Robot Control using Animation and Bluetooth

Parvathi Ram, Shivadharshne Ashokkumar, Ishaa Mane, Nila Sekar

National Public School, Indiranagar, Bangalore, India

And Vishnu Ram

vishnubright@gmail.com

***Abstract*- Wireless control of robots is becoming popular and so are mobile handsets. In this paper we describe the control of a robot using an Android mobile. We use Bluetooth as the wireless interface between the mobile and the robot. An android application serves as the graphical user interface for the application. Different user inputs are possible in the mobile application. We explain the use of animation to represent the actions of the robot. The commands given by the user in the animation file are analysed by an command-analyser and granular tasks are then transmitted to the robot by a communication module. This communication module is hosted in an android mobile. The robot includes an interpreter which will convert the commands from the mobile into motor controls. This is in addition to a manual mode where we enter commands. We use animation to input the actions to be performed by the robot. We demonstrate our ideas using a prototype, which is built using open source animation software, an android mobile and a Robot Control Board with ATMega16. Problems faced by us are described and future work is outlined.**

I. INTRODUCTION

Use of robotics to teach students various concepts is becoming popular. To sustain the interest of kids in scientific concepts and get them to explore the world of technology, we require a fun-filled approach. We understand the problems faced by kids, and have developed an application which allows children to program the robots easily. Our application, developed as of now for entertainment purposes, can be suitably modified with ease to for several other uses. This application, Toonbots, enables the translation of some animated files to codes decipherable by the robot, because of which the robot can mimic the actors on the animation screen. This makes programming of the robot simple and fun to do.

In general, to achieve the results of actions performed, robots have to be precisely controlled . Existing methods of controlling the robot includes: serial connection, sensors, etc. This requires three main components.

1) Inputs from the environment: Robots normally work in a specific environment. The interaction of the robot with its environment needs to be specified. Typically this is done using input from sensors such as Infrared sensor.

2) Processing the information in the robot processor: This could be done using embedded programs in robots e.g. programming of processors like Atmel processor using languages like C.

3) Dissemination of these commands to its various parts: This internal communication of messages to robot parts, for e.g. motors in arms or motors under wheels fall under this category. This is done using internal messages or implemented as low-level Application Programmer Interface (API) libraries.

Emergence of wireless protocol like bluetooth has added a new dimension to this robot control space. Programmers now have the option of using these wireless protocols to interface with the robot.

Nowadays, programming of robots has become child play, using plug and play mechanisms such as LEGO® MINDSTORMS® EV3 [8]. However, mapping of user input to the robot actions in the environment in which it acts remains a challenge. Specifically, in our paper, we try to open up the world of programming to inexperienced programmers for example, children by using animation techniques. In our prototype we demonstrate the use of an open source animation software to specify the actions for the robot. This makes it interactive and convenient for first time robotics enthusiasts.

Mobiles have become ubiquitous. Control of robots using a GUI Interface has been discussed in [1]. The combination of short range wireless networking and mobiles is a potent interface towards robots. Extensible open source development, like android, makes it easier for us children to develop powerful applications in mobiles to control robots. In our prototype, we use an android application running on the mobile to control the robot. The inputs to this android application comes from the animated avatars.

Animation software, for example, Alice 3.1 [9], gives mechanisms to model and build virtual environments, specify characteristic for avatars or characters and corresponding action sequences. When these sequences are represented in openly specified and implemented using open source, that gives a powerful platform to model, represent and specify the corresponding actions for robots in real life.

In this paper, we describe the mechanisms to control a robot using bluetooth wireless, sending commands to it using an android app running on mobile and specifying these action sequences using open source animation software. Our paper has the following sections: problem statement, existing work in the area, our solution, implementation, advantages, conclusion, references.

II. PROBLEM STATEMENT

Programming remains a major hurdle for children and first time robotics enthusiasts. In recent times, high level programming languages have emerged. These help in alleviating the problem but does not bridge the gap between a user's imagination and the robot's actions. Mapping the environment to the robot and relating the robot’s actions with respect to the environment and other actors remains a challenge. Micro-level programming languages, while being powerful, does not solve the bigger problem of representing the environment and modeling the actions of other actors and the robot in a single platform. Weaving together a complete story involving various actors and the robots is essential to the accuracy of robot programming. The current mechanisms available in the embedded programming of robots, for e.g. LEGO, are inadequate to address this problem.

