WWW-based System for LEGO Robots Control

Georges Meguro
University of Aizu
Tsuruga, Ikki-machi
Aizu-Wakamatsu, Japan
+81-242-372717
m5201110@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 a WEB-based LEGO Robot control system with JAVA technologies. This work is based on the original Virtual-Model-View-Controller design pattern, and focused to exchange data between server and robot. The designed system can manage the robot and reflects the collected data by robot to the client on web page.

Categories and Subject Descriptors
I.2.9 [**Artificial Intelligence**]: Robotics – *operator interfaces, sensors, workcell organization and planning*.

General Terms
Algorithms, Design, Experimentation

Keywords
LEGO Robots, WWW interface, Data Exchange Protocol.

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 the 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].

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 by 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 to develop WEB-based applications for a 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. The best solution is to teach students how to program, and then to build and test their models. The first visual programming environment was called LEGOsheets, and was created by the University of Colorado in 1994.

The work presented is devoted to designing the WEB-based Robot control system in the framework of the Apache Tomcat environment. This system is an open source software implementation of the Java Servlet, Java Server Pages, Java Expression Language and Java WebSocket technologies. During this work a Robot control server was designed allowing communication with an internal robot program, accumulating data and visualizing them. More specifically, the paper is devoted to designing the mechanism of data exchange between a WEB server and LEGO robot controlled by the client side.

This paper has the following structure. Section 2 contains the description of the environment, which is used for creating the system. Explanation about the data exchange mechanism design is presented in Section 3. Results of testing are presented in Section 4. The conclusion and future work are discussed in Section 5.

ICAIT'16, Oct. 6–8, 2016, Aizu-Wakamatsu, Japan.

2. LEGO ENVIRONMENT

2.1 Hardware

LEGO MINDSTORMS is a robot construction kit which is designed by Lego. It's constructed mainly from a computer brick, which can run programs, various sensors for gathering data, and a motor for dynamic force (Fig. 1).

EV3 is the third generation product of LEGO MINDSTORMS. Its computer brick has an ARM9-type CPU, 64 MB of RAM memory and 16 MB of flash memory, MicroSDHC card slot; Linux is installed. In addition, EV3 also supports Bluetooth and Wi-Fi for wireless communication [7].

Figure 1. Constructed LEGO MINDSTORMS EV3

2.2 LeJOS Firmware

LeJOS is a firmware replacement for LEGO MINDSTORMS. It includes a Java virtual machine (Java Runtime Environment 7) and allows Java program execution on LEGO MINDSTORMS. This means it can work with other Java-based technology. Typically it includes:

- A library of Java classes (classes.jar) that implement the LeJOS EV3 Application Programming Interface (API) and provides an alternative Java Runtime (packages java.*) that is optimized for the EV3.
- A linker for linking user Java classes with classes.jar to form a binary file that can be uploaded and run on the EV3.
- PC tools for flashing the firmware, uploading programs, debugging, and many other functions.
- A PC API for writing PC programs that communicates with LeJOS EV3 programs using Java streams over Bluetooth or USB, or using the LEGO Communications Protocol (LCP).

In this research we are using the LeJOS EV3 0.9.0-beta version [8].

3. APPLICATION SCENARION

3.1 Architecture

Usually, a WEB-application is a client-server software application in which the client (or user interface) runs in a web browser. There are several ways to include a robot into a client-server environment. In our investigations we are using the thin client approach in which a robot has direct connection with the server. This architecture is shown in Figure 2.

In order to establish this web-architecture, we are using the Apache Tomcat system that is an open source software implementation of the Java Servlet, Java Server Pages, Java Expression Language and Java WebSocket technologies. We are using these technologies to create a web application in conjunction with the Lejos Java program. In the presented work the Apache-Tomcat-8.0.28 is used [9].

