Visual Learning Tool for Lego Robots Programming

Georges Meguro
University of Aizu
Tsuruga, Ikki-machi
Aizu-Wakamatsu, Japan
+81-242-372717
m5201110@u-aizu.ac.jp

Kenji Abe
University of Aizu
Tsuruga, Ikki-machi
Aizu-Wakamatsu, Japan
+81-242-372717
s1200222@u-aizu.ac.jp

Alexander Vazhenin
University of Aizu
Tsuruga, Ikki-machi
Aizu-Wakamatsu, Japan
+81-242-372717
vazhenin@u-aizu.ac.jp

ABSTRACT
This article proposes visual learning tool for Lego MINDSTORMS EV3 using LEJOS firmware. It has an icon-based style of introducing robot activities with on-line reflection of corresponding code. This allows beginners to enjoy and make EV3 programs. The designed tool can launch and control the robot actions and check/debug them even if a program isn't complete. It also enabled that a client operates on the web page and moves a robot through a server.

Categories and Subject Descriptors
D.1.7 [**Visual Programming**]

General Terms
Design, Human Factors, Languages.

Keywords
Visual Programming, Robotics, Icon-based Language.

1. INTRODUCTION
The current state of research in robotics has moved from a limited capability single robot to multi-robot systems equipped with a plethora of sensors, leading to de-facto multi-processor and distributed applications. Developing ad-hoc solutions based on low level communication libraries is not an efficient approach and makes reuse and sharing of software difficult. Software frameworks can be considered as an important paradigm for next generation of robotics applications [1]. Actually, robots need to use information from many sources. For example, the scan system recognizes objects, a mapping system builds an environment map, a human may give commands, and the robot loads data and knowledge from different sources. For making use of this knowledge, a robot needs to integrate these different pieces of information, represent them in a common format and language including its semantics. Recent robotics studies are investigating how robots can exploit the World Wide Web in order to offer their functionality and retrieve information that is useful for completing their tasks [2, 3]. The effective way to realize this is represented by the Service Oriented Architecture (SOA), where the Service Component Architecture defines a component-based implementation [4]. Robotic development environments leverage these emerging Web-programming paradigms [5].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*ICAIT'*16, Oct. 6–8, 2016, Aizu-Wakamatsu, Japan.
Copyright 2016 University of Aizu Press.

Our investigations are based on the original Virtual-Model-View-Controller (V-MVC) design pattern that is a combination of two well-known approaches: SOA and the Model-View-Controller (MVC). Currently, we are designing the VMVC-based technology for developing SOA-applications that reduces the developer's efforts concentrating mostly on creating the processing logic and the UI design, facilitating the debugging process of the view and model separately [6]. The goal of this research is in developing WEB-based applications for Robot Control System, visualization of robot activities and data in the framework of Service-Oriented Architecture and the VMVC-based technology.

LEGO Education EV3 robotics is the third generation of LEGO Educational robotics and comes with an improved intelligent brick, new motors, new sensors and improved software. It can be considered as a perfect solution allowing to teach students how to program, build and test their robot models. The first visual programming environment was called LEGO sheets [1]. It was created by the University of Colorado in 1994.

The paper presented is devoted to design a visual tool allowing users study how to program robot activities. It has an icon-based style of introducing robot activities with on-line reflection of corresponding code. This allows beginners to enjoy and make the EV3 program. The designed tool can launch and control the robot actions and check/debug them even a program isn't complete. It also enabled that a client operates on the web page and moves a robot through a server.

The organization of this paper is as follows. Section 2 describes the Lego MINDSTORMS environment and proposes the way of improving it. Section 3 describes the proposed application. Section 4 describes the visual robot program made with the application and the experimental results of its testing. Section 5 contains conclusion and future directions of work.

