J. Astron. Space Sci. 19(2), 97–106 (2002)

DESIGN AND FLIGHT SOFTWARE EMBEDDING OF KOMPSAT-2 SIMULATOR

Sanguk Lee†, Sungki Cho, and Jae Hoon Kim
Communications Satellite Development Center, ETRI, Daejeon 305-350, Korea
E-mail: slee@etri.re.kr

(Received February 7, 2002; Accepted April 10, 2002)

ABSTRACT

The design feature of KOMPSAT-2 simulator based on object oriented design methodology in terms of unified modeling language (UML) has been discussed in this paper. Also, we present how to embed flight software into the simulator. Flight software embedding for KOMPSAT-2 simulator is compared to that of the KOMPSAT-1 simulator.

Keywords: KOMPSAT, simulator, flight software embedding, object-oriented design, UML

1. INTRODUCTION

KOrea Multi-Purpose SATellite-1 (KOMPSAT-1) had been launched in December 1999 and has been being operated normally by the Mission Control Element (MCE), which was developed by Electronics and Telecommunications Research Institute (ETRI). Currently, we are designing and developing the MCE for KOMPSAT-2, which will be equipped with Multi-Spectral Camera (1m panchromatic and 4m multi-band). KOMPSAT-2 MCE is consists of SOS (Satellite Operations Subsystem) (Mo et al. 2000), MAPS (Mission Analysis and Planning Subsystem) (Won et al. 1999), TTC (Tracking, Telemetry, and Command Subsystem), and SIM (Satellite Simulator Subsystem) (Choi et al. 2000) as shown in Figure 1.

The KOMPSAT-1 MCE system has been verified via operation of KOMPSAT-1 from Launch and Early Orbit Phase (LEOP) to normal operation phases, and long run operations so far.

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, TC (TeleCommand) and TM (TeleMetry) Processing, and comprehensive visualization tool for the satellite dynamics. The onboard flight software should be embedded into the satellite simulator for its simulation accuracy and fidelity. Depending

†corresponding author

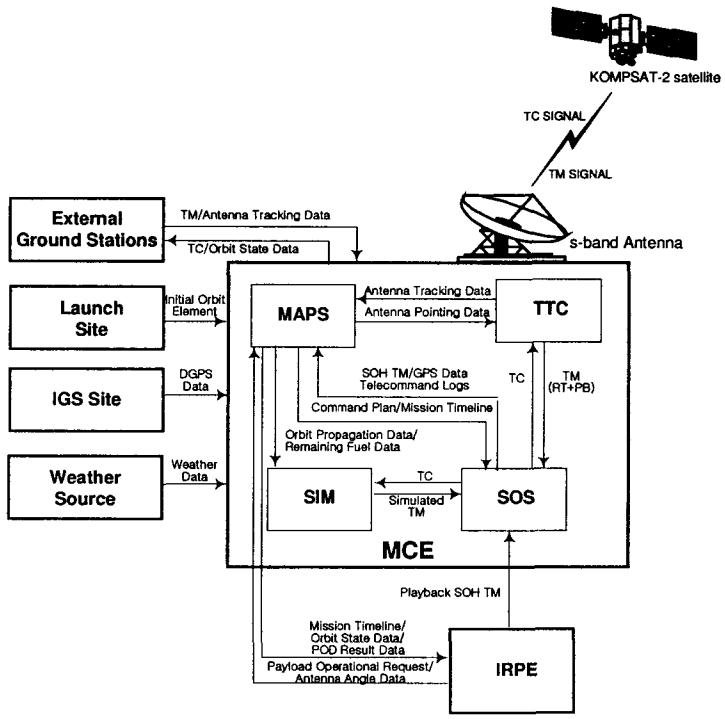

Figure 1. Configuration of KOMPSAT-2 MCE.

on 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 such as (Williams & Irvin 2001)

- 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.

In this paper, we will discuss about the KOMPSAT-2 simulator design and the re-compilation method which is employed for KOMPSAT-1 and KOMPSAT-2 flight software embedding.

Conventional structured design method that was applied to KOMPSAT-1 SIM is straightforward. However it may not satisfy modern S/W development and maintenance requirements such as reusability, extensibility, and reliability. For the KOMPSAT-2 SIM, OOA/OOD processes are employed to maximize those requirements.

