Proceedings of the 13th
KACC, October 1998

An Efficient Method to Develop Control Software of A Research Purpose Legged Mobile Robot

Hiroshi MIZOGUCHI, Ken-ichi HIDAI, Yoshiyasu GOTO,
Masashi TESHIBA, Takaomi SHIGEHARA, and Taketoshi MISHIMA
Department of Information & Computer Science, Faculty of Engineering, Saitama Univ.
255 Shimo Okubo, Urawa 338-8570, Japan
Tel & Fax: +81-48-858-9034
Email: hidai@me.ics.saitama-u.ac.jp

Abstract

This paper proposes a novel method to efficiently develop GUI based control software for a legged mobile robot. Although GUI is convenient, it is a very burden to both a computer and its developer. In case of the mobile robot, these problems are more serious. The proposed method solves these problems by separating GUI from control software. An implementation based upon the proposed method demonstrates its effectiveness.

1. Introduction

The authors are studying human robot communication [1] utilizing a research purpose legged mobile robot, TITAN-VIII [2]. As a part of the study, we have developed a network transparent control GUI (graphical user interface) program for the TITAN-VIII.
In these years GUI becomes very popular as Windows and Mac OS widely spread. Consequently even in robot control, GUI program is preferred. But it is difficult to develop GUI, because there are some problems.
This paper makes these problems clear and proposes solutions to them. The paper also confirms the proposed solutions by their implementation.

2. Problems to implement GUI program

Although GUI is very convenient for its users, it is a very burden to both a computer and its developer. Especially, in case of a mobile robot, these problems are more serious for the following reasons.

Problem 1: Execution load
GUI consumes a lot of CPU power and deprives a robot control process of the power. Hence a high-performance computer is necessary for GUI. Such computer is generally bigger, heavier and more expensive. This is the problem for the mobile robot, because onboard computers should be compact and light.

Problem 2: Developing load
Size of GUI program tends to be larger than character-based interface program. Debugging GUI is also harder. Thus more effort is needed to develop GUI program. Such effort is not desirable, because it wastes researchers' power. From the viewpoint of the research GUI is a merely means instead of purpose.

3. Solutions

To these problems we propose the following solutions.

Solution 1: Load distribution
Fig. 1 shows a solution to the first problem mentioned above. The idea of the solution is to separate GUI part from a control program for the mobile robot.

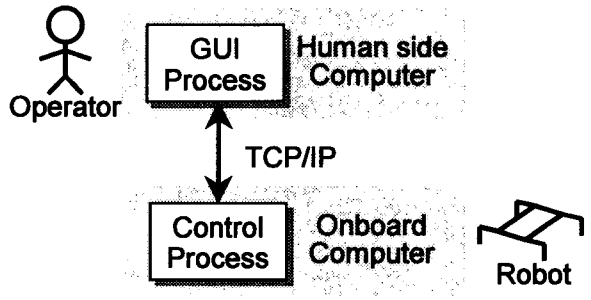

Fig.1: Separation of GUI from control process

We prepare two computers. One is a human operator side computer. And the other is a robot side computer. We divide a control program of the mobile robot into

two independent sub-programs. The sub-program on the operator side computer concentrates to process only GUI functions. And the other sub-program on the robot side computer processes all other functions except GUI. These two sub-programs communicate each other utilizing TCP/IP protocol.
In consequence of this, we can free the robot side computer from the load of the GUI functions. In addition, the program obtains network transparency. The human operator is possible to control the mobile robot through the Internet.

Solution 2: Selection of programming language
The control program is divided into two sub-programs as described above. These sub-programs are not necessary to be written in the same programming language. Thus you can select a proper programming language to write the GUI sub-program. And you can also select a proper language according to developer's skill. As a result, you are possible to develop the GUI sub-program easily. However, you can apply the above-mentioned method only if the programming language provides TCP/IP interface.

4. Implementation

Based upon the solutions mentioned above, the authors implement a network transparent control program for TITAN-VIII. Structure of the control program is shown in Fig. 2. Table 1 describes correspondence of programming languages to functions of sub-programs.

