Automatic Driving System Using LEGO

Chuanxi Zhou[1], Yujian Fu[2], and Mezemir Wagaw[2]
[1]Department of Computer Science, Alabama A&M University, Normal, AL, USA
[2]Department of Food & Science, Alabama A&M University, Normal, AL, USA

Abstract **-** *LEGO is not only a toy, but also a tool for education. By analyzing and solving the all sorts of problems with LEGO, students have better preparation for a career in science and engineer. Our LEGO-robot is built to pick up a ball, find its way to follow a color curved line, put down the ball at the destination, and show the sound values on the NXT screen. A touch sensor, light sensor, sound sensor, Ultrasonic sensor and three motors are used for the robot to complete this challenge. Our LEGO-robot automatic driving system is designed with UML. UML as a graphical modeling language is a standard way for visualizing, specifying, constructing, and documenting an Object-oriented software system. A class diagram in the UML is used to describe the information and relations about seven classes, including the CorrectingDirection class, the LegoMotor class, the SoundSensor class, the UltrasonicSensor class, the LightSeonsor class, and the TouchSensor class. A use case diagram is also used to describe our robot system's behavior as it responds to a request from the outside of the system. A state diagram is used to describe the internal behavior in a class with natural states. This project is implemented using Java with leJOS. The system works steady with expected design functionalities, which indicates that we have successfully designed the LEGO-robot automatic driving system using UML.*

Keywords: UML, class diagram, use case diagram, state machine diagram, LEGO robot, software design

1 Introduction

LEGO is not only a toy, but also a tool for education. With LEGO, students can build models, utilize motors and sensors, learn how to design, program and control models. By analyzing and solving the all sorts of problems with LEGO, students move forward for a better preparation for science and engineer career.

Every year, the First LEGO League (FLL) sponsors a challenge tournament around the world. It is an international robotics team competition for children aged from 9 to 16. Today, FLL tournaments take place in more than 40 countries worldwide with participation of over 10,000 teams [4]. The tournament challenges vary in each year with the basic skills and concepts hold the same. For example, the challenge of following color line appeared in both "The Smart Move Robot Game" (2009) and "The Body Forward Challenge" (2010). Therefore, in this project, we will focus on this challenge. Our LEGO-Robot moves forward following a color line, picks up a ball when the touch sensor is pressed, adjusts the direction if it needed, determines the terminus by pre-programmed distance, puts the ball down when it arrives the terminus, and detects the sound value and shows the value on the NXT screen. Our robot and the mat are shown in Fig 1.

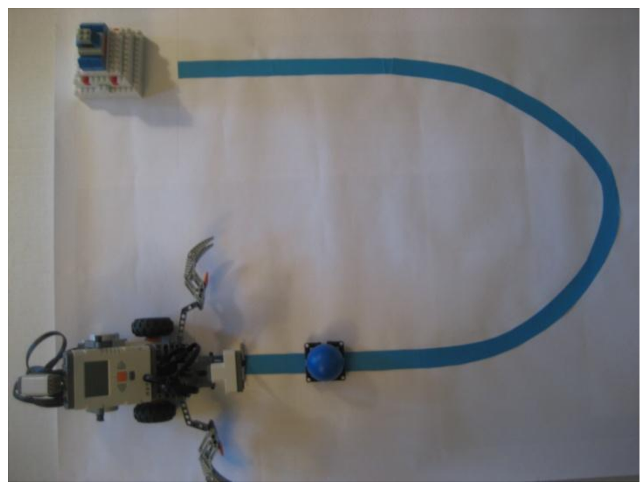

Figure 1, LEGO-Robot and LEGO-pad

This LEGO-Robot Automatic Driving System is designed with the Unified Modeling Language (UML). UML is a graphical modeling language which has been widely recognized as a standard way for visualizing, specifying, constructing, and documenting the artifacts of an Object-oriented software system [6]. It can also be applied for defining Software Architecture, modeling of business processes and their design, managing complexity, etc.

The remainder of this paper is organized as follows. We discuss the related works in Section 2. After that, the system design of automatic driving system (ADS) is presented in Section 3. The implementation in Java using LEGO NXT toolkit is shown in Section 4. Conclusion and future investigations are discussed in the Section 5.

