The LISA PathFinder DMU software, a global overview

Lluís Gesa, Aleix Conchillo, Alberto Lobo, Ivan Lloro, Víctor Martín
Institut d'Estudis Espacials de Catalunya (IEEC)

Introduction

The IEEC team has developed the **LISA Pathfinder** Data Management Unit (**DMU**). During six and a half years, the software team has implemented its emmbedded software. The **ESA** guidelines and standards, have been followed ensuring and documenting each development stage along its quality.

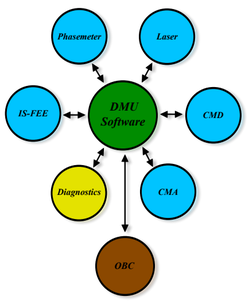
DMU, Data Management Unit

Picture of the EQM model of DMU, used in hardware and software tests. The DMU is composed by two sets (Nominal and Redundant) of :

- *Two PDUs: Power Distribution Unit*
- *Two DPUs: Data Processing Unit*
- *Two DAUs: Data Acquisition Unit*

Concept

The **DMU** is a critical piece of engineering in the **LISA Pathfinder** mission, in charge of the processing of data generated by the Phasemeter, acting also as an interface between the whole LISA Technology Package (LTP) and the On-Board Computer (OBC). It also manages and controls the diagnostics systems (Heaters, Coils, Magnetometers, Radiation monitor, Thermometers). The software is composed of two modules: the Boot Software, in charge of booting the system, and the Application Software, the full operating application.

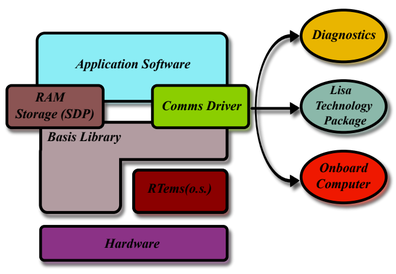


Basic concept

Design

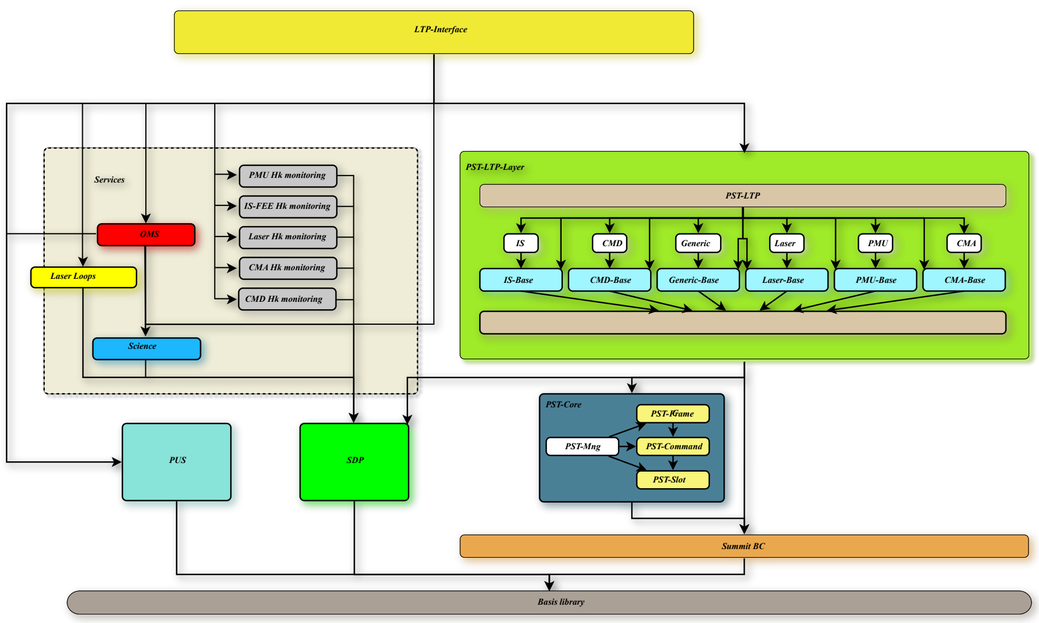
Software was designed to be high modulable, with 5 main blocks:

- **Basis library :** Common routines used in both BSW and ASW.
- **RAM storage :** Using a System Data Pool strategy (SDP). Shared between BSW and ASW.
- **Communication Drivers:** Built from scratch over Sμmmit chipsets using MIL-STD-1153 bus protocols for communications. Data layer using the **ESA** Pus Utilitzation Packet (PUS) Standard.
- **O.S. layer :** In BSW a minimalistic Operating System was completely implemented, whereas in ASW the ***RTems*** is used.
- **ASW :** The core of the control & alghoritms. Main software. Patchable by ground. Running at Real-Time, ensuring phasemeter processing and laser loops computing at 100hz, and providing information to OBC at 10hz.



