From LPF to eLISA: new approach in payload software

Ll. Gesa, V. Martin, A. Conchillo, J.A. Ortega, I. Mateos, A. Torrents, J.P. Lopez-Zaragoza, F. Rivas, I. Lloro, M. Nofrarias, CF. Sopuerta
Space Sciences Institut (CSIC-IEEC) - Barcelona, Spain
E-mail: gesa@ieec.cat

Abstract. eLISA will be the first observatory in space to explore the Gravitational Universe. It will gather revolutionary information about the dark universe. This implies a robust and reliable embedded control software and hardware working together. With the lessons learnt with the LISA Pathfinder payload software as baseline, we will introduce in this short article the key concepts and new approaches that our group is working on in terms of software: multiprocessor, self-modifying-code strategies, 100% hardware and software monitoring, embedded scripting, Time and Space Partition among others.

1. Introduction
The Gravitational Wave Astronomy Research and Technology group of CSIC-IEEC has developed the LISA Pathfinder Data Management Unit (DMU). During several years, our group has implemented its embedded software following the ESA guidelines and standards to ensure each development stage along its quality. Now it’s time to, from lessons learnt from more than 8700 hours of real operativity in flight, start to define the new software approach for the future space based gravitational wave observatory: eLISA.

2. Lisa Pathfinder Software conclusions
The LPF DMU Software has shown its complete robustness during the LPF operations. But from the daily operative and monitoring we have learnt that things could be improved:

• Hardware status could be better informed to ground and a best internal diagnostic would be useful.
• A more flexible upgrade/modification mechanism it is desired: Now science operation mode must be stopped to have time for upgrades for example. On ground observatories the non-productive time should be avoided or at least reduced, therefore is desired to have the same aim in a space observatory.
• Historicals: An internal status history could help to analyze problems reducing the offline time needed for that.

Consequently, if the future software development adds improvements in those points it will maximize the support of the payload software for the scientific return. Hardware development also should take care of it, for example with more data room and CPU speed.
3. Our approach

Next figure deploys our approach: multiple applications running together in parallel sharing hardware resources and data. Each one with its own operating system and isolated security, and each one covering some specific functionality needed to control the payload system.

Our proposal is to group each needed behavior in separate applications to be executed isolated from the rest, using a technology called Time and Space Partitioning (TSP). Our solution is to have next applications:

- The **Basis partition**, running at highest security level, is in charge of overall system monitoring

- Data **application** is handling the internal data and its telemetry packing. The subsystem manager and housekeeping monitor is performed by the **Diagnostics** part.

- The **Science application** is the main core performing the specific science.

- And our innovative bet; the **Dinamic**, an application partition enabled to mirror any other partition, adding the capacity of system upgrades while system is in operational mode and also enabled to run scripting sequences, useful for debugging or in-place testing of new functionalities before to be official.

All of them running over a hardware abstraction layer in charge of managing the resources, for example the physical communication layers or controlling the memory data sharing mechanism between applications. These layers are known as bare metal hypervisors. Several solutions are already present, none of them in a maturity level to be used in space mission. Several among them are currently under study such as: AIR, an ESA sponsored project based on RTEMS, and XtratuM hypervisor system. Other commercial solutions could be VxWorks 653 or PikeOS.

In relation with the real-time operating systems on partitions, LISA Pathfinder Payload Software has been based on RTEMS, so our first approach is to still using it, but other options like Xluna, or the provided with XtratuM: LithOS are not discarded. Specially the use of standard GNU/Linux could be useful on the dynamic partition to enable the use of scripting execution for fast development, execution and diagnostic.

4. The key: ARINC-653

To accomplish our approach, the use of the standard ARINC-653 is under study. This Avionics Application Standard Software Interface is a specification for Time and Space Partitioning (TSP) in safety-critical avionics Real-Time Operating Systems (RTOS). It allows the hosting of multiple applications of different software levels on the same hardware in the context of an Integrated Modular Avionics (IMA) architecture. This technology is recently under ESA studies to be applied in space [1].

---

**Figure 1.** Architecture under study
Several software applications with its own characteristics and security levels can co-exist over same hardware exchanging (or not) information and the failure of one of them doesn’t affect the operativity of the others.

5. Development & Qualification
Development should follow ESA quality guidelines, the ECSS standards defines the technological development for space missions. Specifically ECSS-E-40 and Q-80 for software. In new researches in TSP area from a whole system architecture the guidelines and processes proposed by Space Avionics Open Interface aRchitecture (SAVOIR) should be followed. Since some ARINC-653 implementations are part of several ESA studies, projects and researches, full space qualification is not yet reached. Our work could be a partner in this process. Our approach of Dynamic APP partition is a new concept not yet studied, therefore a further work in terms of development and qualification will be required.

6. Conclusion
The work for eLISA is just starting. Many open points and much work ahead, but the challenges of all these new software technologies and technical proposals give us the energy to kick off as soon as possible to start to see results in short. Study phase is not yet finished, and this is important to enter in the developing phase ensuring the final goal. Validation and acceptance phase is the expected more complex part. These technologies are new in space and the qualification process will take its time.

7. Acknowledgments
We acknowledge support from Project AYA2010-15709-Espacio of the Spanish Ministry of Science and Innovation (MICINN)

References
[1] V. Bos, P. Mendham, P. Kauppinen, N. Holsti, A. Crespo, M. Masmano, J.A. de la Puente, J. Zamorano, “Time and Space Partitioning the EagleEye Reference Mission” in Data Systems In Aerospace (DASIA 2013), Porto, Portugal; pp. 1-7, May 2013.