Development and Testing of Satellite Operation System for Korea Multipurpose Satellite-I

Hee-Sook Mo[a)], Ho-Jin Lee, and Seong-Pal Lee

The Satellite Operation System (SOS) has been developed for a low earth orbiting remote sensing satellite, Korea Multipurpose Satellite-I, to monitor and control the spacecraft as well as to perform the mission operation. SOS was designed to operate on UNIX in the HP workstations. In the design of SOS, flexibility, reliability, expandability and interoperability were the main objectives. In order to achieve these objectives, a CASE tool, a database management system, consultative committee for space data systems recommendation, and a real-time distributed processing middle-ware have been integrated into the system. A database driven structure was adopted as the baseline architecture for a generic machine-independent, mission specific database. Also a logical address based inter-process communication scheme was introduced for a distributed allocation of the network resources. Specifically, a hot-standby redundancy scheme was highlighted in the design seeking for higher system reliability and uninterrupted service required in a real-time fashion during the satellite passes. Through various tests, SOS had been verified its functional, performance, and interface requirements. Design, implementation, and testing of the SOS for KOMPSAT-I is presented in this paper.

Manuscript received November 1, 1999; revised January 25, 2000.
a) Electronic mail: hsmo@etri.re.kr,

I. INTRODUCTION

Korea Multipurpose Satellite-I (KOMPSAT-I) was launched in December, 1999. The main mission of KOMPSAT-I is to perform cartography of Korean peninsula during the design life of three years. Other missions include taking large-scale multi-spectral images of the ocean, measuring the ion layer, and detecting high energy particles. KOMPSAT-I is designed to operate at an altitude of 685.13 km, sun synchronous orbit, and a local time of ascending node of 10:50 am. KOMPSAT-I passes over the KOMPSAT-I ground station (KGS) in two pass sequences. Each sequence is made up of two or three S-band contacts and one or two X-band contacts. For the monitoring and control purposes of KOMPSAT-I, a satellite ground facility was built in Taejon, Korea. This ground control station for a remote sensing satellite has been designed and implemented by Korean engineers for the first time in Korea. The KGS is comprised of Image Reception and Processing Element (IRPE) and Mission Control Element (MCE). MCE monitors and analyzes the satellite, plans the mission, and controls the satellite, whereas IRPE is responsible for collecting and processing the image data from KOMPSAT-I [1].

MCE is made up of four systems; the Tracking, Telemetry, and Command (TTC) System, the Satellite Operation System (SOS), the Satellite Simulator System (SIM), and the Mission Analysis and Planning System (MAPS). Figure 1 shows the overall configuration of the KOMPSAT-I MCE. The design of the KOMPSAT-I MAPS has been discussed in [2]. The design of an advanced Real-Time Satellite Simulator for KOREASAT has been discussed in [3].

SOS monitors the state of the satellite by receiving, processing, and displaying satellite telemetry. SOS also creates and transmits commands to the satellite. SOS can be said the essential core of the

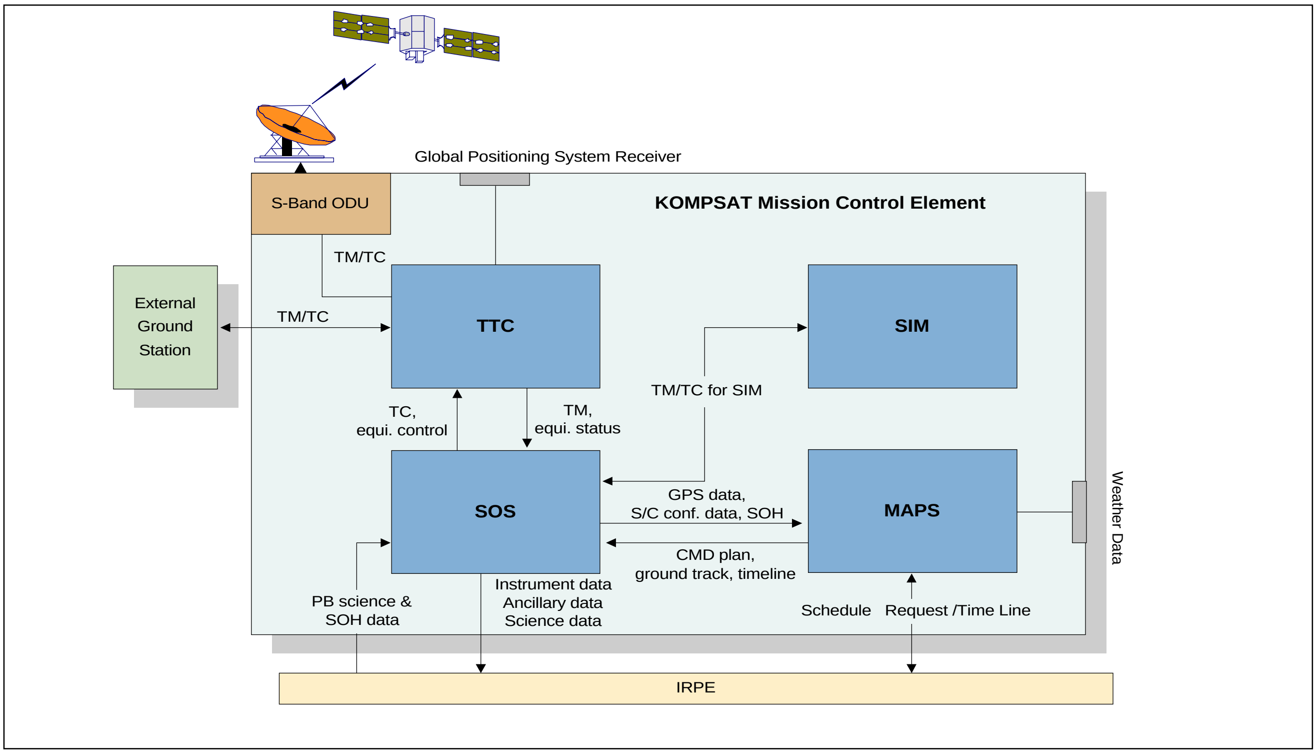

Fig. 1. The MCE overall configuration.

ground control system in the sense that it carries out the processing of actual monitoring and controlling the satellite. The major functions performed by the KOMPSAT-I SOS include

- Extraction of satellite state of health (SOH) data in real-time from telemetry data received from TTC or external ground station and display them for monitoring of KOMPSAT-I,
- Sending telecommands to TTC for implementing the planned mission operation on KOMPSAT-I,
- Storage of real-time and playback data, reprocessing them for analysis by trending of the data,
- Monitoring the status of ground equipment of TTC,
- Preparation of telecommand procedures in order to realize the mission plan generated by MAPS, and
- Distribution of the SOH data, space instrument data and mission data to MAPS and IRPE for mission planning.