Brief Design

Each of that blocks are designed with a large number of submodules with its behavior completely dedicated. Next diagram shows a portion of the elements in charge of LTP monitoring and control.

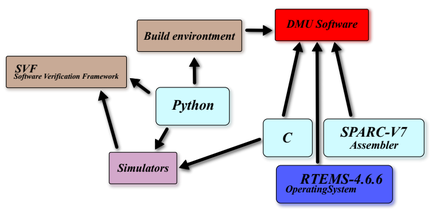


Software module interactions related with LTP

Implementation Technology

Since first commit on Mon Dec 12, 2005 15:09, system has been built basing its development on several technologies:

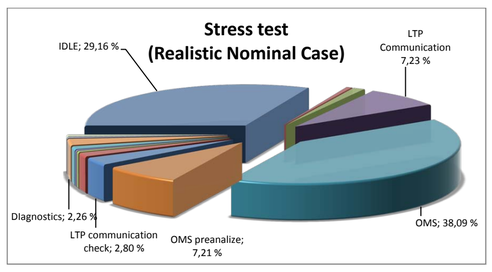
- C, as main programing language (C99) for:
  - Applictation itself.
  - Simulators (LTP and Diagnostics) for test and check.
- Python, as main support language for:
  - Build system.
  - Testing Framework.



Technologies used

RTems is used, in ASW, as operating system, a well-known system in Real-Time embedded software for space mission. Due hardware is based in *ERC32*, some pieces has been built directly in *SPARC-V7* assembler, basically for boot software.

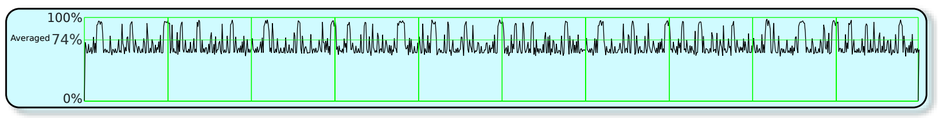
Test



Main actions during nominal stress operation

Software has been deeply tested:

- Unit Test as first stage, where each individual code method has been tested individualy.
- Validation Test, where each requirement or group of them, has been tested using a hardware flight equivalent model.
- Stress Test where software is executed in several full load scenarios during large time periods in order to stress the system.



CPU Load during 10 seconds in nominal stress operation

Application in numbers

Some numbers about Main Application (Boot Software not included)

ASW (Application)

Module	*Lines of code*
Basis (ANSI C)	16411
ASW (ANSI C)	40263
Total	56676

TESTING

Module	*Lines of code*
Unit Test (ANSI C)	83207
Validation	
Common (ANSI C)	8400
Common (Python)	18697
Stress Test (Python)	5263
Comms Validation (Python)	10219
Functional Validation (ANSI C)	16012
Functional Validation (Python)	208174
Total (ANSI C)	24412
Total (Python)	242353
Total	266765

323441 Lines of code to produce DMU Main Application.

- **4.7** lines of test per line of functional code.
- 42512 lines of source comments (headers plus inline comments) (ASW + Basis) this means **0.75** lines of comments per line of code.
- 622 files in ASW source structure (387 for binary and 237 for Unit Test).
- 423 files in Basis source structure (292 for binary and 131 for Unit Test).
- 137272 comments in whole Validation (Unit Test not included), this means **0.51** lines of comments per line of validation code.
- 1506 files in Validation source structures (Unit Test files not included).
- 23 ASW Releases.
- 1591 commits in ASW repository.
- 1368 commits in Basis library repository.

Conclusion

On March 2012, the release v3.4 of the application has been officially proposed to be accepted as flight software for **LISA Pathfinder**.
As a critical piece, the **DMU** software has been designed, implemented, and tested with special effort in guarantee its correctness. All the targets for this complex system have been achieved:

- **Fully functional :** All requeriments have been acomplished.
- **Reliable :** It has been thoroughly tested, with almost 5 lines of test code per software code.
- **Robust :** All possible scenarios have been tested, with different load, with expected results.
- **Qualified :** ESA standars compliant, and accepted as a validated software.

So, it is expected to be a system whose architecture could be re-used in part in future satellite mission software.

WWW: http://gwart.ice.cat
Mail: gesa@ieec.cat / vmartin@ieec.cat
Institut de Ciències de l'Espai (CSIC-IEEC) - Barcelona, Spain