Asian Journal of Information Technology 7 (2): 58-63, 2008
ISSN: 1682-3915
© Medwell Journals, 2008

Development of a Tele-Robot Programming System for Programming Education on Ubiquitous Environment

[1]Inhwan Yoo and [2]Youngkwon Bae
[1]Department of Computer Education, Daegu National University of Education, Namgu Daegu, 705-715, South Korea
[2]Department of Computer Education, University of Mokwon, Seoku Daejeon, 302-729, South Korea

Abstract: Robot programming requires special knowledge of both robot and programming. In addition, for teaching-learning robot programming, various kinds of hardware and software are necessary and they are usually so expensive and complex that it is not easy to operate. To overcome these limitations, the present study developed a tele-robot programming system. This system provides solutions for controlling robots in ubiquitous computing environment using several freeware robot programming languages. Learners can connect the system and code robot controlling programs in different programming languages. Users can transmit byte codes generated from compiling to a server robot through infra-red or Bluetooth and get debugging information by watching the operation of the robot through a Web camera. In addition, if a user has a robot, he can download compiled codes from the server and execute them. Using the system only with PC or PDA connected to the Internet, learners can learn robot programming without additional hardware or software. In particular, the system implemented a language processor so that elementary students in non-English-speaking regions can approach and learn programming easily in their own language. Because these functions make it easier for learners to make the initial approach to programming, the system is expected to stimulate learners' interest in programming and improve the efficiency of learning.

Key words: Robot programming, programming education, tele-robot programming, ubiquitous environment

INTRODUCTION

There have been arguments for against the educational value of programming. According to previous researches, some argue that programming with computer languages is too difficult and complicated, so does not have effect on the development of problem-solving abilities, but most researchers have reported that it has a positive effect (Beer *et al*., 1999; Hwang and Blandford, 2000; Lee *et al*., 2001; Choi, 2002).

In addition, negative opinions usually point out problems caused by educational methods or inadequate educational environment rather than limitations in the educational value of programming. From the selection of adequate programming languages to purchase, installation and teaching, programming education requires a lot of efforts, time and budgets.

Accordingly, it is highly likely that the effect of education appears negative, because it is difficult to execute proper programming education due to these practical problems. After all, today's school education is gradually getting away from programming and being focused on the application of computers.

Thus, the present study purposed to develop a tele-robot programming system. This system provides solutions for controlling robots in ubiquitous computing environment using several freeware robot programming languages.

Related work: Programming robot provides some advantages in computer science. Several researcher used robotics programming to teach problem-solving technologies, basic computer programming concepts and embedded computer systems, computer simulation, etc (Flowers and Gossett, 2002; Fagin and Merkle, 2003; Garrett and Thornton, 2005; Klassner and Continanza, 2007). Furthermore, robotics has proven to be a powerful educational device of encouraging cooperative problem solving and team work in computer science (Beer *et al*., 1999; Hwang and Blandford, 2000).

Corresponding Author: Youngkwon Bae, Department of Computer Education, University of Mokwon, Seoku Daejeon, 302-729, South Korea

Some of the most compelling studies have focused on two aspects motivation of learning (fun factor) and understanding as students could see with their eye the execution of their algorithm in physical objects (Fagin *et al.*, 2001; Flowers and Gossett, 2002; Stevenson and Schwarzmeier, 2007). This study implies that programming education using robots can teach not only programming but also information technologies and is an integrated education for enhancing problem-solving abilities. In addition, Barnes (2002) conducted a research using robots in the course of introduction to Java and reported that robots are a kind of physical models controlled by computers and are much more educationally meaningful than simulation models in learning programs and learners take them very interestingly.

