Flight Software Modeling in Satellite Simulator

Sanguk Lee — Communications Satellite Development Center, ETRI Gajung-dong 161, Yusong-gu, Taejon 305-350, Korea

Sungki Cho — Communications Satellite Development Center, ETRI Gajung-dong 161, Yusong-gu, Taejon 305-350, Korea

Jae Hoon Kim — Communications Satellite Development Center, ETRI Gajung-dong 161, Yusong-gu, Taejon 305-350, Korea

Seong-Pal Lee — Communications Satellite Development Center, ETRI Gajung-dong 161, Yusong-gu, Taejon 305-350, Korea

ABSTRACT

Based on flight software modeling experiences on satellite simulator developments so far, three different approaches for modeling the flight software within the satellite simulator such as utilization of a processor emulator executing the actual flight software image, re-compilation of the flight software sources within the simulator infrastructure, and development of a set of abstract models representing the required flight software functionality are presented.

1 INTRODUCTION

KOrean Multi-Purpose SATellite-1(KOMPSAT-1) had been launched in December 1999 and has been being operated normally by Mission Control Element(MCE), which was developed by Electronics and Telecommunications Research Institute(ETRI). Now, we are in design phase of development of MCE for KOMPSAT-2, which is equipped Multi-Spectral Camera (1m panchromatic and 4m multi-spectral band).

KOMPSAT-2 MCE is consist of SOS(Satellite Operations Subsystem)[1], MAPS(Mission Analysis and Planning Subsystem)[2], TTC(Tracking, Telemetry, and Command Subsystem), and SIM(Satellite Simulator Subsystem)[3] as shown in Figure 1.

The KOMPSAT-1 MCE was verified via operation of KOMPSAT-1 from Launch and Early Orbit Phase(LEOP) to normal operation phases.

The KOMPSAT SIM, which is a comprehensive application software, includes flight software and satellite subsystem mathematical models of the KOMPSAT. Major functions of the SIM are the validation of command, functional validation and operation check of the SOS, training of operators, anomaly analysis support, functional validation of the on-board flight software, and validation of spacecraft control laws and mission scenario, etc.

The KOMPSAT SIM provides real-time and non real-time simulation capabilities for AOCS, EPS, TC&R, Ground Antenna Ranging & Tracking, Telecommand and Telemtry Processing, and comprehensive visualization tool for the satellite dynamics.

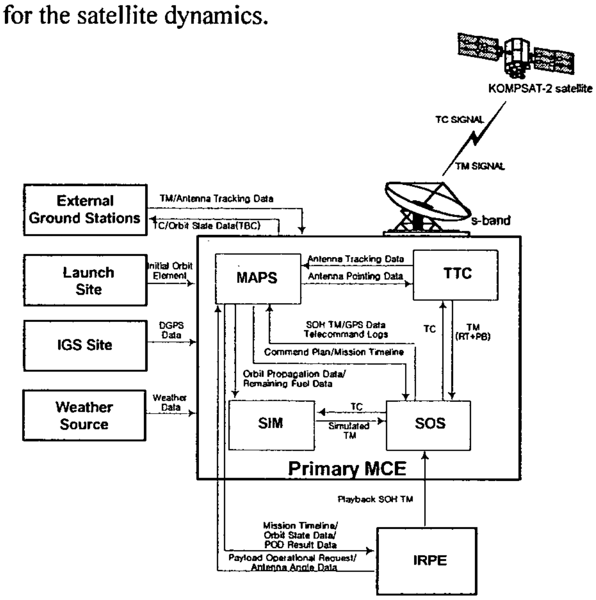

Figure 1. Configuration of KOMPSAT-2 MCE

An onboard flight software should be embedded into the satellite simulator for its simulation accuracy and fidelity. According to development situations such as availability of flight software level, documentation quality, target infrastructure, performance of target system, and so on, there are generally three different approaches for modeling the flight software within the satellite simulator as follows :

- Utilization of a processor emulator executing the actual flight software image
- Re-compilation of the flight software sources within the simulator infrastructure
- Development of a set of abstract models representing the required flight software functionality

Each of these approaches provides differing degrees of modeling fidelity to the end users. In addition to this, each approach has advantages and disadvantages, and also it is exposed into different risks to the simulator development.

2 FLIGHT SOFTWARE MODELING

2.1 Utilization of a processor emulator executing the actual flight software image

A software emulator provides a means of incorporating the functionality of the onboard software within a simulation of a spacecraft or payload by providing a virtual machine within which the real onboard software image can be run. This emulates the instructions of the target processor by translating them into instructions for the host processor.