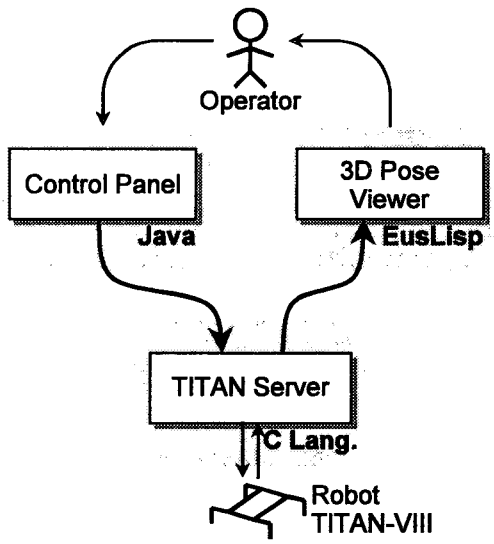

Fig. 2: Structure of control program

Table 1: Correspondence of languages to functions

Module name	Function	Language
Control Panel	GUI	Java
TITAN Server	Walking gait generation	C Language
3D Pose Viewer	3D geometry modeling	EusLisp

Terms in Fig. 2 are explained as follows.

TITAN-VIII: It is a research purpose 4-legged mobile robot kit. It is originally developed by Prof. Hirose of Tokyo Institute of Technology (TITECH) and now commercially available. Its specification is shown in Table 2.

Table 2: Specification of TITAN-VIII [3]

Dimension	400 * 600 * 250 [mm]
Numbers of leg	4
D.O.F	12 (Each leg has 3 D.O.F)
Weight	19 [kg] (Including motor driver. Not including computer and battery.)
Payload	5 to 7 [kg] (12[kgf]: In a short time.)
Limitation of walking velocity	0.3 [m/s] (B=0.75) 0.9 [m/s] (B=0.5) B: Duty Factor

Java: It is an object oriented programming language spreading rapidly in recent years [4]. Advantages of Java are as follows.
Firstly, Swing [5] enables you to implement advanced GUI simply and easily. Swing is GUI parts library included in Java. Secondly, Java makes it possible for you to implement TCP/IP interface with little effort. Lastly, using Java you are possible to execute your program on almost major platforms. The same binary code can execute even on different architecture.
However, as weakness of Java, it needs much CPU power.

EusLisp: It is an object oriented LISP developed by Dr. Matsui of Electrotechnical Laboratory (ETL), Japan [6]. EusLisp provides three-dimensional geometry modeler. It enables you to implement advanced three-dimensional geometry modeling easily. EusLisp has both interpreter and compiler. The interpreter enables you to develop

programming rapidly and interactively. But the interpreter sacrifices runtime performance. The compiler solves that problem. By compiling a program, you can execute it fast.

The network transparent control program consists of three sub-programs, TITAN Server, Control Panel and 3D Pose Viewer, as shown in Fig. 2. These sub-programs act in the following way.
TITAN Server continues to generate walking gait. It makes the robot to move.
Through Control Panel human operator inputs instructions. Control Panel sends the instructions to TITAN Server. TITAN Server changes walking gate according to the instructions.
3D Pose Viewer shows a robot pose of three dimensions. It sends requests of pose information to TITAN Server at regular intervals. In response to the requests TITAN Server sends the pose information to 3D Pose Viewer.
Among these three sub-programs, Control Panel and 3D Pose Viewer interact with human operator through display screen as shown in Fig. 2. Screen shots of these two sub-programs are shown in Fig. 3 and Fig. 4.

