Experimental Evaluation of Data Flow Control on Three-Layered Architecture for Planetary Rover

Yasuharu Kunii, Masaru Furukawa, Yoshiki Matsui, Noriaki Mizukami

Chuo University, Tokyo, Japan
e-mail: {kunii, masaru_12, yoshiki_13, mizukami}@hmsl.elect.chuo-u.ac.jp

Abstract

Firstly, our proposed software architecture for tele-operated systems such as planetary exploration rovers is introduced. The architecture modularizes each function as a software module. It makes advanced extendibility and flexible task management by connecting with a database via a network. And the database manages the modules. As a result, it improves efficiency in development and operation stages of tele-operated systems. In this paper, we evaluate validity of the architecture from a view of variability. In order to evaluate it, the problems that should be solved in tele-operator are considered, and then we discuss and show how the proposed architecture responds to these evaluation criteria. Finally, the rover testbed with the prototype software shows the feasibility and validity of the architecture.

1 Introduction

Tele-operators (i.e., remote mobile robots) are used in extreme environments such as planetary surface, disaster sites, and other dangerous zones. In general, they are required to achieve a stable performance during advanced missions in those various environments. Several system architectures for robots have been proposed for achieving the capability [1]-[3]. Tele-operators are composed by several functions (e.g., action planning, recognition and motion control) and various subsystems (e.g., moving mechanisms, communication system, and various sensors). Since their systems are multifunctional, they tend to become bulky and complex. Conversely, the systems are required to be scalable and efficiently adapt to any situation. To solve these problems, their control and operating software should enable users to freely combine installed elements on them via a network, and to allow modification of system components. In order to flexibly respond to environmental changes or problems that are unpredictable, it is necessary to change the system configuration or add new functions, from a remote site over a network. Most of existing software architectures for tele-operator have some difficulties to operate a robot if a failure occurs in a remote site [4]-[6]. It arises from difficulties of dynamic modification of robotic functions. From this point of view, architecture with advanced scalability and variability is highly required for an actual mission operation. Tele-operated systems should also address the problem concerning to info-communication, i.e., the transmission of sensory information from a remote site to a human operator. Especially, for the safety operation of a robot, information of system conditions should be known. However, the complexity of the system makes it difficult to grasp the states of the system. To overcome these limitations, we have discussed and proposed a system architecture that emphasizes variability in the structure of functions, as well as data transparency [7].

In this paper, we discuss variability for a remote robot system first. Next, we show how the proposed architecture achieves variability. Finally, it is confirmed by experiment.

2 Proposed Software Architecture For Tele-operated Systems

In general, a tele-operator is operated at an area that is not amenable to human activity. Moreover, the environment of these locations may not be well known. Therefore, some failures on the system may occur because of the nonconformity or uncertainty of parameters or algorithms. In addition, harsh environmental conditions often cause hardware problems. In such cases, the system should be alterable by the software itself without physical restrictions; that is, the system should be flexible and adapt the structure of its functions to suit the situation. Moreover, as mentioned above, the safe operation of the robot requires knowledge of its state in the system during its operation. Therefore, we designed the system that emphasizes the flexibility (variability) in the structure of functions and transparency (accessibility) of data. In our proposed architecture as shown in Fig.1, each function is modularized, and connected via a network. Advanced variability is achieved by defining real and virtual connections within different layers. Each layer of the architecture is detailed in the following subsections.

2.1 Physical Layer

In a bottom layer, all hardware is connected via a network as shown in Fig.2. Thus, any function can be directly accessed, and its connections can be changed using the software operation, which imposes no physical restrictions. Thus, it is accessible and has an advanced variable structure.

2.2 Connection Layer

A robot operating from a remote site should switch among multiple tasks (comprising a module's behavior logic) in response to the situation. However, switching connection is too costly to become a reality in practice. For this reason, a middle layer of the proposed architecture manages the actual modules of the system, and virtually achieve task dependencies defined in a top layer. This action is performed by the database node module (DNM), which relays information between the functions of the modules. In particular, all modules are connected to the DNM as shown in Fig.3, and data are exchanged at a high speed via shared memory. DNM transmits the destination addresses of each module containing task dependencies defined by the user in the logical layer. In this way, the DNM builds out a network list. Hence, module connections can be switched by changing reference pointers, when DNM manages the timing of the switches.

2.3 Logical Layer

The top layer enables users to intuitively compose tasks. Therefore the efficiency of task development improves. Users collect the necessary modules and connect them according to the intended task flow as shown in Fig.4. Our method allows free swapping, addition, replacement, and deletion of modules. Thus, the system can effectively rebuild its functions and respond quickly to changing situations or problems encountered.

