International Journal for Educational Media and Technology
2011, Vol.5, No. 1, pp. 97-106

VRES: A Web-based Education System for NXT Robot

Xiong Hongyu
Sunchon National University, KOREA
Jun Youngcook
Sunchon National University, KOREA
So Wonho
Sunchon National University, KOREA

In this paper, we introduce a web-based development environment called Virtual Robot Education System (VRES). VRES is designed to assist students to learn programming with the Lego Mindstorms NXT Robot. In this development environment, students can write codes online through a web browser. They can also remotely compile, download, and run their program, then monitor the behaviors of the robot. Students can use this system to learn robot programming during or after regular classes with Internet and a web browser. We also design a new simple programming language and a corresponding compiler to facilitate programming. With that language and compiler, middle and primary school students can also easily create a program for NXT robot.

Keywords: web-based, e-learning, Lego Mindstorms NXT, robot

Introduction

LEGO Mindstorms Robot is very popular in robotics education. The latest version of it is called NXT (Kim, S. et al. 2007), which is a very expensive commercial product. It is difficult to give every student a robot to learn. Therefore, teachers find ways to share the robot and increase the utilization ratio of the robot. Recently, there are many web applications available in the Internet. These web-based applications overcome the limitations of time and space.

This article introduces a web-based programming environment for robotics education. Figure 1 depicts the basic scenario of the proposed VRES. This education system can serve huge number of students, who may be located in different regions. With this system, users can create, compile, and download programs to control NXT robot. Users can monitor the state of robot remotely through two videos in different views and an audio on the web. Initially, the system supports Java programming only. However, it is a little difficult for middle and primary school students to write code in Java. Therefore, a new script-like language has also been designed, and an interpreter is offered by VRES.

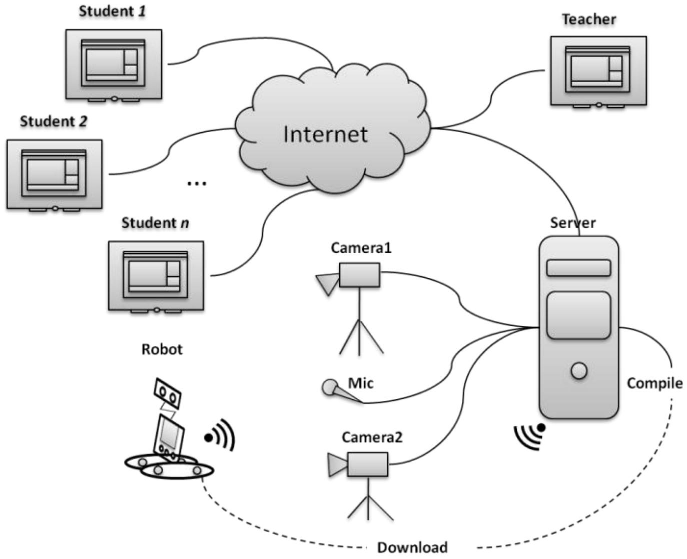


Figure 1. *Application Scenario of VRES system for distributed students and teacher*

Related Work

Some existing studies and projects related to education or robots are reviewed in this study. A web-based education system for predicate logic is showed in (Hatzilygeroudis et al. 2004). It is an example with basic frame of web-based education system. A JINI based distributed platform can remotely control the LEGO Mindstorms RCX robot is proposed in (Gonzalez 2002). But that was not a web-based system. There is a web robot learning powered system by Bluetooth implemented in (Sagiroglu 2006). That project implements a web based robot control center which can control robots through web. However it does not support programming with robots. (Cedazo 2007) introduces a platform called Ciclope, which is a free software project for web-based laboratories which include robotics. The aims of Ciclope projects are to build educational web labs to optimize universities and to improve the courses delivered at technical universities.

According to the existing work (Hazel 2009, Khen 2009, & Foord 2009) for web based programming environments, it is practical to create a web-based Java programming environment for NXT robot. There is an open source project called LeJOS (Solorzano et al. 2009) which provides a Java based firmware can be deployed in NXT robot. With LeJOS, Java program can be used to control NXT robot.