Robots are used for various applications, where the environment is hazardous to humans. For e.g. in a nuclear power plant there are materials which can be easily handled by robots, but is unsafe for humans. An interactive approach to control the robot is required in this case. Control stations are used to control the robots in such situations, but handheld, portable controllers makes it easier for humans to interact with robots. The proliferation of mobiles makes it a natural choice for hosting these control applications.

In entertainment applications, where robots do standalone role-playing, it is cumbersome to use wired interface like serial cables. This makes it imperative for us to use wireless interface to control the robots [2]. Wired interfaces also have the disadvantage of limited length and reachability. Wireless interfaces also gives us an ability to get around obstacles and gives a better manoeuvrability to robots. Our approach is to solve these problems with a combination of animation software to aid in programming and wireless control of robot from the mobile.

II. EXISTING WORK IN THIS AREA

Traditional methods of programming the robots include specific high level programming languages like Robot C, low level programming languages like C, or assembly.

The following mechanisms are popular among robot enthusiasts to control the movement of robots.

Pre-assembled robot kits: These kits provide building blocks for making robots. They are popular because of their ease of use. Different components come pre-assembled with all software. For e.g. making a LEGO [8] based robot is easy due to availability of pre-built software components which accompany the robot.

Sensor based Robot Control: Different sensors such as infrared and ultrasound sensors can be mounted on a robot to determine obstacles, presence of objects, etc. Movement of robot parts can be real-time controlled using the input from these sensors. The following algorithms could be used in combination with the sensors:

- Line Following Algorithm- Line Following Algorithm [3] is a foolproof way of navigating for robots. The distinction between the line and the rest of the ground is used to navigate the robot. Every cell in the sensor is made up of a sensor-receiver pair. Reflection of light is used to determine the path.

- Obstacle Avoidance Algorithm- Another important algorithm is the object avoidance [5]. This is done by assuming that the ground is of a single, even colour. Then the robot filters the colour out to reveal only the edges. Mapping these edges and its final goal point, the robot proceeds to the goal without touching the edges.

Use of animation software in the educational domain has been studied in detail. [6]

IV. OUR SOLUTION

Our solution comprises of the following components:

1) A manual/animation-based mechanism to model the environment and specify the actions of the robot.
2) A sequence analyzer software module which translates the animated steps to granular tasks.
3) Distribution of these granular tasks to the robot using an android software module.

Figure 1 explains the architecture of our solution. We propose 2 mechanisms to control the robot.

1) A manual sequence of calibrated messages are sent from a Graphical User Interface (GUI) to the robot over Bluetooth. This results in predetermined actions in the robot. Eg. Move, Turn, etc.

2) Alice-based animation program [9] is used to represent an avatar of the robot and generate a sequence of procedures to control the robot. These are sent over Bluetooth to the robot which results in the robot mimicking corresponding actions of the avatar.

An android program hosted on a Bluetooth capable mobile is used to send control messages to the robot. Details of our prototype implementation are given in section V.

