

Integrated Environment for Launch Vehicle Telemetry Data Management

Arjun R

Dept. of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala-673601
arjrs123@gmail.com

Mohammed Shahim MI

Scientist/Engineer-SF, Section Head, MSSG/VSSC
Mission Simulation and Synthesis Group
Vikram Sarabhai Space Centre (ISRO)
Thiruvananthapuram, Kerala-695 022
m_shahim@vssc.gov.in

Santhosh Kumar G

Assistant Professor
Department of Computer Science
Cochin University of Science and Technology
Cochin, Kerala-682 022
san@cusat.ac.in

Abstract—The telemetry data processing operation intended for a given mission are pre-defined by an onboard telemetry configuration, mission trajectory and overall telemetry methodology have stabilized lately for ISRO vehicles. The given problem on telemetry data processing is reduced through hierarchical problem reduction whereby the sequencing of operations evolves as the control task and operations on data as the function task. The function task Input, Output and execution criteria are captured into tables which are examined by the control task and then schedules when the function task when the criteria is being met.

Keywords- Data Management, Telemetry processing, Automation

INTRODUCTION

Telemetry systems form a crucial for the on-flight and post monitoring and analysis, which is key element in the launch vehicle missions. In order to achieve a high level of quality and reliability, very large amount of flight transmitted telemetry data is received and stored. The data of previous missions forms vital and significant source for the analysis and design in later missions.

The flight transmitted telemetry data, which is received and stored by ground stations, is a binary file with a pre-defined format. The format information is available in database form and is processed to extract information required to process the telemetry data. Data is organized in records, which contains various parameter values monitored in flight and other formatted sub-streams posted by the onboard computers etc. The data stored in processing system is transformed into a usable form through a series of operations, which begin with the ambiguities removal and merging of the ground station data, which ends up with delivering data in a usable form for analysis.

The application addresses in the solving present telemetry data processing issues comprehensively, which encompass the following salient features:

I. Domain Wisdom: The consistency checks for the format database, processing of format database and sequencing of the core operations are captured in the implementation so that the program functions with domain wisdom of the human expertise. The implementation is flexible so that changing the underlying principles does not affect the main program.

II. Full Automation: Telemetry data processing for a mission begin with an empty workspace and ends with version controlled output data. [2] The program has interfaces to the format database, raw telemetry data from ground stations, and user performs operations in a pre-defined response to events that accomplish the task of telemetry data processing.

II. LITERATURE SURVEY

An optimal application architecture is designed following a brief literature survey. ISPD (“ISRO Software Process Document”) and ISES-92 (“ISRO Software Engineering Standards”) were standards initially studied in formulating ideal solution. (Please refer the references section for complete list). The different telemetry processing sub elements being identified are-

- a. **TM Data**- The raw telemetry data from LV, which is logged in the ground station, is being stored in using the file extension *.dat*. Data in this format is unusable for analysis. The default processing strategy is being followed in different launch missions.

- b. **PCM Format-** The PCM plan and format for each mission is available as well in advance for the format implementation to be validated with the telemetry configuration and point out errors or ambiguity in the configuration format. The PCM format i.e. pulse code modulation format information is contained in separate document that identifies and applies the standard of encoding to be applied in the telemetry data files. The data received in RU is being encoded in pulse coded stream that has multiple streams of data. Here it needs to be demultiplexed. It is usually included in data file named *TMRUI.dbf* which holds data for specific RU numbering information.
- c. **Scale factor-** Scale factor information is the process by which the data received in manner of attenuation and compressed by the offset deviation can be reconstructed to the original data sample and identified with standard engineering value. The scale factor information is used to regain the compressed value to the original scale, which is standard engineering unit information. Standard scale factor files are being applied to the processing stream, with files of extension *.sf* for each mission named accordingly. For ex. PSLV mission files named PSLV-D1 and corresponding scale factor file *pslv_d1.sf* and likewise.
- d. **Workspace archive-** The repository of the processing information logs. Primarily information like addition of the input data, scale factor file used, the flight measurement plan are store usually denoted a unique configuration file which can be reused for further missions. The workspace folder also retrieves backup copy with file extension (*.bak*) of original file content.
- e. **Input and Output tables-** It defines and maintains the various Input and Output tables that are interfaced to the application. In the input, the raw TM Data files, configuration files, Scale factor, PCM data format are applied. The output module of application are carries the final processed data.
- f. **TM Data CD-** TM Data compact disc is the final processed and the version controlled telemetry data that is in the usable form for analysis. Application provides options to save processing history and make final data to be stored in CD (compact disc).