This approach provides high level modeling fidelity and full functionality of the onboard flight software in simulator such as accurate modeling of the TC processing, and corresponding TM generation. Also, new version of flight software can be uploaded into the simulator directly or through telecommand. It makes simulator be possible to test software loading operational procedures and control, and verify new software either complete software image or some RAM patches. By the way, this approach is dependant on availability and quality of interface documentation, popularity of hardware, and performance of host computer.

Once an emulator is developed for the specific infrastructure and the specific microprocessor, it is very powerful tool for developing satellite simulator in the same environments. But it requires a lot of expenses for its first development.

2.2 Re-compilation of the flight software sources within the simulator infrastructure

Re-compiling the flight software within the simulator environment is an attractive alternative to using an emulator. The level of fidelity is high, in fact it can be very close to that provided by the emulator approach. However, new software images cannot be uploaded to the simulator and dumped to ground. The changes introduced in the new software image cannot be validated using the simulator. A common approach used is to parameterize the functionality of software expected to change frequently via satellite operations. The behavior of the software is controlled by key values contained in data tables. These tables can be accessed by load and dump TCs, without modifying the software. Hence, the behavior of the software functional models in the simulator can be changed based on the uploaded data values

The approach of re-compiling the flight software into the simulator relies on the software design being compatible with the simulator infrastructure. It should be noted that the potential compatibility problems are specific to the architecture and infrastructure of the simulator.

2.3 Development of a set of abstract models representing the required flight software functionality

This approach provides the lowest modeling fidelity. The functionality of the flight software is modeled in a set of abstract models. These models will typically be developed using information extracted from the flight software documentation, and perhaps the actual flight software. The level of abstraction that can be applied during the modeling is constrained by the end user simulator requirements, and the interfaces to other simulator models.

It is important to note that even for low fidelity simulators, the simulator end users require accurate modeling of the onboard TC handling and TM generation. Any abstract models of these portions of the flight software are therefore likely to duplicate the flight software functionality. There is, therefore, an option to re-compile specific parts of the flight software into the simulator, and develop abstract models for other parts. It can be difficult to isolate a single functional area of the software due to dependencies on other parts of the software.

Typical problems with this approach include lack of available information, incorrect information, and over-abstraction in the simulator requirement specification.

3 APPLICATION EXAMPLES

3.1 KOMPSAT SIM-1 Flight Software Modeling

The H/W configuration of KOMPSAT-1 SIM[3][4] is shown in Figure 2.

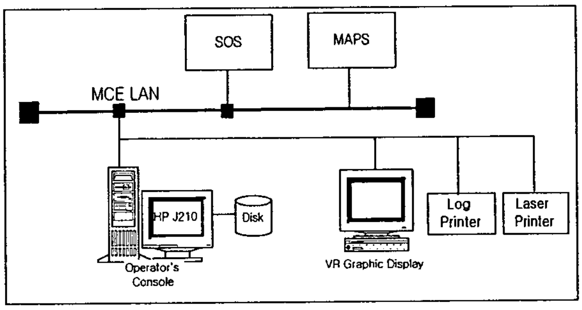

Figure 2. H/W configuration of KOMPSAT-1 SIM

Main computer was HP J210 workstation, Pentium PC is for VR display of Satellite Dynamics, and those were connected with other subsystems via LAN with TCP/IP protocol.

HP-UX is used as an Operating System for HP workstation. Figure 3. shows run-time configuration of processes in KOMPSAT-1 SIM. Data exchanges between processes is made through shared memory and message queue. The task execution sequence of the SIM is **sch** task, **tcr** task, **sdc** task, and **eps** task via message queue as shown in Figure 3.

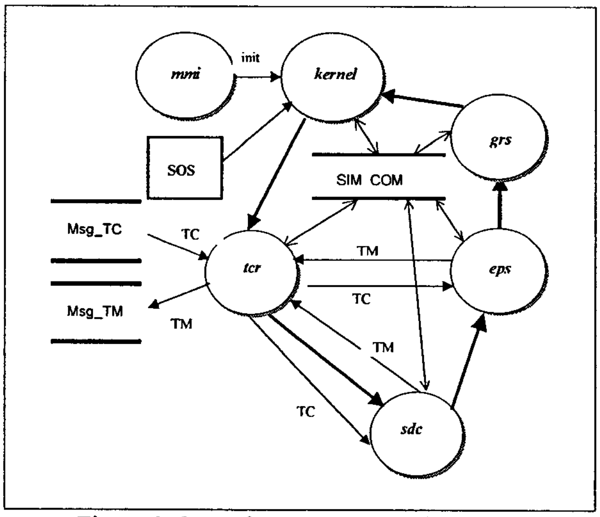

Figure 3. Run-Time Process Configuration of KOMPSAT-1 SIM

KOMPSAT-1 has three 80C186 microprocessors onboard named as OBC, RDU and ECU and those are connected with one another via 1553B data bus. VRTX is used as their operating system. Their configuration is shown in Figure 4.