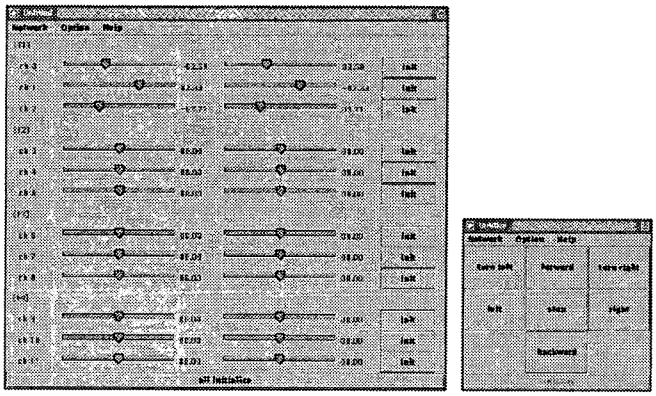

Fig. 3: Control Panel

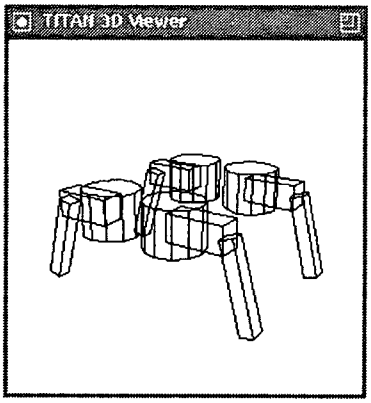

Fig.4: 3D Pose Viewer

We have succeeded to develop these sub-programs with little trouble for a short period. Size of Control Panel sub-program is about 300 lines, and 3D Pose Viewer sub-program is 170 lines. Both sub-programs have been developed for a few days.

5. Experiment

The authors conduct a simple experiment in order to confirm effectiveness of the solution 1. This experiment is done in the following way.
Control Panel sends a request of a joint angle to TITAN Server and receives value of the angle. Control Panel sub-program repeats this process. Fig. 5 illustrates flow of the experiment. The result is shown in Table 3.

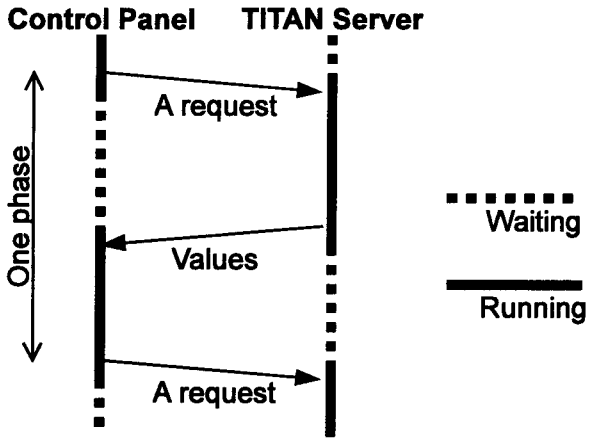

Fig. 5: Flow of this experiment

Table 3: CPU running rate

Module name	weighted CPU percentage	CPU
Control Panel	97.0~99.0	PentiumII 300MHz
TITAN Server	under 0.1	Pentium 75MHz

This result shows that GUI processing consumes CPU power approximately 100 times than robot control processing. It leads that the problem 1 is very serious in this program. In other words, effectiveness of the solution 1 is remarkably high.

6. Conclusion

The authors have developed a GUI control program for a legged mobile robot, TITAN-VIII. There are two

problems to implement and execute GUI. One is that the GUI consumes a lot of CPU power. The other is that developer needs to pay much effort to construct GUI. To these problems, the authors proposed two solutions. One solution is that we divide the program across the network into two different computers. The other solution is that we select proper programming language, which enables us to implement each program easily. The authors have also showed an implementation and confirmed the effectiveness of the solutions.

7. References

[1] Hiroshi Mizoguchi, Masashi Teshiba, Yoshiyasu Goto, Ken-ichi Hidai, Takaomi Shigehara and Taketoshi Mishima, "Security Problems and Protection Methods in Remote Control Communication for Mobile Robots Using Wireless IP Network (in Japanese)", Proceedings of the 16th Annual Conference of Robotic Society Japan, 1998

[2] Keisuke Arikawa and Shigeo Hirose, "Development of Quadruped Walking Robot TITAN-VIII", Proceedings of International Conference on Intelligent Robots and Systems (IROS'96), Vol.1, pp.208-214, 1996

[3] "TITAN-VIII", http://mozu.mes.titech.ac.jp/research/walk/TITAN8/TITAN8.html

[4] Ken Arnold and James Gosling, "The Java Programming Language (Java Series) 2nd Edition", Addison-Wesley Publishing, 1997.

