Awakening to Reality

Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 100 (2016) 43504-43508

Elixir
ISSN: 2229-712X

Artificial Neural Trend Analysis Package of Telemetry Tracking and Command Network for Linux Workstation System

K.Karthikeyan and R.Mala
Department of Computer Science, Marudupandiyar College, Thanjavur.

ARTICLE INFO

Article history:
Received 3 October 2016;
Received in revised form:
3 November 2016;
Accepted: 10 November 2016;

Keywords
Telemetry Tracking,
Command Network,
Satellite Subsystems,
Space Craft,
LINUX Script,
Performance Analysis,
Navigation.

ABSTRACT

A satellite has many supporting subsystems like sensors, solar panels, thrusters, thermal control system, reaction wheels, batteries, etc. These subsystems are integral and functional parts of a satellite that serve different purposes. For example, batteries supply power during eclipse time, thermal control system maintains an optimum temperature; thrusters are used to maintain satellite attitude and orbit. The subsystem must be in good condition and help the satellite to perform mission operation. Therefore the subsystems are analyzed to determine whether they are in good condition or not or showing a tendency to degrade. To do this analysis, archive data files are used, these files contains actual information about the subsystems. The subsystems are represented in the data files as parameters. The checking manifestations of the satellite health status are carried out to yield exact result. Based on these results further steps that facilitate the continuity of the satellite activities are implemented. Thus the spacecraft analysis is a vital activity and one of the objectives of *"Telemetry Tracking and Command Network"* for space research organizations among various other activities out to provide uninterrupted space services. The purpose of this research is to develop application software for analyzing spacecraft data and provide trend analysis package to benefit spacecraft subsystem specialists and designers at various centers of space research connected by Space NET.

1. Introduction

The satellite is only one component in the larger system required to provide the service for which it was built and launched. There are three specific segments that must work together for the larger overall system to provide communication, navigation, or any other service of interest (Army Space Reference Text 1993[1])

- The space segment consisting of all satellites required for the application and the launch vehicles used to deliver those satellites to orbit.
- The command segment consisting of all the personnel, facilities, and equipment that are used to monitor and control all the assets in space.
- The user segment consisting of all the individuals and groups who use and benefit from the data and services provided by the payloads of the satellite.

Onboard each satellite, the connection between the spacecraft and the command segment is achieved by the Telemetry, Tracking, and Control (TT&C) subsystem. As can be deduced from its name, this subsystem has three specific tasks that must be performed to ensure the ability of the satellite to successfully achieve any application:

a. Telemetry. The collection of information on the health and status of the entire satellite and its subsystems and the transmission of this data to the command segment on the ground. This requires not only a telemetry system on the spacecraft but also for a global network of ground stations around the world to collect the data, unless, of course, the application satellite network includes inter-satellite links that are capable of relaying the data to a central collection point.

b. Tracking. The act of locating and following the satellites to allow the command segment to know where the satellite is and where it is going. Again this requires a ranging system on the spacecraft and a data collection network on the ground that allows this ranging and tracking function to work.[3].

c. Control. The reception and processing of commands to allow the continuing operation of the satellite in order to provide the service of interest. Again a ground system is required.

These tasks must be performed for both of the major components of the satellite: the payload and the spacecraft bus. Earlier chapters detailed the principles and technologies related to specific payloads and applications and the previous chapter introduced the key subsystems present in any spacecraft bus required to support the payload (structure, attitude control, power, thermal controls, etc.).

The 'Spacecraft subsystem Health Analysis and Report Package (SHARP) on SpaceNET' is to provide an environment to carryout automatic spacecraft subsystem parameter health analysis, generate summary report and manage the data of spacecraft subsystem parameters through SpaceNET which is a private network owned by space research organizations.

1.1 Access data module

Accesses the archived data using offline SCHEMACS utility for identified parameters.

1.2 Analysis module

Which contains functions of various subsystem modules and perform statistical and subsystem analysis.

Tele:
E-mail address: kirithicraj@gmail.com

1.3 Report generate module

Takes the outputs of analysis module as input and generates the output in a report format.

1.4 Remote data updating module

Updates the spacecraft subsystem health database.

1.5 Remote data access and report generation module

Access and manage the spacecraft health data maintained at SpaceNET server.

The tables, plots, charts for the requested time segment will be made available to the subsystem designers.

1.6 Mimic display

Mimic display of spacecraft subsystems will be made available to the users.Subsystem data for the requested time/date segment will be made available to the subsystem designers.

