Test, debug and development environment of Subaru Observation Software System

Junichi Noumaru[a], Yoshihiko Mizumoto[b], Toshiyuki Sasaki[b], George Kosugi[b],
Jun A. Kawai[c], Yasuhiro Morita[c], Itsushi Akiyama[d], Yasutoshi Kusama[d] and Shigeaki Imai[d]

[a]Subaru Telescope, National Astronomical Observatory of Japan
650 N. A'ohoku Pl., Hilo, Hawaii 96720, U.S.A.

[b]Subaru Project, National Astronomical Observatory of Japan
2-21-1 Osawa, Mitaka, Tokyo 181, Japan

[c]Science Systems Division, Fujitsu Limited
1-9-3 Nakase, Mihama, Chiba, Chiba 261, Japan

[d]SEC Co. Ltd.
NES Bldg. 22-14 Sakuragaoka, Shibuya, Tokyo 150, Japan

ABSTRACT

Subaru Observation Software System (SOSS) provides observers with so called high-level user interface, scheduler and data archival system. This paper presents both software and hardware environment for test, debug and development of instrument controller (OBCP) and SOSS. The environment is composed of instrumentation software toolkit, instrumentation software simulator, telescope simulator and summit simulation computer system.

Keywords: Subaru Telescope, software, simulator, toolkit

1. INTRODUCTION

Since the concept of Subaru Observation Software System (SOSS) was released[1], we have set distributed process and scheduling observation as one of the key of this system[2,3]. The other major function of the system is data acquisition and handling to the archive system[3,4] which transfers the data to Hilo base facility and even to Japan.

In order to realize flexible scheduling observation, unified control of both telescope and instruments is indispensable. On the other hand, since instruments are manufactured by various groups and configuration varies instrument by instrument, observation system need acceptable interface between them. Before instruments are brought up to the summit, interface check and test must be done. Every step of check and test has must be done in the manufacturing process, before shipping and just before taking the instrument to the summit. With the combination of instrumentation software toolkit, which facilitates instrumentation group to make the complied interface, and instrumentation software simulator, interface test can be done without other computers than an instrument control computer(OBCP). We have a computer system in Japan which simulates the network of the summit system in part. Software test and even some hardware test of OBCPs can be done before shipping to Hawaii. Unpacked instruments can be tested again in Hawaii by using the telescope simulator and summit simulation system.

Further author information -

J.N.: Email noumaru@naoj.org, Telephone (808)934-7788, Fax (808)934-5984, WWW: http://www.naoj.org

2. OVERVIEW OF INSTRUMENTATION SOFTWARE TOOLKIT

Subaru software consists of the telescope control software, instrument control software, and observation control software which controls former two kinds of software. Since instrument control software is developed by instrumentation groups for the specific instrument, interface to observation control software should be clear and easy. Observation data from instruments are transferred to data storage of data acquisition subsystem where preliminary data analysis is available.

But fast read-out instruments, such as infrared and mid-infrared instruments or instruments which generate large format data, such as mosaic CCD camera may require browsing and data analysis within their instrument control system. The function and method of browsing and data analysis for each instrument are similar and have a lot of components in common. In order to save manpower for coding in instrumentation groups, Subaru observation software development team provides instrumentation groups with common software in data analysis so called "instrumentation software toolkit."

Toolkit is divided in two categories. One is related to the communication between observation control software and instrument control software. Software for communication interface, provided as a toolkit can be installed in instrument control workstations. The contents of the toolkit are communication tool, status data acquisition tool, FITS format edit tool, and image monitoring tool.

The other toolkit is related to data browsing and analysis and is also installed in instrument control workstations. The contents of the toolkit are log generating tool in terms of preliminary data analysis, FITS file I/O and image data calculation, WCS(World Coordinate System) keyword generating tool and data browser.

The detail of this toolkit was introduced in the previous paper[4].

3. INSTRUMENTATION SOFTWARE SIMULATOR

Since most instruments of the first generation are developed, assembled and tested in Japan, the test environment for the instrument controller (OBCP) is required in Japan as well as in Hawaii where some instruments will be developed in future. Moreover, sites of development spread over the country and their resources for the test are limited. Communication test with Subaru Observation Software System (SOSS) without any computers other than the OBCP itself is required.

