



GLAST Large Area Telescope

**Instrument Flight Software
Flight Unit Design Review
16 September 2004**

**Telecommand and Telemetry Data
Dictionary Services**

**Sergio Maldonado
Stanford Linear Accelerator Center
smaldona@slac.stanford.edu
650-926-4355**



Telecommand and Telemetry Data Dictionary: Requirements

- **Flight Software General Requirements (2 new of 198):**
 - **LAT Housekeeping Data to SC (5.3.5.1.1)**
 - The SIU FSW shall provide housekeeping data to the SC, via the CTDB, on a schedule defined in GLAST 1553 Bus Protocol ICD.
 - **Diagnostic Telemetry (5.3.5.2)**
 - The FSW shall send diagnostic telemetry messages, as defined in the LAT Flight Software Telecommand and Telemetry Formats document, to the SC via the CTDB. Diagnostic telemetry is sent in response to a command and downlinked immediately while the SC in contact with the ground. If not in contact, the telemetry is downlinked at the next contact.
 - **Alert Telemetry (5.3.5.3)**
 - The FSW shall send alert telemetry messages, as defined in the LAT Flight Software Telecommand and Telemetry Formats document, to the SC via the CTDB. Alert telemetry is generated internally by the LAT and sent to the SC to be downlinked immediately. If not in contact, the SC recognizes the telemetry as Alert and activates a TDRSS link for immediate downlink.

- **Flight Software General Requirements (derived):**
 - The Telecommand and Telemetry Data Dictionary function shall generate a data dictionary defining the full set of telecommands and telemetry associated with the instrument FSW component of the mission. This data dictionary shall be provided in a format that can be processed to produce ITOS, or other representations of the data.



Telecommands and Telemetry Data Dictionary: The Issue

- **Constraints**
 - During the mission, MOC will use the Integrated Test and Operations System (ITOS)
 - During satellite integration, Spectrum Astro will use an ITOS compatible system
 - LAT FSW must produce a telecommand and telemetry data dictionary
 - LAT FSW must produce code corresponding to the telecommand and telemetry data dictionary document
 - This code defines all types of commands consumed and telemetry packets issued by FSW tasks:
 - commands to tasks
 - housekeeping telemetry
 - alert telemetry
 - diagnostic telemetry
- Many products, but all derived from the same basic information
 - A maintenance nightmare, if all products are maintained independently
- The FSW solution: the LCAT (LAT Command and Telemetry) tool



The CCSDS Packet Format

- LCAT doesn't concern itself with details of the 1553 bus protocol and CCSDS packet assembly, which are handled by the CTDB and CCSDS FSW packages, respectively. However, it's worth pointing out a few details about the CCSDS format for context.
- CCSDS command and telemetry packets have a header, a payload, and a checksum.
 - The header contains:
 - Fields that define the packet type, length, sequence order and count, timestamps, etc.
 - An APID that identifies destination tasks (in the case of telecommands) or source tasks (for telemetry) for routing purposes
 - For telecommand packets, a function code that identifies a routine to be executed by a destination task
 - Payloads can be variable length. Payloads can be segmented into bitfields or fields.
 - A checksum.
- Using LCAT, developers:
 - Define the payload structure of the telecommand packets received by or telemetry packets issued by their subsystems.
 - Define the APID (and function code) to be included in the packet header.



CCSDS Payloads: Field Data Types and Constraints

- **Command and telemetry fields can have one of 7 data types:**
 - * (unsigned) char
 - * (unsigned) int
 - * float
 - * string
 - * (unsigned) short
 - * (unsigned) long long
 - * double
- **Values in some fields should naturally be constrained or converted.**
 - For instance, temperature telemetry data may be compared against thresholds -- green (safe), yellow (watch), and red (warning) -- and should be reported in telemetry accordingly.
 - Values for these thresholds (limits) must be specified.
- **Command field constraints:**
 - Some telecommand field data is enumerated
 - Some fields should be restricted to an accepted range of values
- **Telemetry field constraints:**
 - Limit sets: some values in telemetry fields should be mapped to green, yellow, and red threshold limits
 - Telemetry fields can also be converted using discrete and analog conversion rules
 - Telemetry fields may have engineering units (e.g., °C) that must be identified