This Space NET application allows two types of users

1. Administrator.
2. Spacecraft subsystem specialists and designers.

In order to understand the functionality of the existing system, we will go through the whole process how the satellite revolves around the earth and how it will send the stored data to the ground stations.

Normally the satellites which revolves around the earth are called as LEO-Satellites, the satellite can only send the stored raw data to the ground station if that particular ground station is visible to that satellite, in order to have a continues space service, space research organizations has made five main ground stations around the globe.

This data that are onboard a spacecraft are dumped by it to the visible ground stations in real time at a very fast rate, these data are used as archival data for further subsystem analysis.

The data which are stored at various ground stations will be transferred to the File Server through communication network(SpaceNET), this File Server is situated in Spacecraft Control Center – ISTRAC. The data from the File Server is sent to various shared terminals where each terminal is meant to store data for a specific satellite.

For analyzing subsystems, an archive data containing all its parameters is used. These files are handled manually to check each parameter value in a sequential manner and to perform calculations on them. If the calculated values are erroneous then the entire file must be read line by line to spot the error.

- Manual intervention is time consuming and leads to duplication of works.
- Subsystem specialists are using utility files to retrieve the archive data files used for analyzing subsystems and the obtained is unfiltered data.
- Specialists are developing their own utilities to filter the data and analyze.

This system is designed to overcome the drawbacks of existing system and provides standard environment for subsystem analysis. The proposed system facilitates the formatting, filtering and utilization of filtered data files for generating the report package.

The proposed system is having two parts- first part (**terminal base**) is having subsystem analysis module on LINUX workstation; second part (**application base**) is a web application based trend analysis package.

Terminal Base

Programming Languages: LINUX Shell Scripts,
C-Language.

Platform: Digital LINUX Workstation.

Application Base:

Programming Languages: JAVA(Jsp, Servlets, Java Scripts, Jdbc, Applets).

Platform: Windows System (98/ NT/ 2000/7).

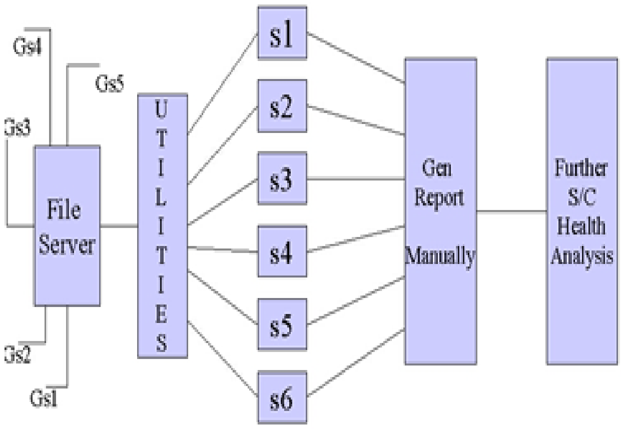

Figure 1. Structure of Terminal Base.

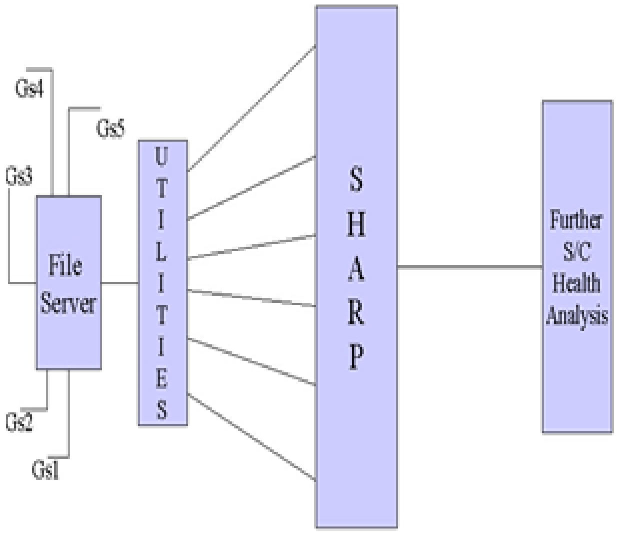

Figure 2. Structure of Application Base Model.

The system will have facility to view subsystem trend analysis report as per user request on Spacenet, the tables, plots, charts, for the requested time segment will be made available to the subsystem designers.