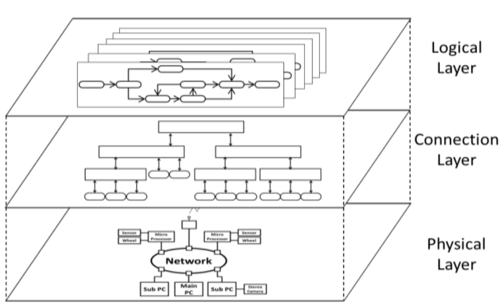

Figure 1. Proposed three-layered architecture for tele-operator

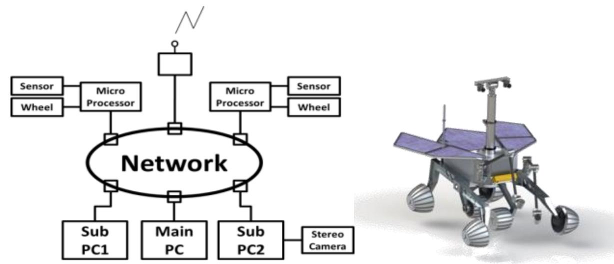

Figure 2. Network of hardware connections.

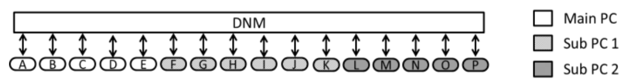

Figure 3. Connection of modules in DNM

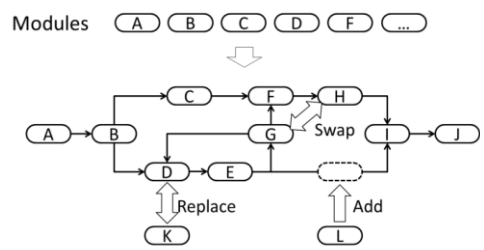

Figure 4. Example of task flow operation.

3 Discuss of Tele-operator System

In the first, evaluation items are considered in order to evaluate the validity of the architecture. And then, we show how this architecture can respond to evaluation criteria. Although a subject can be considered to tele-operated system on a development phase and an operation phase, we discuss an operation phase, because of an interest with tele-operation in this section.

3.1 Discussion of Evaluating Items for Tele-operation

Tele-operator is often operated in the environment where man cannot survive. Since man cannot fix remote robots, tele-operator itself needs to adapt a situation, if an abnormality of hardware or software is happened. Therefore, in order to continue missions, performing a system change from a remote site is required. Moreover, since the communication may be unstable and low speed, it is necessary to cope with a trouble through a low-speed communication. Therefore, the troubleshooting also needs to be performed efficiently. From the above, the architecture is required understanding states of the robot remotely, and coping with the troubles flexibly through a low-speed communication.

3.2 Approach of Proposed Architecture

In order to perform troubleshooting flexibly from a remote site, the advanced variability that can easily change a system configuration is required. The variability in the proposed architecture is achieved with the function that can easily change module connections. In the conventional method communicating between modules directly, data flow is realized by actual connection between modules. Therefore, since it is necessary to change actual destination of each module connection, large operating cost occurs. Moreover, the transmitting data and internal information of a module may be lost, if a trouble occurs on a module. Therefore, it is difficult to recover the system and its condition in emergency process.

In the proposed architecture, each functional module connects through special management module called DNM in which can virtually realize each connection of modules (Fig.5-7). It realizes software and hardware configurations with a net-list of module connections, and we can change the configuration by sending the net-list from a remote site through low capacity network. That is, data flow is independent from actual module connection, and furthermore it can arbitrarily switch the connections of hardware or software. Thereby, regardless of physical connections, the architecture realizes a system configuration management from a remote site under a low-speed communication. Furthermore, we can expect reprogramable function with installed modules on a robot by advancing unification of interface of module data communication. Moreover, since DNM is in charge of all data transmissions, it can preserve system information even if a functional module is down or terminated somehow (Fig.6). Thereby, the advanced variability for trouble management is achieved.

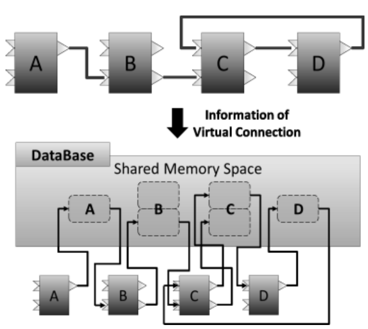

Figure 5. Analysis of virtual connections by DNM.

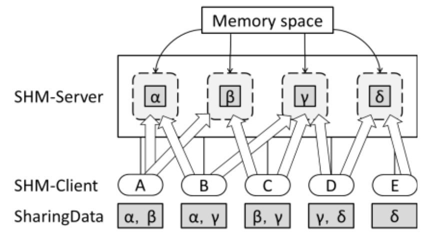

Figure 6. Shared Memory System.

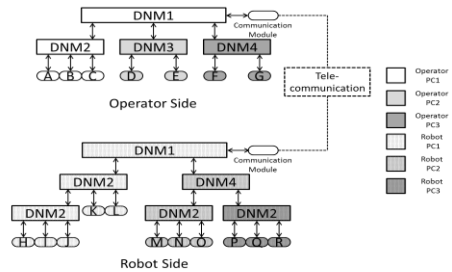

Figure 7. Connections between two systems in a tele-navigation system.