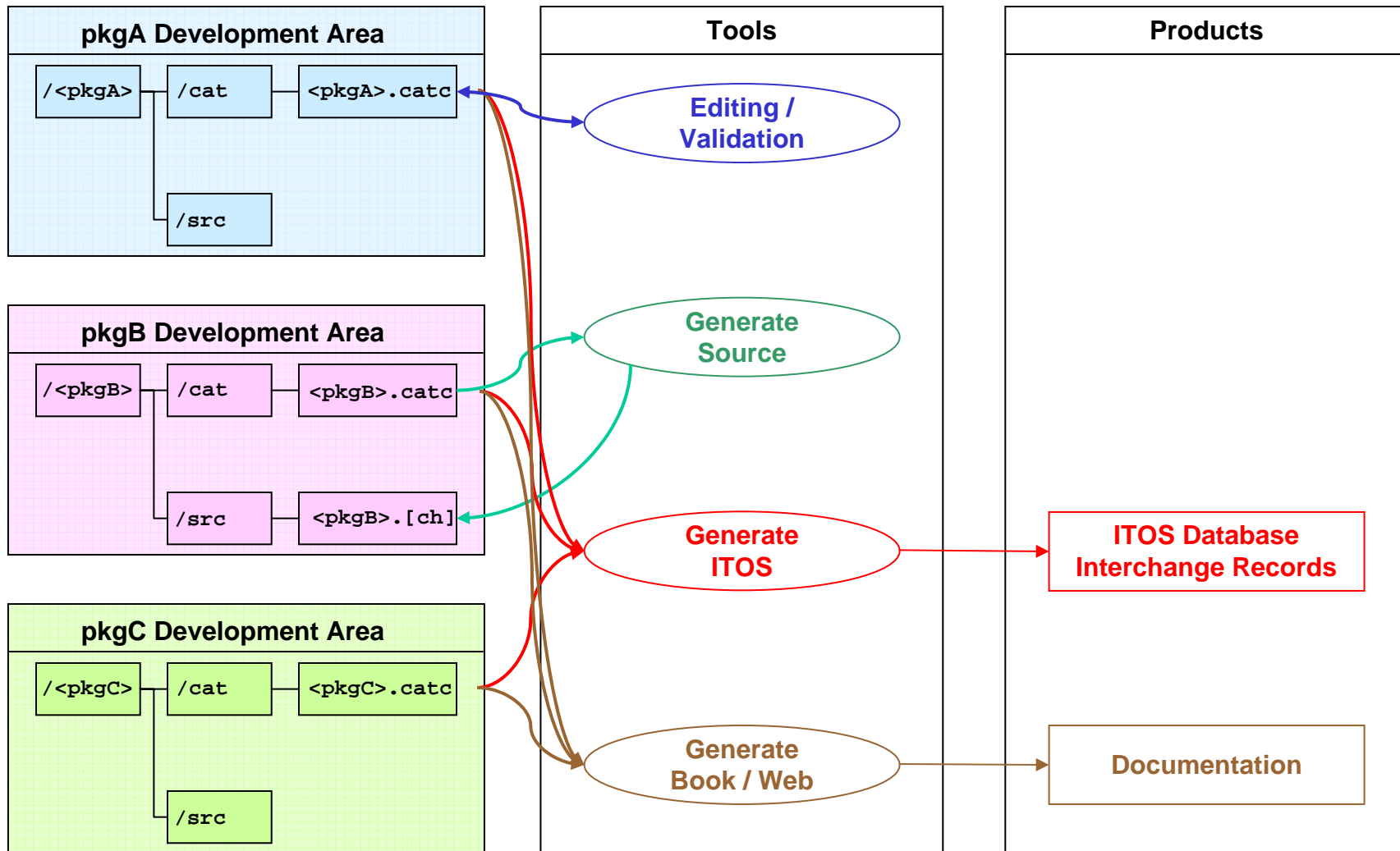


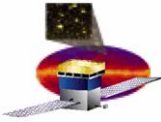
Telecommands and Telemetry Data Dictionary: The Design

