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 embedded software. The ESA guidelines and standards, have been followed ensuring and documenting each development stage along its quality.

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.

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.
- **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 & algorithms. 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.

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

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

Software has been deeply tested:

- Unit Test as first stage, where each individual code method has been tested individually.
- 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.

Application in numbers

Some numbers about Main Application (Boot Software not included):

<table>
<thead>
<tr>
<th>Module</th>
<th>Lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASW (ANSI C)</td>
<td>40263</td>
</tr>
<tr>
<td>Total</td>
<td>56676</td>
</tr>
</tbody>
</table>

**TESTING**

<table>
<thead>
<tr>
<th>Module</th>
<th>Lines of code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation (ANSI C)</td>
<td>8400</td>
</tr>
<tr>
<td>Functional Validation (ANSI C)</td>
<td>16012</td>
</tr>
<tr>
<td>Total (ANSI C)</td>
<td>24412</td>
</tr>
<tr>
<td>Total (Python)</td>
<td>242353</td>
</tr>
</tbody>
</table>

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 0.78 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 files 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 Released.
- 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 requirements have been accomplished.
- **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 standards 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.