GLAST Large Area Telescope

Instrument Flight Software
EM2 Review
26 February 2004

Instrument Configuration by File

James Swain
Stanford Linear Accelerator Center

jswain@slac.stanford.edu
650-926-2721

Instrument Configuration by File: Requirements

- **Flight Software General Requirements:**
  - **Req. 5.3.4.6.2: FSW shall read back and record the configuration of the CAL, TKR, ACD, T&DF and LAT power supply subsystems by reading back the T&DF provided configuration registers.**
  - **Req. 5.3.4.6.1: FSW shall configure the CAL, TKR, ACD, T&DF, and LAT power-supply subsystems by writing to the T&DF-provided configuration registers.**

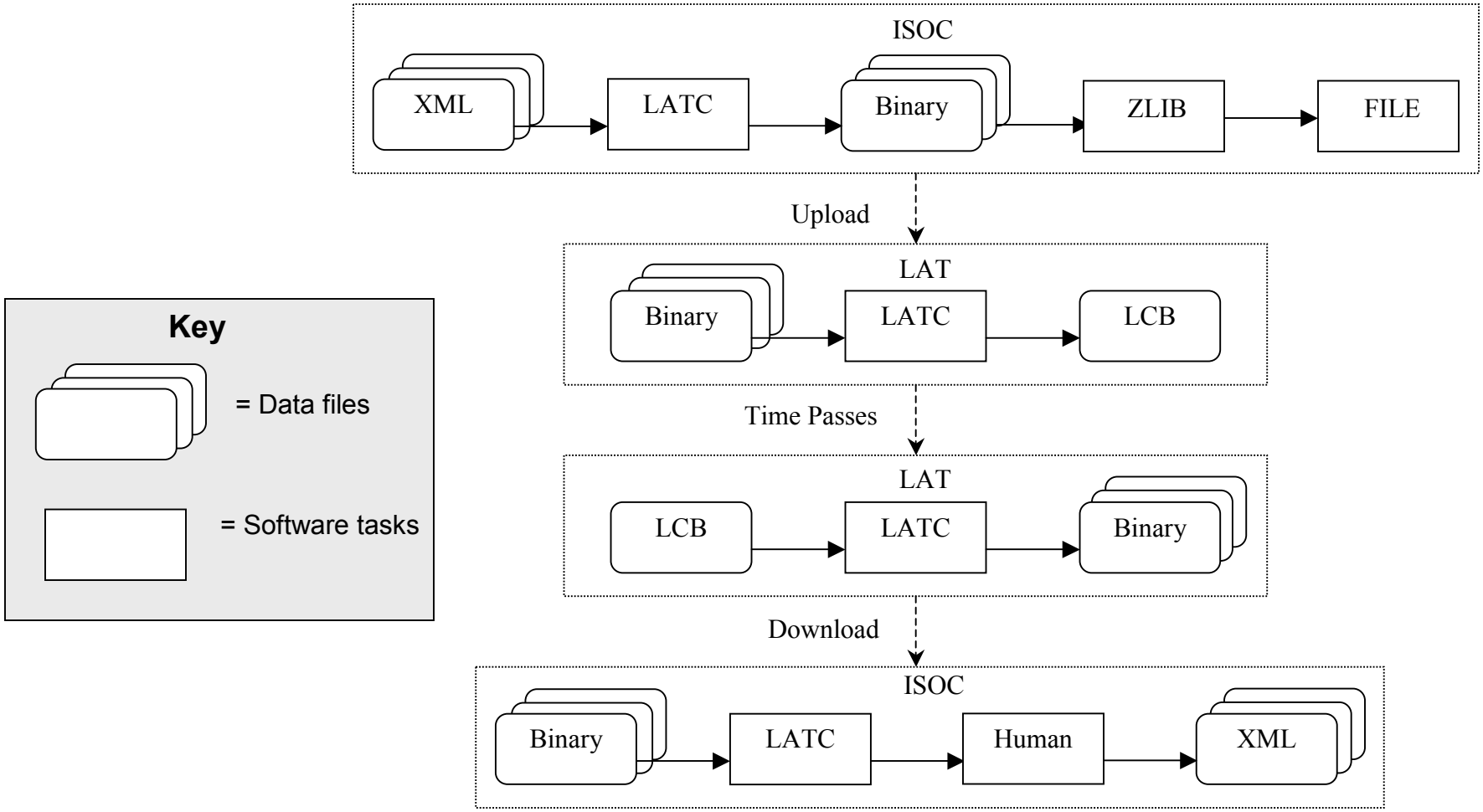
Instrument Configuration by File: Functional Components