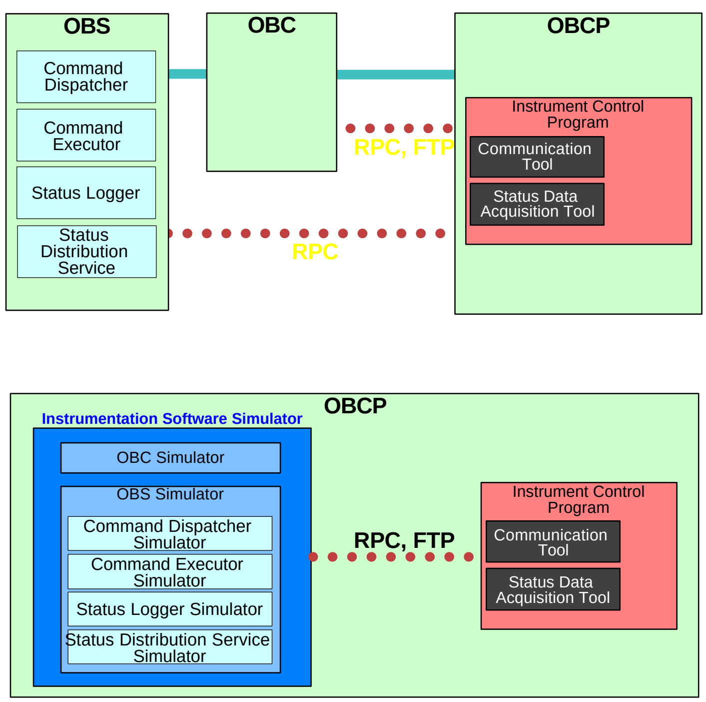

Figure 1 Simulator and toolkit

Figure 1 show how the simulator works. The diagram above is a configuration of OBCP, OBC and OBS in the operation at the summit. Both communication tool and status data acquisition tool are components of the instrumentation software toolkit and logically communicates with OBC and OBS. Physically, OBCP is connected only with OBC. Communication method among them is shown. The diagram below shows an OBCP with instrumentation software simulator installed. Communication via LAN at the summit is converted to the communication among processes inside the machine.

The software simulator which works in OBCPs has been developed and were distributed among instrumentation groups. It simulates the interface to OBCP in SOSS, sends commands to OBCPs and receives replies from them.

Networking test using more than two machines and network among them is the next step. This test will be done in the simulation environment located in both Japan and Hawaii. Even this simulation environment, however, lacks some components which are related to the telescope controller. In order to compensate data from/to them, SOSS subsystems have a "test mode", which simulate the telescope controller function and enables OBCPs to test their network performance at the simulation environments.

3.1. HARDWARE REQUIREMENTS FOR INSTRUMENTATION SOFTWARE SIMULATOR

The following hardware and software are required for simulator:

- Sun SPARCstation 20
- Sun Ultra 1
- Memory size > 64MB
- Disk space > 10MB
- Solaris 2.4 or 2.5
- SPARCompilerC 3.0.1 or 4.0
- Tcl7.4/Tk4.0

3.2. COMPONENTS AND FUNCTION

OBCP is connected with OBC(data acquisition and preliminary data analysis) by Ethernet and/or FibreChannel and communicates with OBC and OBS(scheduler and status logging). Instrument software simulator works in OBCP and simulates OBC and OBS.