System Design

The functions required by VRES are provided in following modules: *Login, File Manager, User Manager, Editor, Compiler, Loader, Monitor, Robot Access Controller, and Instant Commander*. (Figure 2) *Login* guarantees the user is a registered user through ID and password verifying.

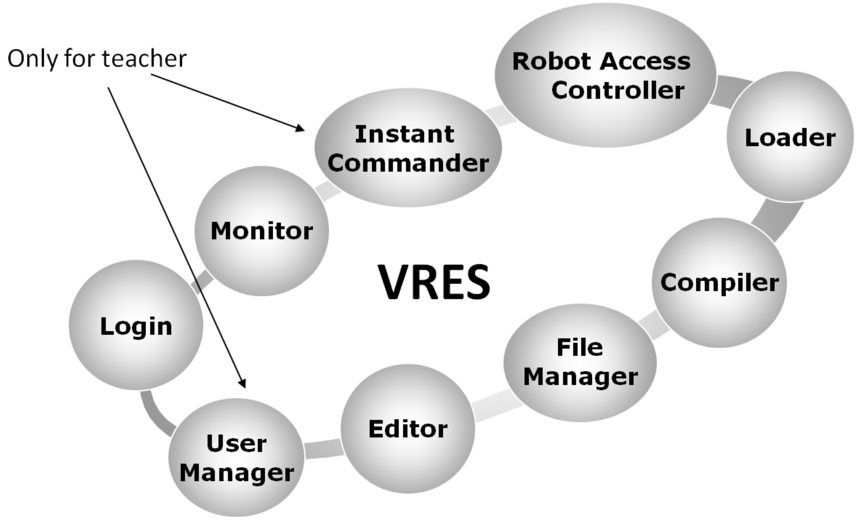


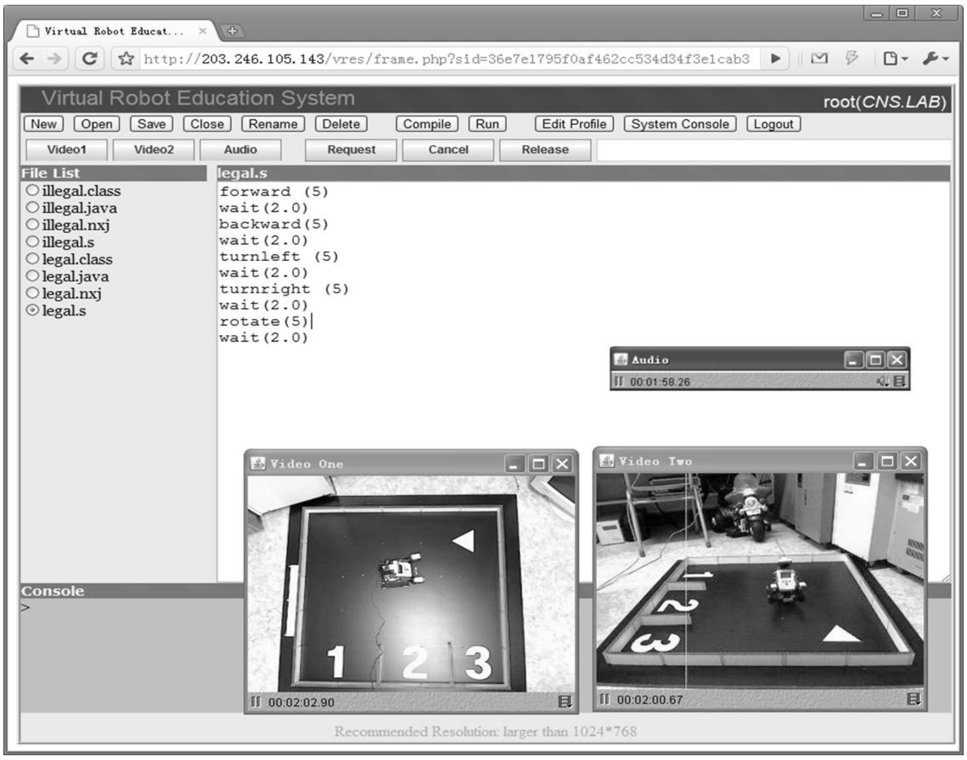
Figure 2. *Function modules of VRES system*

