DESIGN OF COMMON TEST HARNESS SYSTEM FOR SATELLITE GROUND SEGMENT DEVELOPMENT

Seok-Bae Seo+, Su-Jin Kim, In-Hoi Koo, and Sang-Il Ahn

Korea Aerospace Research Institute
P.O. Box 113, Yuseong, Daejeon, Korea
sbseo@kari.re.kr

ABSTRACT: Because data processing systems in recent years are more complicated, main function of the data processing is divided as several sub-functions which are implemented and verified in each subsystem of the data processing system. For the verification of data processing system, many interface tests among subsystems are required and also a lot of simulation systems are demanded. This paper proposes CTHS (Common Test Harness System) for satellite ground segment development which has all of functions for interface test of the data processing system in one PC. Main functions of the CTHS software are data interface, system log generation, and system information display. For the interface test of the data processing system, all of actions of the CTHS are executed by a pre-defined operation scenario which is written by purpose of the data processing system test.

KEY WORDS: Data Processing, Interface Test, Ground Segment Development, Test Scenario

1. INTRODUCTION

Because data processing systems in recent years are more complicated, main function of the data processing is divided as several sub-functions which are implanted and verified in each subsystem of the data processing system. For the verification of data processing system, many interface tests among subsystems are required and also a lot of simulation systems are demanded.

Ground segment of satellite also has many subsystems for complicated data processing, therefore a lot of interface test are required, which is very important for the completion of ground segment system development.

Complicated processing systems can be realized as the distributed system. The distributed system is a method of computer processing in which different parts of program run simultaneously on tow or more computers that are communicates with each other over a network. The distributed system has a lot of merits, short schedules and low cost for development, but design and realization for the sharing of data and resources are very important matters. File interface is a simple solution for the distribute system realization, and which has strong point, file system is independent in platform of system.

This paper proposes CTHS (Common Test Harness System) for satellite ground segment development based on file interface.

The CTHS has all of functions for interface test of the data processing system development in one PC. Main functions of the CTHS software are data interface, system log generation, and system information display. For the interface test of the data processing system development, all of actions of the CTHS are executed by a pre-defined operation scenario written by purpose of the data processing system test.

2. CTHS SOFTWARE STRUCTURES

The CTHS can be divided as software part and hardware part. In a case of hardware part, it is possible for all of PC system to use if the PC system has more than one LAN port. Software part designed as 6 mangers; master manager, scenario manager, process manager, display manager, connection manager, and file manager. Detailed design of managers of software part is following.

2.1 Master Manager

The master manger is the supervisor of 5 managers in CTHS. All of managers are controlled by the master manger in the CTHS except the master manager.

2.2 Scenario Manager

The scenario manager manages scenarios of the CTHS, its main functions are as following.

- Scenario generation: Generated scenarios operate the CTHS. GUI of scenario manager can help to generate well made scenario.
- Scenario grammar check
- Scenario Image generation from scenario text files

2.3 Process Manager

The process manager manages all of processes in the CTHS. Its main functions are as following.

- Master process generation/control : The master process controls its sub process.
- Sub process generation/control

2.4 Display Manager

The display manager manages all of displays in the CTHS. Its main functions are as following.

- Display layout decision using the scenario file
- CTHS status display
- CTHS log display

2.5 Connection Manager

The process manager manages all of processes in the CTHS. Its main functions are as following.

- FTP connection/disconnection
- SOCKET connection/disconnection
- Connection insurance during the CTHS operation

2.6 File Manager

The file manager manages all of files and folders in the CTHS. Its main functions are as following.

- Scenario information control: Scenario information archiving in the scenario management file when scenario is started.
- Folder management
- File management

Figure 1 shows composition of managers in the CTHS software.