2 Related Works

LEGO NXT tool kit is widely used for the educational and research purposes in academic now [2]. The Mindstorms NXT brick uses a 32-bit ARM processor as its main processor, with 256 kilobytes of flash memory available for program storage and 64 kilobytes of RAM for data storage during program execution. To acquire data from the input sensors, another processor is included that has 4 kilobytes of flash memory and 512 bytes of RAM. Two motors can be synchronized as a drive unit. To give the robot the ability to "see, the ultrasonic sensor, which is accurate to 3 centimeters and can measure up to 255 centime ters, and the light sensor, which can distinguish between light and dark, can be attached to the brick. A sound sensor that can be adjusted to the sensitivity of the human ear can be used to give the robot the ability to hear and react, if programmed, to noises. Finally, the two touch sensors give the ability for a robot to determine if it has been pressed, released, or bumped, and react accordingly [2].

Although LEGO NXT is a highly integrated and low cost educational settings, currently, there is not much work has been done for the designing of a reliable LEGO robot to realize the expected functionalities in literature. There are several approaches available for the software intensive design, UML is one of the most popular methods that is currently widely used in both industry and academic. In this section, we discuss some existing works on design of embedded systems using UML.

Grady Booch et al. [6], [1] introduced the UML 2.0 concepts and notations. Its advantages of allowing users to model everything from enterprise information systems and distributed Web-based applications to real-time embedded systems.

Saxena et al. [7] presented a UML model of multithreaded programs on a Dual Core processor. Performances of the programs in JAVA and C# on the basis of UML design were compared and evaluated.