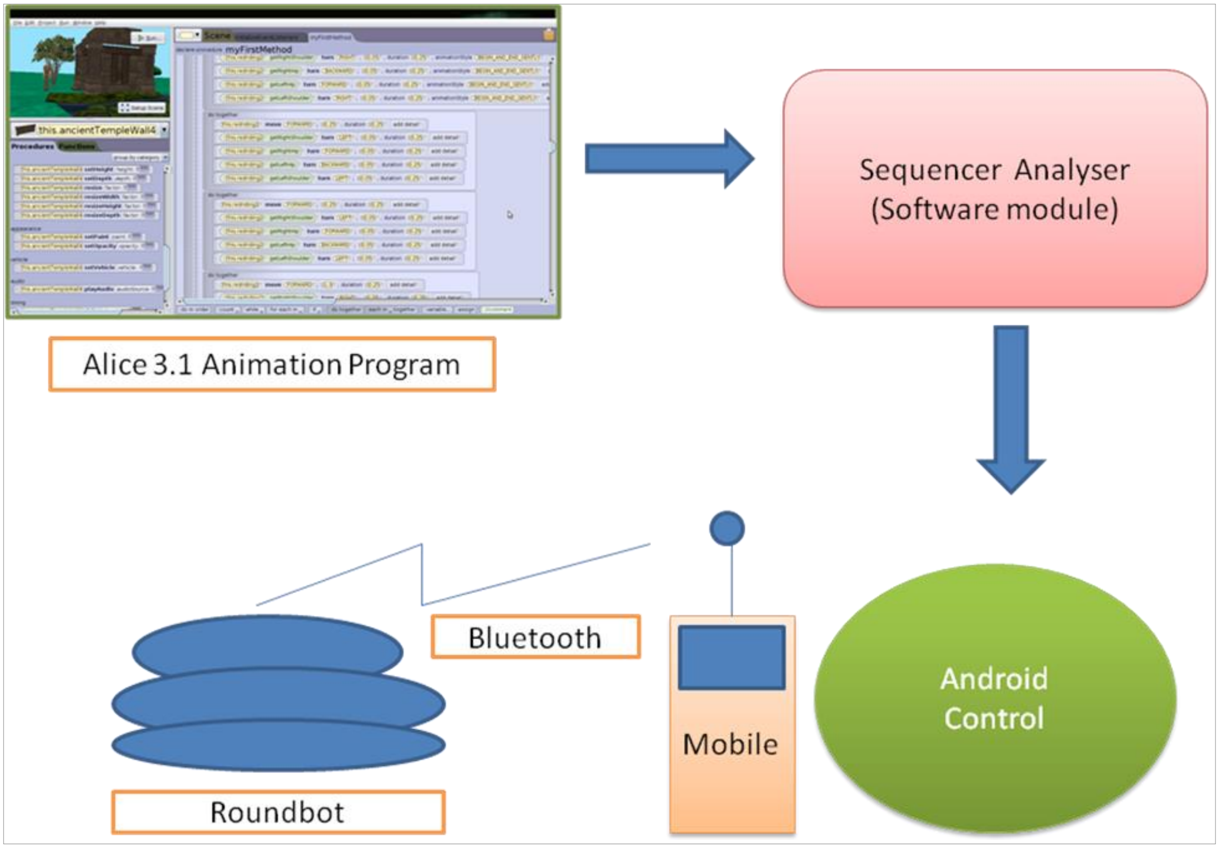

Figure 1: Bluetooth and Animation Control of Robot

V. IMPLEMENTATION

V.1 CREATING ROBOT CONTROL SPECIFICATIONS USING ANIMATION

We used the animation software Alice 3.1 to create virtual environments, actors and specify actions to them. As shown in figure 2, Alice provides mechanisms to specify certain characteristics of avatars like gender and age, and certain characteristics of environments such as Grass, Desert, Underwater, etc. We can make a list of actions that the avatar has to perform such as Turn, Move, Point At, etc. These actions can be compiled into a narratable sequence using the `Do In Order` or `Do Together` functions. Thus we are able to weave together this list of procedures by individual avatars to form a verifiable sequence. This can be then played using the run command to form a complete animation strip. The strip can be iteratively verified and tuned to adjust the actions and behavior of avatars which represent the robot.