SOS should meet the requirements of KOMPSAT-I mission operations during the relatively short ground contact time of the satellite. This emphasizes the real-time performance requirements of SOS.

In Section II, the SOS configuration and design concept are described, followed by description of each block of the SOS to some extent. Section III describes the integration and implementation of SOS. In Section IV, various levels of tests and corresponding results for SOS are presented.

II. DESIGN CONCEPT AND SOS CONFIGURATION

1. Design Concept

During the SOS design and development phase, TeamWork/SD was used for the structured analysis and design. The work method applied through design has been resorted to a worldwide aerospace practice used for many space qualified product programs. Interested readers can refer to [4].

As one of important requirements in the SOS design, it should have a real-time processing capability for monitoring and controlling the satellite in any conditions during a relatively short contact time of KOMPSAT-I. Therefore, the primary emphasis of the SOS design was put on reliability, flexibility, and expandability. For a higher reliability of real-time processing software, four approaches in summary have been taken. One of them turns out a redundancy scheme adoption as used most widely. The real-time data processing was designed in such a duplicated way that backup processes could instantly take roles of primary ones in case of failure. This feature enables an uninterrupted real-time operation and protects system from any loss of data packet due to processing failures. The redundancy features will be described separately in some detail. The second method was a robust design of input data processing and user interface. In other words, every possible error condition check was designed for critical modules so that any erroneous output shouldn't give any further impact on the system otherwise put into any vulnerable situation. Since system may also have erroneous external or user input or operational mishap through user interface, data limit or range checks were inserted at appropriate points along with double confirmation of user clicks as necessary. Of course the testing has been repeatedly conducted for the system robustness stimulated by various error conditions raised from either telemetry input or GUI input. Redundant scheme was also tested vastly to verify instantaneous switching, continuous operation, and data integrity. The third consideration was an embedding of a well-proven interprocess communication (IPC) support package into the system. A mature technology in message passing can give a higher reliability and relieve developers of a low level testing and debugging as well. For this sake a commercial-off-the-shelf (COTS) product was selected as a backbone for process interactions. A distributed process structuring was also one of the approaches to enable distribution of functions and workloads for a higher reliability.

Flexibility provides the ability to "tailor" or "modify" a system to achieve replacement or modification or upgrade of end user applications for meeting mission specific needs. For a reason of cost saving or operational requirement, by virtue of distributed structure and software modularity, the system can be reconfigured for a smaller implementation on fewer platforms. Conversely the system could be reallocated or distributed over network to accommodate different operational environments or missions. In short, the system has flexibility to be reconfigured or utilized easily for another applications without significant change. In addition, satellite specific information has been separately populated as database. This gives the system more flexibility to be migrated for another mission simply by changing the mission specific database.

Expandability is obviously the ability to expand itself to accommodate the changes in satellite platforms, mission requirements, system responsiveness, and operational requirement [5]. The capacity can be also increased linearly by adding additional modules. User interface or terminals can be added additionally as required in a distributed fashion as long as the total workloads are allowed. Software design techniques like distributed software structure, modularity, and server-client model enabled the capability of expansions. A typical example is an expansion for Launch and Early Orbit Phase (LEOP) telemetry displays. Contrast to 3 or 4 displays required for normal operation, about 30 or 40 displays are necessary in this phase because of overwhelming telemetry monitoring for SOH checkout and performance evaluation. During 2 months of LEOP, the system can meet these requirements simply by adding telemetry display clients as many as required to the existing system.

In order to achieve these objectives, a COTS middle-ware called 'RTworks' was adapted for design. RTworks consists of a family of software products for building high-performance client/server applications that manage time-critical data applications for monitoring, analysis, display, logging, and control of complex systems [6].

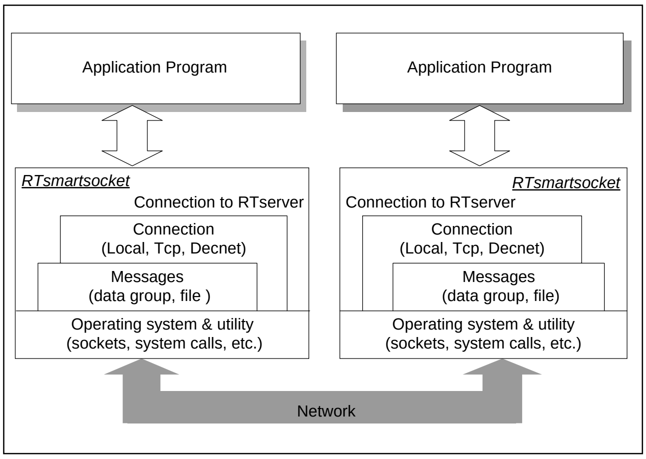

Fig. 2. IPC mechanism of SOS.

A. IPC Features

This distribution layer concept introduces a middle-ware 'RTsmartsocket' is chosen to implement the mechanism for passing data between applications. With this, a logical address based IPC scheme can be implemented to give a freedom to allocate or map process over the network resources. SOS uses the logical address of message for IPC. A message is a structured packet of information sent between processes. A message is broadcasted using logical datagroup from one process to another particular process or to a number of other processes. This feature provides the advantage of an easy expansion and restructuring of the system independent of physical location or platform type. Figure 2 shows the mechanism of IPC. A client/server application distributed processing software architecture relies on the existence of message passing software that provides a logical functional separation.