João M. Fernandes [5] demonstrated the utilizing of UML to

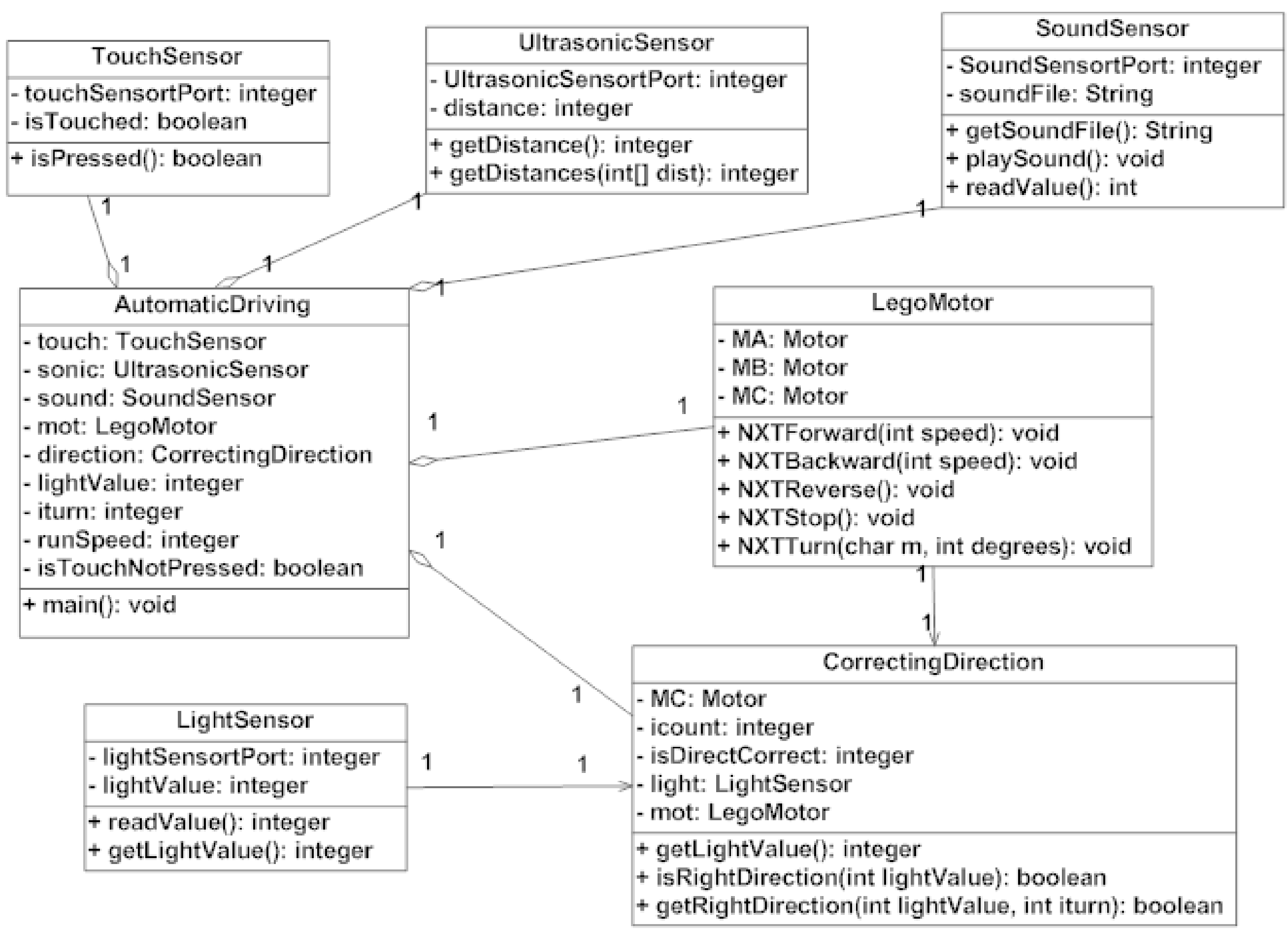

Figure 2 – Class Diagram for the LEGO-Robot Automatic Driving System

model industrial embedded systems. Using a car radios production line as an example, this study illustrated the modeling process following the analysis phase of complex control applications. The authors used some guidelines to transform the use case diagrams into a single object diagram to guarantee the continuity mapping of the models.

Our work is different from above works in following aspects: a) This work aims at developing a reliable robotics system using object oriented methodology – UML. Robotics systems are complex, concurrent embedded systems that include multiple objects and multiple interfaces. Our work has demonstrated that it is important to produce reliable robots for using UML design method. b) Our work applied UML design on the LEGO NXT settings, which provides a new successful application of UML on the embedded system design.

3 System Design – UML Model

In this section, we present our approach to design LEGO-Robot Automatic Driving System with UML.

3.1 Class Diagram

A class diagram in the UML is used to describe static information about classes, with operations and data (attributes), and to describe relations (including inheritance, aggregation, association, etc.) between different classes. As mentioned by Michael Blaha and James Rumbaugh [3], a “class diagram provides a graphic notation for modeling classes and their relationships, thereby describing possible objects. Class diagrams are useful both for abstract modeling and for designing actual programs.”Class diagrams are the mainstay of object-oriented analysis and design.

Fig. 2 shows the class diagram of our LEGO-robot automatic driving system. The NXT is the brain of a LEGO-robot. It’s an intelligent, computer-controlled LEGO brick that makes a LEGO-robot alive and perform the programmed activities. The NXT has three output ports to attach motors and four input ports to attach sensors. The three motors are grouped into the LegoMotor class, which including the LightSensor class, The TouchSensor class, the UltrasonicSensor class, and the SoundSensor class corresponding to the light sensor, the touch sensor, the Ultrasonic sensor, and the sound sensor, respectively. For detecting and correcting the moving direction of the robot, the light sensor and motors are grouped into the CorrectingDirection class. The AutomaticDriving class is consisted of the CorrectingDirection class, the LegoMotor class, the SoundSensor class, the UltrasonicSensor class, and the TouchSensor class.

3.2 Use Case Diagram

A use case is a description of a system’s behavior as it responds to a request that originates from outside of that system. In other words, a use case describes "who" can do "what" with the system in question [1]. The use case diagram of our LEGO-Robot Automatic Driving System is shown in Fig. 3. This figure clearly shows that a person can turn on the NXT, find certain program, run certain program, and turn off the NXT.

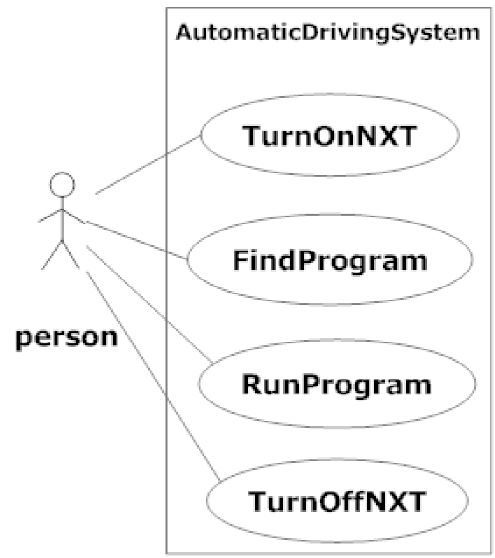

Figure 3 – Use Case Diagram for the LEGO-Robot Automatic Driving System.