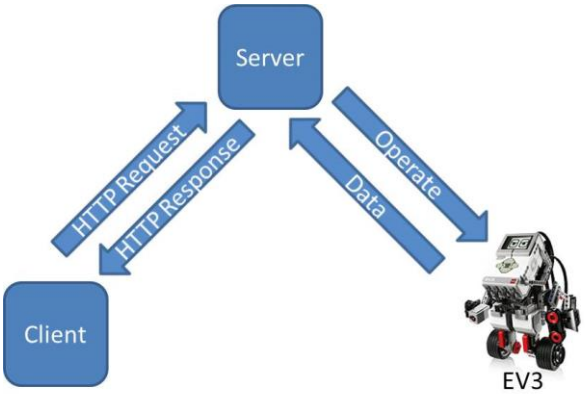

Figure 2. Basic WEB-architecture

3.2 Application Scenario

Consider the following application of the LEGO robot. It is oriented to scan designated areas, accumulate some data from the LEGO sensors, collect these data at the server, and show results on the client site as well as control robot movement and actions. An example of area is shown in Figure 3.

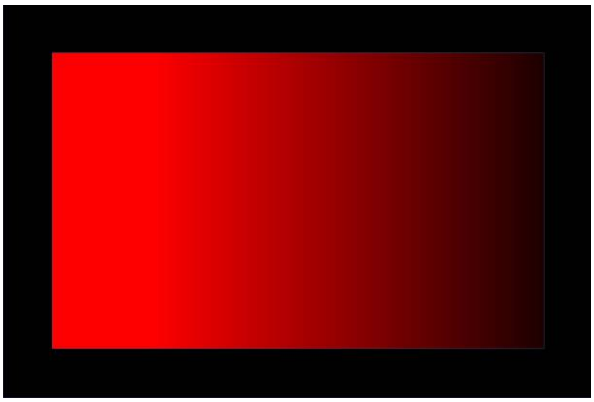

Figure 3. Scanning Area

Importantly, a robot can have different algorithms for its movement defined by the EV3 program. For example, the LEGO robot can move across this area with a zigzag trajectory by changing its direction after reaching area boundaries.

Figure 4 shows a sequence diagram realizing the implementation scenario as follows. First, the application is launched by client request to an application server. Next, the server connects to EV3 and orders it to start exploration. Then, EV3 sends the explored data to the server. Finally, the server generates the web page to show the result and sends it to the client.