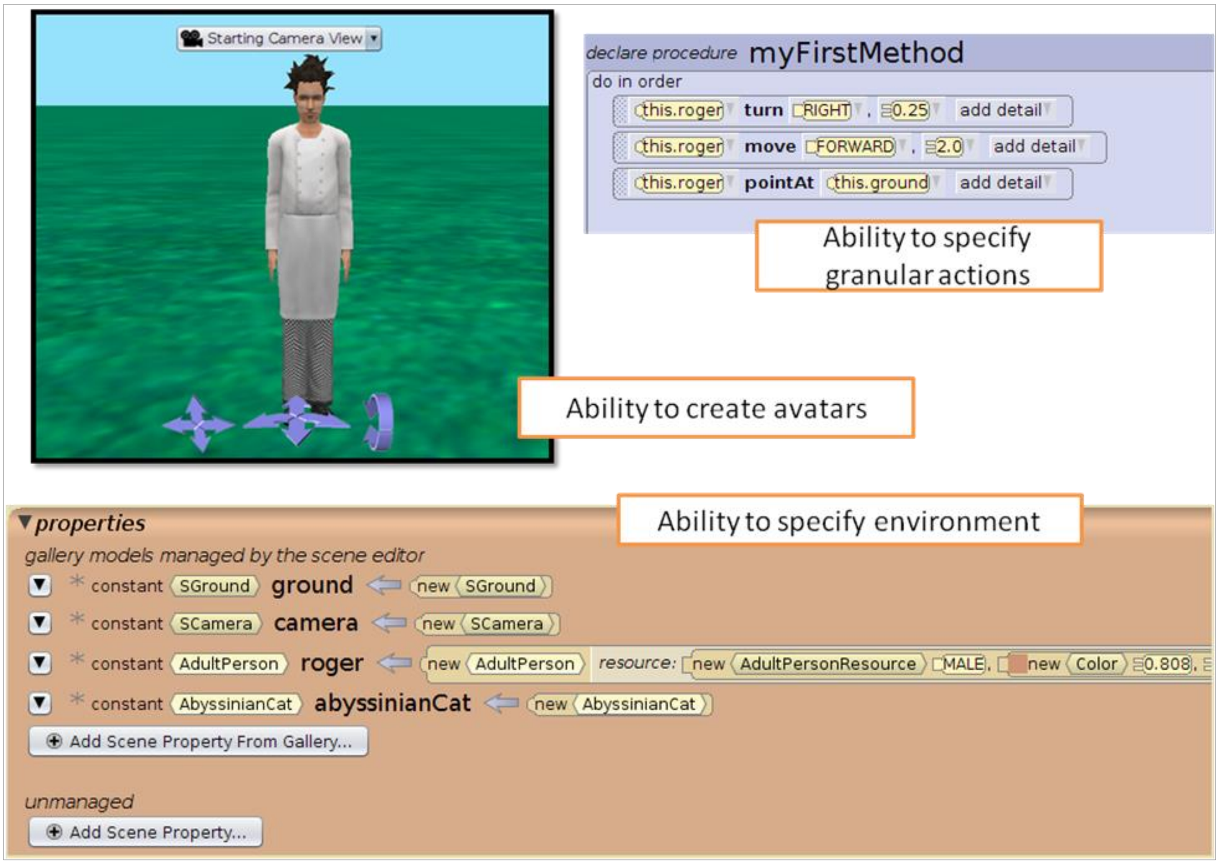

Figure 2: Alice 3.1 Screenshot

V.2 CONVERTING ANIMATION SPECIFICATIONS TO CONTROL COMMANDS

In our approach, the animation software saves the animation sequence in an open format using Extended Markup Language (XML)[10], a sample of which is given in figure 3. This shows the definition of an avatar named "Roger". Figure 4 shows the corresponding action definition for Roger. Figure 5 is an example of a particular action "Turn".

```xml
</property>
<property name="field">
<node key="44" type="org.lgna.project.ast.UserField" uuid="f35d4194-78d1-4d67-bd6e-8f08cd4bf1f1">
<property name="name">
<value type="java.lang.String">roger</value>
</property>
```

Figure 3: Avatar Definition in XML for "Roger"

```xml
<node key="7b" type="org.lgna.project.ast.JavaType" uuid="9a409466-c938-4ea8-a5a3-e024903152a1">
<type name="org.lgna.story.event.SceneActivationEvent"/>
</node>
</property>
</node>
</collection>
</property>
<property name="body">
<node type="org.lgna.project.ast.BlockStatement" uuid="24d75534-e003-4c1f-a1d3-bd0edcf5169c">
<property name="statements">
<collection type="java.util.ArrayList">
<node type="org.lgna.project.ast.ExpressionStatement" uuid="892a5c99-16c3-4136-95bd-981523d97ab0">
<property name="expression">
```