B. Redundancy Features

Redundancy has been an important factor in design for enhancing the system reliability. With the RTworks server-client model, SOS implements two types of redundancy scheme, workstation level and process level redundancy. In other words, SOS can switch from failed resources to newly activated ones in the unit of workstation or a process over network. Since the telemetry and telecommand processing is most critical and surely secured for safe operations, redundancy scheme is applied to these processes. Of the four workstations assigned for SOS, two work-

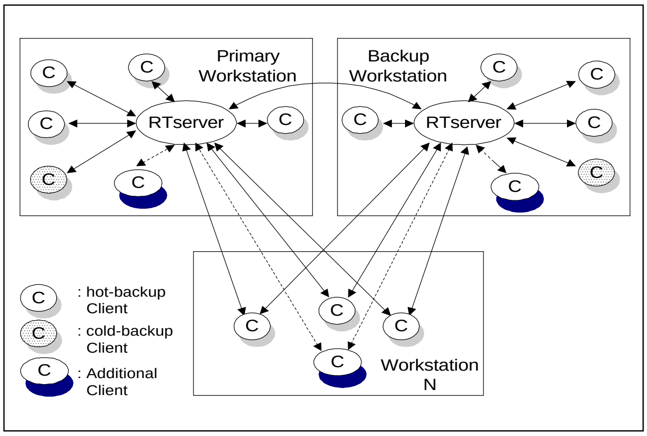

Fig. 3. Redundancy configuration of SOS.

stations are configured to work in this scheme where telemetry and telecommand processing processes are installed in a duplicated way. Figure 3 shows the redundancy configuration of SOS. Two RTserver processes reside one on each workstation respectively and are directly linked together for communication and sanity monitoring of each other. Each local client process is connected to its server as well as its clone located in the other workstation through RTserver communication channel. These two workstations are connected to the rest of the workstations via network in a cross-coupled way that the client processes are configured to access external client-pairs through two RTservers. Each RTworks process uses a startup command file for its initialization. At initialization, one of the RTservers is assigned as primary and all the local clients attached to this are activated as primary clients. At the same time the other one is configured as backup server automatically. When the primary server fails, the backup server detects this and the role is switched over. In spite of this switchover, the client process's primary ownership doesn't change unless the client itself fails. On the other hand, if one client in the primary workstation fails, its clone process in the other workstation takes over the active role on a client basis.

SOS has adapted two operation modes in redundancy; hot backup and cold backup. Since the clients can be activated automatically or by external trigger, one of two modes can be assigned. Note that RTservers always start the processing at the same time and run concurrently but only the primary one can communicate with clients. The hot backup refers to an immediate switch over done automatically on process basis. The hot backup processes are usually associated with telemetry and telecommand that has real-time interface with the satellite. On the contrary, the cold backup mode can work in near real-time by external command for some configured processes. This mode of backup is used in ground equipment control and graphical user interface. The failed processes can be rebooted by external command and configured as backup mode. Even though it can be reconfigured as active one, it is generally not necessary for LEO satellites since the pass time lasts up to 15 minutes at most and there lefts little chance of another subsequent failure. Enough time will be available to reconfigure whole processes or workstations to initial configuration by rebooting the system, during inter-pass gaps.

C. Database Management System (DBMS)

To achieve flexibility with respect to replacing or upgrading an application, a database driven structure was adopted as the baseline architecture in order to formulate a generic machine independent, mission specific database. This approach allows the generic process engine to be used from one program to another and enhances the expandability by making provision to add or replace the mission specific parts. The SOS application program was designed to access a database to get satellite-specific data through network SQL primitives. Of course satellite-dependent processing was minimized and separately implemented. The DBMS provides and maintains data relationships required for intelligent or autonomous functional implementation, as well as database security and integrity. The databases populate the satellite parameters and characteristics, the spacecraft operational information like alarm limit and display configuration, telemetry and telecommand characteristics data, and some relations featuring an automatic detection of a failure and telecommand verification. The telemetry state-of-health data of KOMPSAT-I is also managed by the DBMS for concurrent access from multiple users.

D. Consultative Committee for Space Data Systems (CCSDS) Implementation

One of the features in designing the KOMPSAT-I SOS is an implementation of CCSDS format in telemetry and telecommand processing. As currently most of the ground control systems take this as a standard in packetizing. Although it is a recommendation, this implementation permits inter-operability of SOS in telemetry reception and telecommand transmission with international ground stations or global control network [7]. This can support KOMPSAT-I by several external ground stations via bent pipe data channel especially in LEOP. Based on the KOMPSAT-I requirements in transmitting a large amount of telemetry data in variable length, bit-rate, and format, CCSDS processing implementation is distributed over SOS and TTC according to layer concept considering performance optimization. SOS takes responsibilities of upper level functions, like post processing for telemetry and of preprocessing for telecommand. Note that telecommand CCSDS specification is adopted for highly reliable command data transmission and effective transmission of large amount of command data. CCSDS packet and Advanced Orbiting Systems (AOS) options pertaining

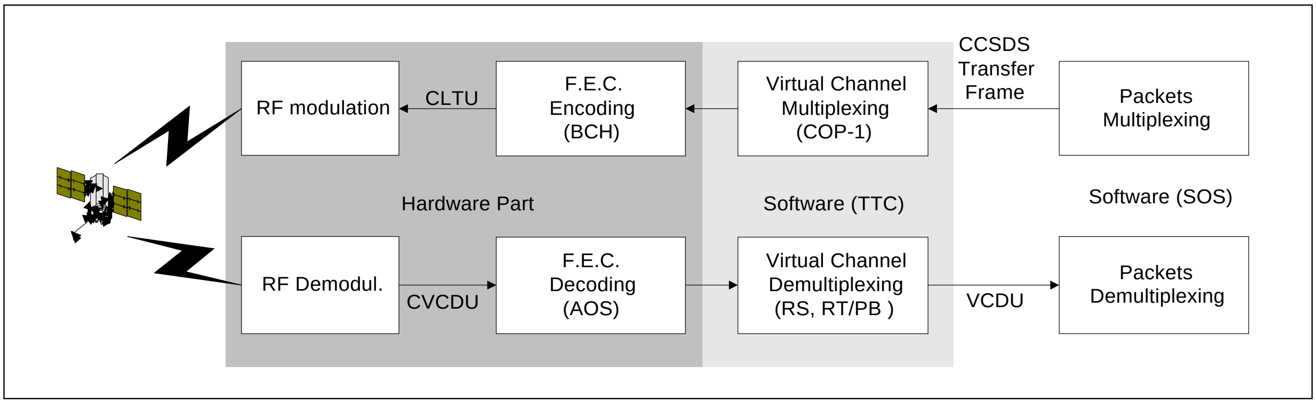

Fig. 4. CCSDS telemetry and telecommand processing implementation.

Table 1. The KOMPSAT-I telecommand and telemetry characteristics.