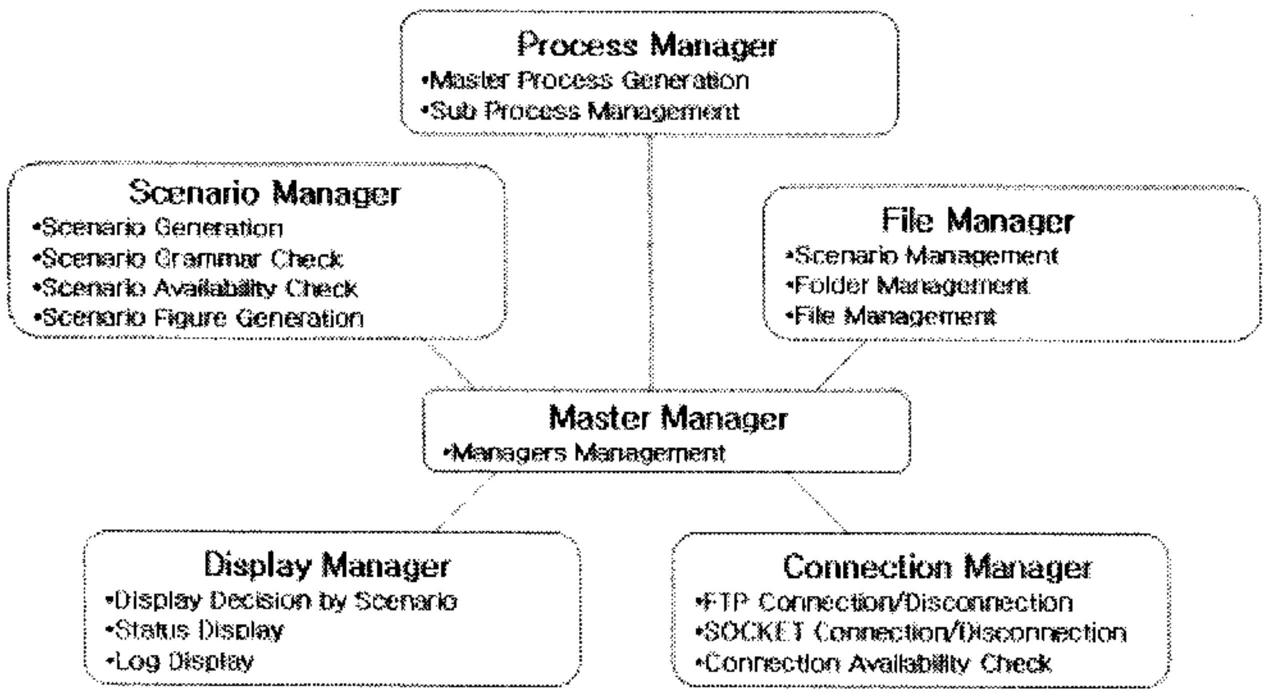


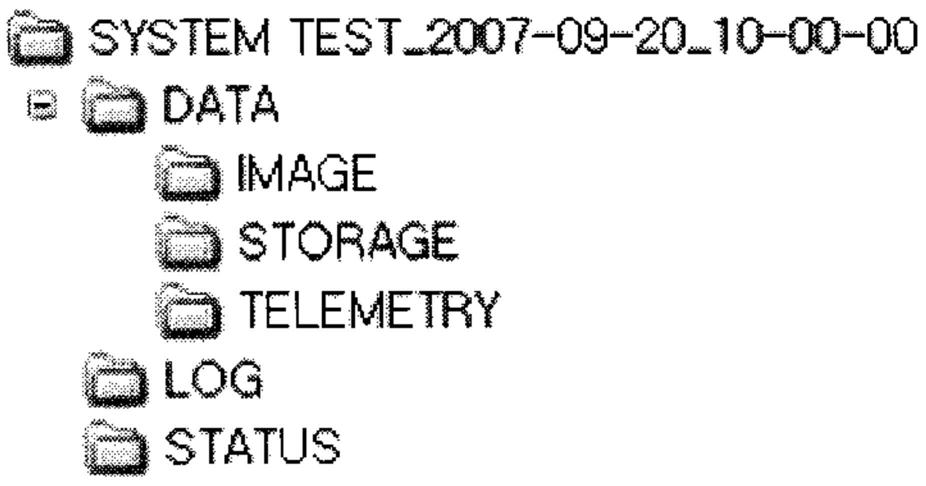
Figure 1. Composition of managers in the CTHS software

3. CTHS DIRECTORY DEFINITIONS

A root folder(SYSTE TEST_ 2007-09-20_10-00-00) of current scenario is generated at the time of start of a scenario. At the same time, DATA folder is generated under the root folder as sub folder of the root folder. The Data folder has sub folders, which contains test files for each pre-defined external systems in the scenario file. For example, Figure 2 shows that three external systems are required for a interface test in a case of without the CTHS.