2. LEGO ROBOT ENVIRONMENT

2.1 Lego MINDSTORMS

Lego MINDSTORMS is an educational toolkit for robotics that is assembled with Lego Blocks. It is oriented to study how to construct and program different kinds of LEGO robots with some enjoyment (Figure 1). Now, there is the third generation of this toolkit called RCX, NXT and EV3 correspondingly. EV3 is the latest model of Lego MINDSTORMS. It contains four sensor ports, four motor ports, an USB port and Micro SD slot. It may also be extended by a Wi-Fi function allowing to make the wide range of communications in comparison with the NXT version. Table 1 shows some characteristics of the Lego MINDSTORMS EV3.

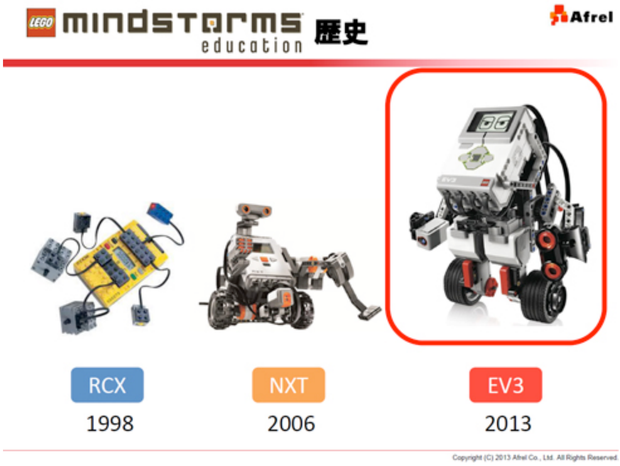

Figure 1. LEGO MINDSTORMS

Table 1. Spec of Lego MINDSTORMS EV3

OS	LINUX
Main Processor	300MHz
Main Memory	64MB
Storage	16MB

2.2 Firmware

The educational software called EV3 is based on LabVIEW of National Instruments Software [7, 8]. The LEGO programming is realized by connecting icons using drag and drop manipulations. Additionally, each icon can have numerical data defining physical parameters of robots like speed, angle, time, etc.

Besides this software, EV3 can also use several programming platforms based on C and/or JAVA languages. The LEJOS is the extension of JAVA language for the Lego MINDSTORMS programmable bricks [9]. To run the LEJOS firmware, it is necessary to prepare the micro SDHC memory card with at least 2GB, and no more than 32GB. The card needs to be formatted with a FAT32 partition.

2.3 Problem

EV3 Software allows making a LEGO program visually and easily without typing the text. So, it isn't necessary to type, and even beginners can be interested. But it doesn't have versatility especially in case when it is necessary to study API and program constructions at the same time.

Conversely, LEJOS uses JAVA language showing a program code directly and allowing to learn basic language constructions in textual mode. Therefore, it makes some difficulties for beginners who is going to study the low level of the LEGO programming. That is why it would be good to have tools combining advantages of visual programming with possibility to study how to specify LEGO robot operations by means of Java API interface.

3. ICON-BASED VISUAL PROGRAMMING TOOL

The presented tool was designed to support the LEGO programming and study combining the icon-based specification of LEGO robot operations with simultaneous reflection of the Java executable code.

The programming panel is shown in Figure 2. Actually, the programming process is reduced to choosing a corresponding icon and specifying its parameters. It can also display a code of EV3 program in real time. This allows learning how to make a sequence of Java instructions with a corresponding reflection of the operation semantics. It is also possible to run the robot program and watch immediately how it works.

After finishing, editing and debugging, the user can save a robot program as a Java code. In future, this code can be edited as an ordinary Java program.

System is saving both the Java code and a list of visual blocks in order to have a correspondence between visual and textual part of a robot algorithm. There are the following panels in the system interface:

1 ***Message Panel*** displays errors and message of preservation completion.

2 ***Port Panel*** allows choosing motor ports and sensor ports.

3 ***Program Table Panel*** displays a code of visual program. After pressing the "RUN" button, program sequence will be sent to the EV3 robot.