Characteristics	Telecommand	Telemetry
Recommendation	CCSDS Telecommand	AOS, Grade 2 service
Packet length	1 packet/1 transfer frame, variable bytes	1 packet/1minor frame
Number of Virtual Channel Data Unit	3 VCDU	5 VCDU
Coding	CCSDS COP1, CRC, CCSDS BCH (63, 56)	Reed-Solomon (255, 223)
Data Rate	2 Kbps	• 2 Kbps (real-time SOH) or • 1.5 Mbps (Playback SOH, science) • 45 Mbps, including coding

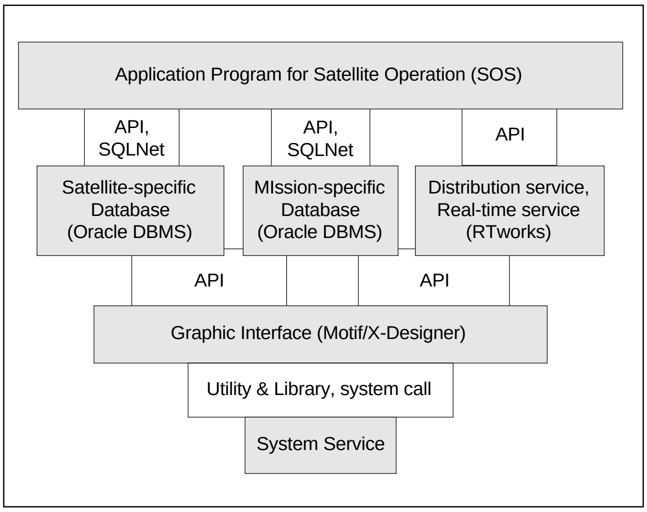

Fig. 5. SOS software architecture.

to KOMPSAT-I requirements can be found in [8], [9]. Table 1 shows the KOMPSAT-I telecommand and telemetry characteristics, whereas Fig. 4 depicts functional implementation of CCSDS between SOS and TTC over LAN.

2. Hardware and Software Configuration

SOS consists of four workstations and peripheral devices such as printers, storage devices, and LAN devices. HP 9000 J210 computers, equipped with 125 MB main memory, 2 GB hard disk drive and optical magnetic disk are used as workstations. External storage is used for archiving the long-term telemetry data and database. SOS uses a LAN-based communications network to interconnect all platforms and external interface elements for expandability. Since SOS is implemented on identical platforms and s/w can be relocated very easily, it can be reconfigured to work on fewer workstations when any workstation needs repair.

SOS utilizes commercial s/w for lower level support or system service implementation. On the HP/UX 10.20 operating system, SOS uses Motif, RT-works® and Oracle DBMS to get advantage of heritage and verification of the system. All the operator interfaces are realized by GUI to support user-friendly human-machine interface. Motif/XLIb and X-Designer are used to support the GUI facility developments. Oracle® DBMS and SQL*Net® are used for database client/servers communication. Figure 5 shows the SOS software architecture.

SOS is made up of five functional blocks; Telemetry Processing Block (TMPB), Telecommand Processing Block (TCPB), System Management Block (SMB), Data Management Block (DMB), and Ground station Control and Monitoring Block (GCMB). Telemetry Processing Block (TMPB) extracts telemetry data and displays the engineering values of the satellite status. Telecommand Processing Block (TCPB) generates, transmits, and validates the telecommands. System Management Block (SMB) performs the start-up procedure of SOS, security check of the operator and shut-down procedure of SOS Data Management Block (DMB) provides such functions

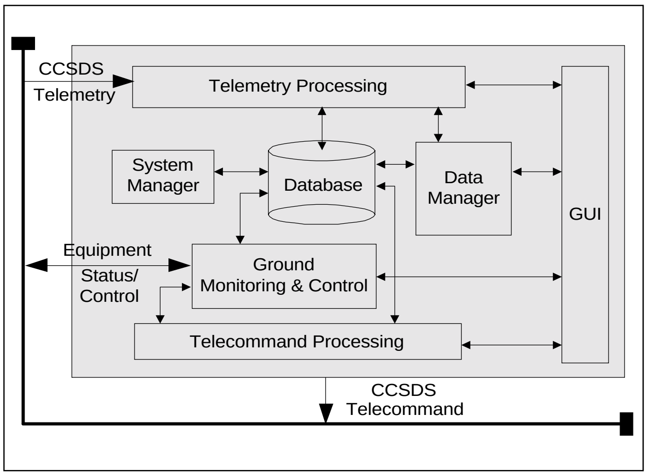

Fig. 6. Software configuration diagram of SOS.

as management of static database tables, backup and restoring the telemetry data. Ground station Control and Monitoring Block (GCMB) monitors periodically the operational status of ground TTC equipment in order to control them. Each block has the interfaces with other blocks via broadcasting of logical data group of real-time messages. Remote Procedure Call (RPC) and Network File System (NFS) are also used for file access and transfer over network. Figure 6 shows the SOS software configuration .

TMPB and TCPB are located on two workstations in a redundant configuration as described earlier. SMB is allocated to the third workstation and provides system logs as well as the capability to be "taught" about processes for monitoring and anomaly detection. The fourth computer contains the DMB and GCMB and manages all the necessary data in SOS. Context diagram, Data Flow Diagram (DFD), State Transition Diagram (STD), and System Software Diagram (SSD) of SOS, as the design results by CASE tool generated in each design phase, are described in [10].

III. SOS INTEGRATION AND IMPLEMENTATION

1. System Management Block (SMB)

As the name SMB suggests, this block manages starting, termination and event handling of SOS. SMB monitors the message stream flow and status of all the process in SOS using RTmonitor utility. It performs necessary corrective procedures if any of the processes is unable to operate.

The start-up of SOS needs a login for authentication of the user and initialization of the graphical user interface. All the event messages sent from the SOS processes are captured and displayed through the event message window and stored as files for analysis. Each process status is coded in color, red for failure and blue for normal on monitor window. It assists operators to monitor, examine, and control the SOS processes.

2. Telemetry Processing Block (TMPB)

TMPB can be divided into two parts according to the operation concept. Real-time processing part works while the satellite is in contact and non real-time processing part works during the non-contact time. The real-time SOH telemetry data acquired locally or from foreign ground station is processed, displayed, and archived in real-time during the contact time. Between the satellite passes, TMPB is activated for processing of the received on-board recorded playback data at a predefined higher rate.