4. CTHS FILE STRUECTURES

This chapter explains file designs for the CTHS; scenario file, status file, and log file, scenario image file, and scenario management file.



```
SYSTEM TEST_2007-09-20_10-00-00
  DATA
      IMAGE
      STORAGE
      TELEMETRY
  LOG
  STATUS
```

Figure 2. An example of CTHS directory design

4.1 Scenario File

The scenario file defines scenario of CTHS operations and is generated in the scenario manager. Grammar of the scenario file is as following.

- The scenario file is generated in the current folder. Where, the current folder means the folder that execute file is existed.
- The scenario file shall be generated as the Unicode txt file.
- File name is same with the scenario name.
- Extension of the scenario file is '.snf'
- Comment mark of the scenario file is '%'
- Main scenario should be express using '[' and ']'
- Separate scenario should be express using '<<' and '>>'
- Detailed action of the separated scenario should be express using '<' and '>'

Detailed grammar for the scenario files is explained in chapter 5 as an example.

4.2 Status File

The status file contains status for file input and output time. Definitions of status file contents are as following.

- The scenario file is generated in the STATUS folder that positioned under the rood directory.
- The status file is generated as the Unicode txt file.
- File name is same with the scenario name.
- Extension of the scenario file is '.stf'
- Only one status is shows in one line which has operation time, file name, departure folder name, and destination folder name.

4.3 Log File

The log file contains log for the CTHS operation. Definitions of log file contents are as following.

- The log file is generated in the LOG folder that positioned under the rood directory.

- The status file is generated as the Unicode txt file.
- File name is same with the scenario name.
- Extension of the scenario file is '.log'
- Only one log is shows in one line which has manger name, generation date, generation time, and manager operation result.

4.4 ETC Files

Except for mentioned files, scenario image file and scenario management file are used in the CTHS.

4.4.1 Scenario Image File

Image file (*.bmp, *.jpg) generated form the text scenario file to help understanding a scenario.

4.4.2 Scenario Management File

The scenario management file is used to manage data size of the CTHS. It has scenario operation data, scenario operation time, and data archiving folder name. Extension of the scenario management file is '.smf' which is a Unicode txt file. (See Figure 8)

5. EXAMPLES

For understanding of CTHS design and file format, simple example is prepared. Figure 3 shows a interface test configuration without the CTHS. We can see that three system or simulators, IMAGE, STORAGE, and TELEMETRY, are required for the interface test. If CTHS can be used in the test, the interface test configuration can be changed to very simple configuration like Figure 4.

The IMAGE generates 10 'images' per every 10 sec after receiving the 'Requrest' from the SYSTEM using the SOCKET protocol. The TELEMETRY generates 3 'telemetries' per every 20 sec from 10:00:00, and transmits that three file to SYSTEM using the FTP protocol in real-time. The SYSTEM generates 2 'products' using the images and the telemetries. Generated products in the SYSTEM are transmitted to the STORAGE for archiving finally.