III. PROGRAM ARCHITECTURE

The program architecture schematic is shown in **Figure 1**. Various components in application program include following:

User: The user of the application software is authenticated with user id and password only is authorized personnel to use the software.

GUI: The TM DataPro application is designed with windows based graphical user interface for the user to operate the application software [1]. At the front-end

different database tables are developed using Microsoft Access 2000, and graphic elements with Microsoft Visual Basic 6.0.

GUI manager: The user interface manager is separate program module that is being used to manage and transact the user information to the application through interactive messages, windows, warning alerts and dialog boxes. Options to add new GS data, Preprocess, Merging etc. are provided in menu buttons.

Task Configuration table: The task configuration table should hold database with process specific information. Logic of each operation is stored with superset of taskmaster table. Each configuration defined create subset task table, which is derived from the task master table. It contains knowledge such as tasks which are under operation and status information.

Task Control table: It is table which depends on the task master. The task control table table functions as controller that enable or disable different tasks that are created by the master logic that is stored in name task master.

Task Scheduler: The main task function and the program loop is function, which runs from the application module and controls the task master table to populate and alter the process specific function table information.

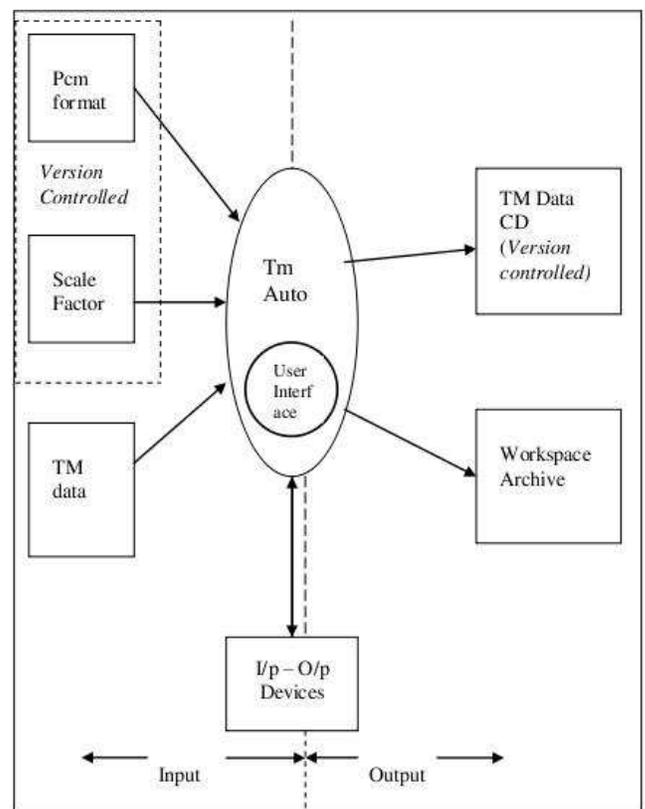


Figure 1

LIST OF ABBREVIATIONS

Abbreviation	Description
LV	Launch Vehicle
GS	Ground Station
TM	Telemetry
FMP	Flight Measurement Plan
QLD	Quick Look Data
AMC	Advanced Mission Computer
OBC	Onboard Computer
ICU	Integrated control unit
DSU	Data Storage Unit
RU	Receiving Unit
PCM	Pulse Code Modulation

IV. MULTILAYER FUNCTIONAL DECOMPOSITION

The various operations that culminate into processed, version controlled telemetry data have a direct bearing on the onboard telemetry configuration, ground station visibility matrix, and the measurement format / scale-factor. Hence if appropriate interfaces are identified to input the ground station visibility matrix and onboard telemetry configuration (database tables are used here) then the system can derive the real input requirements and sequence of operation.[4]

The processing operations are hierarchal in nature that each operation requires a preceding operation to be successfully completed and each operation in course gives green signal to a succeeding operation. It is possible to visualize the entire sequence of operations spread over few layers, where there could be more than one operation performed in each layer.

The layer hierarchy can be implemented by giving start criteria for each operation based on the successful completion of the predecessor task. When all tasks are completed, then the task of telemetry data processing is over.

The format and scale factor inputs alter the numerical configuration inputs to the various operations. In order to maintain transparency and correlation with latest version inputs, these inputs are received and integrated to the processing environment as these are released.

Work Space Concept 1 – Requires that user define a work space (an empty folder) and start populating it. The work space inherits all the super set database tables and structures of all other database tables which are to be derived.