Corresponding with four processes in OBS, OBS simulator is composed of the four main processes. Once the simulator is started, main menu appears and one can see components in the simulator (Figure 2).The command dispatcher in OBS issues commands to OBCPs and TSC(Telescope Supervising Controller). The command dispatcher simulator in OBS simulator issues commands, examines replies and measures response time so as to verify if OBCP program is complied with three-way hand shake which is a standard communication procedure in SOSS. The interface of OBCP program to OBS is supposed to be built with software toolkit which facilitates instrumentation group to make an interface. The command dispatcher simulator has its own user interface and users can monitor command characters issued and received, select a file where commands for OBCP are stored, edit

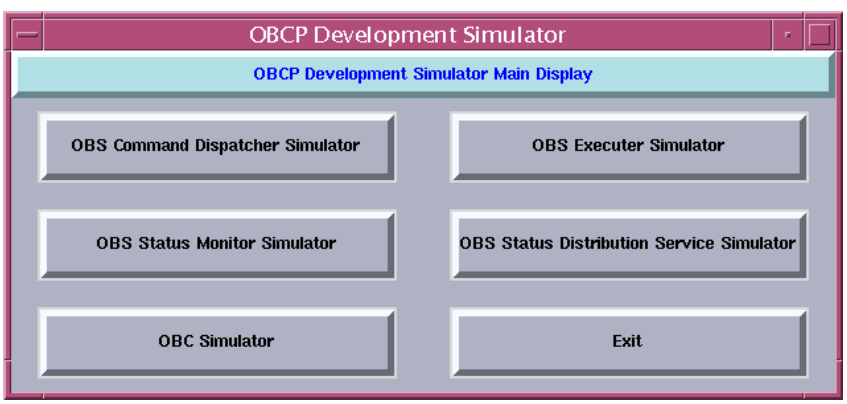

Figure 2 Main Menu

a command to be issued and select sequential command issue or simultaneous command issue. The command dispatcher simulator makes a log file where commands and replies are saved. Figure 3 shows the display of the command dispatcher simulator.

The executor simulator communicates with OBCP in transparent mode when OBCP controls both instrument and telescope. Executor simulator receives commands for telescope control from OBCP and sends acknowledgement immediately and return value to OBCP when the command is done back. The executor simulator has similar user interface with that of the command dispatcher and then users can monitor, select and edit commands and save commands and replies in a log file.

The status monitor simulator receives the status data issued by OBCP program. User interface of this status monitor simulator displays the status characters received.

The status distribution service simulator verifies OBCP commands which request the latest status information the status logger holds and replies to OBCP. Return value from the simulator is fixed and is referred from a file specified in an environment variable.

Note that this simulator checks the communication protocol and the timing but does not check any syntax and