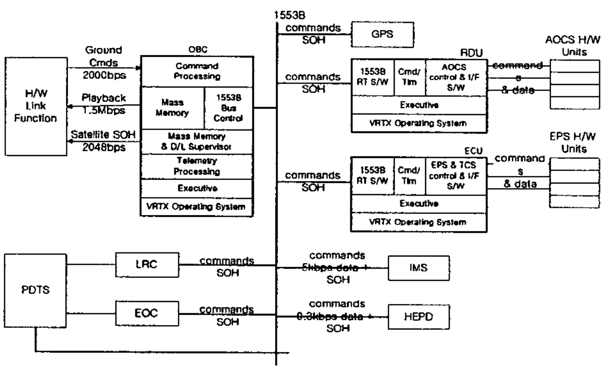

Figure 4. Configuration of KOMPSAT-1 Flight Software

During the KOMPSAT-1 SIM development, we used recompilation method. We designed simulator scheduler that was replacement of onboard scheduler on VTRX.

The routine for swapping byte&bit was used to take care byte&bit-ordering problem because of the difference in infrastructure between real satellite onboard(Intel chip microprocessor) and target system(Motorola chip workstation). We, also, needed to take care data size due to different infrastructure. Byte alignment problem in using UNION on flight software was resolved by compile option change. Data gathering and command assignment were carried out using reserved functions such as inp(), inpw() outp() onboard but those functions were not reserved functions on target system. So, those functions were designed and coded to perform the same functionalities with those sets of onboard flight software. And outpw(). Data exchanges between processors via 1553B data bus onboard was replaced with shared memory.

Development of KOMPSAT-1 SIM in this method had limitations. In other words, it did not support new flight software image or code patch upload, key parameter data(KPD) patch upload, and various memory dumps.

3.2 KOMPSAT-2 SIM Flight Software Modeling

Now, we are in development phase of KOMPSAT-2 SIM[6] in Object-Oriented Analysis/Design Methodologies[7]. Figure 5 shows the SIM H/W Configuration for KOMPSAT-2 SIM. KOMPSAT-2 SIM is developed on a PC server, which communicates with the other MCE subsystems, i.e. SOS, TTC, and MAPS, using TCP/IP protocol via MCE LAN. A PC server as platform including in Window 2000 as an operating system will be used as H/W platform and software environment. The PC server also contains VR graphic display of the KOMPSAT attitude and orbit motion. The SIM VR is implemented using Open GL.

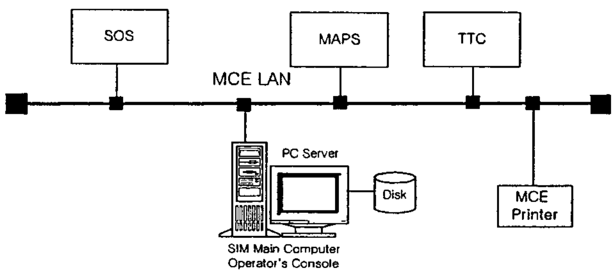

Figure 5. H/W Configuration and KOMPSAT-2 SIM

Figure 6 shows scheduler component of KOMPSAT-2 SIM that manages simulation procedure. When it receives a tick from Ischeduler interface, it sends simulation requests to Models, OBC, RDU, ECU, SCDataCollector through corresponding interfaces in order. Changing time tick interval can control a simulation speed of non real-time simulation mode.

The onboard flight software is compiled into independent process and three interface threads are connected with flight software processes. Scheduler controls simulation sequence through them.