- **A tools-based approach**
 - Define a private, XML-based representation of this information
 - Provide an editor tool to manipulate the XML files
 - Provide extraction tools to produce
 - ITOS database interchange records
 - Telecommand and telemetry web pages and book
 - Source code for inclusion by FSW applications
- **Integration into developer's environment**
 - Any package defining telecommands or telemetry will have a `/cat` directory
 - Developer will create package-specific telecommand / telemetry XML files in `/cat`
 - XML files are code managed under our present system
 - Deriving source code files from XML files will be built into our build scheme
- **ITOS, books and web pages are created by processing `/cat` directories in all packages**



Telecommands and Telemetry Data Dictionary: Workflow





Telecommands and Telemetry: The LCAT Tool

- LCAT, the “LAT Command and Telemetry” definition tool
- LCAT allows developers to:
 - 1) Create a centralized XML representation of all commands and telemetry associated with their software subsystems
 - 2) Generate two general categories of code:
 - Source code
 - C structures corresponding to each command and telemetry packet, each field within those packets, and all constraints on those fields
 - C structures used by the ITC (inter-task communications) system to register APIDs, function codes, and callback functions
 - Callback function prototypes and function stubs for commands
 - Database interchange records for ground-based communications systems
 - For each command and each telemetry message, database records used by the GLAST mission operations folks
 - Equivalent records used by the Spacecraft manufacturer to test LAT-Spacecraft integration
 - 3) Generate command and telemetry documentation
 - 4) Manage all these products centrally with a GUI tool (next slide)
 - 5) Automate source and interchange code production with a command line utility



The LCAT GUI

The screenshot shows the LCAT V1-0-0 GUI with a menu open over the 'TEMPWR' packet. The menu options are:

- Add Telemetry Packet
- Edit Item Payload
- Edit ITOS Mnemonics
- Remove ITOS Mnemonics
- Add to Container
- Delete Item

The main window displays a list of packets with the following columns: Packet ID, APID, and Brief.

Packet ID	APID	Brief
AEM0	402	AEM Environmental Monitor Packe
PDU0	398	PDU Packet 0
PDU1	399	PDU Packet 1
PDU2	400	PDU Packet 2
PDU3	401	PDU Packet 3
TEMPWR0	384	TEM power environmental packet (
TEMPWR1	385	TEM environmental power packet
TEMPWR2	386	TEM environmental power packet 2
TEMPWR		TEM power environmental packet (
TEMPW		TEM environmental power packet 4
TEMPW		TEM environmental power packet 5
TEMTM		TEM environmental power packet 0
TEMTM		TEM environmental power packet 1
TEMTM		TEM environmental power packet 2
TEMTMP3	393	
TEMTMP4	394	
TEMTMP5	395	

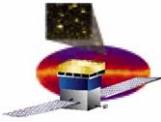
The 'Container' pane on the right shows a tree view of the container structure:

- container (LHK_catc)
 - created (05/26/04)
 - lcat_ver (test)
 - author (smaldona)
 - modified (06/04/04)
 - subsystem (LHK)
 - apid_low (384)
 - apid_high (639)
 - brief (LAT Instrument H...)
 - descr (The LAT instrume...)
 - def_files
 - def_file (LHK_alg.cath)
 - def_file (LHK_cmd_bf.cath)
 - def_file (LHK_cmd fld.cath)
 - def_file (LHK_cmd_pkt.cath)
 - def_file (LHK_cmd_struct.cath)
 - def_file (LHK_dsc.cath)
 - def_file (LHK_enm.cath)
 - def_file (LHK_lim.cath)
 - def_file (LHK_rng.cath)
 - def_file (LHK_tlm_bf.cath)
 - def_file (LHK_tlm fld.cath)