4 ***Program Sentence Panel*** displays a Java program sentence. The user can save a program as a JAVA file.

5 ***Program Button Panel (If and Loop)*** allows selecting and adding/removing program blocks. It supports has if- and loop- statement blocks.

6 ***Program Button Panel*** allows choosing/adding program blocks including the blocks' movement.

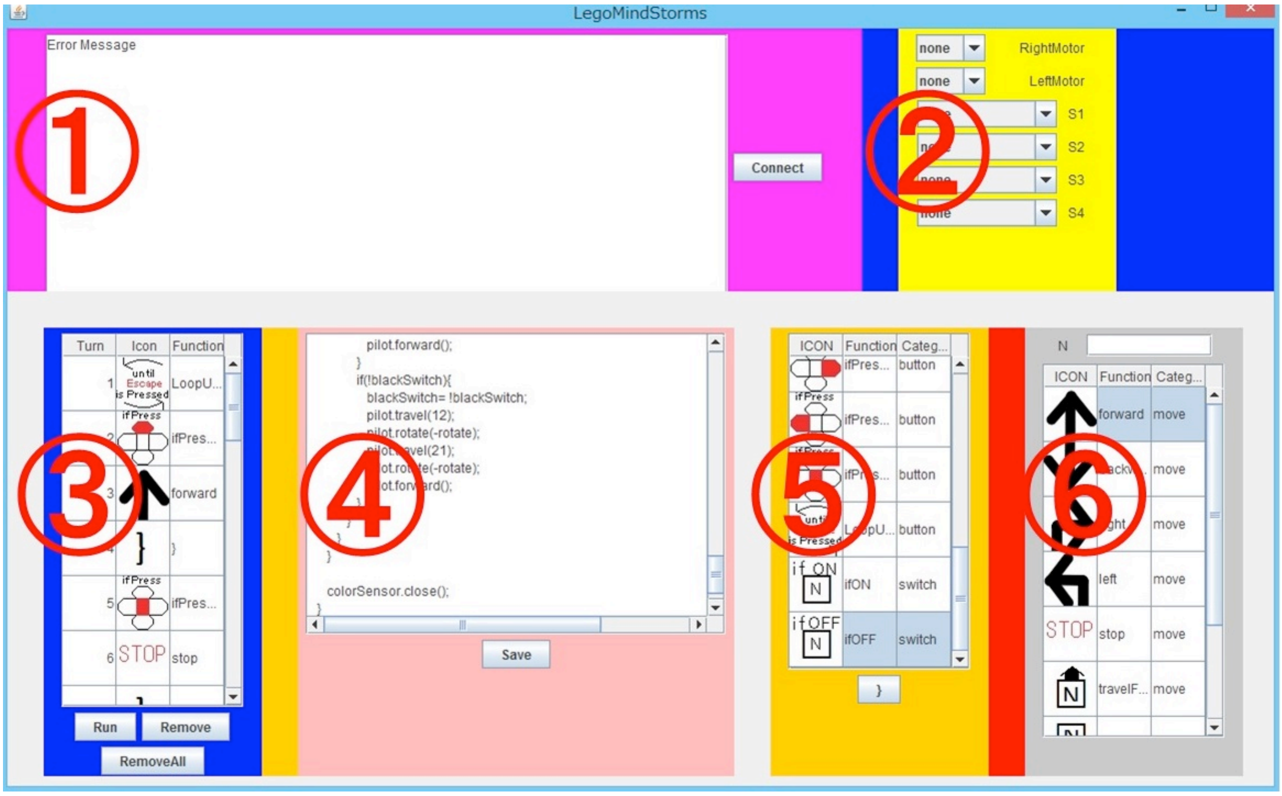

Figure 2. Programming panels of the EV3 visual editor

Figure 3 shows the Data Flow Process that demonstrates how data are transformed in the environment presented.