- **Functional Inputs**
  - Set of LAT configurations defined by personnel on the ground and coded by them into corresponding sets of XML configuration files
  - Configuration data from registers on the CAL, TKR, ACD, T&DF, and LAT power-supply subsystems
- **Functional Processing**
  - On the ground, at the OC, XML configuration files are processed by the function, resulting in multiple binary configuration files that are suitable for uploading to the Spacecraft. The reverse process is also available: binary configuration files describing the state of the LAT are downloaded from the Spacecraft and processed to yield human-readable XML.
  - On the Spacecraft, binary files received from the ground are processed to yield a set of register commands, which are then sent to the LAT and configure its operation. Again, the reverse process is also available.
- **Functional Outputs**
  - Collection of binary files describing register settings on the LAT.
  - XML files reporting the current state of the LAT, for use by personnel on the ground.

Instrument Configuration by File: Design Overview

- Full instrument configuration description is large
  - Impractical to configure instrument with a series of telecommands
  - Impractical to record instrument configurations with a “simple” register dump
- Instrument flight software has developed methods for segmenting and compressing configurations
  - Segmentation
    - Define a file format, describing a hierarchical description of all registers
    - Allow any single file to populate only a small segment of this hierarchy
    - Reading a series of files builds up a full configuration
      - One of the files contains a “golden” instrument configuration
      - Other files then annotate deviations from the golden configuration
      - A particular target configuration (e.g., trigger configuration) is put in a file by itself and reused
  - Compression
    - The above files start in XML format
      - Easy to describe, code manage, and exchange with other LAT groups
      - Flight software and I&T are converging the XML file format
    - A (host/ground based) tool converts the XML to a compact binary format
    - The resulting binary format is compressed using ZLIB

Instrument Configuration by File: Flow of Configuration Data



Instrument Configuration by File: Static and Dynamic Configuration Information

- **The Configuration by File function allows users to plan a dynamic configuration process. The availability of the function allows users to approach LAT registers as though they fall into three categories or *lifetimes*:**
  - **1. Contextual. These registers are set by a flight software application or script when the LAT performs a specific task, e.g. value of charge injection DAC during calibration.**
  - **2. Static. These registers are expected to retain the same value for the life of the experiment having been determined before launch, e.g. the delay registers on the TEM.**
  - **3. Dynamic. These registers are expected to change during the life of the experiment and are the primary focus of the configuration software, e.g. the tracker strip trigger and event masks.**

Instrument Configuration by File: Data Segmentation and Incremental Updates

- Once registers are classified in this manner, configuration files can be segmented accordingly, giving great flexibility in configuration:
  - Settings for static and dynamic registers are separated into different configuration files.
  - Settings for different subsystems or towers can be separated into different configuration files.
  - Default settings are separated from custom settings.
  - Finally, a configuration master file is created to mix and match specialized configuration files as needed.
- This flexibility allows for incremental updating:
  - The XML file(s) forming the input need only contain a description of changed components. For example, if the tracker front end enable masks need to be altered for several TFEs, then the XML files will describe the TFEs and nothing else. Only TFE binary configuration files will be produced and only these files need be uploaded.
  - However, human intervention is needed before the configuration is uploaded. To combine the previously uploaded configuration for the rest of the LAT with the current modifications, the configuration master file must be edited to include the unchanged configuration binary files that are already present on the LAT.

Instrument Configuration by File: Testing

- The Configuration by File function will be tested in several ways:
  - Low-level unit tests will be performed on the LATC code package itself.
  - In operational tests with a functioning SIU (running LATC), a GASU, and functioning LAT subsystems, a test XML configuration file will be sent to the SIU:
    - The LAT's configuration will be independently verified and checked against the test configuration file
    - The LATC package will ask the LAT for the binary file describing its configuration. This file will be unpacked to XML, loaded to memory, and the in-memory representations of the test and output XML files will be compared.

Instrument Configuration by File: Consumers

	Test Stands		ISIS	I&T	FSW Test-bed	FSW Deliverable
	TKR,CAL	ACD				
Configure by compressed file	Y	Y	Y	Y	Y	Y

- **Status:**
  - LATC essentially complete.
- **Test Stands:**
  - Engineers can configure registers on test stand hardware
- **ISIS:**
  - The GSE can configure registers on the CRU, EBM, GEM, and PDU
- **I&T:**
  - I&T is able to fully configure all integrated subsystems.
- **FSW Test-bed:**
  - FSW is able to test its code against various configurations of LAT hardware.