TMPB decommutates and decodes telemetry using CCSDS characteristics as well as telemetry structure database for extracting state of health from data points. It distributes the state of health to the logical address datagroup over the network in order to display the telemetry in various visual forms. Predefined page script and interactive on-line selection of display items are possible for alphanumeric and graphical display. For a quick-look analysis of telemetry and monitoring of the telemetry downstream reception status, the telemetry of KOMPSAT-I is also displayed in a raw format. It provides the archiving of the processed telemetry data, raw telemetry data, and general memory dump data for off-line processing and trend analysis. Figure 7 shows the telemetry processing flow. In case of a failure in TMPB, the backup TMPB takes over immediately so that no data loss occurs.

3. TeleCommand Processing Block (TCPB)

TCPB carries out real-time processing and off-line processing

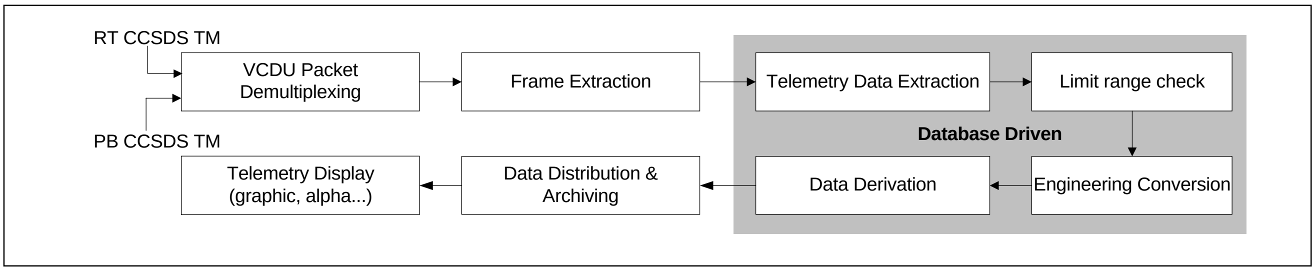

Fig. 7. Telemetry processing flow.

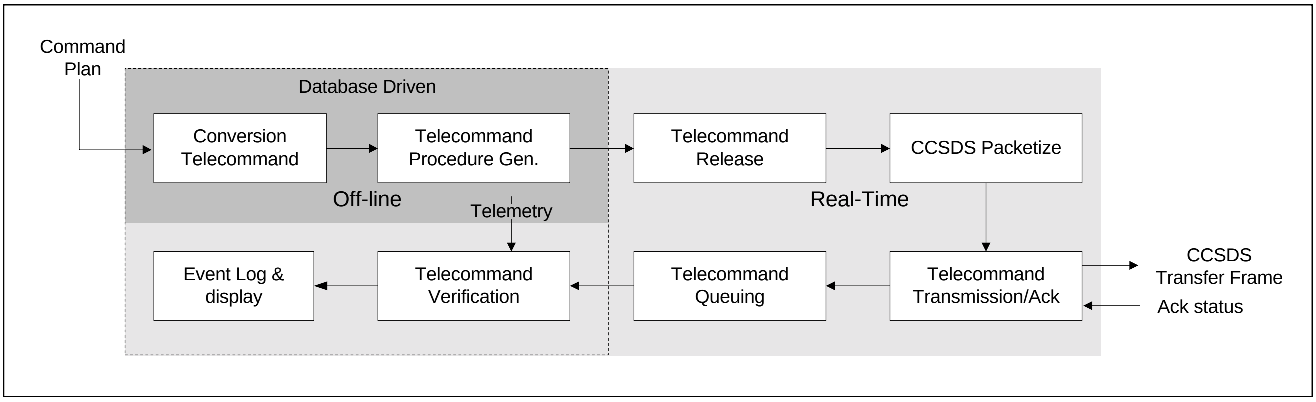

Fig. 8. Telecommand processing flow.

of telecommand as in TMPB. The real-time telecommand processing is primarily responsible for the successful transmission of the telecommands. User level mnemonic is translated into binary codes through database and formatted in on-board compatible CCSDS telecommand packet to be transmitted to TTC for further lower level encoding. Figure 8 shows the telecommand processing flow. Regarding the telecommand transmission, two kinds of verification follow, i.e., transmission verification and execution verification. Transmission status is checked through an acknowledgement of telecommand fed back from TTC. On-board execution verification is accomplished through the resulting telemetry by comparing them with the predefined values in the database. This process is linked with telemetry processing and thus it takes time proportional to the telemetry update frequency. This feature requires an autonomous scheme to some extent that execution verification result can control retransmission and proceed forward to transmit the next waiting telecommand in the queue. TCPB implements this function by using RTworks utility and relational database. Due to the short contact periods and the necessity of maximum uplink capability for LEO satellites, two types of telecommand are devised and provided: stored telecommand and real-time telecommand. The stored telecommand is stored in the satellite buffer onboard waiting for execution at later time as defined in the context. Real time telecommands are executed in satellite immediately upon reception. Additionally for the stored telecommands, reception verification is required which is performed through downlink dump telemetry check.

The telecommand procedures, which are script files listing all the required telecommands in a semantic sequence for a specific mission implementation, can be generated and saved off-line using database table before contact. Command plan for mission operation transferred from MAPS is the input for this and implemented by interpretation to lists of telecommands in the generation process. In addition, any memory load files are converted into the on-board processor specific memory load telecommands. Satellite Key Parameter Data (KPD) and GPS ephemeris/almanac load files are examples of this kind but conform to the stored telecommand processing except the argument data is organized in itself according to the predefined internal protocol like X.25.

For off-line analysis, all events for telecommand processing are logged with classification according to verification type or event class. Since TCPB should be able to transmit as many commands as required without failure during the satellite contact time, its transmission performance and verification process sanity is of importance. Redundancy scheme applied to TCPB, is already described in earlier sections.

4. Data Management Block (DMB)

The common satellite database accessed by every block is managed in a centralized way in DMB. DMB stores telemetry data, displays trend graphs, and generates report of the status of the satellite upon the operator's request. DMB processes are written in Pro*C/C++ and are connected to the Oracle DBMS. This DBMS contains the database of telemetry frame formats, telemetry conversion coefficients, limit ranges, command frame formats, command characteristics, and sequences of commands, alarm/event message formats and run-time display formats.