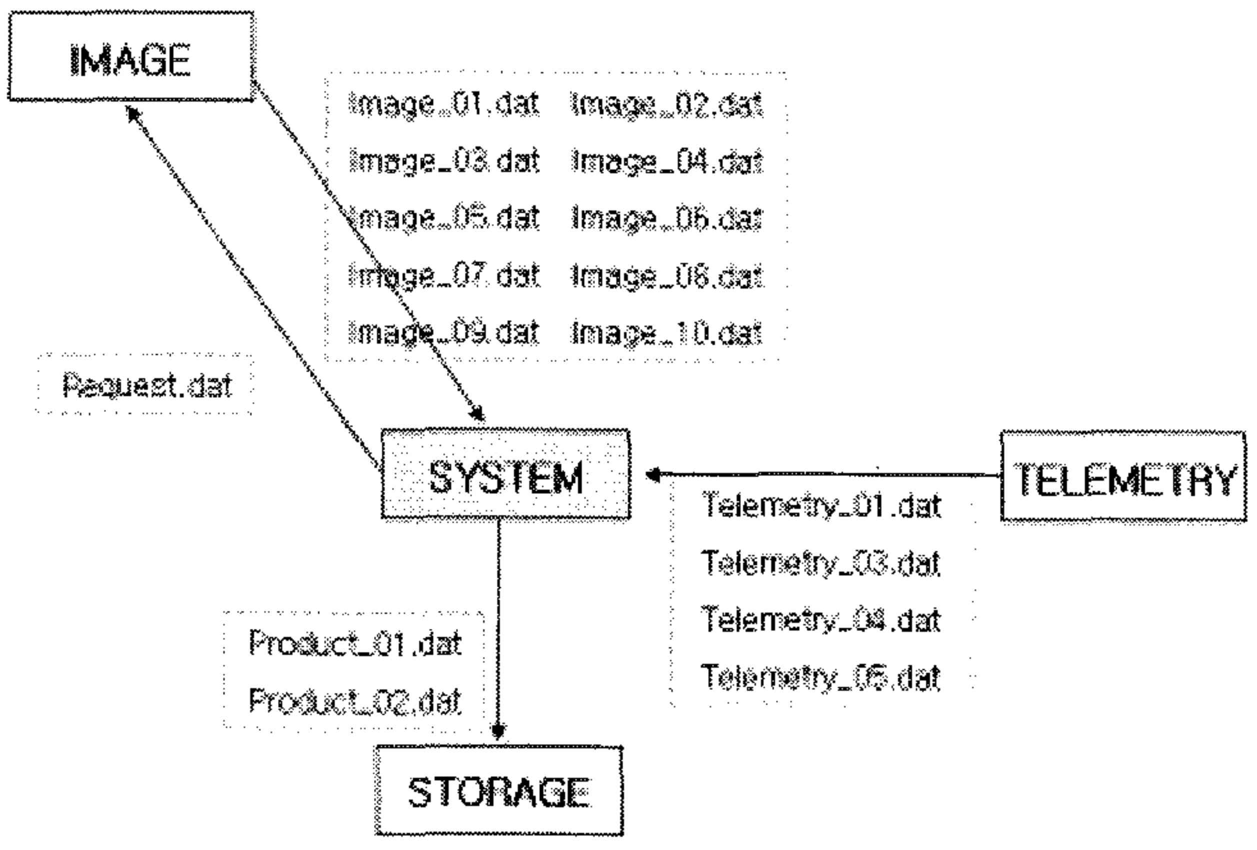


Figure 3. Interface Test without the CTHS

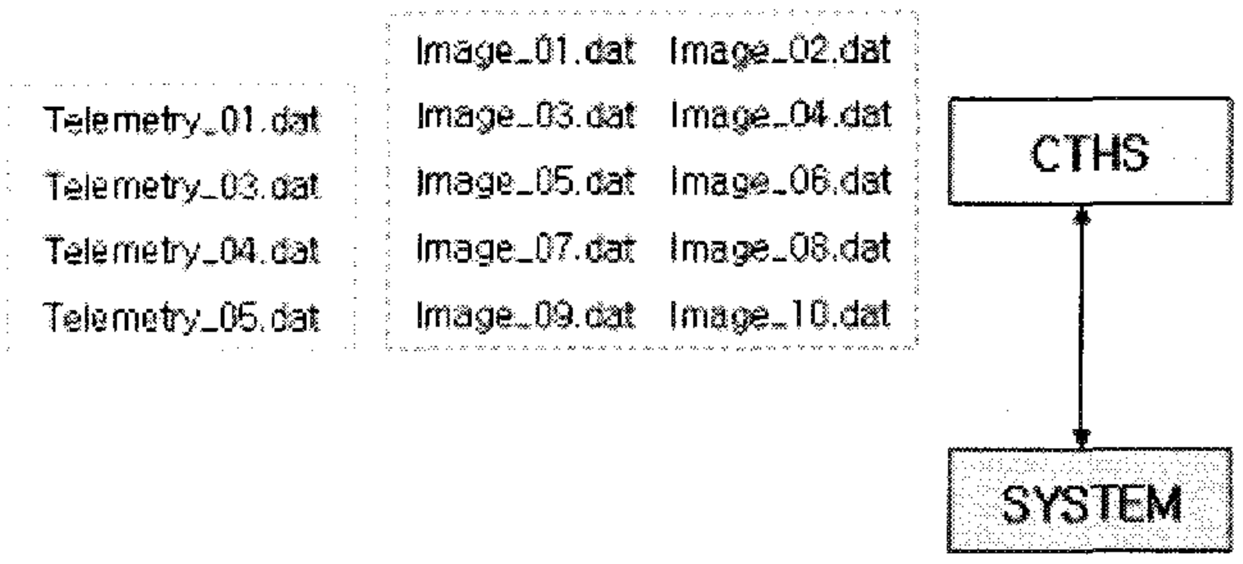


Image_01.dat Image_02.dat
Image_03.dat Image_04.dat
Image_05.dat Image_06.dat
Image_07.dat Image_08.dat
Image_09.dat Image_10.dat
Telemetry_01.dat
Telemetry_03.dat
Telemetry_04.dat
Telemetry_05.dat
CTHS
SYSTEM

Figure 4. Interface Test with the CTHS

The scenario file is very important to the CTHS operation. Figure 5 shows an example of scenario file for test that is configured like Figure 4.

```
[Title] SYSTEM TEST
[Date] 2007/09/20
[Start Time] 10:00:00
[END Time] 10:02:00
[NET Margin] 20%
[Scenario No] 2
[CurrentDir] C:\SYSTEM TEST

[DataDir] IMAGE %C:\SYSTEM TEST\DATA\IMAGE
[DataDir] TELEMETRY
          %C:\SYSTEM TEST\DATA\TELEMETRY
[DataDir] STORAGE %C:\SYSTEM TEST\DATA\STORAGE

[CTHS SOCKET] 222.222.222.222.:60000
[CTHS FTP] 222.222.222.222; user2; userpassword2;


[Scenario Start]
%------------------------------------------------------------
<<ScenarioID: IMAGE>>

<<Departure>> IMAGE

<<Destination>> C:\ImageData
<<SOCKET>> 111.111.111.111:60000

<<ProcessNo>> 3
<10:00:00, 10:01:10, 10, 8, Image_01.dat, RF07FE08,UP>
<10:01:30, 10:01:20,  1, 1, Image_09.dat>
<10:01:40, 10:01:30,  1, 1, Image_10.dat>
%------------------------------------------------------------
<<ScenarioID: TELEMETRY>>

<<Departure>> C:\TLM
<<FTP>> 333.333.333.3333; user3; userpassword3;