The implication which can be drawn from previous researches is as fallows. First, robot programming has many educational values and particularly for students at the introductory stage of programming, they can provide opportunities to learn programming in an easier and more interesting way (Cole and O'Connor, 2003; Cobb, 2004; Kumar, 2004). Second, robots are related to various engineering technologies they enable integrated education of basic technologies. Robots can provide a new paradigm of computer education (Portz, 2002; Matson *et al.*, 2004; Garrett and Thornton, 2005). Third, robot education programs can be meaningful themes for developing advanced abilities such as creativity, problem-solving ability, decision-making ability, cooperative problem solving and critical thinking ability (Beer *et al.*, 1999; Hwang and Blandford, 2000; Yoo, 2005; Bae, 2006). Fourth, robots are good educational materials that meet the nature of humans as instinctive desire for manipulation. It used robotics as a method of visual and tangible programming solutions (Choi, 2002; Fagin and Merkle, 2003). Accordingly, the present system aimed at the development of a system through which learners at the introductory stage of programming can approach robot programming easily.

POSSIBILITY OF ROBOTS IN PROGRAMMING EDUCATION

Robot programming is helping learners understand basic engineering concepts and principles including computers through self-directed inquisitive activities and this shows that the use of robots can play a significant role in selecting and organizing the contents of computer education and applying methods.

It is highly possible for the educational use of robots to be settled as meaningful and interesting educational activities for understanding technologies and the world in the cognitive, affective and psychomotor domains (Vanhorn, 2005). Breaking away from the conventional frame of education based on software, it can be a new paradigm of computer education. Through his pedagogical analysis on mathematics teaching-learning in computer environment. Lee (1999) pointed out that computer-based class should induce the students to focus their thinking and acting on the manipulation of objects rather than on things displayed on the computer screen and that it is necessary to emphasize students' meta-cognition functions controlling and adjusting their activities and knowledge in order to form new knowledge in computer environment.

If a robot is structured and programmed through trials and errors, it is hardly likely to run properly. Learners should assemble a robot based on a design for the 'operation' of the robot and program the robot without even a minor error for its accurate operation. It is not an activity of manipulating an object on the screen unconsciously but requires the use of meta-cognition functions. Thus, the robot making and programming process is similar to Piaget's reflective abstraction concept that reflects on and abstracts one's own behaviors. In the robot programming process, learners understand concepts and principles by themselves and achieve creative inventions. In order for computer education to be free from existing criticisms, it should pursue cognitive development through reflective abstraction in teaching the concepts and technologies of information communication and for this robot programming can be a good alternative.

In addition, robot programming can develop an attitude of thinking rationally and scientifically. It promotes abilities to learn the principles of basic science and think logically in the process of predicting engineering principles and programming using gears, levers, pulleys, wheels, axles, etc. In addition, children's development can be supported by their experience in making a robot by themselves rather than simulating it on the computer. What is more, the learning process involves many discussions on the design of robot structure, assembly and programming and such discussions can refine the attitudes of respecting others' opinions and solving problems creatively. In this way, robot programming is matched well with the aims of computer education in the cognitive, affective and psychomotor domains and thus is expected to contribute to raising creative intellectuals equipped with information communication technologies.

EDUCATIONAL ROBOTS AND PROGRAMMING TOOLS

The present study made a technological examination of commercial educational robots and decided to develop a system using two kinds of robots-Lego Mindstorms and

Ez-RoboMaster-that can be handled easily by elementary students and can make various activities using several sensors and other means.

The Mindstoms system is a product for the development of robots and automation applications. The Mindstorms NXT is a new product released in 2006. This set has a 32-bit programmable microcontroller, sensors, interactive Servo Motors, wireless Bluetooth and USB connection. With this system, students can easily design-program their robots and test it to see if it matches their anticipations. Furthermore, students can rebuild and reprogram it as their wish (Ferrari *et al*., 2002; Yoo, 2005).