This application saves data in an array list including the visual blocks list named a program table and the Java code named a program sentence. The user can change the program table by inserting/removing a visual block and specifying its parameters. Synchronously, the corresponding changes will be made in the program sentence. At any time, the user can launch execution of the robot program. In this case, the program sentence is loading into EV3, and the editor will launch its execution.

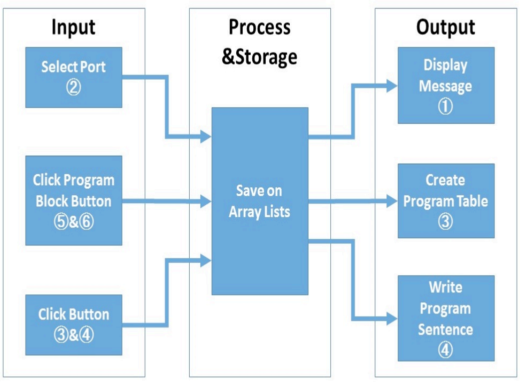

Figure 3. Input-output data flow

The robot program can be executed via a stand-alone and WEB-based application correspondingly. In both cases the connection between EV3 and a host is realized using WI-FI connections.

4. EXAMPLE OF APPLICATION

The features of the presented system were tested using the algorithm that is oriented to find a part with the given color on the rectangular area as shown in Figure 4. The LEGO robot is moving across this area by a zigzag trajectory. The area is bounded by a black color, and a robot is changing its direction as shown in Figure 4. During this movement, the robot saves and displays colors of all areas visited. The movement will be stopped when the area with given color is reached.

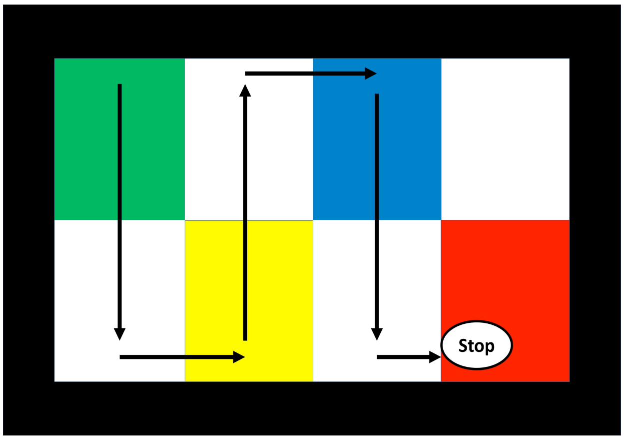

Figure 4. The route where EV3 runs

The block diagram of this algorithm is presented in Figure 5. As shown in Figure 6, the visual program realizing this robot's behavior consists of twenty-four visual blocks. It shows practical examples of usage the icon language especially IF-statement allowing programming according to states of the robot sensors. Figure 7 illustrates the program logic and shows how visual constructions are nested. Obviously, in our experiments we used the color LEGO sensor allowing to distinguish sixteen different colors.

Figures demonstrate examples with the red given color and black colored boundaries. At first, EV3 moves forward. Robot stops its movement when the area with given color is reached. It changes direction when a black boundary will be found and continues movement. It implements counterclockwise or clockwise turns realizing the zigzag movement.