2. LINUX and JAVA in "SHARP on SpaceNET"

The main server, which gets the data from various Ground Stations and Satellites, is a Digital LINUX Workstation. The shell scripts under **LINUX** system helps us to fetch the real time data, which are received, from various Ground Stations, after fetching the data, the required data should be filtered and these filtered data should be stored in a separate file. This kind of filtering the data needs the program to be very flexible with the file operations. Hence we used C-language which is a very flexible in case of programs related to file operations.

The second part of our project takes the output of first part as input and generates the output in the form of reports, here the tables, charts for the requested time segment will be made available to the subsystem designers. The only way to achieve dynamic reports is to write the Dynamic HTML(JSP/ Servlets) using JAVA. We integrated JSP with the Applets in some situations where we wanted to display the Dynamic Graphs or charts. These Dynamic Graphs will be plotted by fetching the required data from the database.

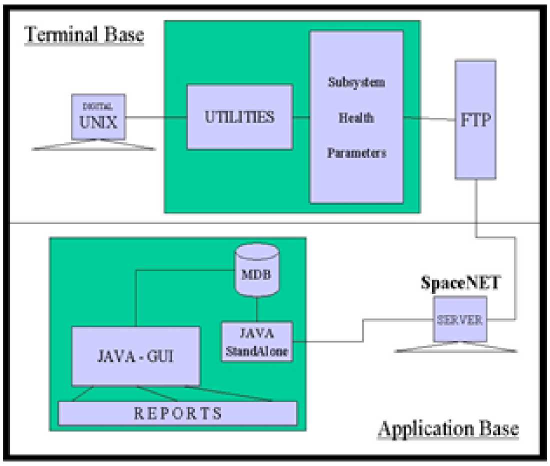

Figure 3. Structure of Terminal Base and Application Base.

2.1 System Level Module

a. Copy the schedule file and get the orbit number

Copy schedule file from the scheduling division of SCC.

b. Updating all files and Generating corresponding data files

Number of orbit accessed from schedule file, that many times the OI files are updated and that particular parameter file of the subsystems are generated.

c. Filtering data files

From the accessed file, required data is copied by filtering the original file.

d. Processing filtered files for date and time validation, limit checking and performing calculations on parameter values

Converting hours, minutes, seconds and milliseconds into seconds format and calculating the difference as local time rate(LT_Rate), epoch date, report date from current date.

e. Generating the report

Generate the report in required format.

2.2 Application Level Modules

a. Login module

User authentication, login registration for administrator only.

b. Options module

Provides options like Latest, Required Date, Monthly, Yearly, Date Segment.

c. Latest module

Select satellite/satellites with parameters according to Latest Date.

d. Required date module

Select satellite/satellites with parameters according to Selected Date.

e. Monthly module

Select satellite/satellites with parameters according to particular Month and Year.

f. Yearly module

Select satellite/satellites with parameters according to particular Year.

g. Date segment module

Select satellite/satellites with parameters according to selected Start Date and End Date.

3. Table of Structure

Table 1. Login Table

Field Name	Data Type	Field Size	Description
Uid	Varchar	20	User-ID
Pwd	Varchar	10	Password
Desg	Varchar	20	Designation

Table 2. Ground Track Shift for IRS-1C

Field Name	Description
Dd	Day
Mm	Month
Yyyy	Year
Gts	Ground Track Shift
Gts_rate	GTS_Rate
Lthh	LocalTime_Hours
Ltmin	LocalTime_Mins
Ltsec	LocalTime_Seconds
Lt_rate	LocalTime_Rate

Table 3.Keplarian Element for IRS-1C (kep1c)

Field Name	Description
Dd	Day
Mm	Month
Yyyy	Year
A	Semi major axis
E	Eccentricity
I	Inclination
Omega	Omega
Somega	Small_Omaga
Manomaly	Mean_Anomaly_Value

Table 4.Mean Element for IRS-1C (mean1c).

Field Name	Description
Dd	Day
Mm	Month
Yyyy	Year
A	Semi major axis
E	Eccentricity
I	Inclination
Omega	Omega
Somega	Small_Omaga
Manomaly	Mean_Anomaly_Value

Table 5. State Vector for IRS-1C (svector1c).

Field Name	Description
Dd	Day
Mm	Month
Yyyy	Year
X(km)	X – Direction
Y(km)	Y – Direction
Z(km)	Z – Direction
Xd(km/s)	X Direction Deflection
Yd(km/s)	Y Direction Deflection
Zd(km/s)	Z Direction Deflection

The Tables explained above is for only one Satellite “IRS-1C”, the Tables for Satellites IRS-1D, IRS-P3, IRS-P4, IRS-P6 and IRS-T1 contains similar fields but only difference is the Table names are combined with Satellite-ID. Since we are working on six Satellites where each Satellite contains four parameters, totally we are making use of 26 Tables whose fields are similar as in above-mentioned Table Fields.