There are two types of user for that system. One is normal user for student. The other is *administrator* for teacher. *File Manager* shows files belong to the login user. And the owner of files can rename or delete them by this module. *User Manager Module* is used to add, delete, or modify a user of this system. *Editor* is used to edit the source code on the web page. *Compiler* converts the source code organized in a simple programming language into Java object code. In this paper, the simple programming language is called NXJ Script (NXJS). *Loader* is used to download the executable program file to the robot, and run it in the robot. *Monitor* is implemented in Java which has consists of a remote server and a client. The server captures video and audio of the place where the robot is located. The client can play video and audio on the web. With this mechanism, users can monitor the action of the robot remotely. *Robot Access Controller* is designed to assign the robot for users. It provides two model of service. Under *Auto Mode*, there is a FIFO (First In First Out) queue system working in the server, users follow the schedule to access the robot one by one. Under *Manual Mode*, users can only access robot when teacher assigns the robot to them. *Instant Commander* provides the instant controlling interface for the robot. Teachers can control the robot instantly.

The web interface for both users and administrators is showed in Figure 3. The interface frame consists of various elements such as buttons, name list of the files, Java applets for monitor, text-area for source code, and text-area for responses. Figure 4 shows the frame of system console which can only used by system administrator. In system console teacher can create new user, show online users and existing users, show robot state, and instantly control the robot.

