Astronomical Data Analysis Software and Systems XXI
ASP Conference Series, Vol. 461
Pascal Ballester, Daniel Egret, and Nuria P. F. Lorente, eds.

Operations and Data Processing for the Planck Low-Frequency Instrument: Design Strategies and Practical Experience

F. Pasian,[1] A. Zacchei,[1] M. Frailis,[1] S. Galeotta,[1] M. Maris,[1] D. Tavagnacco,[2,1] C. Vuerli,[1] M. Tuerler,[3] R. Rohlfs,[3] N. Morisset,[3] M. Meharga,[3] T. A. Ensslin,[4] J. Knoche,[4] A. Gregorio,[2] D. Maino,[5] A. Mennella,[5,1] M. Tomasi,[5] F. Cuttaia,[6] G. Morgante,[6] L. Terenzi,[6] G. Maggio,[1] F. Gasparo,[1] E. Franceschi[6]

[1]*INAF - Osservatorio Astronomico di Trieste, Via G.B.Tiepolo 11, 34143 Trieste, Italy*

[2]*Dipartimento di Fisica, Universitá di Trieste, Italy*

[3]*ISDC, Chemin d'Ecogia, Versoix, Switzerland*

[4]*Max Planck Institute fur Astronomie, Garching, Germany*

[5]*Dipartimento di Fisica, Universitá di Milano, Italy*

[6]*INAF - IASF Bologna, Italy*

Abstract. Planck is an ESA mission launched in May 2009, which is mapping the microwave sky in nine frequencies and accurately measuring the anisotropies of the Cosmic Microwave Background (CMB) with its complement of two instruments (HFI and LFI), covering respectively the far infrared and the radio domains. The operations and data processing of the Planck instruments are carried out by Data Processing Centers, one for each instrument. The DPCs need to support both a day-by-day quasi-real-time calibration workflow and high-throughput pipelines for a high-volume data flow. The LFI DPC has been designed to be a centralized facility built by geographically distributed institutions, in a funding scenario based on multiple funding agencies and, in most cases, on a fixed budget in the presence of launch delays. A strategy for managing effectively the distributed and collaborative software development and maintenance has been developed, based on the use of open source and off-the-shelf software, and on the reuse of systems developed ad-hoc for other missions. Product and quality assurance has been supported throughout development, integration and testing. The effectiveness of the design choices has been proven by the readiness of the system at launch time and by the extremely smooth operations phase.

1. Data Processing Center (DPC): Definition and Development Constraints

The Planck DPCs are responsible for the production, archiving and delivery to ESA of the following scientific data products (Pasian & Sygnet 2002): calibrated time series data (including attitude reconstruction and removal of systematics), photometrically and astrometrically calibrated maps of the sky in the observed bands, sky maps of the main astrophysical components, catalogs of sources detected and the CMB power spectrum. The LFI and HFI DPCs were developed separately, sharing a set of standards (e.g. a common data format) and a number of utilities (e.g. documentation and software repositories, and a change tracking tool).

The design and development of the LFI DPC had to comply with a number of constraints. First, the fact that the developer's community is sparse: over 20 institutes in a dozen countries in Europe and North America. This distributed development maps onto a centralized approach for operations, which are run at INAF-OATrieste. As for the data, the flow goes from the spacecraft to the Ground Station in Perth, Australia, to the ESA-operated Mission Operations Center (MOC) in Darmstadt, Germany, and finally to the DPC. While there is no distribution of data to outside community (which is c/o ESA), there is exchange of data between Instrument Consortia (i.e. between the LFI and HFI DPCs). The main constraint refers to the budget, which is very limited: as a consequence there are no resources to re-code algorithms into an agreed-upon language.

The DPC processing can be logically divided in four Levels: S (Simulation), 1 (Telemetry Processing and Interface with the MOC), 2 (Data Reduction and Calibration), 3 (Component Separation and Optimization, and Generation of final products).