<<Destination>> TELEMETRY

<<ProcessNo>> 1
<10:00:00, 10:01:20, 25,-1, Telemetry_01.dat, RF11FE12>

[Scenario End]
```

Figure 5. Example of the Scenario File

The scenario file explained in Figure 5 can be divided as two parts; (first part) scenario definition/setting and (second part) actual scenario performs. The actual scenario perform part is between [Scenario Start] and [Scenario End].

In the first part, the scenario operation is prepared. Scenario title is 'SYSTEM TEST' that start at 2007/09/20 10:00:00 and finished at 10:02:00 of the same date with start time. [NET Margin] means network margin for test. For example, the connection manager gets a real network speed of the CTHS, if its value is 300Mbps and NET Margin is 20% then the connection manager check the entire file can communicate 240Mbps by this scenario. If the connection manager thinks that all of file can't be communicated by that network speed, then the connection manager informs that information to test engineer. In this case, the test engineer should change test scenario to meet network speed. After the connection manger checks that network environment has no problems, the file manger makes data directory for the test; C:\SYSTEM TEST\DATA\IMAGE @ the CTHS hard disk, c:\SYSTEM TEST\DATA\TELEMETRY @ the CTHS hard disk, c:\SYSTEM TEST\DATA\STORAGE @ the CTHS hard disk. And then, the connection manager connects SOCKET and FTP of the CTHS.