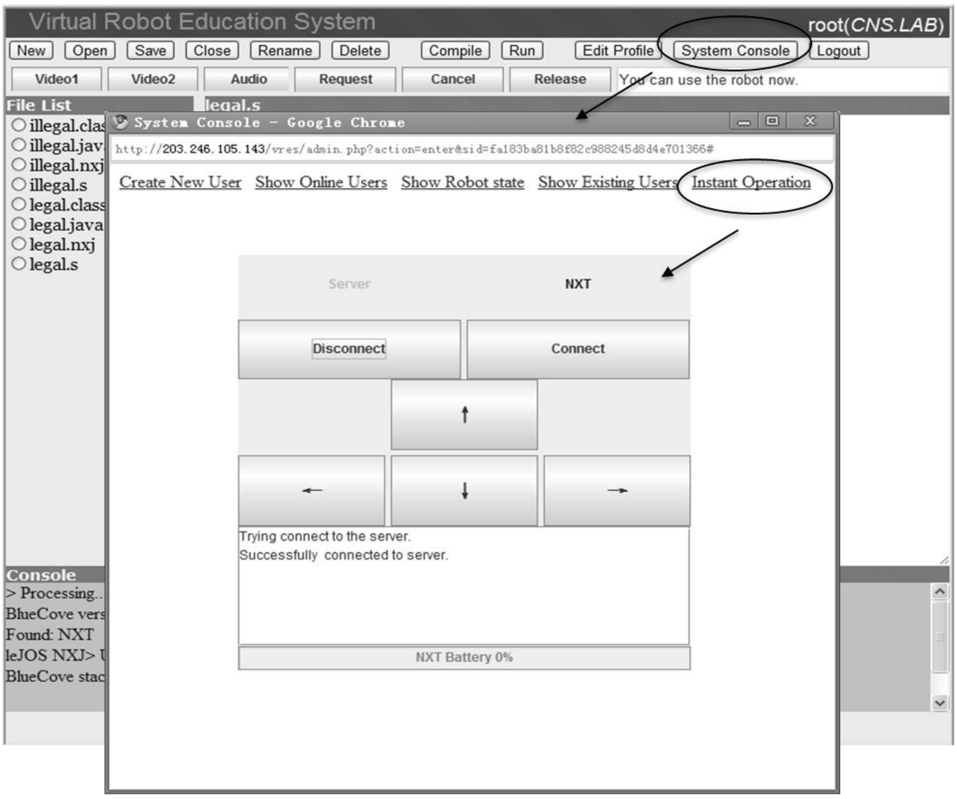
Implementation Methods

Application programming interface and a collection of developing tools are supplied by an open source project called LeJOS NXJ. It provides a plenty of packages for programming NXT robot. Programmers can write source code in Java, and then compile it into Java byte codes which can be executed by LeJOS in NXT robot.



```
forward (5)
wait(2.0)
backward(5)
wait(2.0)
turnleft (5)
wait(2.0)
turnright (5)
wait(2.0)
rotate(5)
wait(2.0)
```

Figure 3. *Interface of VRES front-end with two videos and one audio*



You can use the robot now.

Trying connect to the server.
Successfully connected to server.

```
> Processing..
BlueCove vers
Found: NXT
leJOS NXJ> U
BlueCove stac
```

Figure 4. *Interface of system console with instant command tab*

In VRES, apache web server is used to provide web service, and PHP dynamic web page is applied to support interacting with client. Many web pages contain PHP, HTML, and JavaScript programs were created for each module. *Login, File Manager, Editor, User Manager*, these modules can be implemented through very conventional methods. Thus, the following paragraphs describe the details about the implementation for *Compiler, Loader, Monitor, Robot Access Control,* and *Instant Commander* modules separately.

Compiler: When users click the compile button on the web page, the php program will invoke 3 external executable files in the server to do the compiling. The schema is showed in Figure 5.