4. Details of Parameters

The major data structures and parameters used in the Health Analysis are specified here.

Table 6. General Schedule File.

Field	Description
DATE	To store the date
SAT_ID	Satellite Identification
STN	Station Name
ORBIT	Orbit Number in which spacecraft is rotating
MAX_ELE	Maximum Elevation
LOS_TIME	Loss of signal time
AOS_TIME	Acquisition of signal time
OPERATIONS	Operations that are schedule, carried, archived and payload operations.

The detailed design document is the developer’s blueprint. It provides precise directions to software programmers about how basic control and data structures will be organized. Typically, the Detailed Design Document consists of tables and diagrams that translate the functional specifications into data structures, data flows and algorithms.

The Detailed Design Document is written before programming starts. It describes how the software will be structured, and what functionality will be included. This document forms the basis for all-future design and coding.

The system study is a process of planning for the near future. The system design can be stated as the process of planning a new system or one to replace or complement an existing system. The design specification describes the features of the system, the components, the elements of the system and the physical appearance. Certain design standards are to be considered for any new development designs. Some of the common design standards are data standards, coding standards, documentation standards, and screen standards.

Structural standards have to be considered in developing the module. The major factors include the policies in software modularization, coding structures like indentation, highlighting convention.

5. Design Strategy

5.1 Input Design

In the input design, the requirements from SRS are considered and according to that the screen layouts are designed. Input screens form the primary interface between the user and the software. The input screens are designed in such way that they have a simple, intuitive & user friendly layouts. Input validations are performed to enforce all the necessary system data constraints. The input screens are designed in such a way that if any erroneous data are submitted for processing, it is detected even during input so that corrections can be made before the storage.

Some of the purposes of input design are

- To reduce redundancies in recording data
- To control work flow
- To increase clerical accuracy.
- To allow easier checking of data.
- The Input forms designed here are:
- Different Input forms for different modules are

5.2 Output Design

Output design of the system emphasizes on preparing the reports as required by the user. Users evaluate the system using the output obtained from the system. The output is designed in the form of reports and files.

5.3 Database Design

Database design aims at storing the data and enabling one or more users to share the common data, there by eliminating redundancy and maximizing the efficiency of data processing system. Normalization techniques are used to reduce the redundancy and improve data consistency.

5.4 Verification

There are few techniques available to verify that the detailed design is consistent with the system. The focus of verification in the detailed design phase is on showing that the detailed design meets the specifications laid down in the system design.

In this review method it is very desirable to detect and remove design errors early as the cost and effort of removing them later can be considerably more than the cost of removing them at the design time.

The use of checklist is considered important for the success of the review.

The Checklist used in the review

- Does each of the modules in the system design exist in the detailed design?
- Have the exceptional conditions been handled?
- Is the design structured, and does it conform to standards?
- Is the module logic too complex?
- Are the loop termination conditions properly specified?
- Are the conditions?
- Is the nesting proper?

6. System Implementation

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal document. So that conformance of the code to its specification can be easily verified and so that debugging, testing and modification are erased. This goal can be achieved by making the source code as clear and straight forward as possible. Simplicity, clarity and elegance are the hallmarks of good programs.

The first part of software is developed in development area such as C and LINUX and second part of the software is developed in Java, Java script, jsp, java servlets, MS Access, WindowsNT. The analyst accesses the software and carryout the desired analysis using terminal basis in the first part, in the second part the analyst accesses information with in the campus of ISTRAC.

6.1. Access_system_date ()

Description: It accesses the system date and system time. For accessing system time, there is inbuilt function is there which returns a structure type of data.

Assigning:This accessed date from system is assigned to a variable, curr_date and also assigned to submit_date variable.

Generate:Using this curr_date, this function also generate report_date.

6.2. Getno_orbits_completed ()

Description:It accesses the schedule file from file server and searches for report_date.

Generate:This function generates number of orbit completed by make using report_date as a key data.

6.3. Getno_TTC_passes ()

Fetches: This function fetches the number of Telemetry and Telecommand are passed.

6.4. Get_orbital_element ()

Generate: It generates Epoch date.

6.5. Get_gtslt ()

Description: This function block fetches GTS values, gts-rate, local time and local time rate.

6.6. Getno_commands ()

Description:This function block generates number of command of DEC1 and DEC2 are issued.