Work Space Concept 2 – Workspace has a uniform structure irrespective of the mission. The system can identify additions to the work space based on the ground station visibility matrix and onboard telemetry configuration.

Work Space Concept 3 – At any given point of time, work space status could be (1) being developed, (2) ready, (3)a retraction from ready state or a retraction from accomplished state, (4) accomplished state.

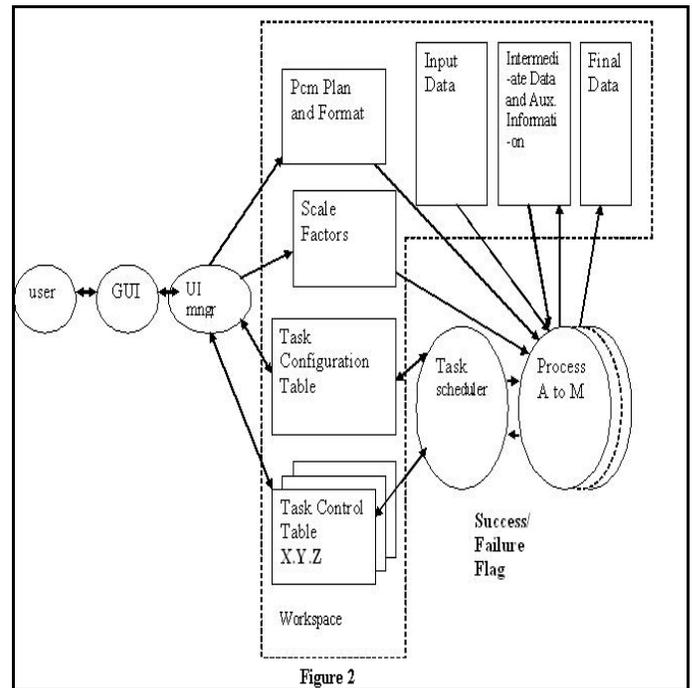
Work Space Concept 4 – work space status ready means (1) The ground station visibility matrix and onboard telemetry configuration are input and verified / approved, -AND- (2) The sequence of operations (Auto Sequencing) generated by the system is verified / approved, -AND- (3) The format and

scale factor inputs are added to the work space and the subsequent numerical configuration inputs to the various operations are verified / approved.

Work Space Concept 5 – Once the work space status attains ready state, it may proceed to process TM data and accomplish the task. However, it may be required to implement a change in (1) onboard telemetry configuration, (2) The sequence of operations, or (3) The format and scale factor inputs, in which case it is a retraction. A retraction may occur either from ready state or accomplished state.

Work Space Concept 6 – A change of work space status from ready or accomplished state to retraction and then back to ready state makes an incremental change in version number. This way version control is a property of the workspace and is automatically done.

V. AUTO SEQUENCING IMPLEMENTATION



The **Figure 2** represents the detailed software application the Here the Auto Sequencing is implemented by way of tables operating at two different levels. One at the upper level, called Task Master has a one to one correspondence with the hierarchal operation sequence. This table also specifies the predecessor task, start criteria, whether start criteria is based on a single event or multiple events, and whether the task is a multi input task. This table also gives a pointer to another table which actually controls the task execution at minute level. For each operation in the hierarchal operation sequence there will be an entry in the Task Master. Every operation in the hierarchal operation sequence is unique by way of <Function> - <Layer No> - <Instance No> or (X.Y.Z). That is 'A – IV – 3' uniquely identifies the 3rd instance of function A in layer 4.

This is how the pointer is generated to the lower level table which actually controls the task execution. All tables other than Task Master, operating at the next lower level are called task control. There is no uniform structure for these tables. Structure of the table depends on the operation (task or function) to be performed. Structure will be uniform across the same type of operation (irrespective of layer and instance). Structure will be copied to workspace with the pointer X.Y.Z and populated with relevant entries. Execution of the main task – The task of telemetry data processing – is controlled by the task master. Each task has an entry (row) in this table. For each such task that is in wait mode, the scheduler opens the task control table and verifies if any start criteria is satisfied so that the task could begin. This is progression. In the same way if the scheduler finds that a Task Control table is completely executed, then it sets the status of the entry in Task Master as ‘Success’. This is retrogression. Though these progressions and retrogressions, the entire rows of Task Master will be executed, whereby, the task of telemetry data processing is accomplished