Next step explains the second part of the scenario file. There are two actual scenarios; IMAGE and TELMETRY. Where, scenario names, IMAGE and TELMETRY, are meaningless for scenario operation but it can help test engineers to understand scenario. The first actual scenario is that 10 image data are transmitted from the CTHS to SYSTEM. Among 10 image data, 8 image data are transmitted by '*<10:00:00, 10:01:10, 10, 8, Image_01.dat, RF07FE08,UP>*'. It contains start time, end time, action period, repeat time, image file name, and two repeat rules. The first repeat rule 'RF07FE08' means that from 7th character of file name to 8th character of file name can change for sequential file transmission. For example, 'image_01.dat, image_02.dat, image_03.dat, image_04.dat, image_05.dat, image_06.dat, image_07.dat, and image_08.dat' can be transmitted by '*RF07FE08*', '*10:00:00(start time)*', '*10:02:00(end time)*', '10(repeat period, 10 sec)', and '8(repeat file no)'. If value of 'repeat file no' is less than 0 then that transmission is repeated infinitely from the star time to the end time.

If 'UP' is changed to 'DOWN', sequential file transform is done like 'image_08.dat, image_07.dat, image_06.dat, image_05.dat, image_04.dat, image_03.dat, image_02.dat, and image_01.dat'. Because that the others actual scenarios can be easily interpreted by above descriptions, the explanations for remained part of the scenario file are omitted.

Figure 6 shows an example of status file which contains all of file status for the interface test with the CTHS. In the first line, 'GET' means that the CTHS receives the 'Request.dat' at IMAGE folder.

```
2007-09-20, 09:59:59, GET, Request.data, @IMAGE
2007-09-20, 10:00:00, PUT, Image_01.dat, @IMAGE
2007-09-20, 10:00:00, PUT, Telemetry_01.dat, @TELEMETRY
2007-09-20, 10:00:10, PUT, Image_02.dat, @IMAGE
2007-09-20, 10:00:20, PUT, Image_03.dat, @IMAGE
2007-09-20, 10:00:25, PUT, Telemetry_02.dat, @TELEMETRY
2007-09-20, 10:00:30, PUT, Image_04.dat, @IMAGE
2007-09-20, 10:00:40, PUT, Image_05.dat, @IMAGE
2007-09-20, 10:00:50, PUT, Image_06.dat, @IMAGE
2007-09-20, 10:00:50, PUT, Telemetry_03.dat, @TELEMETRY
2007-09-20, 10:00:55, GET, Product_01.dat, @STORAGE
2007-09-20, 10:01:00, PUT, Image_07.dat, @IMAGE
2007-09-20, 10:01:10, PUT, Image_08.dat, @IMAGE
2007-09-20, 10:01:15, PUT, Telemetry_04.dat, @TELEMETRY
2007-09-20, 10:01:20, PUT, Image_09.dat, @IMAGE
2007-09-20, 10:01:30, PUT, Image_10.dat, @IMAGE
2007-09-20, 10:01:45, GET, Product_02.dat, @STORAGE
```

Figure 6. Example of the Status File

Figure 7 shows an example of log file that contains all of log for the CTHS operation. For the better test, time sync and socket connection should be prepared before the defined test time 10:00:00. And we can see that after all of interface test are finished the connection manager disconnects all of connection in the CTHS.

```
2007-09-20, 09:59:00, Sucess, Time Sync
2007-09-20, 09:59:05, Sucess, SOCKET Connected, 222.222.222.222
2007-09-20, 09:59:11, Sucess, SOCKET Connected, 111.111.111.111
2007-09-20, 09:59:12, Sucess, FTP Connection, 222.222.222.222
2007-09-20, 09:59:14, Sucess, FTP Connection, 333.333.333.333
2007-09-20, 09:59:15, Sucess, Network Ready
2007-09-20, 10:02:00, Sucess, SOCKET Disconnected
2007-09-20, 10:02:01, Sucess, FTP Disconnected
```

Figure 7. Example of the Log File

Figure 8 shows an example of scenario management file (Scenario_History.txt) which contains start date, star time, scenario name, and operation direction. One text line is added in the established scenario management file for a scenario execution.

```
2007-09-20,   10:00:00,   SYSTEM   TEST,C:\SYSTEM   TEST_2007-09-
20_10-00-00
```

Figure 8. Example of the Scenario Management File

6. CONCLUSIONS

In this paper, the CTHS is designed for interface tests of satellite ground segment development. After GUI design of the CTHS is finished, the CTHS can be realized and then will be used in interface tests of the satellite ground segment development.

Reference

[1] http://en.wikipedia.org/wiki/Distributed_system