Ez-RoboMaster provides DC motors, step motors for driving the robot and can be mounted with a driving part and controlled easily through its built-in driver. It also has ports for receiving signals from sensors as well as for serial communication and Bluetooth communication. MPU, which is the most important component of the controller, uses ATmega128, one of AVR® series. Ez-RoboMaster is programmed basically using GNU GCC as a compiler and supports convenient programming by providing integrated development environment called Ez-CPU. The integrated development environment contains various API functions for robot control and functions to control through GCC. The biggest advantages of Ez-RoboMaster are that it can share sensors and components with Lego Mindstorms, that it can control sensors, step motors, servo motors, etc. made by different manufacturers and thus sophisticated robots can be made and that programs can be coded efficient using the C language.

Mindstorms and Ez-RoboMaster have various programming tools, but tools for developing Web-based systems are NQC, BrickOS (GCC), WinAVR (GCC), NXC, etc. Not Quite C (NQC) is a programming language, API and byte-code compiler for the Lego Mindstorms, but NQC has limitations as a robot programming language. To overcome the limitations, brickOS (GNU GCC) are used together brickOS is an operating system for the Lego Mindstorms RCX Controller. It is a C/C++ development environment for RCX programs using gcc and g++, so it can be used to download programs to the RCX. Next Byte Codes (NBC) and Not eXactly C (NXC) can be used to program the NXT programmable brick. NBC is an easy language with an assembly language syntax. NXC is a hard level language, built on top of the NBC compiler. WinAVR is used to program AVR, the MPU of Ez-RoboMaster, on a Windows platform. It is an executable, open source software development tools for the Atmel AVR series of RISC microprocessors. It has the GNU GCC compiler for C and C++.

SYSTEM DESIGN AND IMPLEMENTATION

This system provides a service, with which learners can experience robot control programming using PC or PDA and a Web browser without expert knowledge of robotics and related hardware and software. In addition, it can control various kinds of robots, supports various compilers and allows programming using Korean keywords. Connecting to the system, learners can choose a robot and a compiler and code a program for controlling the robot. The generated byte code can be transmitted to and executed in a robot linked to the server through infra-red or Bluetooth communication. In addition, learners monitor the operation of the robot using a Web camera and get debugging information. What is more, if a user has a robot, he can download the compiled code into his robot and execute it.

In this way, learners can have programming activities for controlling finished robots provided in a server-based system or can design and make robots by themselves and make programs to control the robot in a client-based system. This can be diagramed as follows (Fig. 1).

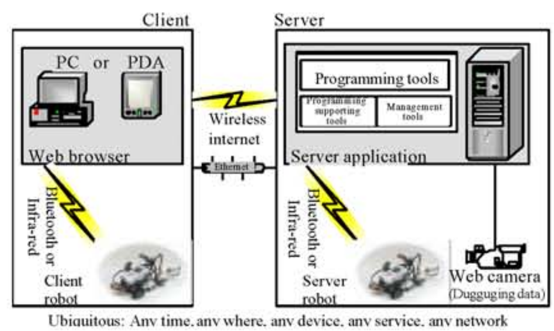

Fig. 1: Schematic diagram of tele-robot programming system

Fig. 2: Tele-robot programming and monitoring

This system is composed of major modules such as programming tool, programming support tool and management tool (Fig. 2).

The programming tools module executes key tasks of robot programming such as coding and compiling and the programming support tools module provides functions such as file management, programming helper, Korean multiple keywords and sharing among users. In addition, the management tools module provides various support tools for system management. As in Fig. 2, a robot linked to the server can be monitored through a web camera.

LANGUAGE PROCESSOR

Keyword translator module: This module enables coding and compiling with English or Korean keywords as in Fig. 3.

For non-English-speaking students, it is difficult to understand the meaning of keywords and memorize the spelling and this hinders them from concentrating on implementing algorithms. To solve this problem, the system provides a function that converts a source code from Korean to English. This function will be helpful to beginners so that they can learn coding easily using Korean keywords and then practice English keywords step by step. With this function, users can designate Korean keywords arbitrarily and multiple Korean keywords for each English keyword. As in Fig. 4, if a user defines ' main =몸통=메인' he can use '몸통' or instead of 'main' as a Korean identifier.