2. Scheme for Development–Integration–Release

The general structure of the Planck LFI development cycle used during pipeline development was defined (Zacchei et al. 2004). Scientists devise algorithms based on requirements raised from the data products needed to analyze and develop the relevant prototype code. Such software is not expected to have the robustness, documentation and maintainability that the DPC pipeline software is expected to guarantee. The DPC Integration Team therefore engineered the prototypes and pipeline elements in robust DPC-quality software. As soon as a frozen version of a prototype was produced the code was engineered, optimized and integrated at the DPC as a module harmonized into the pipeline (e.g. using common environment and data structures), and a release was issued.

The initial delivery of prototype software (code sources, scripts, make files, etc.) from both the Software Prototyping Team and the Modeling and Simulations Team to the DPC Team was accompanied by documentation forming the basis of the URDs for the Scientific Pipeline (one for each Level). The LFI DPC Integration Team was responsible of integrating the code, so as to produce pipeline software satisfying all requirements defined in the URDs. The Pipeline integration phase follows the ESA PSS-05 software development standards, including Product and Quality Assurance. The performance results and testing results of the integrated software was provided to the Software Prototyping Team and the Simulations and Modeling Team at the time of each official release. Integrated source code was provided back to the originators.

In some cases, scientific tests on the final product (release) modified the main requirements, so this implies that this cycle was applied iteratively with the scope of reaching the best scientific results.

3. Design, Development and Validation

The LFI DPC was designed and developed so as to be ready for launch, and is currently in operations (Zacchei 2011).

For Levels 2 and 3 of the pipeline, and for all simulations (Level S) run at the LFI DPC, the development followed the logical approach described in the previous section. The processing is harmonically arranged in such a way that data Pipelines, driven by an environment called Process Coordinator, are set up. Since there were no resources to re-code algorithms into an agreed-upon language, the Process Coordinator was designed with the capability of wrapping code written in many languages. A common infrastructure was necessary for all the DPC processing levels to define, archive into and retrieve the data from a DBMS. Such system, called Data Management Component (DMC), consists of two logical separated components: the DMC Interface (DMCI), an API independent of the physical storage used; and the DMC physical implementation, based on the Java Data Objects (JDO) standard and providing access to several database management systems.

For Level 1, which is more tightly related to operations, a different approach was chosen (Morisset et al. 2010). Direct development was used in this case, also reusing some of the components developed at ISDC for the INTEGRAL mission.

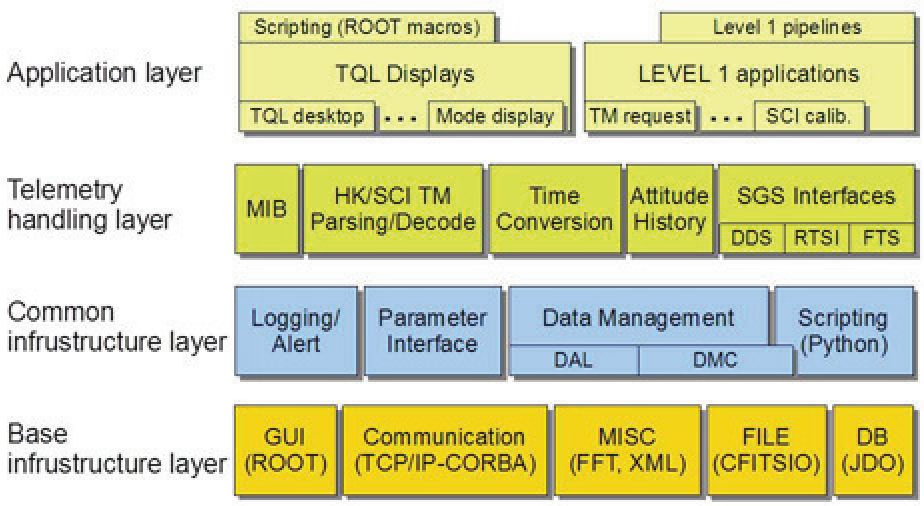

Figure 1. The logical architecture of Level 1: each layer groups software components sharing the same level of abstraction and stability of the interfaces.