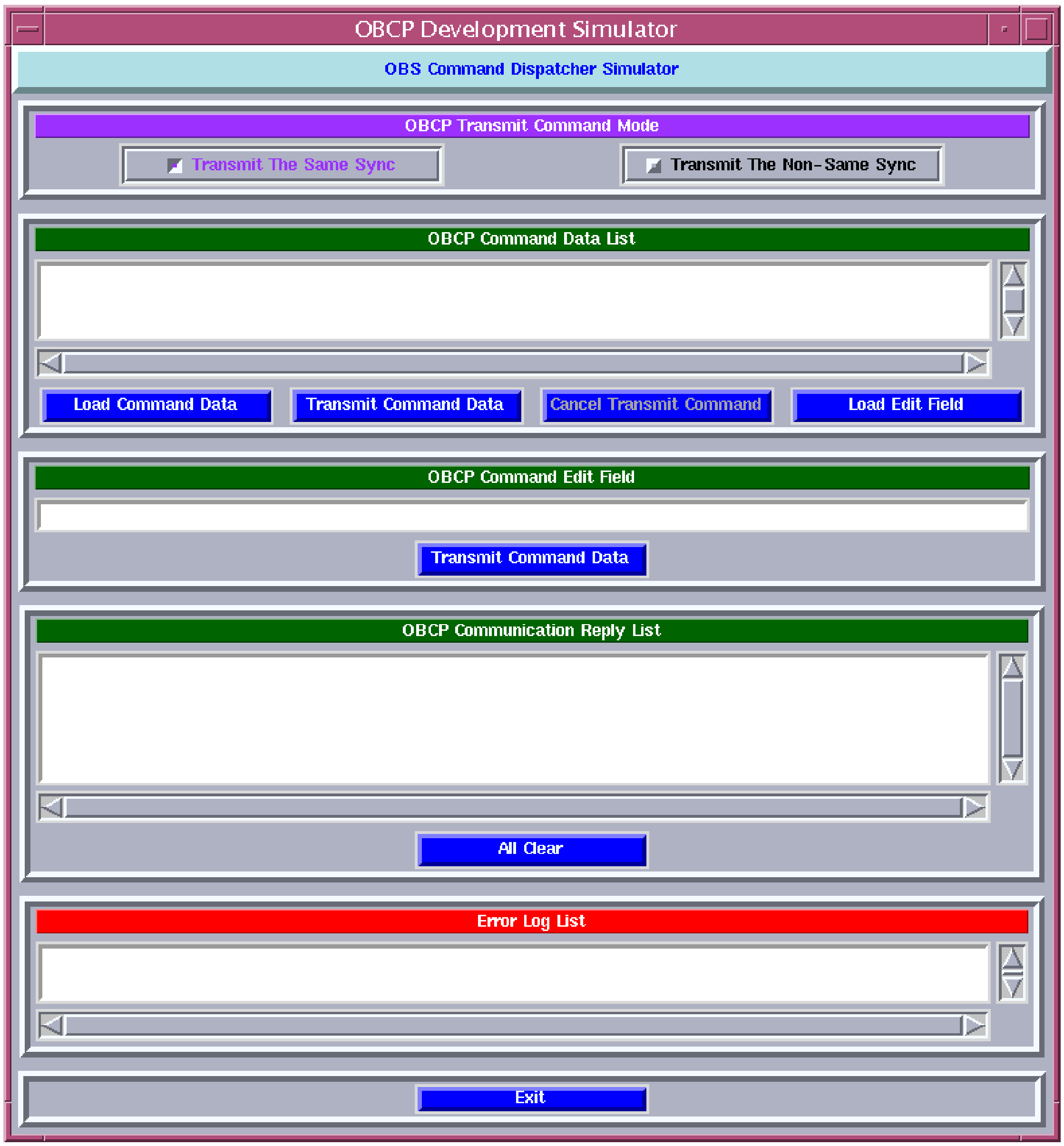

Figure 3 User interface of Command Dispatcher Simulator

logical consistency of commands issued and received.

OBC simulator also works in OBCP and gets data from OBCP program. When OBCP requests OBC simulator to get observation data, the FTP client in OBC simulator access FTP server in OBCP program and then get data. The user interface displays commands from and to the OBC simulator, file list which has been taken from OBCP program (Figure 4).

4. SUMMIT SIMULATION SYSTEM

Base facility of Subaru Telescope opened in April 1997 (Figure 5) and computer and network system including a super computer, a parallel processing server and 174TB tape library was installed there. This system includes the summit simulation system which is substantially a subset of the SOSS at the summit. This system includes two sets of Sun Ultra Enterprises, three sets of Sun Ultra 2, a DEC Alpha 600, a 60GB disk array, an 1.5TB tape library, a 266Mbps FibreChannel switch and an ATM switch with two OC12 ports and twelve OC3 ports. The summit simulation system has a LAN independent of general LAN in the building which extends to laboratories and the simulator room where “Telescope Simulator” is located. Detail on the telescope simulator is mentioned in the next