Keyword definition rules are as follows:

- Lines beginning with '//' are comments.
- Lines beginning with '#__' are system variables.
- A Korean keyword and its corresponding English keyword are coded one-to-one on the same line.
- Special signs and empty lines are not allowed.
- Only Korean keywords can be multiple (support up to 2) and are distinguished by '='.
- Keywords should not be redundant.
- A keyword used in a compiler can be used redundantly in a different compiler.

Messages translator module: For non-English-speaking students, the major compile error messages are presented in their first language (Pre-defined compile error message DB) so that beginners and children can understand their meanings easily.

One of difficulties in learners' programming is that error messages are produced in English not understandable to the learners. To solve this problem, we designed a module as in Fig. 5 that presents compile error

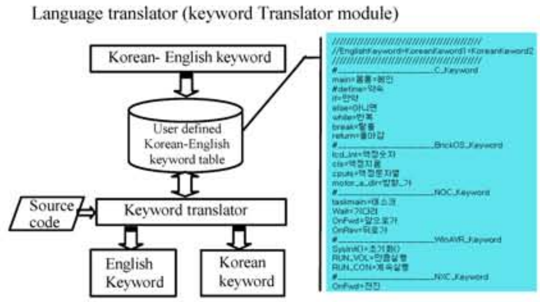

Fig. 3: Keyword Korean-English conversion

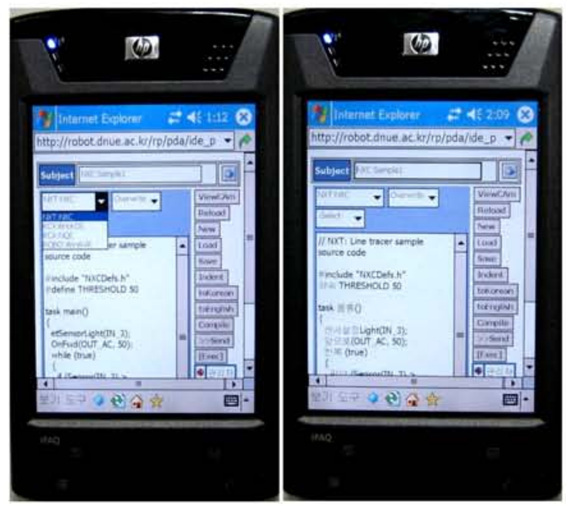

Fig. 4: a) English keyword, b) Korean keyword

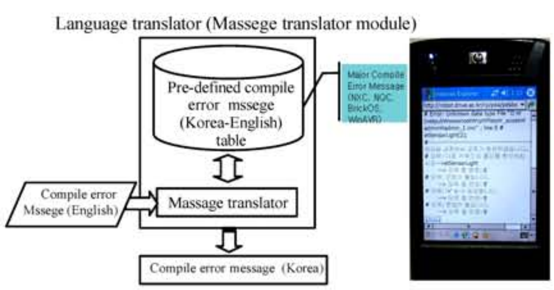

Fig. 5: Compiler error messages in Korean

messages in Korean using a database containing errors happening frequently in students' programming.

Compile process: This module compiles a source code and generates an execution code (Fig. 6). It supports compliers such as NXC, NQC, BrickOS and WinAVR and two types of robot controllers RCX and Ez-RoboMaster. The major functions of this module are compiler selection

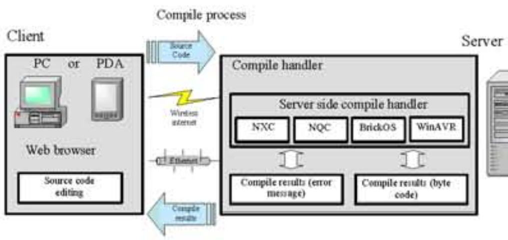

Fig. 6: Compile process

and compiling. This system interface we have developed uses a client-server architecture. This process is as fallows. First, the learner logs on the system. Second, the learner selects one robot and one compiler (NXT/NXC, RCX/NQC, RCX/BrickOS and Ez-RoboMaster/WinAVR). Third, the learner writes source code and sent it to server. Fourth, The compile handler at the server execute compile through selected compiler and send the results to client. The results are error message or byte code.