3.3 State Machine Diagram

A state diagram has states, events and guards. They are used during the design phase to describe internal behavior in a class with natural states. State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented in series of events, that could occur in one or more possible states [1].

Fig. 4 shows the state diagram of our LEGO-robot Automatic Driving System. When the actor presses run on NXT, the LEGO-robot will firstly wait 5 seconds and let the actor to adjust the robot to the right location. After that, the robot is programmed to move forward. When the touch sensor is pressed along its way forward, the robot will firstly move backward to pick up and hold the ball. Then the robot will continue follow the color line which includes a half circle in the middle of the mat. The light sensor is used here for the robot to find the correct way to follow the curve. Here is how it does that, if the light sensor detects a different color value from the previous one, that means the robot is in the wrong direction, the robot will then turn around to look for the previous color value. If the light sensor cannot find the same value after turning a certain degree (here we set it as 3400), the robot will then simply stop. The Ultrasonic sensor is used to find the destination of the robot. When the Ultrasonic sensor detects the pre-programmed distance, the robot will just put the ball down and stop. During the whole moving activity, the sound sensor is detecting the sound values in the environment and shows them on NXT screen.

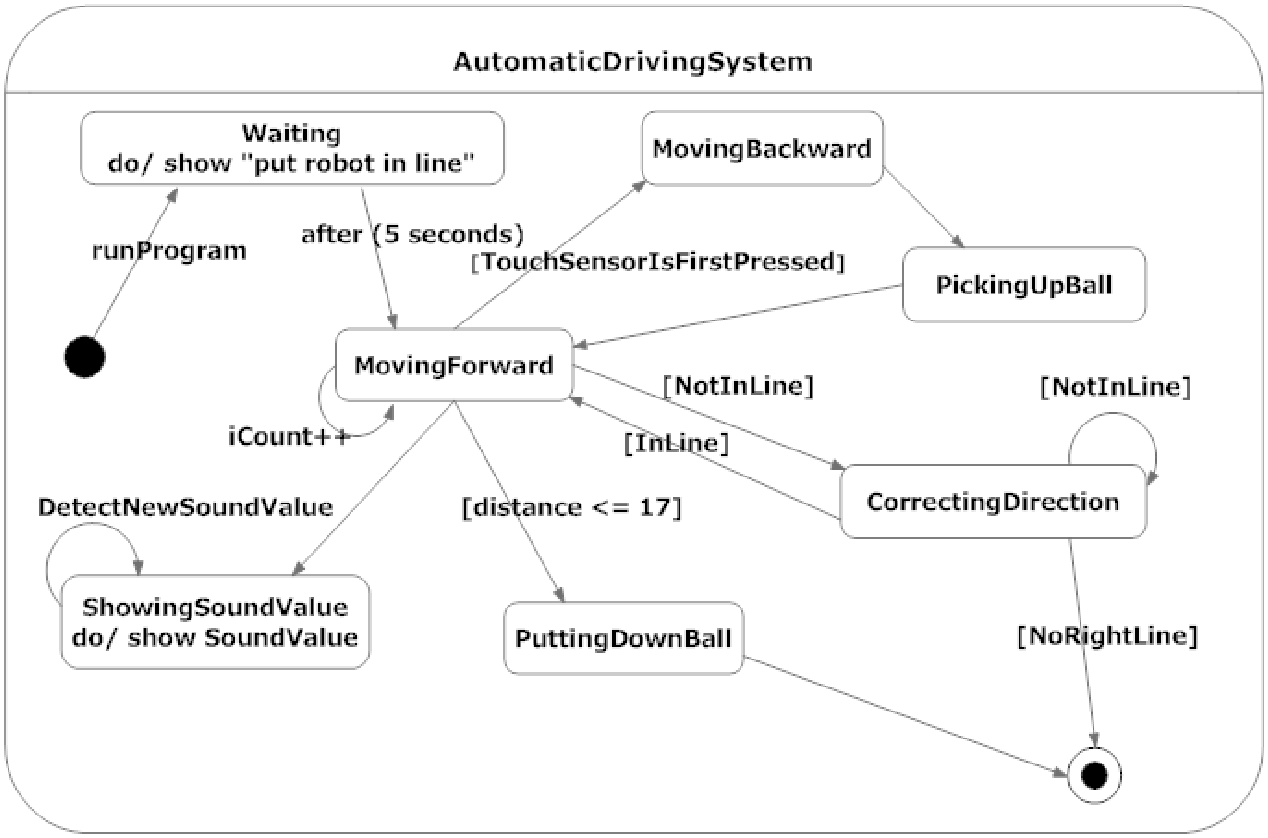

Figure 4 – State Diagram for the LEGO-Robot Automatic Driving System.

4 Implementation in Java

Java/leJOS is used in this project to implement our program. leJOS is a tiny Java Virtual Machine [8],[2]. LeJOS NXJ supports the NXT brick which allows us to code the LEGO-robot Automatic Driving System with the Java programming language. The Java coding gives the LEGO-robot a lot of abilities. Characteristics of Java are utilized in the coding, e. g., object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, and dynamic.

There are some issues that should be pay additional attention to when coding the program. 1) The robot will pick up the ball only when the touch sensor is pressed for the first time. The touch sensor could be pressed many times if the mat is not flat or something stuck underneath. But only the first time pressing will allow the ball to be picked up. 2) The light value for the light sensor should be set as a region not a single number. Because the color values on the line are not a constant number. There may have a place where is slightly darker or lighter than other places. In addition, under different illuminating conditions could influence the light sensor to define different values for the color line. 3) When the robot moves to a region that has a different color value from the previous one, we programmed to let the robot turn left at a small angle, then right at a small angle, and so on, until the previous value is found. Or, the robot will stop if no correct way is found.