Access authorization is performed for users according to the login privileges. For example, only the database administrator can update, delete, backup and insert the database tables whereas operators can only retrieve the database tables. The events of update or change to the database are recorded in the log file. To investigate some unexpected status or condition recently observed, trend analysis is provided on the selected items for a specified period. An example of output data is shown in Fig. 9. Users may display multiple plots per page, by choosing grid option, axis lengths and plot mode. The textual reports may be prepared for the limit values or the min/max/ average values, which are displayed in a chronological order. The telemetry data is stored for two weeks in the short-term storage, and transferred to an external storage for a long-term archive. For external interface with MAPS

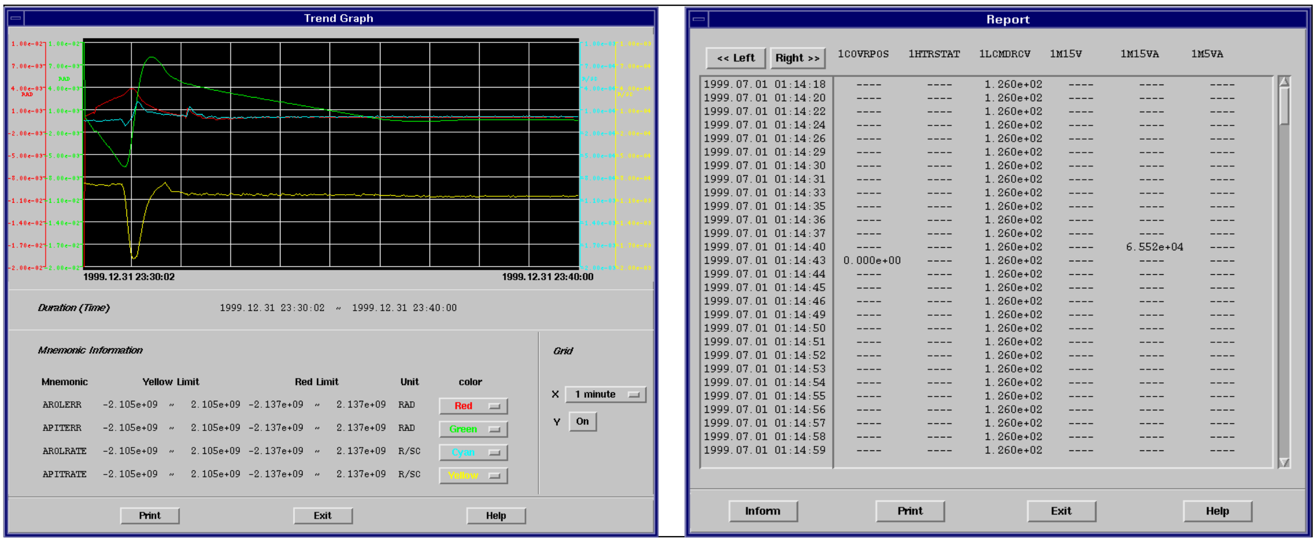

Fig. 9. Telemetry data trend plot and report window.

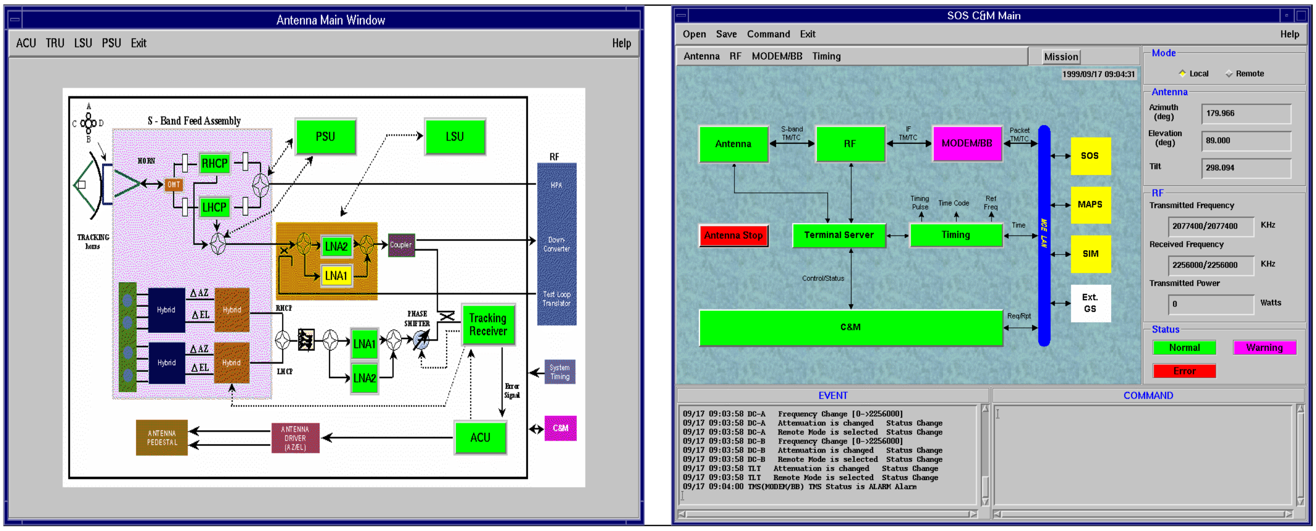

Fig. 10. Ground station control & monitoring window.

and IRPE, DMB also generates the ancillary data file, instrument data file, fuel data file, attitude data file, GPS navigation solution data file, and so on.

5. Ground Station Control & Monitoring Block (GCMB)

GCMB controls the equipment in TTC by sending remote commands to TTC in TCP/IP protocol via ground station LAN. It also receives equipment status and displays them on the screen. TTC can monitor and control that equipment either locally or through SOS according to the mode selected in real-time by the users. In order to indicate any event related to TTC equipment operation, the equipment status and event messages are displayed on the windows as shown in Fig. 10. GCMB is configured by default to work in a cold backup configuration.