VI. TEST RESULTS

For test run we choose and loads a sample data from the user defined directory path at location “E:\Arjun\test data\12bit\12bit.mdb”. The select telemetry data files are copied on the newly created workspace “E:\TM DataPro\Sample” in separate folder. It is auto-checked with a **maintask()** function program called when the Taskmanager window is clicked with a **Start** button. The output data for 1st operation selected is logged at different file, named as per naming convention.(.dat for data, with GS_Id, Link Id as file pointer)

TmAutolog.txt

```
***** New Process begins!***** 02/03/2011 12:40:01 PM
***** Main task pass number=1 02/03/2011 12:40:01 PM
AA_01_01_IG1 , 12:40:01 PM , 5 , 0: 40: 6 , S1P11B.DAT
AA_01_01_IG2 , 12:40:01 PM , 5 , 0: 40: 6 , S1P12B.DAT
AA_01_01_IG5 , 12:40:01 PM , 5 , 0: 40: 6 , PB1B.DAT
AA_01_01_IG6 , 12:40:01 PM , 5 , 0: 40: 6 , PB2B.DAT
From output folder of JJ_05_01 the following files can be
retrieved:
***** Main task pass number=8 02/03/2011 12:44:45 PM
S1S2PB1B_PRE.ngr, Shar1, Shar2, Portblair- 1B redundant
pre-processed (navigation guidance)
S1S2PB1B_PRE.np, Shar1, Shar2, Portblair- 1B primary
channel processed (navigation guidance)
S1S22B_PRE.ngp
S1S22B_PRE.np
S1S22B_PRE.nr
```

Here, TM DataPro application implements integrated program execution, so the pre-process function in the workspace folder “E:\TM DataPro\Sample” is executed in the same process function and the output be used to store in database table “AA_01_01” and the input criteria be

set to “Wait” condition. The process “BB_01_01” gets filled with required details as and when process AA_01_01 is checked by the program scheduler loop if all the rows of AA_01_01 are set to True,(*indicated by flag bits*). In this way the sequential processing can be achieved and each process wise tables can be given feed backed to the loop of Tasks scheduler.

CONCLUSION

While real-time space mission telemetry processing scenario poses challenges to be overcome in future, by the present development a software prototype, basic level tasks for off-line processing including addition of GS data, preprocess initiation and merging of telemetry data were successfully implemented. Further development by automation can augment the processing of the launch vehicle telemetry data and improves the mission critical systems used in launch vehicle operations.

ACKNOWLEDGMENT

I express invaluable gratitude to my mentor, internal supervising guide Dr. Santhosh Kumar G, Assistant Professor, Department of Computer Science. Also thanks to all members of faculties and staff at the Dept of Computer Science, CUSAT. In VSSC, immense gratitude and thanks to my external supervising guide Shri.Mohammed Shahim MI, Scientist/Engineer-SF, Section Head, for providing facilities and being so considerate and helpful. Also thanks to Mr. Narayanan Namboodiri MN, Sci/Engineer-SG. Special thanks to Dr. Purushothaman Nair R, Sci/Engineer-SE, and also to Shri. Jaimon KC, Sci/Engineer-SE for help and assistance throughout. Also thanks to my friends Mr. Prabhat Kumar Dubey, Sci/Engineer-SC, Mr.Premchand MP, Mr. Anil, Sci/Engineer-SC, and all members in MSSG/VSSC.

REFERENCES

- [1] A. Li, E. Hu and C. Olson, “Continuing to Improve Telemetry Data Accessibility”, in 3rd IEEE International Conference on Space Mission Challenges, IEEE 2009.
- [2] Kenny C. Gross, Scott McMaster, Adam Porter, Aleksey Urmanov, Lawrence G. Votta, “Towards Dependability in Everyday Software Using Software Telemetry”, in 3rd International Workshop on Engineering of Autonomic & Autonomous Systems, IEEE 2006.
- [3] Philip M. Johnson, Hongbing Kou, Michael Paulding, Qin Zhang, Aaron Kagawa, and Takuya Yamashita, “Improving Software Development Management Through Software Project Telemetry”, in IEEE Software, IEEE Computer Society, IEEE 2005
- [4] Mouna Samaan, Stephen Cook, “The Generation of Telemetry Frame formats in User-Friendly Environment”, in Proc. of AUTOTESTCON, 97. Autotestcon, IEEE 1997

***The paper is part of the original research work carried out for the Master’s thesis project during period of June 2010 to May 2011 at MSSG/VSSC (ISRO), Thiruvananthapuram,Kerala,India.Pin-695022**