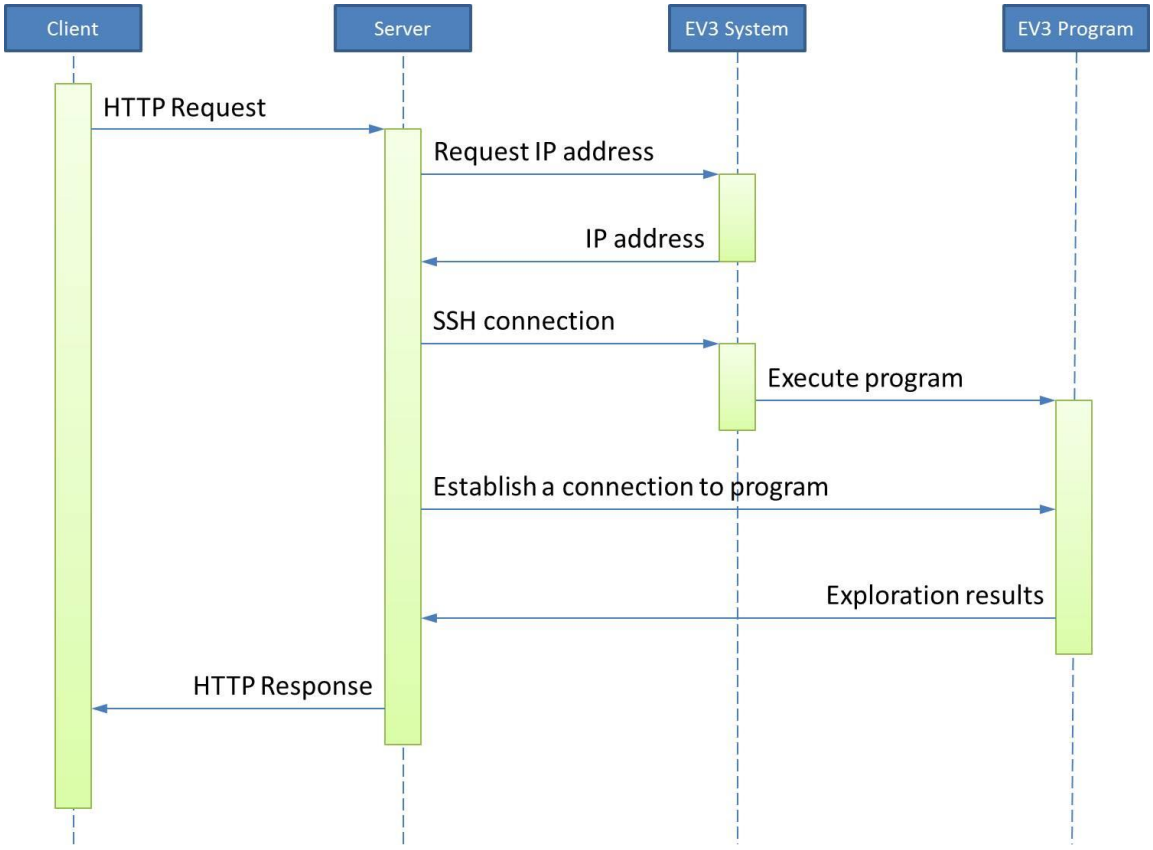

Figure 4. Implementation Diagram

3.3 Communication between Server and LEGO Robot

The Wi-Fi network allows high-speed connections between devices supporting on-line control for the robot behavior and data exchange. The efficient way to support security of Wi-Fi connections is in using encryption of exchanged data. SSH-protocol provides a secure channel over an unsecured network in client-server architecture. Accordingly, the communication protocol between server and robot is realized as follows. First, the program in the server searches connectable EV3 and obtains its IP address. Next, the server connects to EV3 by SSH using obtained an IP address to launch program in EV3. Finally, each program makes a connection through Java socket and starts exchanging data to solve task.

To realize this protocol, it is necessary to grant remote access from one device to operate on other device from the server, run programs or manipulate with files. We realized it using the Ganymed SSH-2 for Java that is a library which implements the SSH-2 protocol in pure Java and realizes connecting to other devices by SSH from Java programs. It supports SSH sessions (remote command execution and shell access), local and remote port forwarding, local stream forwarding, X11 forwarding, SCP and SFTP. There are no dependencies on any JCE provider, as all crypto functionality is included. Ganymed-ssh2-build210 is used in this paper [10]. Figure 5 describes the basic Java code implementing a connection between a server and an external device.

```
//Connect to device and log in.
Connection conn = new Connection(IPADDRESS);
boolean connected
      = conn.authenticateWithPassword(USERID, PASSWORD);

//Execute the command line at logged-in device
if (connected) {
   Session session = conn.openSession();
   session.execCommand(COMMANDLINE);
   session.close();
}
conn.close();   //Disconnect
```

Figure 5. Connecting to Device via the Ganymed SSH-2 Library

4. IMPLEMENTATION AND TESTING

To test the communication protocol designed, a WWW-based prototype was developed. It has a rather simple end-user interface allowing operation to be launched inside the robot from a browser. The current version of the program allows having access to the color robot sensors.

Figure 6 shows a flowchart of a prototype EV3 program which operates the robot to explore the field shown in Figure 3, and

send the red density of the current position to the server. As shown in Figure 3, the area has a red colored area with different color density. It also bordered by black colored lines.