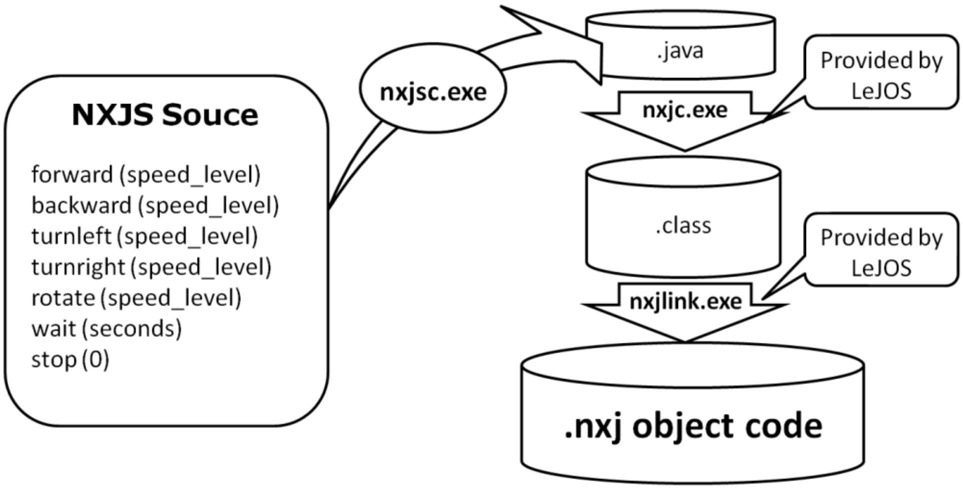


Figure 5. *Process and tools for compile NXJS source*

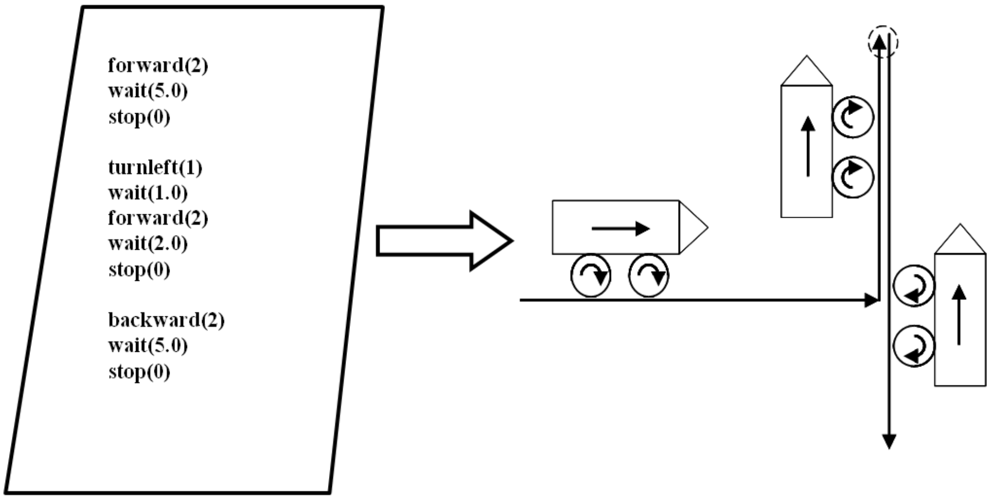
There are 3 external programs— *nxjsc, nxjc* and *nxjlink*, which are located in the server. NXJ Script (NXJS) is a simple language proposed in this project. It is designed as a script language. Each line contains one instruction to control the robot. NXJS currently has seven instructions listed in Table 1. Figure 6 shows the grammar of NXJS. The program *nxjsc* is an interpreter we created in C with Flex and Bison (Vermeir, D. 2009). It compiles the source code written in NXJS into Java source code as .java file. Figure 7 shows an example of source code in NXJS and the expected movements of robot. The external program *nxjc* and *nxjlink* are provided by leJOS NXJ command line tools. Command *nxjc* compiles the Java source code into Java byte code. A successful compiling of *nxjc* can generate a .class file. Command *nxjlink* can link the .class file with other necessary libraries to generate an *.nxj* file.

Table 1. *Instruction set*

Instruction	Action of Robot
forward (speed_level)	keep forwarding in specified speed_level
backward (speed_level)	keep backwarding in specified speed_level
turnleft (speed_level)	keep turnlefting in the specified speed_level
turnright (speed_level)	keep turnrighting in specified speed_level
rotate (speed_level)	keep rotating clockwise in specified speed_level
wait (seconds)	keep the current action for specified seconds (float number)
stop (0)	Stop the robot

```
line:= instruction
instruction:= keyword (number)
keyword:= forward | backward | turnleft | turnright | rotate | wait
```

Figure 6. *NXJS grammar based on regular expression*



```
forward(2)
wait(5.0)
stop(0)

turnleft(1)
wait(1.0)
forward(2)
wait(2.0)
stop(0)

backward(2)
wait(5.0)
stop(0)
```

Figure 7. *An example of NXJS programming*

Loader: Similar to the compiler, when users click the run button in the toolbar, the php program will invoke another external program to download the .nxj file to robot and start running. leJOS NXJ command line tools provide a command tool *nxj* to download and run the .nxj file. It is illustrated by Figure 8.

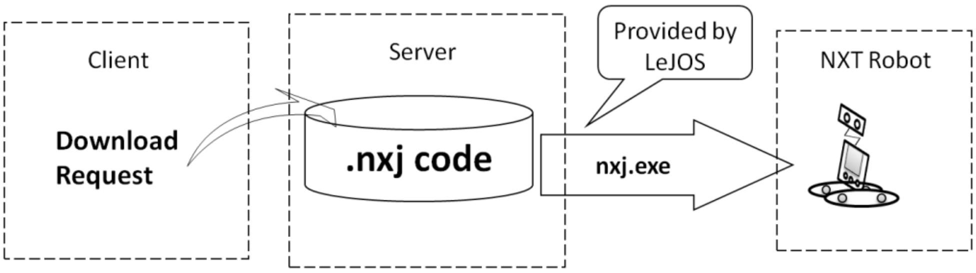


Figure 8. *Schema of* loader *module*

Robot Access Controller: The auto mode is based on a FIFO mechanism. The FIFO queue in the server registers all the requests from clients. Any time, only the first user in that queue can use the robot. After finishing an experiment, the user who owns the robot should release the robot. The scheduler server is implemented in Java. The applet embedded in the web communicates with the server by socket. Supervising teacher can set the access control mode.