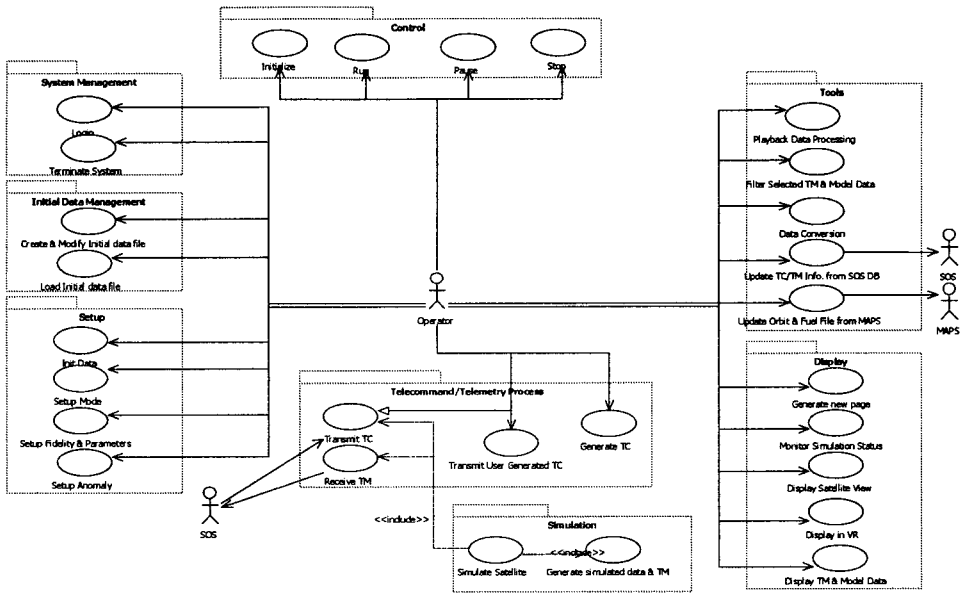

Figure 2. KOMPSAT-2 SIM use case diagram.

2. KOMPSAT-2 SIMULATOR OBJECT-ORIENTED DESIGN

The UML notation is common language to specify, construct, visualize, and document for designing object-oriented software system. The standard OOA (Object-Oriented Analysis)/OOD (Object-Oriented Design) procedure (Odell & Martin 1995) has been made to design KOMPSAT-2 SIM. According to Object-Oriented Analysis process, Use Case Modeling and Domain Modeling were carried out. Then, Architecture Design and Component Design have been performed. The more detailed modeling and design processes in OOA/OOD can be found in references (Rumbaugh et al. 1991, Pooley & Stevens 1999).

2.1 Use Case Modeling

Use Case Diagram is provided to describe system requirements in the users (operator) external view points as shown in Figure 2.

The KOMPSAT-2 SIM has Use Cases that are categorized as several groups such as system management, control, initial data management, setup, TC/TM process, simulation, display, and tools. Use Case Specification is required for each Use Case after generation of Use Case Diagram.

2.2 Domain Modeling

Domain Modeling describes how the Use Case is realized in the Use Case Model. Domain Modeling can be expressed by the Class Diagram, which describes how these classes interact with each other. Collaboration (or Sequence) Diagram is also used to describe flows of massages and sequences between objects, structural relationship, and control flows.