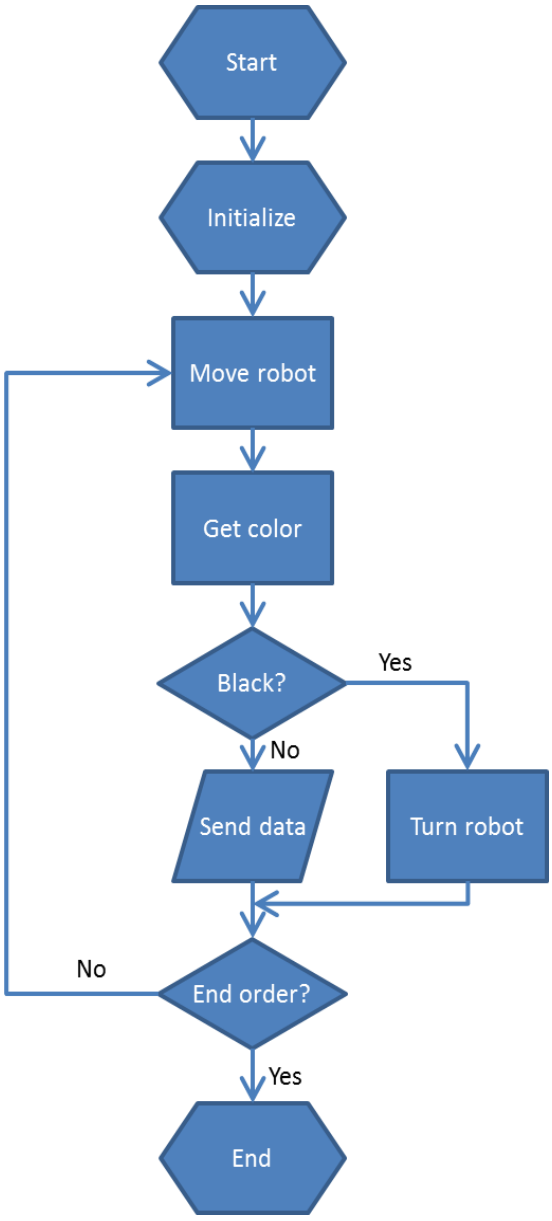

Figure 6. Flowchart of prototype program

The program execution is started from the initialization step. During initialization, the program makes a connection to the server using a Java socket. It also collects information about the robot motor and sensor, which is connected to the computer brick and will participate in operation. After the initialization, the robot starts movement in a straight direction. At the same time, it starts the acquisition of data from the color sensor. Acquired color information is transmitted to the server.

If the sensor detects the black line of the outer edge, the robot starts rotation to avoid jumping out from the field. The program keeps exploring until the server sends the stop instruction.

By moving the LEGO robot over the designated area, it is possible to get codes of corresponding colors and show them in the browser's window. The experiments provided confirm the correctness and accuracy of the designed protocol.

5. CONCLUSION

The work presented is devoted to designing a WEB-based Robot control system in the framework of the Apache Tomcat environment. More specifically, the mechanism of data exchange between a WEB server and LEGO robot was designed and tested. The experiments provided confirm the correctness and accuracy of the designed protocol. The communications between server and the EV3 program were realized via SSH-protocol that provides a secure channel over an unsecured network in the client-server architecture. The experiments provided confirm the correctness and accuracy of the designed protocol.

This work can be considered as a first step in designing the SOA-based Robot control system based on a V-MVC paradigm. In future work, we are planning to improve the test program and create a V-MVC model of this system.

6. REFERENCES

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

[2] Blake M.B., Remy S.L., Wei Y. and Howard A. M. 2011. 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] Erl Th. 2009. *SOA Design Patterns*. PrenticeHall.

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

[6] Cortez R. and Vazhenin A. 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] Lego Mindstorms - LEGO.com. http://mindstorms.lego.com/.

[8] Lejos, Java for Lego Mindstorms. http://www.lejos.org/.

[9] Apache Tomcat. http://tomcat.apache.org.

[10] The Ganymed Project. http://www.ganymed.ethz.ch/ssh2.