Inputs: Start time and End Time.

6.7. Getsuprt_problem ()

Description: This function block is a user interface; It checks the support problems of the spacecraft, Problem details, remarks of problem.

6.8. Get_archival_status ()

Description: It checks, whether any archival is pending.

Process: If any archival pending it takes orbit no, station,
Type of data, reason for pending and action taken.

9. Results and Discussion

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The implementation of the source code was done in the development area. Once source code has been generated software must be tested to correct as many as error possible before delivery to the customer. This can be done using the software testing techniques.

This software was tested in the operational area to see if it produced the required output. The test conducted was successful in generating the health summary report. The errors detected during the test were also rectified.

In the system no error should be generated, if there is an error in the system then it should be displayed by the system as a message to the user and the error cause is to mentioned. The error occurrences are as followed.

9.1 Copying of Schedule file

We have to copy this file from Schedule division of Spacecraft Control Centre of ISTRAC, if file not found then we have to display the Error message "Schedule file not found"

9.2 Accessing spacecraft subsystem parameters file

To access the spacecraft subsystem parameters file, if those files are not there it shows error messages, if error message shows "cannot execute command", this is a system command, this is due failure in connecting network to the file server.

9.3 Date validation

If the report date is wrongly entered as the advanced date, then an error of date mismatch takes place, so the report date should be of valid day which is less than the submit date. The format of date enter should be of day-month-year. If is entered wrongly it shows "Date Format Invalid".

10. Conclusion and Future Enhancement

The SHARP project has two parts- terminal base and web application base. In terminal base application, the sharp software will be installed in the LINUX workstation server. It has a limitation, users have to give report date and spacecraft id by manually entering through keyboard, so it is command based process not a graphical user interaction. There is a much scope to overcome this command based process and an enhancement of the current command based process to graphical user interaction through by using X motif language. In the second part, currently the outcome of first part can be accessed as a text file and stored in LINUX main server, through FTP, specialist have to select this file (which is in appending of daily created record mode form) and forward it into windows workstation. So this is a one limitation, problem can be overcome by automate the file forward process. Currently this web application can be accessed within the ISTRAC campus, this can be enhance by installing this application in space net so that all over the Space Research organization can access this health report content web pages. By making the package a J2EE compatible, we can make the package a generic one, so that any other Space Research Organizations can also make use on this package.

References

[1] *Army Space Reference Text*. (US Army Space Institute, Fort Leavenworth, 1993), www.fas.org/spp/military/docops/army/ref_text/index.html

[2] W. Ecke et al., Fiber optic sensor network for spacecraft health monitoring. Measure Sci. Technol. J. 12(7), 974

[3] C.J. Keesee, *Satellite Telemetry, Tracking, and Control Subsystems*. (Massachusetts Institute of Technology, 2003), http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-851-satellite-engineering-fall-2003/lecture-notes/l20_satellitettc.pdf

[4] W. Larson, J. Wertz, *Space Mission Analysis and Design*, 3rd edn. (Microcosm Press, 1999)

[5] V. Pisacane, *Fundamentals of Space Systems*. (Oxford University Press, New York, 2005)

[6] *Glossary of INFOSEC and INFOSEC Related Terms*. Compiled by Corey D. Schou. Pocatello, Idaho: Idaho State U Simplot Decision Support Center, 2006.

[7] *Capability Maturity Model® Integration (CMMI^{SM})*. Version 1.1. CMU-SEI-2002-TR-011. ESC-TR-2002-011. Pittsburgh, Pennsylvania: Carnegie Mellon University, 2002. <http://www.sei.cmu.edu/publications/documents/02.reports/02tr011.html>

[8] Willis H. Ware, ed. *Security Controls for Computer Systems: Report of Defense Science Board Task Force on Computer Security*. 1970. Rand Report R-609-1. Reissued, Santa Monica, California: The Rand Corp., 1999. <http://www.rand.org/publications/R/R609.1/R609.1.html>

[9] *An Introduction to Computer Security—The NIST Handbook*. Federal Information Processing Standards Special Publication 800-12. Gaithersburg, Maryland: NIST, October 2011. http://csrc.nist.gov/publications/nistpubs/800-12/handbook.pdf

[10] *Handbook of Satellite Applications*, Joseph N. Pelton, Scott Madry, Sergio Camacho-Lara, 2013, pp 1067-1078: ISBN: 978-1-4419-7670-3 (Print) 978-1-4419-7671-0 (Online).