DISCUSSION

We introduced the system to 8 specialists (4 teachers who had experiences in teaching programming and 4 researchers or doctors in computer education), had them try the system and conducted a questionnaire survey.

The meaningful major contents are summarized as follows:

- The developed system provides various compilers in integrated development environment and supports the use of Korean keywords, Web camera, etc., which distinguish the system from existing programming education environments.
- The developed system can solve problems in the purchase and installation of compilers for programming education.
- The developed system contributes to learners' deeper understanding of robot programming and easier robot programming.
- Remote robot programming is a good educational method. However, it requires students to assemble a robot and operate it by themselves. Thus, remote support technology can be utilized efficiently in basic learning or home-based learning of robot programming.
- As the system provides environment for learners to practice programming at any time and in any place, it expands opportunities for learning and improves the continuity of education.

On the other hand, we found limitations and points to be improved as follows. First, UI (User Interface) needs to be improved. As this system was developed for research, its UI is hardly acceptable to elementary students. Thus, we need to design UI to be more friendly and interesting to elementary students and more convenient for their learning. Second, we need to improve system stability and the speed of compiling. The speed is slow compared to compiling on a PC, because codes are compiled on a Web server. Thus, it is desirable to use this system at the introductory stage of robot programming and to use a PC for actual programming. Third, learners need to be provided with opportunities to make robots. This system supports robot programming without a robot. In actual learning, however, we need to find a way of providing learners with opportunities to make robots by themselves. Fourth, the Web camera and the server robot used in this study have limitations. The Web camera allows the learners to observe the operation of a robot but if the robot makes unwanted operation the error should be corrected by human. Thus, this function can be fully utilized only when there are operators stationed. This problem may be solved by providing simulation instead of the operation of a real robot.

CONCLUSION

In the knowledge and information society, computer is the base and tool of creative knowledge activities. Thus, developed countries are making huge investments in the information of education and the education of talented students. In Korea as well, various institutions are being founded and operated to educate students talented in computer and most of them are teaching programming languages, but their teaching methods are not much distinguished in enhancing creativity. At higher educational institutions such as universities, on the other hand, programming languages are mostly compulsory courses for computer-related majors, but the grammar-oriented teaching of such courses has limitations in enhancing students' interest in and understanding of programming languages and developing their abilities to use the languages.

Robot making and programming activities are reflecting on and abstracting one's own behaviors, similar to Piaget's reflective abstraction concept. In robot programming, the learners understand concepts and principles by themselves and make creative inventions. In order for computer education to be free from existing criticisms, its teaching method should promote cognitive development through reflective abstraction and robot programming was proposed as an alternative in this study.

The characteristics of this system are as follows. First, on the Web-based system, users can perform basic programming works including coding, compiling and debugging. Second, the system supports various types of educational robots and robot compilers. Third, the system provide environment in Korean so that learners can learn programming and do coding easily in Korean. Fourth, the learners can use the system conveniently without installing or setting up software. Fifth, the system is adaptable to future environment such as ubiquitous environment, supporting not only PC but also PDA.

ACKNOWLEDGEMENT

This research was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-041-B00611), in which main calculations were performed by using the supercomputing resource of the Korea Institute of Science and Technology Information (KISTI).

REFERENCES

Bae, Y.K., 2006. A Study of the Robot-using Education Model for Supporting a School System with a Five-day School Week. J. Curculum and Evaluation, 9: 95-119.

Barnes, D.J., 2002. Teaching introductory Java through LEGO MINDSTORMS models. Proceedings of the 33rd SIGCSE technical symposium on Computer science education. Cincinnati, Kentucky, USA., pp: 147-151.