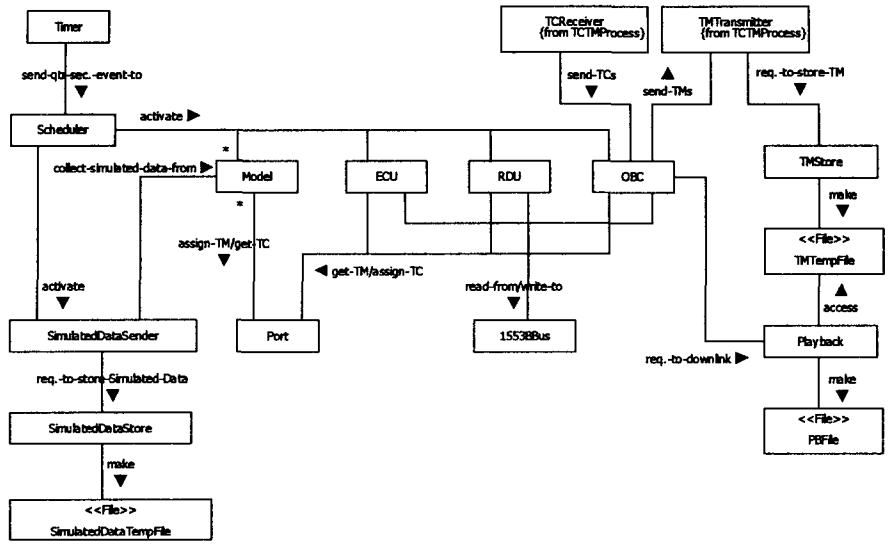

Figure 3. Class diagram of KOMPSAT-2 SIM domain modeling.

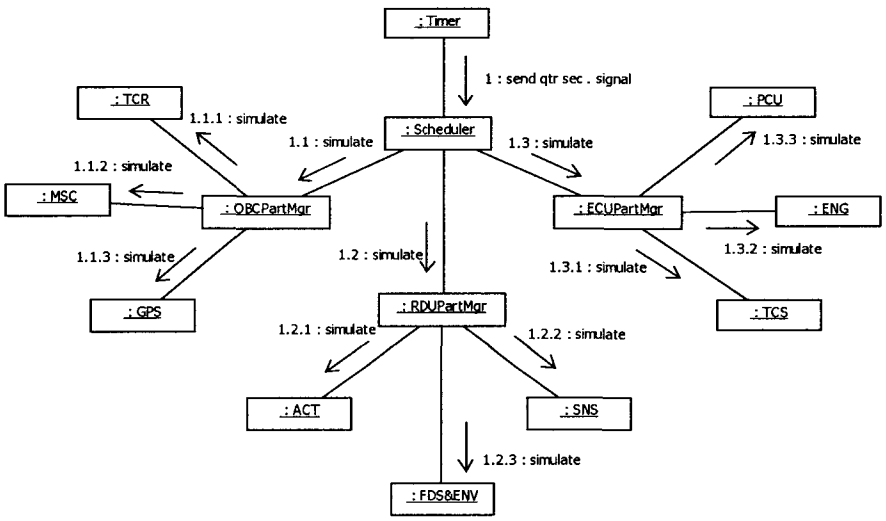

Figure 4. Collaboration diagram of KOMPSAT-2 SIM domain modeling.

Figure 3 shows Class Diagram of KOMPSAT-2 SIM that describes interaction and message flows between objects. Communications between three flight software and S/C Model are implemented by using Port object, and the onboard 1553B bus communication is implemented by introducing 1553B Bus object. OBC, RDU, and ECU in Figure 3 are named after three sets of onboard flight software, respectively.

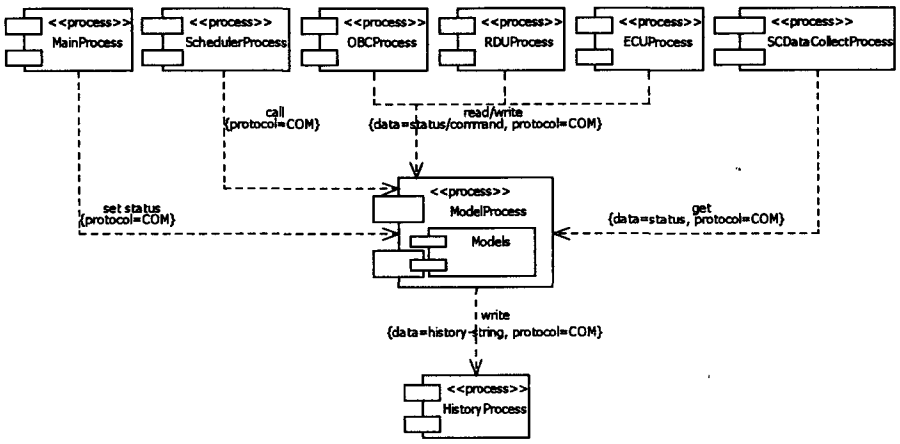

Figure 5. Process view of KOMPSAT-2 SIM architecture design (Mode l process).