The LFI Level 1 software (its schema is shown in Figure 1) includes two main subsystems: a TeleMetry Handler system (TMH), which retrieves and processes the raw telemetry packets, and generates time series of each scientific and housekeeping parameter (also called Time Ordered Information, TOI); a Telemetry Quick-Look system (TQL), providing interactive analysis tools specific for the LFI instrument. In the telemetry handling layer, a component is dedicated to the modeling of the Mission Information Base (MIB), which defines the structure of each telemetry packet, according to the PUS standard. Another component handles the interfaces (protocols) with the Mission Operation Centre (MOC) and the other Science Ground Segment (SGS) centres. As for the TMH, there are four main pipelines: the Data Receive pipeline implements the client side of the MOC interfaces, such as the Data Disposition System (DDS) and the Real-time Science Interface (RTSI). The TM Handling pipeline is triggered when a new segment of telemetry data is received; it groups the TM packets according to their source and type, decodes the housekeeping packets and converts the parameters into engineering units, checks if some parameter values are out of range, decompresses and decode the science TM and stores the timelines into the Level 1 archive. The two remaining pipelines handle auxiliary data such as attitude information, time correlation coefficients, and command history data.

Development efforts have also been specifically concentrated on the possibility of correct recovery in the case of DPC hardware failures, and on the concept of providing warm or cold backup capabilities at operations time.

All pipelines, and especially the Level 1 one (Frailis 2009), were obviously hard-tested before launch, and during operations new pipelines were installed and verified in a dedicated cluster/storage, to avoid interfering with the nominal operations. Only after approval new releases were installed on the nominal machine, interfaced with the nominal storage/database and officially released as new baseline. This process takes up to a few months, depending on the complexity of the tests to be performed.

4. Conclusions

The entire software development and integration cycle was tested and applied since the production of the very first LFI pipeline (the Bread-Board Model), and all the way to the Demonstration and Operations Models. The re-use of a good fraction of the system used for the INTEGRAL mission by the LFI DPC Level 1 allowed to optimize development efforts, and maximizing the use of freeware as off-the-shelf software allowed to keep costs low. The prototype-integration loop between scientist and DPC staff, and the possibility of wrapping code written in a variety of languages within a data processing environment was a strong point of the development strategy, and allowed to use in the most efficient way the limited amount of resources available.

The principles used to develop, integrate and engineer the code necessary to the Planck LFI pipeline can easily applied to any distributed software development project: in particular, they plan to be used, modified where necessary, in the development of the Science Ground Segment of the recently-approved Euclid ESA mission.

Acknowledgments. The development of the Planck LFI DPC has been supported by the various national agencies participating in the project, while the operations have been mainly supported by ASI and INAF. CINECA has provided the HPC cluster used during operations, and is gratefully acknowledged. Many thanks to Marco Molinaro for his support.

References

Frailis, M. 2009, Journal of Instrumentation, 12, T12021

Morisset, N., Rohlfs, R., Tuerler, M., Meharga, M., Binko, P., Beck, M., Frailis, M., & Zacchei, A. 2010, in ADASS XIX, edited by Y. Mizumoto, K.-I. Morita, & M. Ohishi, vol. 434 of ASP Conf. Ser., 143

Pasian, F., & Sygnet, J.-F. 2002, in Astronomical Data Analysis II, edited by J.-L. Starck & F. D. Murtagh, vol. 4847 of Proc. SPIE, 25

Zacchei, A. 2011, A&A, 536, A5. arXiv:1101.2040

Zacchei, A., Vuerli, C., Lama, N., & Pasian, F. 2004, in ADASS XIII, edited by F. Ochsenbein, M. G. Allen, & D. Egret, vol. 314 of ASP Conf. Ser., 396