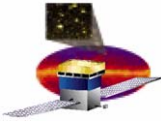
LCAT C&T Definitions: Product Hierarchy

- The primary storage mechanism for LCAT definitions are XML files.
 - All information is organized into a strict hierarchy, which is enforced by the application's XML parser and the associated Document Type Definition (DTD) file.
 - A single top level "container" file holds a reference to each definition file and allows for the sharing or reuse of data content between them.
 - Command and telemetry definitions are stored in multiple "definition" files. Once created, these definition files can be imported into other command and telemetry containers/projects:
 - Fields (cmd and tlm)
 - Bitfields (cmd and tlm)
 - Structs (cmd and tlm)
 - Packets (cmd and tlm)
 - Ranges (cmd only)
 - Enumerations
 - Discrete Conversions
 - Analog Conversions
 - Limit Sets



Production of ITOS and AstroRT by LCAT

- **LCAT outputs ITOS database interchange records as character-delimited dbx files, one file for commands and one file for telemetry.**
 - **These interchange files are used to populate the databases that support the ITOS and AstroRT systems.**
- **ITOS interchange format:**
 - **The cmd_itos.dbx file contains a record that defines the subsystem, and records for each command definition and command enumeration definition.**
 - **The tlm_itos.dbx file contains a record for the subsystem definition, and records for each telemetry definition, each conversion definition (discrete and analog), and each limit definition.**
- **AstroRT interchange format:**
 - **The dbx files are nearly identical to the ITOS versions. However, they do not contain records to define subsystems.**
 - **AstroRT records are not generated by the LCAT tool; the AstroRT software converts from ITOS**
- **Individual programmers produce database interchange records for the subsystems they write.**
 - **Since APIDs do not overlap, the whole set of records is imported into ITOS or AstroRT to create the master command and telemetry database.**



LCAT Command Line Tool

A command-line version of the tool is provided for build automation.

Command-line version capabilities:

- Validate XML container files against the DTD
- * Generate C source
- * Generate ITOS
- * Print dependency list for container
- * Print xml dump of the container including all referenced definition files

```
lcat validate <container_file> [--local | --global]

<container_file>  Filename of container to validate
--global           Validate against all global packages
--local           Validate local container
```

```
lcat generate source <container_file>
  [--cmd=<cmd_header_filename>]
  [--template=<cmd_src_template_filename>]
  [--tlm=<tlm_header_filename>]

<container_file>  Filename of container
```

```
lcat generate itos <container_file>
  [--cmd=<itos_cmd_filename>]
  [--tlm=<itos_tlm_filename>]

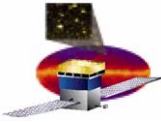
<container_file>  Filename of container
```

```
lcat generate dependencies <container_file>

<container_file>  Filename of container
```

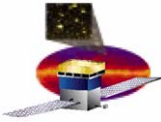
```
lcat generate xml <container_file>

<container_file>  Filename of container
```



LCAT Development Environment

- **For the full LCAT GUI:**
 - **Python 2.2.2 or later**
 - **PyXML version 0.8.1 or later**
 - **4Suite version 1.0a3 or later**
 - **Qt/X11 GPL/Non-Commercial version 3.1.1 or later – A platform independent C++ application development framework**
 - **QScintilla version 1.53-x11-gpl or later – A port to Qt of the Scintilla C++ editor class**
 - **SIP version x11-gpl-3.6 or later -- Python bindings for C++**
 - **PyQt gpl version 3.6 or later – A python interface to Qt libraries**



Telecommands and Telemetry Data Dictionary: Testing

- **By and large, this function will be tested in use, through verification and inspection:**
 - **As FSW code packages that consume telecommands and return telemetry are refined and completed, developers will check that valid source code was generated by the T&T database tool set.**
 - **As FSW hardware subsystems are refined and completed, engineers will send telecommands generated using the T&T tool set and verify in action whether the appropriate telecommands are sent and correct telemetry is received.**
 - **As consumers such as the ISIS and the ISOC utilize the data dictionary, they will validate that the correct representation of T&T data has been delivered.**