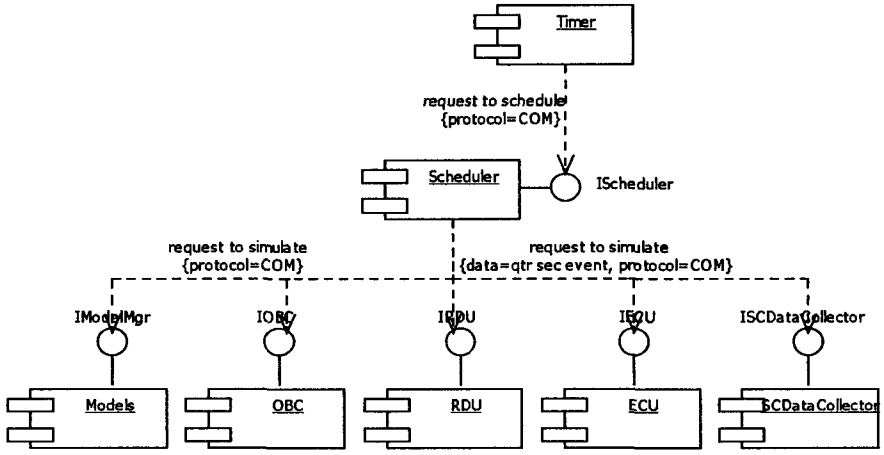

Figure 6. Scheduler component of KOMPSAT-2 SIM.

Figure 4 shows an example of Collaboration Diagram that describes interactions between objects and their sequences. In Figure 4, Timer requests Scheduler to generate Events every quarter second, and then Scheduler requests Models, OBC, RDU, and ECU to perform simulation. Requested Models perform simulation and write the results of simulation on Port. Port is the only interface between S/C hardware models and onboard flight software.

2.3 Architecture Design

SIM architecture can be expressed in four different viewpoints. Logical View decomposed into conceptual components or logical components and describes connector between them. Development View describes modules and their dependency. Process View describes processes in the system and communications between them. Physical View describes deployment of process in hardware

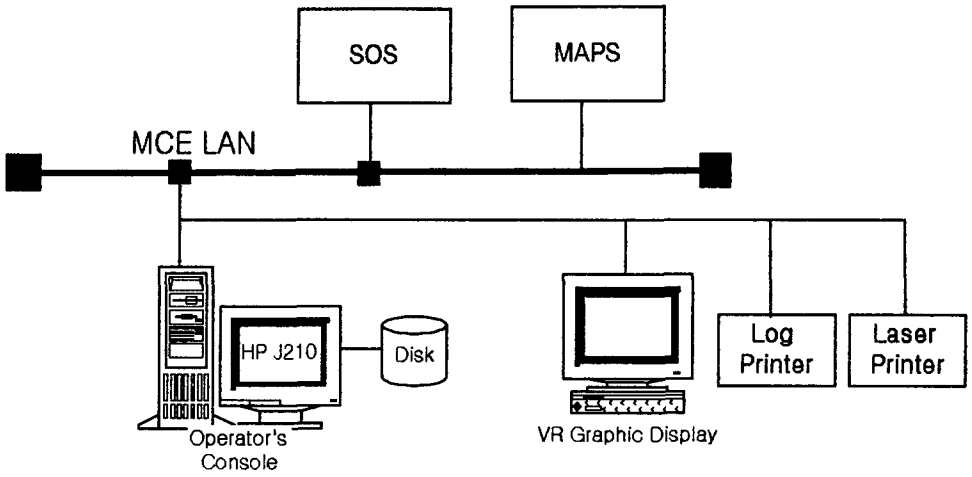

Figure 7. H/W configuration of KOMPSAT-1 SIM.

and communication between them. Figure 5 shows process view of KOMPSAT-2 SIM architecture design that describes model process and its related processes. Communications between processes is implemented by using COM (Microsoft Component Object Model), Communication with SOS is implemented by using socket, and H/W interface is implemented by introducing Port.

2.4 Component Design

Component Design is to describe model configuration and dependency in static implementation view. It describes interfaces, static structure, dynamic behavior functions, format, model, and persistent data of components. 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, and SCDataCollector through corresponding interfaces in order. Changing time tick interval can control simulation speed of non real-time simulation mode.

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

3. KOMPSAT-1 SIMULATOR FLIGHT SOFTWARE MODELING

The H/W configuration of KOMPSAT-1 SIM (Lee et al. 1999) is shown in Figure 7. Main computer for KOMPSAT-1 SIM was HP J210 workstation, Pentium PC is for VR (Virtual Reality 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 8 shows run-time configuration of processes in KOMPSAT-1 SIM. Data exchange between processes is carried out through shared memory and message queue. The task execution sequence of the SIM is kernel task, tcr task, sdc task, and eps task via message queue as shown in Figure 8.