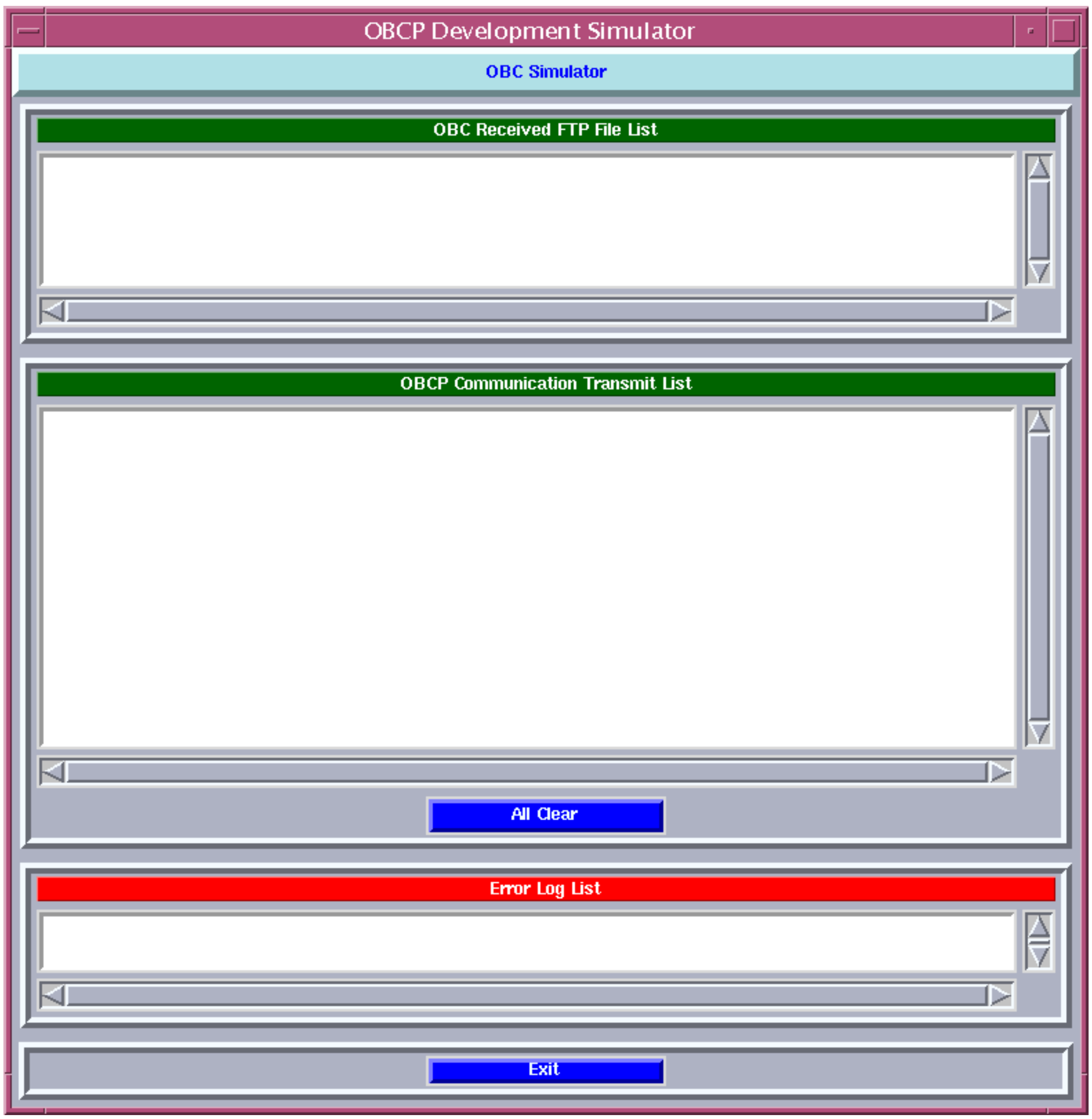

Figure 4 User interface of OBC Simulator

section.

One of the role of this summit simulation system is to provide instrument groups with test and debug environment.But debug of the SOSS itself and development and upgrade work for SOSS are more important for the system. This system prevents telescope time from being wasted and minimizes the annoying work at the summit. Since the summit simulation system has the identical or quite similar hardware as the summit system, it works as “Hot Standby” machines in case of hardware trouble of the summit system. Two-hour drive takes standby machines at the summit.

5. HARDWARE ENVIRONMENT FOR TEST AND DEBUG OF INSTRUMENTS

Instruments under construction in Japan need to be fully tested before shipping to Hawaii. Software simulator and other test environment support the test and debug of instrument software. As for hardware, we have built a special jig called “Telescope Simulator” or “Optical Simulator” shown in Figure 6. This simulator has a flange which shape is the same as that of the Cassegrain focus to hang Cassegrain instruments. The flange rotates like field derotator and the center section tilts like elevation movement of the telescope. The telescope simulator comes with emission lamps for wavelength calibration and continuum light source simulating a single star so that mechanical offset such as flexure can be measured. A telescope simulator was already built in Mitaka, Japan and another will be built in Hilo, Hawaii this year. These simulators will be remodeled so that prime focus instruments can be mounted on when rotating the center section by 180 degree. Cables such as fiber optic and twisted pairs are installed in the simulators and instruments on it are ready to hook up with their controller. Both simulators are controlled locally through PC but position or angle which an instrument is directed to are sent to an OBS in an simulation system and can be referred by OBCP just like status of the telescope.

