources is underway.

7. Conclusions

The initial phase of GDAAS has provided the necessary experience in initially setting up a complex system, based on a highly flexible architecture on which to further build upon in the coming years prior to launch. It was important to have selected an open architecture, one that is maintainable over a long period of time, and at the same time flexible enough to evolve over time. The choice of technology will certainly change in the coming years, however, the base of the architecture should remain very much stable.

During the next phase of the project, a collaborative environment will be set up allowing the GAIA community to propose solutions to various aspects of the mission. Also, a rigid Configuration Control will be put in place, thereby allowing this effort ample time to evolve into an operational system. The tests have been promising in that the performance issues have not been as critical as

one may have thought. However, the technological development in handling this enormous database will be the actual challenge.

References

González, L.M. Serraller, I., Torra, J., et al. 2002, GMV-GDAAS-RP-001 (Livelink)

Lindegren, L. 2001, GAIA-LL-34 (Livelink)

Lattanzi, M., Luri, X., Spagna, A., Torra, J., Jordi, C., Figueras, F., Morbidelli, R., & Volpicelli, A. 2001, GDAAS-TN-005 (Livelink)

Masana, E., Jordi, C., Figueras, F., Torra, J., & Luri, X. 2001, Highlights of Spanish Astrophysics II, pag. 389

Torra, J., Chen, B., Figueras, F., Jordi, C., & Luri, X. 1999, Bal. Astron. 8, 171