Beer, R.D., H.J. Chiel and R.F. Drushel, 1999. Using Autonomous Robotics to Teach Science and Engineering. Commun. ACM., 42: 85-92.

Choi, Y.H., 2002. Development of the Program Model for Educational ROBOT. J. Korean Pract. Arts Edu., 16: 75-90.

Cobb, C., 2004. Education by robot. Tech Directions, EBSCOHost database (Academic Search Premier, AN: 12911265) http://www.epnet.com., 63: 15-17.

Cole, L. and J. O'Connor, 2003. The nuts and bolts of robot building with kids. Technol. Directions, EBSCOHost database (Academic Search Premier, AN: 9082248) http://www.epnet.com., 62: 19-22.

Fagin, B.S. and L.S. Merkle and T.W. Eggers, 2001. Teaching computer science with robotics using Ada/MINDSTORMS 2.0. ACM SIGAda Ada Letters, Proceedings Annual ACM SIGAda. Int. Conf. Ada, 21: 73-78.

Fagin, B. and L. Merkle, 2003. Measuring the effectiveness of robots in teaching computer science. ACM SIGCSE Bull. Proc. 34th SIGCSE Technical Symp. Comput. Sci. Edu., 35: 307-311.

Ferrari, G., A. Gombos, S. Hilmer, J. Stuber, M. Porter, J. Waldinger and D. Laverde, 2002. Programming Lego Mindstorms with Java. Rockland, MA: Syngress Press.

Flowers, T.R. and K.A. Gossett, 2002. Teaching problem solving, computing and information technology with robots. J. Comp. Sci. Colleges, 17: 45-55.

Garrett, A. and D. Thornton, 2005. A Web-based Programming Environment for LEGO Mindstorms Robots. Proc. Ann. Southeast Regional Conf. Kennesaw, Georgia, USA., pp: 349-350.

Hwang, D.J. and D.K. Blandford, 2000. A multidisciplinary team project for electrical engineering, computer engineering and computer science majors. In: Proc. ASEE Annual Conference, St. Louis, Missouri, USA, Session 2553.

Klassner, F. and C. Continanza, 2007. Mindstorms without Robotics: An Alternative to Simulations in Systems Courses. Proc. 38th SIGCSE Technical Symp. Comput. Sci. Education, Covington, Kentucky, USA., pp: 175-179.

Kumar, A.N., 2004. Three years of using robots in an artificial intelligence course: Lessons learned. J. Edu. Res. Comp., 4: 1-15.

Lee, J.Y., 1999. A Didactical Analysis of the Mathematics Teaching and Learning in the Computer-based Environment. Doctoral dissertation, Seoul National University Graduate School. Seoul.

Lee, T.W., I.H. Yoo and C.H. Lee, 2001. ICT Educational Theory. Seoul: Hyung Seul Press.

Matson, E., S. DeLoach and R. Pauly, 2004. Building Interest in Math and Science for Rural and Underserved Elementary School Children Using Robots. J. STEM Edu. Innov. Res., 5: 35-46.

Portz, S.M., 2002. Lego League. Tech Directions, EBSCOHost database (Academic Search Premier, AN: 6691603) http://www.epnet.com., 61: 17-19.

Stevenson, D.E. and J.D. Schwarzmeier, 2007. Building an Autonomous Vehicle by Integrating Lego Mindstorms and a Web Cam. Proceedings of the 38th SIGCSE technical symposium on Computer science education. Covington, Kentucky, USA., pp: 165-168.

Vanhorn, R., 2005. Marvelous toys and educational robots. Phi Delta Kappan, EBSCOHost database (Academic Search Premier, AN: 15611507) http://www.epnet.com., 86: 408-409.

Yoo, I.H., 2005. The Possibility of Robot Programming to Enhance Creative Problem-Solving Ability. J. Edu. Stud., 36: 109-128.