Figure 5 Subaru Telescope Base Facility in Hilo, Hawaii

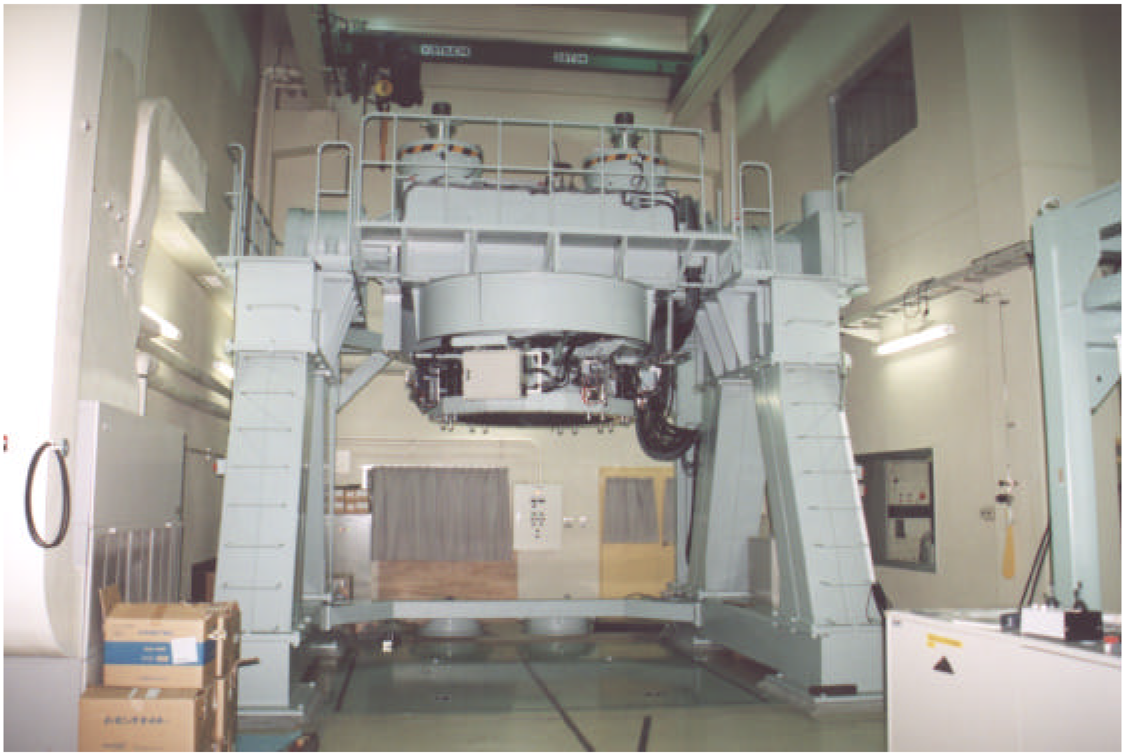

Figure 6 Telescope simulator built in NAOJ at Mitaka, Japan

REFERENCES

1.. T. Sasaki et al., “Concept Design of a Control System for the Subaru Telescope,” *Nucl. Inst. Methods in Phys.* Res. A 352, P. 75, 1994

2.. G. Kosugi et al., “Observation Scheduling with Subaru Control System,” *Proceedings of “Instrumentation for Large Telescope in the 21st Century”*, pp. 379 – 384, 1995

3.. J. Noumaru, G. Kosugi, T. Sasaki, T. Aoki, J.A. Kawai and A. Kidou, “Observation Scheduling and Data Acquisition with Subaru Control System,” *Proc. SPIE*, Vol 2479, pp. 453 – 461, 1995

4.. J. Noumaru and Subaru Computer Team, “Control and Data Acquisition System of Subaru Telescope,” *Proc. SPIE*, Vol. 2871, pp. 1041 – 1052, 1997