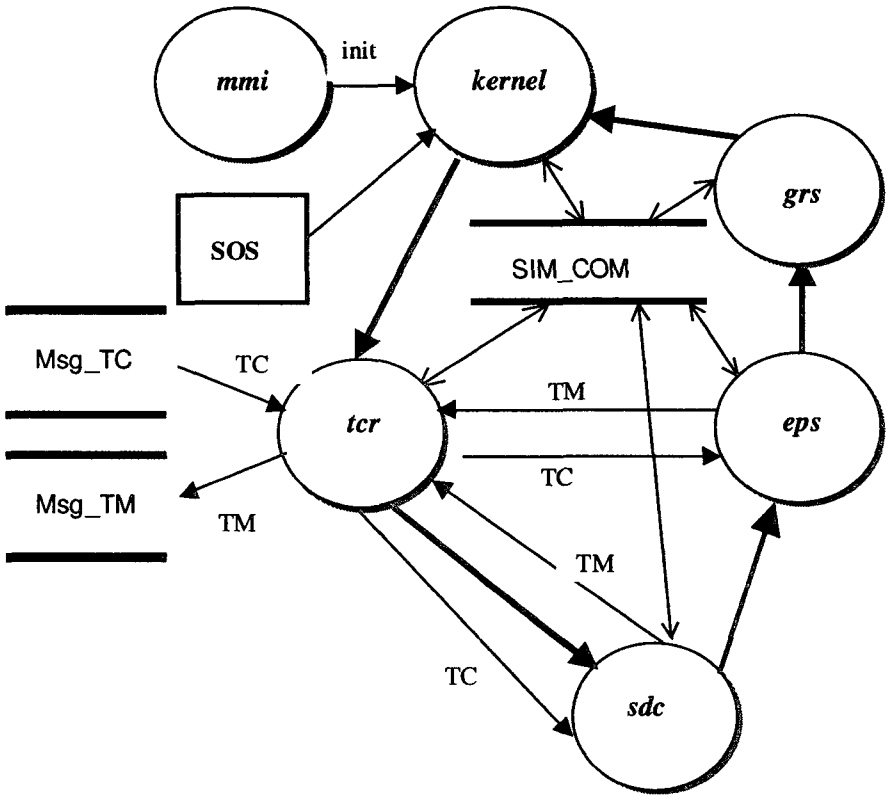

Figure 8. Run-Time process configuration of KOMPSAT-1 SIM.

KOMPSAT-1 has three 80C186 microprocessors onboard named as OBC, RDU and ECU and they are connected with one another via 1553B data bus. VRTX is used as their operating system. Their configuration is shown in Figure 9. During the KOMPSAT-1 SIM development, we used re-compilation 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 processor) and target system (Motorola processor). We also needed to take care data size due to different infrastructure. Byte alignment problem in using UNION of C-language 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. Data exchange 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,