Monitor: Java Media Framework (JMF) is a library which provides real time multi-media service on the web (Sun Microsystems, 1999). In VRES a Java program uses JMF to capture image and sound. The collected information is sent to Internet. Two computers are used to provide video and audio services individually. One computer only captures video, and the second computer captures video and audio simultaneously. Both of these two media servers send the media information through Real Time Transport Protocol (RTP) (Schulzrinne et al. 2003). Two videos can give users a two-aspect view of the robot. Audio can strengthen the *feeling of reality* and interactivity. Each user will access the video and audio service through the Java applet with JMF embedded in the webpage. The service model is illustrated by Figure 9.

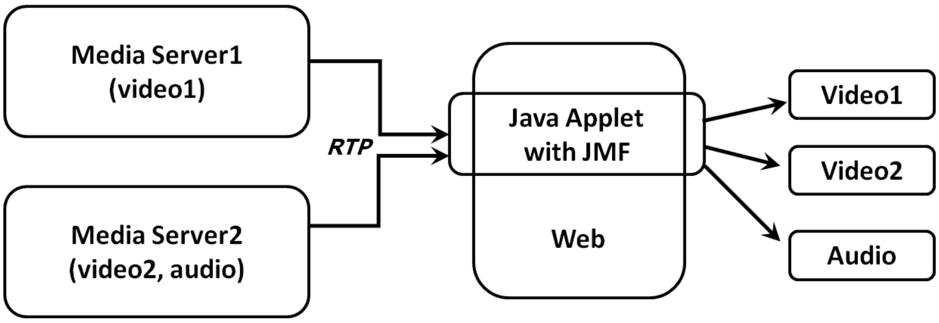


Figure 9. *The service model of monitor module*

Instant Commander: There is an open source package called *icommand* which can control NXT robot instantly. In VRES, a command processor is running at the system server. NXT robot connects with that command processor through Bluetooth. A Java applet embedded in the webpage can communicate with the command processor. When an arrow button in the applet is clicked, the corresponding action command will be sent to the command processor. Finally, the action of NXT robot will follow that command. Figure 10 shows us the model of this approach.

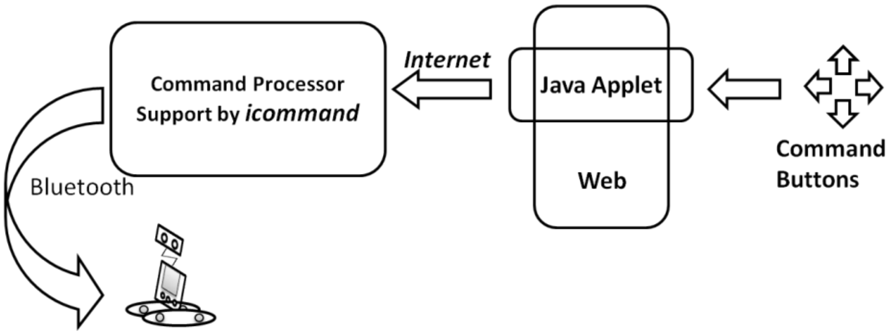


Figure 10. *The model of* instant commander

Evaluation

In order to evaluate this education system, we carried out a questionnaire with 20 students in computer education department of Sunchon National University. The questions were designed to measure whether the system can provide certain benefits. Questions of that investigation are listed in Table 2.

We get 14 responses from students, but only 13 of them are available. The average score of estimation in terms of each question are showed in Table 3. The scores are separated into three grades. The scores less than 3.00 are in grade one, and that means users don’t agree the benefit described in the corresponding questions; The scores bigger than 3.0 and less than 3.50 are in grade two, and those assessments indicate the facilitations described in corresponding questions are not significant; The scores bigger than 3.50 are in grade three, and that can be considered as the advantages are acceptable.

According the result, we can say that users agree VRES facilitates in programming with NXT, but the interactivity component needs to be improved. A significant problem is VRES does not have enough help information for students.