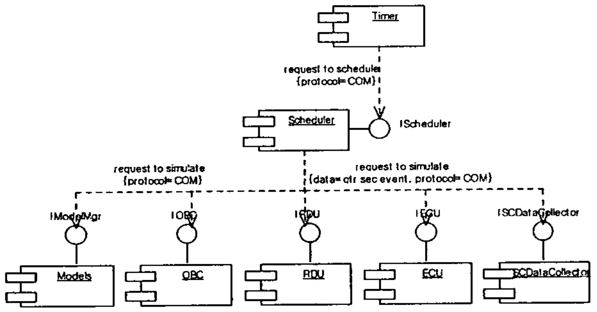

Figure 6. Scheduler Component of KOMPSAT-2 SIM

The PC server will be used for avoiding byte&bit ordering problem due to infrastructure difference. Object-Oriented Programming technique will be used to minimize flight software revision. For example, reserved I/O functions like inp(), inpw(), outp(), and outpw() will be functions named after the same function name using function overloading technique. Some system calls on VRTX operating system, also, will be replaced with functions produced by function overloading. Also, the same technique will be used for scheduler onboard and process manager. Data exchanges between processors will be implemented with component object model(COM) as shown in Figure 6. KPD in EEPROM will be emulated with file that contains KPD data to support KPD patch upload. For KOMPSAT-2 SIM, some new features and changes will be made to improve performances, to problems encountered in KOMPSAT-1 development, and to take user request into account. The object oriented design and programming technique will be employed for extensibility, reusability, and portability.

3.3 KOREASAT SIM Flight Software Modeling

Laboratory model of KOREASAT Ground Control System was developed by ETRI from 1990 through 1994. During the that time, KOREASAT SIM, Advanced Real-Time Satellite Simulator(ARTSS)[8] was developed from 1993~1994. It was hybrid simulator that was software simulator combined to H/W communication payload system developed by ETRI. Figure 7 shows the H/W configuration ARTSS that is consist of VAX workstation as main computer, 3D VR display PCs, and terminal server to provide I/F with Real-Time System(Satellite Operation System) for TC and TM Links and I/F with Payload Control PC for Payload Commanding and Telemetry Gathering.

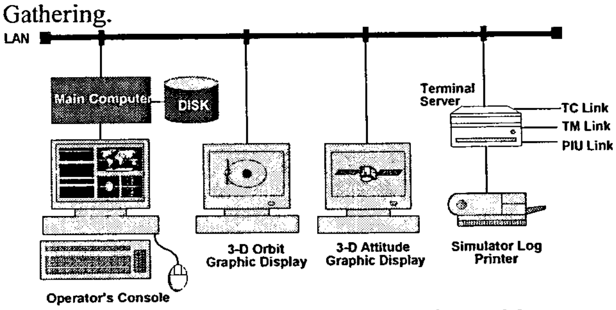

Figure 7. H/W Configuration of ARTSS

Figure 8. shows the Architecture of ARTSS for KOREASAT-1. Abstract models representing the required flight software functionality was developed for the ARTSS of KOREASAT-1. On the contrary KOMPSAT-1, LEO satellite, KOREASAT-1 is the GEO satellite and it is contacted with ground control system all the time. For this reason, flight software for KOREASAT-1 is less complicated than that for KOMPSAT-1.

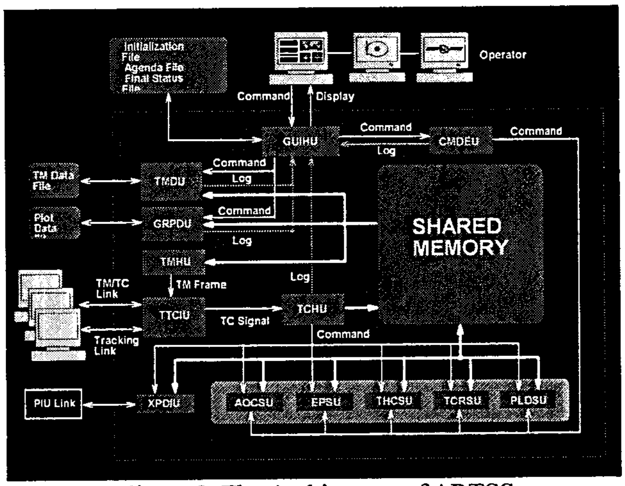

Figure 8. The Architecture of ARTSS

Several functions of KOREASAT-1 flight software spread into units of ARTSS. TeleCommand Handling Unit(TCHU) contains TC processing function, TeleMetry Handling Unit(TMHU) contains TM processing function, and Attitude Orbit Control Subsystem Unit(AOCSU) includes attitude processing electronics(APE) that has attitude control logic. Real-time scheduling was developed as it has almost the same functionality with the onboard. Communications between processes were made by shared memory. This approach is easy to implement if the detailed document for flight software is provided but it does not provide hi-fidelity simulation function.