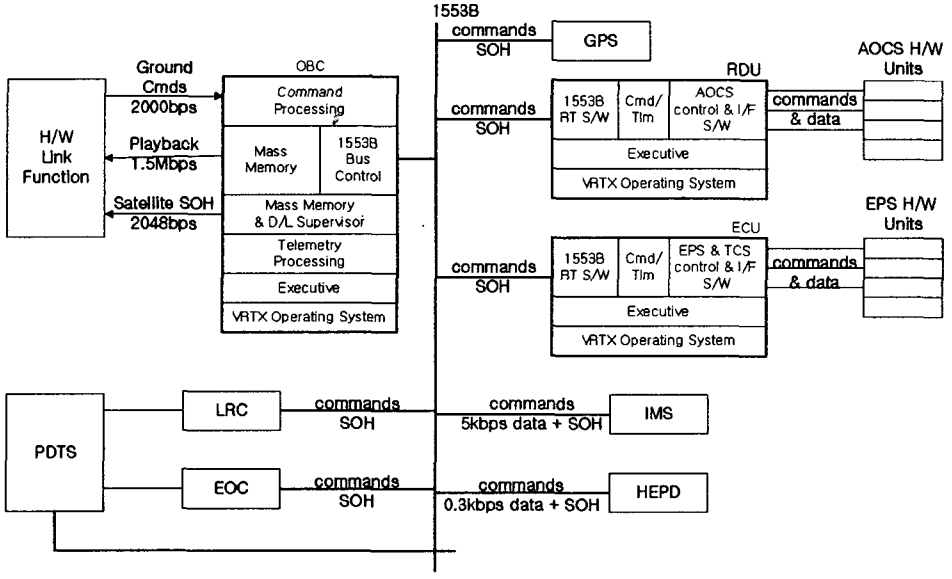

Figure 9. Configuration of KOMPSAT-1 flight software.

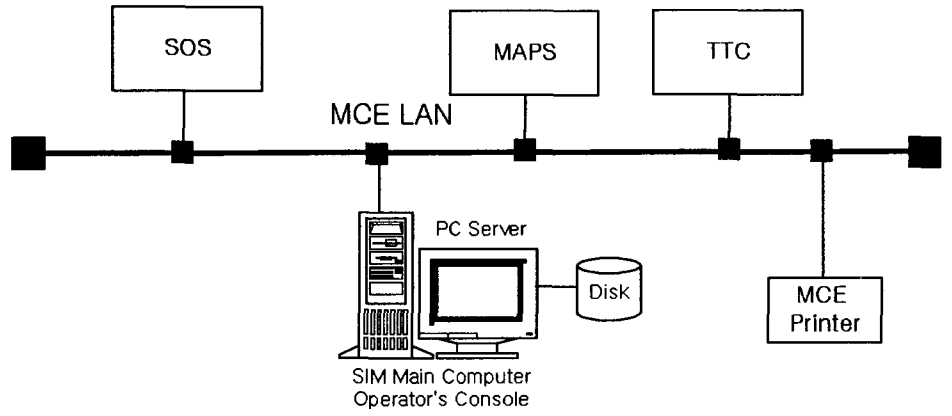

Figure 10. H/W Configuration and KOMPSAT-2 SIM.

and various memory dumps.

4. KOMPSAT-2 SIM FLIGHT SOFTWARE MODELING

Now, we are in development phase of KOMPSAT-2 SIM in Object-Oriented Analysis/Design Methodologies (Odell & Martin 1995). Figure 10 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 including 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.

The PC server equipped Intel microprocessor 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 (Lee et al. 2001a, 2001b). 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 exchange between processors will be implemented with Component Object Model (COM) as shown in Figure 6. KPD in EEPROM will be emulated with the 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 solve the 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, portability, and less code changes in flight software embedding.

5. CONCLUSIONS

The KOMPSAT-2 SIM design according to Object-Oriented Design Methodologies and flight software embedding for KOMPSAT-2 simulator by recompilation method were presented in this paper. These were employed for better performance with fewer efforts in development of satellite simulator. Flight software embedding discussed in this paper had limitations 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 work was supported by Korea ministry of information and communications.

REFERENCES

Choi, W. S., Lee, S., Eun, J. W., Choi, H., & Chae, D. S. 2000, KSAS International J., 1, 50

Lee, S., Chae, D. S., & Choi, W. S. 1999, KSAS, 27, 145

Lee, S., Cho, S., Kim, J. H., & Lee, S.-P. 2001a, in Proceedings of the 14th International Conference on System Science, eds. Z. Bubnicki & A. Grzech (Wroclaw: Oficyna Wydawnicza Polotechniki Wroclawskiej), III, 184

Lee, S., Kwon, O., & Lee, H.-J. 2001b, KSAS, 29, 125

Mo, H.-S., Lee, H.-J., & Lee, S.-P. 2000, ETRI J., 22, 1

Odell, J. J., & Martin, J. 1995, Object Oriented Method: A Foundation (New York: Prentice-Hall)

Pooley, R., & Stevens, P. 1999, Using UML: Software engineering with objects and component (New York: Addison-Wesley)

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. 1991, Object-Oriented Modeling and Design (New Jersey: Prentice-Hall)

Williams, A., & Irvin, M. 2001, Technical Report. NO ETRIREP.007 (Darmstadt: VEGA Informations-Technologien GmbH)

Won, C.-H., Lee, J.-S., Lee, B.-S., & Eun, J.-W. 1999, ETRI J., 21, 29