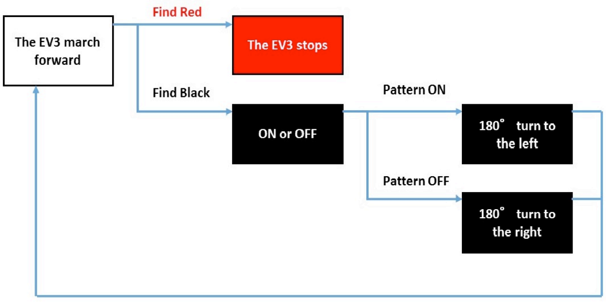

Figure 5. Find color program algorithm

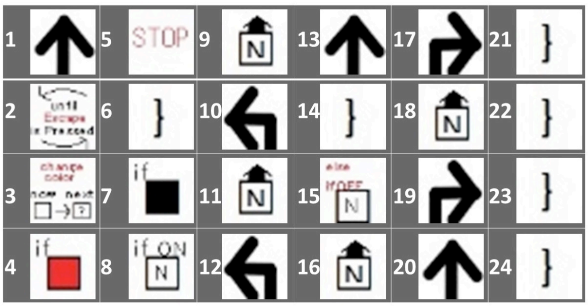

Figure 6. Icon of Find color program algorithm

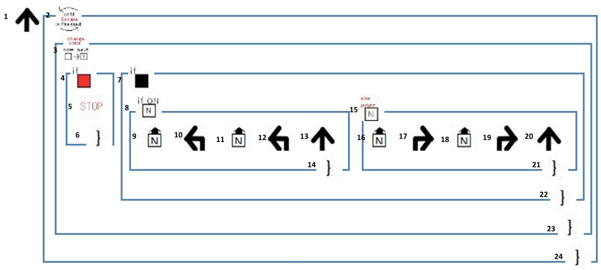

Figure 7. Visual blocks of Find Color Program

The size of the area is an important parameter influencing on the search results. This requires changing the width of zigzag. Figure 6 demonstrates also possibility to specify time of forward movement. Other experiments were provided to evaluate possibility to find area with sizes of 18, 6 and 3 centimeters. On-line changing of this parameter allowed us to optimize this search procedure.

5. CONCLUSION

In this research, we propose a new application based on a Java language which enables to program visually. It allows to make a program by selecting icons and specifying physical parameters of each operation. The system generates a Java-program that can be loaded in the LEGO robot. The main feature of the system is that any visual object is immediately displayed by representing a corresponding Java code. It is also possible to confirm so how the sentence is written. The presented work is made on cooperation with another research oriented to make robot control program via WEB browser.

To test the system, we asked six participants. It was confirmed the simplicity of programming and on-line study process. This opinion confirms workability the program presented and defines future directions in its evolution based on some improvement of the user interface.

6. REFERENCES

[1] D. Chugo and S. Yokota (Eds). 2012. *Introduction to Modern Robotics II. Syst.* iConcept Press Ltd. 294.

[2] M. B. Blake, S. L. Remy, Yi Wei, and A. M. 2011. Howard, Robots on the Web. *IEEE Robotics & Automation Magazine* (June 2011). 58-68.

[3] M. Tenorth and Ul. Klank, D. 2011. Pangercic, and M. Beetz, Web-Enabled Robots. *IEEE Robotics & Automation Magazine* (June 2011). 33-43.

[4] Th. Erl. 2009. *SOA Design Patterns*. PrenticeHall.

[5] S. L. Remy and M. B. Blake. 2011. Distributed service-oriented robotics. *IEEE Internet Comput* 15/2, (Mar./Apr. 2011). 70–74.

[6] R. Cortez and A. Vazhenin. 2015. Virtual model-view-controller design pattern: Extended MVC for service-oriented architecture. *IEEJ Trans. On Electrical and Electronic Engineering* 10 (July 2015), 411-422.

[7] *Ltd Afrel, Educational Lego Mindstorms regular agency*. Ltd. Afrel. http://www.afrel.co.jp/lineup/mindstorm-ev3/.

[8] 2014. *Education version EV3 Java programming guide.* Ltd. Afrel.

[9] *LEJOS, Java for Lego Mindstorms.* Lejos. http://www.lejos.org/.

[10] 2013. *LEGO MINDSTORMS EV3*. LEGO group. 4-31.

[11] R. Cortez, and A. Vazhenin. 2013. Developing Re-usable Components Based on the Virtual-MVC Design Pattern. *Databases in Networked Information Systems*. 132-149.

[12] N. Teslya, and S. Sergey. Smart-M3-based robot interaction in cyber-physical systems. *Open Innovation* (FRUCT16, 2014 16th Conference of. IEEE). 108-114.