4 CONCLUSIONS

Three different methods to model flight software of satellite for satellite simulator were discussed. Among them, application examples for two methods, abstract modeling on KOREASAT-1 simulator(ARTSS) and recompilation(KOMPSAT-1 and 2 SIM), were presented in this paper. Both methods had limitation on simulation coverage and on achievement of hi-fidelity simulator. In near future, we will employ most attractive method, utilization of a processor emulator executing the actual flight software image for our satellite simulator to avoid tedious code-modification.

ACKNOWLEDGMENTS

This paper was prepared based on the KOMPSAT2 Ground Segment Consultancy report[9] and KOREASAT-1 and KOMPSAT-1 Simulator development experiences.

REFERENCES

[1] Hee-Sook Mo, Ho-Jin Lee, and Seong-Pal Lee. 2000. Development and Testing of Satellite Operation System for Korea Multipurpose Satellite- I. *Etri Journal* 22(1) :1~11.

[2] Chang-Hee Won, Jeong-Sook Lee, Byoung-Sun Lee, and Jong_won Eun. 1999. Mission Analysis and Planning System for Korea Multipurpose Satellite-I. *Etri Journal* 21(3) : 29~40.

[3] Wan Sik Choi, Sanguk Lee, Jong Won Eun, Hanjun Choi, Dong Suk Chae. 2000. Design Implementation and Validation of the KOMPSAT Spacecraft Simulator. *KSAS International Journal* 1(2) :50-67.

[4] Sanguk Lee, Dong Seok Chae, and Wan Sik Choi. 1999. Implementation of An Interface Between Flight Software and Satellite Subsystem Models on the KOMPSAT Simulator *KSAS* 27(3) : 145-150.

[5] Sanguk Lee, Oryong Kwon, and Ho-Jin Lee. 2001. Feasibility Study of Satellite Simulator Development by Using Multithreading and Shared Memory. KSAS 29(2) 125~131.

[6] Sanguk Lee, Sungki Cho, Jae Hoon Kim, and Seong-Pal Lee. 2001. Object-Oriented Design of KOMPSAT-2 simulator including onboard Flight Software. *14th International Conference in System Science*. Wroclaw, Poland. To be presented in Sept. 13.

[7] Odell, J. J. and Martin, J. 1995. *Object Oriented Method: A Foundation.* Prentice-Hall

[8] Ja-Young Kang, Jae-Moung Kim, and Seon Jong Chung. 1995. Design and Development of an Advanced Real-Time Satellite Simulator. *Etri Journal* 17(3) : 1-14.

[9] Andy Williams and Mike Irvin. 2001. KOMPSAT-2 Embedding Flight Software and Use of Store and Restore Function within Satellite Simulators. Technical Report. NO ETRIREP.007. VEGA Informations-Technologien GmbH Julius-Reiber-Str. 19, 64293 Darmstadt, Germany

AUTHOR BIOGRAPHIES

SANGUK LEE is a senior research engineer of Communications Satellite Development Center at Electronics Telecommunications Research Institute, Taejon, Korea. He received his BS from Yonsei University, Seoul, Korea, in 1988 and MS and Ph. D. in Aerospace Engineering from Auburn University, Alabama, U. S. A, in 1991 and 1994, respectively. His research interests are in satellite dynamics and control, modeling, simulation, ground control system, and optimal control. His email address is <slee@etri.re.kr>.

SUNGKI CHO is a Senior Research Staff of Communications Satellite Development Center at Electronics Telecommunications Research Institute, Taejon, Korea. He received his BS in Astronomy from Yonsei University, Seoul, South Korea in 1990, and MS and Ph.D in Aerospace Engineering from Auburn University, Alabama, US in 1995 and 1999, respectively. His research interests are in Dynamics and Control of Satellite. His email address is <soocho@etri.re.kr>.

JAEHOON KIM is a Project Manager in the Communications Satellite Development Center at ETRI. He received the Ph. D. degrees in computer engineering from Chungbuk National University, Cheongju, Korea. His research interests are satellite simulation, security in the satellite communications, fault diagnosis of satellite using AI technologies, and system modeling using objected-oriented technologies. His email address is jhkim@etri.re.kr

SEONG-PAL LEE is a Director of Communications Satellite Development Center at Electronics Telecommunications Research Institute, Taejon, Korea. He received his B.S. from Seoul National University, M.S and Ph.D. from Polytechnic Institute of New York. His interests include Satellite Communications System, Mobile Satellite Communication Service, Satellite Tracking & Telemetry & Command(TTC) System and Stratospheric System. His email and web addresses are spallee@etri.re.kr.