4 Experimental Evaluation

4.1 Micro6 Rover

Our planetary exploration rover test-bed Micro6 is shown in the right hand side of Fig.2. Micro6 has the four-wheel steering system in which each carries out an independence drive, and another fifth wheel is installed in the center of its bottom side for improved traversability. It is also possible to mount a manipulator that holds an object and puts in to an own pocket. Two stereo camera systems are mounted, which are for navigation with a pan-tilt gimbal and for manipulators without a gimbal respectively.

4.2 Software Implementation

Logic Layer is shown in Fig.8 (left) and Connection Layer is shown in Fig.8 (right). Virtual connection is built using the tool called Operation GUI, and it transmits to DNM. DNM controls modular data flow based on this information. In Physical Layer, the network assigns two PCs and the microcomputers of robots to the same LAN by a cable LAN, and each module is assigned to a separate PC. An operator performs tele-operation by accessing this network from a remote site.

4.3 Evaluation in a operation process

In the proposed architecture, the variability is achieved by control of the data flow between modules. Therefore, the data flow control as the fundamental function for the variability is discussed in this session.

In order to confirm variability of the architecture, module connections are switched in the actual system, as feasibility test. Based on connection information of modules in Logic Layer, DNM makes reference pointer list which modules refer to get its connecting information. If a module connection is changed, the pointing address will be updated immediately by DNM, module will get data from new reference, and as a result, the data flow will be changed. Table 1 shows the modules database in DNM during the data flow management process. This is a result of an operation under the assumption dealing with a system failure, in which a module using camera was changed to another module using LRF on Micro6. The actual operating result of data flow control is shown in Fig.9. After this data flow switching operation, sensor was successfully switched and Micro6 was able to continue its mission. As a result, the software and the architecture can realize stable data flow control, and it supported the confirmation of its variability.

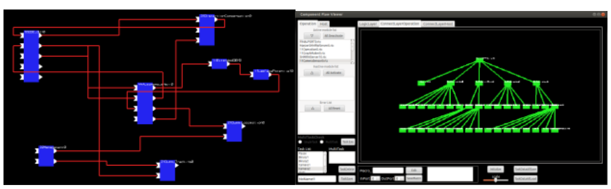

Figure 8. Implementation of Three-Layered Architecture.

Table 1. Modules database in DNM

ModuleDataBase	OutportNumber (C)	InportNumber (C)	OutportNumber (L)	InportNumber (L)
Camera_GW	1,2,3,4		1,2,3,4	
PanTiltController	5	2	5	2
ImageTransmittrer	6	1	6	1
VOLandmarkTracker	7	3,4,5,8	7	3,4,5,12
ChamereonCapture	8,9	10	8,9	
ShutterSpeed	10	9	10	
ParticleFilter	11	7	11	7
LRF_Capture	12,13		12,13	14
Laser_Power	14		14	13
CreateStereo	15		15	

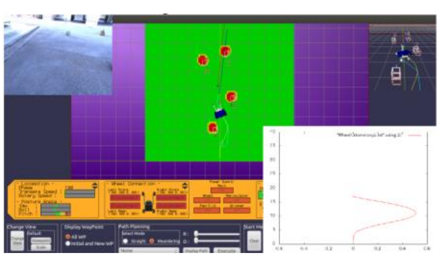

Figure 9. Operating result of data flow control

5 Conclusion

In this research, we introduced the architecture aiming at the increase in efficiency of the development and operation in a remote robot. And we evaluated the proposed architecture using the mobile robot in trouble situations. In the foundational evaluation in this paper, the proposed architecture can cope with trouble situations and difficult problems that are required to solve for the remote robot systems in recent years. Because we showed the advanced variability by the data flow control used as the proposed architecture for a remote robot system. We plan to evaluate a further development that is based on it in the future.

References

[1] H. Ahn, D. Lee, and S. Ahn, “A hierarchical fault tolerant architecture for component-based service robots,” Proc. of Int’l Conf. on INDIN, pp. 487-492 (2010).

[2] N. Medvidovic, H. Tajalli, J. Garcia, I. Krka, Y. Brun, and G. Edwards, “Engineering heterogeneous robotics systems: a software architecture-based approach,” Computer, vol. 44, pp. 62-71, May 2011.

[3] R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, and H. Das, “The CLARAty architecture for robotics autonomy,” Proc. IEEE Aerospace Conf., pp. 121-132 (2001),

[4] T. Estlin, R. Castano, D. Gaines, B. Bornstein, M. Judd, R. C. Anderson, and I. Nesnas, “Supporting increased autonomy for a Mars Rover,” Proc. Int’l SAIRAS (2008),

[5] P. Baranyi, “Vibrotactile force feedback for telemanipulation: Concept and applications,” Proc. of Int’l Conf. on ConInfoCom, pp. 1-6 (2011)

[6] K. Hoshino and Y. Kunii, “Three-Layered Architecture for Tele-operator and Its System Test,” Robot Intelligence Technology and Applications 2012 Advances in Intelligent Systems and Computing Volume 208,, pp 105-114 (2012)

[7] N. Ando, S. Kurihara, G. Biggs, T. Sakamoto, H. Nakamoto, and T. Kotoku, “Software deployment infrastructure for component based RT-systems,” Journal of Robotics and Mechatronics, Vol.23, No.3 pp. 350-359 (2011),