Table 2. *Questions to evaluate the VRES*

Question	very negative	negative	some what	positive	very positive
Q01. This system is helpful to learn robot programming	1	2	3	4	5
Q02. It's easy to use this system without instructions from the teacher	1	2	3	4	5
Q03. This system clearly showed the user's personal information	1	2	3	4	5
Q04. It's easy to manage the files through this system	1	2	3	4	5
Q05. It's convenient to access robot through this system	1	2	3	4	5
Q06. It's convenient to monitor robot through web	1	2	3	4	5
Q07. It's easy to edit the source code online	1	2	3	4	5
Q08. It's easy to compile the source code online	1	2	3	4	5
Q09. It's easy to download program to the robot	1	2	3	4	5
Q10. This system has given enough responses	1	2	3	4	5
Q11. The responses of this system are instant	1	2	3	4	5
Q12. It's funny to use this system to learn robot programming	1	2	3	4	5

Table 3. *Average score of each question*

	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Q11	Q12
AVG	3.62	2.69	2.85	3.23	3.54	3.77	3.54	3.69	3.69	2.92	3.00	3.46
STDEV	0.83	1.46	1.04	0.93	1.01	1.23	1.39	1.31	1.05	1.14	1.35	1.40

Conclusions and Future Work

In this paper, a web-based education system VRES, which is used to assist students in robot programming is introduced. Students can access this system without the restriction of space and time. With VRES, users can create, compile, and download programs to control Lego NXT robot. And users can monitor the state of robot remotely through two videos in different aspects and audio on the web. VRES is an economic solution for schools to conduct robot education with Lego NXT robots. And this system provides a script-like programming language NXJS for the students who are not familiar with normal programming language like Java.

Currently, the interpreter NXJSC can only support motor controlling of the NXT robot. Students cannot make the robot do complex behaviors by this script language. Thus, in the future, the most urgent work is to create enough instructions for this simple language NXJS. We also want to make VRES can arrange two or more robots to students, since students would wait for a long time to get that unique robot. Our final goal is to make VRES to be an open platform as Ciclope. VRES is planned to provide robot programming tools and monitor service for every student who can access internet, and integrate the resources from different universities for robotics education.

Acknowledgement

This work was supported by Mid-career Researcher Program through NRF grant funded by the MEST (No. 2009-0086676)

References

Kim, S., & Jeon, J. (2007). Programming LEGO Mindstorms NXT with visual programming. *A paper presented at International Conference on Control, Automation and Systems*.

Hatzilygeroudis, I., Giannoulis, C., & Koutsojannis, C. (2004). A Web-Based Education System for Predicate Logic. *A paper presented at the Fourth IEEE International Conference on Advanced Learning Technologies*.

Gonzalez, D. (2002). Distributed Platform for Control of Robots at Distance. *A paper presented at the 6th World Multiconference on Systemics, Cybernetics and informatics*.

Sagiroglu, S., Yilmaz, N., & Wani, M. A. (2006). Web Robot Learning Powered by Bluetooth Communication System. *A paper presented at the 5th International Conference on Machine Learning and Applications*.

Cedazo, R., & Lopez, D. (2007). Ciclope: FOSS for Developing and Managing Educational Web Laboratories. *IEEE Transactions on Education, 50*(4), 352 – 359.

Hazel, S. (2009) Codepad. Retrieved from http://codepad.org

Khen, G., Khen,D., & Koubi, G. (2009) Coderun. Retrieved from http://coderun.com

Foord, M. (2009). Trypython. Retrieved from http://www.trypython.org

Solorzano, J., & Andrews, Paul. (2009). LeJOS Java for LEGO Mindstorms. Retrieved from http://lejos.sourceforge.net

Vermeir, D. (2009). An Introduction to Compilers. Retrieved from http://star2.vub.ac.be/~dvermeir/courses/compilers/compilers.pdf

Sun Microsystems, Inc. (1999). Java Media Framework API Guide. Retrieved from http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/guide/JMFPreface.html

Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (2003). RTP: A Transport Protocol for Real-Time Applications. Retrieved from http://tools.ietf.org/html/rfc3550