5 Conclusions

In this project, a LEGO-robot automatic driving system is designed with UML. This robot is built to pick up a ball, find its way to follow a color curved line, put down the ball at destination, and show the sound value on the NXT screen. A touch sensor, light sensor, sound sensor, Ultrasonic sensor and three motors are used for the robot to complete this challenge. The system is designed with UML. A class diagram in the UML is used to describe the information and relations about seven classes, including the

CorrectingDirection class, the LegoMotor class, the SoundSensor class, the UltrasonicSensor class, the LightSeonsor class, and the TouchSensor class. In this project, a use case diagram is also used to describe our robot system's behavior as it responds to a request from people. A state diagram is used to describe the internal behavior in a class with natural states. UML as a graphical modeling language is a standard way for visualizing, specifying, constructing, and documenting an Object-oriented software system. This project indicated with an example that UML can be used to design a system independently. This project is implemented using Java with leJOS. The system works steady, which indicates that we have successfully designed the LEGO-robot automatic driving system using UML.

In this project, the robot is designed to complete some basic activities. For more complex activities, the robot system can be upgraded with more equipment. For example, we also can use two light sensors to detect the correct direction for the robot. A camera can be used to detect a terminus. We can also add parts to control the activities of the LEGO-robot by sound commands.

Acknowledgements

We would like to thank Joseph Shi and Cheng Zhou for helping us to build the LEGO-robot. We would like to thank for all valuable comments of reviewers.

6 References

[1] Unified modeling language (uml), version 2.0. Available from http://www.omg.org/technology/documents/formal/uml.htm.

[2] B. Bagnall. *Maximum Lego NXT: Building Robots with Java Brains*. Variant Press, 2007.

[3] M. R. Blaha and J. R. Rumbaugh. *Object-Oriented Modeling and Design with UML*. Prentice Hall, December 2004.

[4] LEGO development team. LEGO education. Available from: http://www.lego.com/education/competitions/default.asp.

[5] J. A. M. Fernandes, R. J. Machado, and H. D. Santos. Modeling industrial embedded systems with uml. In *Proceedings of the eighth international workshop on Hardware/software codesign*, CODES '00, pages 18–22, New York, NY, USA, 2000. ACM.

[6] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley Professional, July 2004.

[7] V. Saxena and M. Shrivastava. UML modeling and performance evaluation of multithreaded programs on dual core processor. *International Journal of Hybrid Information Technology*, 2, July 2009.

[8] LEGO Team. LEGO NXJ technology. LEJOS APIs. Available from: http://lejos.sourceforge.net/nxj.php.