Figure 4: Action Definition in XML

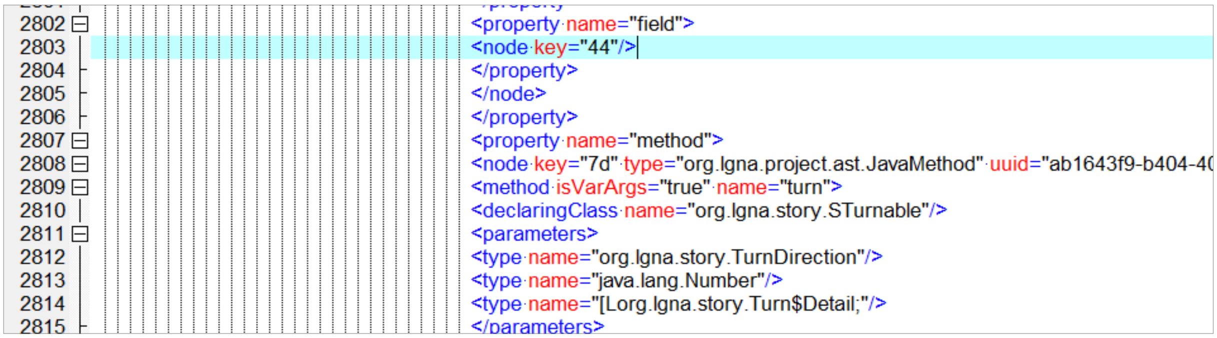

Figure 5: Action Specification in XML for "Roger" to turn

We map these actions corresponding to the avatar into commands to the robot.

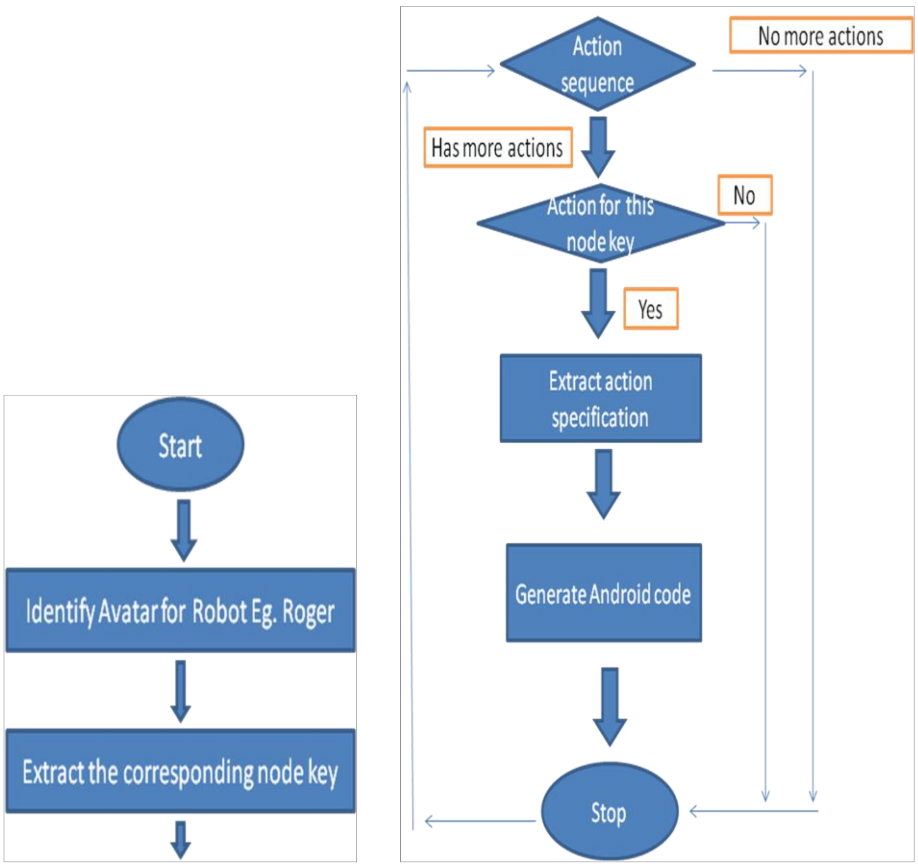

Figure 6: Flowchart for mapping animation sequence to control commands

The flowchart described in Figure 6 has been prototyped using a C based XML parser.