IV. TEST OF SOS

SOS is a prime subsystem of MCE and verification of this subsystem is of prime importance for ensuring the reliability and performance of MCE. From the subsystem level to system-level testing, test plans and accordingly generated test procedures were prepared and subsequent testing activities have been carried out along with the integration of subsystems into system. After the MCE level verification, KOMPSAT system level verification including RF compatibility test and end-to-end test were conducted with the satellite on the ground for a full compatibility of interface and data verification. Figure 11 shows the test hierarchy and configuration of the system from the view of SOS, where various test tools were applied respectively or in combination according to verification level and test

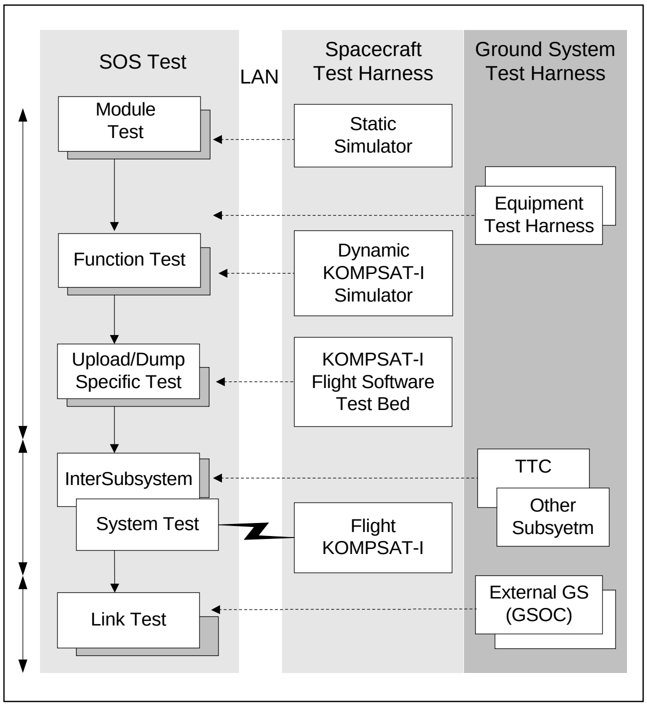

Fig. 11. Test level for SOS test.

phase: static simulator, dynamic simulator, flight software test bed, and flight model KOMPSAT-I. The test procedures, test records, test tools, and system configuration were managed and controlled under configuration management by a commercial tool, HP SoftbenchCM®.

1. Subsystem Test

The SOS test was conducted to demonstrate its capabilities, and to verify whether it satisfies the intended specifications. The trace-driven data set prepared a priori was used to test SOS. The results were displayed on the monitor or in the event message log. The subsystem test was preceded by the module tests using static simulator, to verify correctness of individual modules of SOS. After verifying the telemetry functions with a fixed pattern of telemetry, which was known, SOS was connected with the KOMPSAT-I dynamic simulator for simulated telemetry processing and telecommand processing, where telemetry was provided via LAN. Note that the KOMPSAT-I dynamic simulator (SIM) is another subsystem of MCE and verified via its own test tools before its usage for SOS verification. For TTC interface test, a test harness was developed to provide a simulated TTC equipment status for data processing and external interfaces verification purposes. Also regarding MAPS, all file formats and data transfer interfaces were verified. The results of all the test items were satisfactory, which implied a successful development of the SOS.

The SMB test items included security check function for start and termination of SOS, monitoring of the process, activation of backup process if primary process fails, and log display of event/command message history. TMPB test was specially focused on testing of telemetry processing: telemetry processing for 3 different telemetry formats, archiving of data originated from external interfaces or internally processed data, reprocessing of the stored telemetry data and telemetry display. The important issue of robustness was achieved by testing TMPB against the erroneous event occurrences of telemetry frame in terms of format error, broken frames, and missing frames.

Table 2. Telemetry processing time.

Real-Time Processing		Recorded Telemetry Processing	
TM format	Average Time	Processing Rate	Average Time
Normal format	209.81133 ms	1 time fast	85.545458 ms
Dump format	187.64617 ms	2 times faster	125.197226 ms
Programmable format	204.666426 ms	4 times faster	146.618238 ms
CPU Idle	Without Alarm Display: 62.4 %, Alarm Display: 22.4 ~ 42.8 %		
RTserver CPU rate	8.09~12.3 %		

A processing delay due to GUI display loads had been overcome by a one-to-one assignment of process instance per display window and by displaying only those telemetry items whose values have changed. The test showed that telemetry processing gave a sufficient margin in time to satisfy the real-time processing requirement. Table 2 shows the performance of telemetry processing. In the test, twelve alphanumeric display windows were open in both primary and backup workstations and out-of-range telemetry items change rate were set at about 20 percentage to increase workload in processing. Although the frame to frame variation is typically minor, we have purposely chosen to use items that caused frequent updates to test the limits of the system. It was observed that the processing was completed well within minor frame time without any missing data and the performance of processing was observed to be as fast as 4 times with respect to minor frame rate. Without display on-line, the processing time performance is even more increased so that we have much more time margin with processing of recorded telemtery processing even after replaying at 4 times fast rate.

The TCPB test items comprise of command generation and transmission, validation of real-time, absolute-time, and relative timed commands for OBC, RDU and ECU on-board processor. This test was also performed utilizing the KOMPSAT-I simulator focusing more on functional aspects rather than on the quantitative characterization. Of course it also showed a

Table 3. End-to-end test items list.

ITEMS	DESCRIPTION
CCSDS compatibility	· CCSDS Telecommand packet and Telemetry AOS
SOH telemetry reception and processing	· Normal/Programmable/Dump Telemetry Processing · Playback Telemetry
All spacecraft commands types, memory loads and dumps	· Real-time and Time-tagged command · Memory dump and upload command · Manage Command Input Buffer command · Special H/W command
Instrument command types and telemetry interfaces	· Payload Instrument commands · Payload Instrument SOH monitoring · Payload mission operations
Initial activation and checkout	· Initial Spacecraft Contacts/Deployment Assessment · GPS Activation and Clock Initialization · Earth Acquisition and AHM Assessment · Instrument Activation and Checkouts
Contingency actions	· Watch Dog Timer Fail over Recovery · Under-Voltage Recovery

good performance as in TMPB, for transmitting various types of telecommand groups in a given time slot. For functional verification, selected telecommands according to a mission scenario were transmitted to the KOMPSAT-I simulator and corresponding verifications were tested for the correctness of respective processes and response timing.

In the DMB test, operator authentication, off-line data processing, and database management were tested. In addition, SOS has also been tested for the well-known Y2K anomaly and system time synchronization with the MCE master time implemented using Network Time Protocol (NTP).

2. System Test

System test was conducted to verify that the complete MCE system meets its functional, performance, and interface requirements from an operational point of view. Test cases were developed according to daily operation scenarios, LEOP operation activities, and short-term or long-term activities. In daily operations, contact operation and non-contact operation test cases were tested separately. Tests were performed to ensure that the software-to-hardware integration works properly. All the databases were cross-checked and double-checked for correctness by comparison with the original data sheets. They were also checked up in a realistic way with the actual telemetry data files recorded during the satellite integration and test. Through a repeated test for normal cases and abnormal cases, for which the simulator acts as a test tool and harness, all the MCE functional and performance requirements were successfully verified and were qualified to go ahead to the KOMPSAT-I level testing.