[5] Sun Microsystems, Inc., "The Swing Connection", http://java.sun.com/products/jfc/tsc/index.html

[6] Toshihiro Matui and Isao Hara, "EusLisp version 8.00 Reference Manual Featuring Multithread and XToolKit", http://www.etl.go.jp/~matsui/, 1995.

[7] Satoshi Kagami, Mitsutaka Kabasawa, Kei Okada, Takeshi Matuki, Yoshio Matsumoto, Atushi Konno, Masayuki Inaba and Hirochika Inoue, "Design and Development of a Legged Robot Research Platform JROB-1," IEEE ICRA98 Proceedings, pp.146-151, 1998.

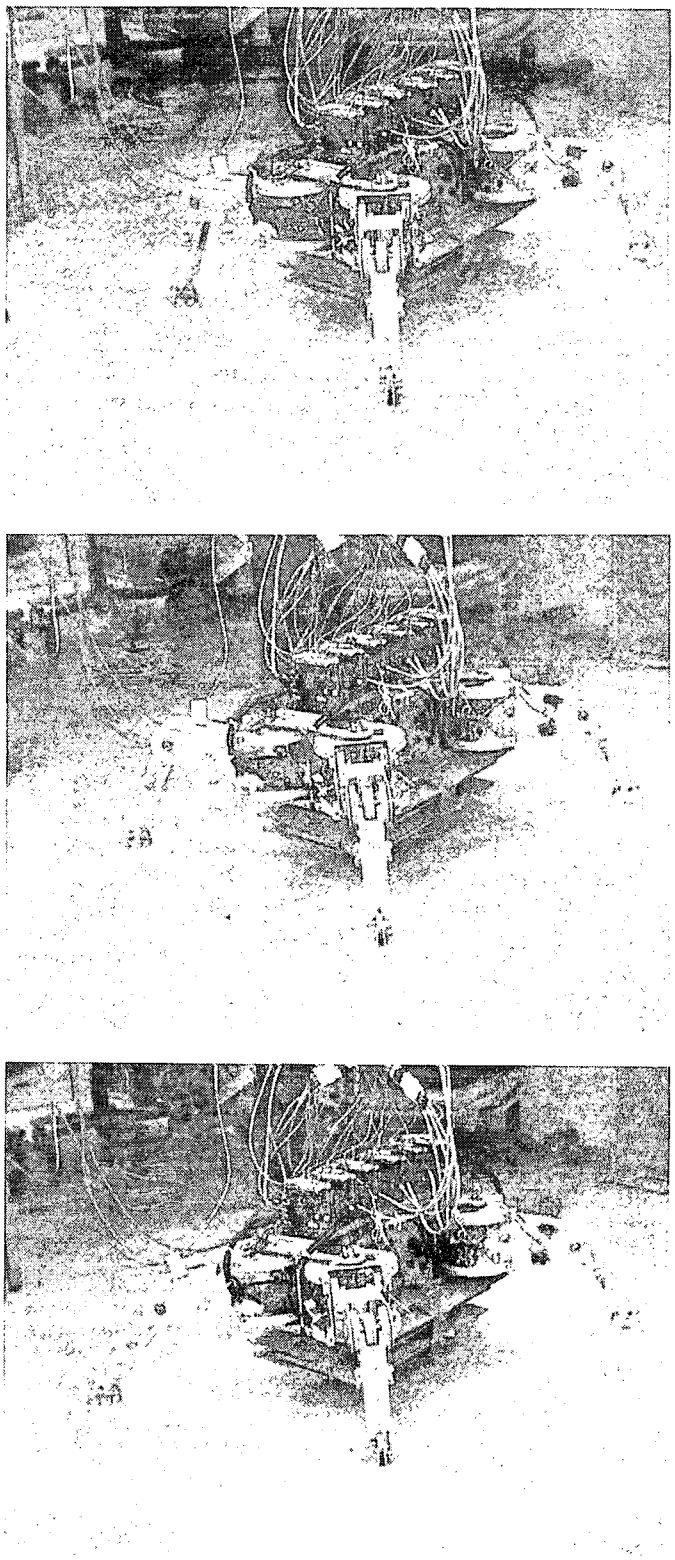

Fig. 6: Walking TITAN-VIII