V.3 DISTRIBUTION OF GRANULAR TASKS IN ROBOT

We use an android program running on the mobile to send the control commands mapped from the animation sequence to the robot. We use a Bluetooth interface to communicate between the android mobile and the robot. Our robot specifications are as below-

1) Robot Controller Board with ATMEGA16 processor mounted on a Roundbot
2) 2 Servo motors and Pan Tilt for arm movement
3) 2 DC motors for lateral movement
4) Bluetooth UART(universal asynchronous receiver/transmitter) module
5) USB-Bluetooth Servo Controller Board

Once the Robot Controller Board has been installed, it can be programmed using an AVR programmer. Once this is done, we can use the Quick C Integrated Development Environment to program the board using high level C statements, a sample of which is given in figure 7.

```
INIT();
UART_INIT(UART_BAUD_RATE);
MOTORINIT();
while(1)
{
    ip1=0;
    ip2=0;
    LED2OFF();

    j=UART_GETCHAR();
    if (j=='W' || j == 'S' || j=='A' || j == 'D' || j=='O')
        {
            if(j=='W')
                {
                ip1=1;                  //Forward
                FORWARD(speed);
```

Figure 7: A snippet of C code which controls the robot

VI. ADVANTAGES

Our approach has the following advantages

1) We use a wireless interface between the controller and the robot. The controller is hosted in an android capable mobile. This makes our system user-friendly and highly maneuverable.

2) In addition to providing a manual control to the robot, we also provide an animation based method to represent the robot and its actions.

3) Open source prototype and an open format for representing the granular tasks in the animation software makes it easier for us to extend and translate the actions of the avatar in the virtual environment into actions by the robot in the physical environment.

4) Use of the animation software for representing the actions is intuitive because of the similarity between the action sequence specifications for the avatar in the virtual environment and the robot in the physical environment. It also gives us the flexibility of iteratively reviewing and tuning of the robot actions without actually investing in building the robot. This we see as the biggest advantage and cost saving factor.

Our intention by this project is to open up the world of robot programming to children like us who are good in art and animation and interested in robotics.

VII. CONCLUSION

Use of robots are becoming popular in many different domains. However, difficulty in programming robots is becoming a hurdle for children, robotics enthusiasts and other first-timers. In this paper, we presented three mechanisms to make their life easier. We proposed an animation based programming approach to control the robot, a Bluetooth- based wireless interface to control the robot and an android mobile to host the controller. We discussed the architecture of our solution, the details of our prototype implementation, and the advantages of our approach.

Our future work includes a standard module for mapping animation to robot control. We intend to enhance our prototype to handle more representations and actions in the sequence analyzer module, and control a group of robots.

REFERENCES

[1] Jong Hoon Ahnn, "The robot control using wireless communication," published.

[2] Rodney A. Brook, A robust layered control system for a mobile robot," published.

[3] https://www.youtube.com/watch?v=4IW3h2ICyr8

[4] http://www.societyofrobots.com/member_tutorials/book/export/html/71

[5] http://www.roborealm.com/tutorial/Obstacle_Avoidance/slide010.php

[6] Ting-Chung Wang, etal. "Teaching Programming Concepts to High School Students with Alice," published.

[7] http://www.ikalogic.com/line-tracking-sensors-and-algorithms/

[8] http://shop.lego.com/en-GB/LEGO-MINDSTORMS-EV3-31313

[9] www.alice.org

[10] http://www.w3.org/XML/

[11] C J Chung and Christopher Cartwright, "RoboParade: a Fun and Effective Way to Promote STEM Education" published.

[12] Maurice Tedder, David Chamulak, Li-Ping Chen, Santosh Nair, Andrey Shvartsman, ITseng, and Chan-Jin Chung, "An Affordable Modular Mobile Robotic Platform with FuzzyLogic Control and Evolutionary Artificial Neural Networks", The Journal of Robotic Systems, Vol 2, Number 8, Aug. 2004, published.