3. End-To-End Test

The end-to-end test was conducted to qualify MCE as an element of the KOMPSAT-I ground segment ready to provide required support for KOMPSAT-I flight operation. In other words, the objectives of this test were to demonstrate the flight readiness of the KOMPSAT-I ground system and to verify end-to-end command and telemetry data flow between spacecraft and MCE. It was intended to verify the adequacy of MCE spacecraft command planning and preparation capability, as well as message and information communication with the ground station elements. The flight model of KOMPSAT-I was connected with MCE ground system via either RF or direct line for this test. Table 3 depicts the end-to-end test items.

Prior to the end-to-end test, the telemetry and telecommand databases were checked once again for confirmation's sake by manually comparing with the KOMPSAT-I documents and Electrical Ground Support Equipment (EGSE) database. Also, the CCSDS compatibility of the telecommand and telemetry were also tested with KOMPSAT-I directly via cable link. External ground station link test and data compatibility test with German Space Operation Center (GSOC) was carried out afterwards. During this test, all the interface incompatibilities are fixed and the system was verified. The development of the MCE, especially SOS, was concluded as a success.

V. CONCLUSIONS

Detailed description of the design, integration and testing for KOMPSAT-I SOS has been presented in this paper. The SOS subsystem, applicable to LEO satellites, has been designed, implemented, and tested by Korean engineers for the first time in Korea. In the design of SOS, the flexibility, reliability, and expandability were the main objectives. In order to achieve these objectives, commercially available CASE tool, database management system, and a real-time distributed processing middleware were integrated into the system. A database driven structure was adopted as the baseline architecture for a machine independent, mission specific database.

CCSDS recommendation was successfully implemented to provide compatibility for both the satellite logical link and external ground station data link. Also a logical address based interprocess communication scheme was introduced for a distributed allocation of the network resources. Specifically a hot-standby redundancy scheme was highlighted in the design seeking for higher system reliability and uninterrupted service required in a real-time fashion during the satellite passes. SOS has been extensively tested at different stages starting from individual block level to total ground system level. SOS has been successfully verified in all of these tests. It has been confirmed that the real-time processing performance of SOS exceeds much more than required. All the functions as well as the databases were verified completely in an end-to-end test with the flight model of KOMPSAT-I. Currently, the SOS has been successfully used for the KOMPSAT-I ground control system. This system can serve as a basis platform applicable to various LEO mission satellites once it is verified for the flying KOMPSAT-I.

ACKNOWLEDGMENTS

The project was funded by the Ministry of Information and Communications of Korea as a part of R&D activity related to the KOMPSAT-I program. The authors gratefully acknowledge the contributions of the Hyundai Space and Aircraft Co. who contributed in implementing the KOMPSAT-I SOS software.

REFERENCES

[1] J. W. Eun, J. I. Choi, W. S. Choi, W. C. Jung, and J. M. Kim, "Architectural Design of KOMPSAT Mission Control Element," *Proceeding of 17th AIAA International Communication Satellite Systems Conference*, Yokohama, Japan, February 23 [1] 27, 1998.

[2] C. H. Won, J. S. Lee, B. S. Lee, and J. W. Eun., "Mission Analysis and Planning System for Korea Multipurpose Satellite-I," *ETRI Journal*, Vol. 21, No. 3, September 1999.

[3] J. Y. Kang, J. M. Kim, and S. J. Chung, "Design and Development of an advanced Real-Time Satellite Simulator," *ETRI Journal*, Vol. 17, No. 3, October 1995.

[4] Department of Defense, *Military Standard Software Development and Documentation*, MIL-STD-498, 5 December 1994.

[5] Scott J. Landis and John E. Mulholland, "Low Cost Satellite Ground Control Facility Design," *IEEE AES Systems Managzine, June 1993*.

[6] Talarian Corporation, *RTworks User Manual*, ver 3.5, 1991–1995.

[7] J. H. Kim and H. S. Mo, "Communications with the KOMPSAT Using the Space Data Communications Standards," *IEEE Malaysia International Conference on Communication (MICC97)*, Kulala Lumpur, Malaysia, November 11, 1997.

[8] Consultative Committee for Space Data Systems, *Telecommand, Part 2: Data Routing Service, Architectural Specification*, *Recommendation CCSDS 202.0-B-2*, Issue 2, Blue Book, November 1992.

[9] Consultative Committee for Space Data Systems, *Advanced Orbiting Systems, Networks and Data Link*, *Recommendation CCSDS 701.0-B-4*, Issue 4, Blue Book, November 1995.

[10] H. S. Mo, K. S. Kim, H. S. Park, J. H. Kim, W. C. Jung, and H. J. Lee, "Design of Satellite Operations Subsystem for Korea Multi-Purposed Satellite (KOMPSAT)," *Proceedings of Annual Satellite Command, Control and Network Management Conference*, Reston, VA, USA, September 3, 1997.

Hee-Sook Mo received her B.S. in computer science from Chonnam National University in 198 9. She joined ETRI in 1990, where she was involved in developing the KOREASAT project. Since 1995, She has been participating in the KOMPSAT-I project as a member of research staff in Satellite Operation Subsystem. Her research interests include satellite communication monitoring and control systems, and network communication. She is a member of Korea Information Processing Society.

Ho-Jin Lee received his B.S., M.S. and Ph.D. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981, 1983 and 1990, respectively. Since he joined ETRI in 1983, he has been involved with development of electronic switching system and GEO/LEO satellite ground control systems. From 1996 to 1997, he worked as visiting system engineer at TRW, CA, USA with KOMPSAT program. He has been leading the KOMPSAT mission control system development project. His research interests include satellite ground control systems, satellite communications monitoring and control systems, and satellite communications applications.

Seong-Pal Lee received his B.S. in electrical engineering from Seoul National University in 1979. He received the M.S. and Ph.D degree in electrical engineering from Polytechnic Institute of New York, USA, in 1986 and 1990, respectively. He joined ETRI in 1980, where he is currently working as a director of Satellite Communications System Department, Radio and Broadcasting Technology Laboratory. His research interests include Satellite Communication System. Mobile Satellite Communication Service, Satellite Tracking Telemetry & Command (TTC) System and Stratospheric System.