MULTIMODAL HUMAN ROBOT INTERACTION

Submitted in partial fulfilment
of the requirements of the degree of

BACHELOR OF SCIENCE (HONOURS)

of Rhodes University

Mark John Swaine

Grahamstown, South Africa
November 2009

Abstract

Human-robot interaction (HRI) is a very broad area of study. This research focuses on multimodal human robot interaction and developing effective interaction techniques to also allow for remote robot manipulation. The aims of this research are to develop a system that combines inputs from different interfaces in order to interact with a robot, and to ascertain which interface or combination of interfaces is best suited to effective human-robot interaction and collaboration. Validation of the objectives is carried out using a LEGO NXT robot. Evaluation reveals that tangible user interfaces are most effective for efficient human-robot interaction, due to enhanced usability and natural methods of control.

ACM Computing Classification System Classification

Thesis classification under the ACM Computing Classification System (1998 version, valid through 2009):

I.2.9 [Robotics]: Operator Interfaces

H.5.2 [User Interfaces]: Interaction Styles

General Terms: Human factors, Performance

Acknowledgements

Firstly, I would like to thank my supervisor Dr Karen Bradshaw for her support, encouragement and patience throughout the year. I also acknowledge the input from my peers in the same research group, Leslie Luyt, Ryan Davey and Ghislain Fouodji.

I would also like to thank the Department of Computer Science of Rhodes University for all the resources that made this project possible. I also acknowledge sponsorship of resources from Microsoft SA, Telkom SA, Comverse SA, Stortech, Tellabs, Amatole, Mars Technologies, Bright Ideas Projects 39 and THRIP through the Telkom Centre of Excellence at Rhodes University.

Lastly, I'd like to acknowledge my parents for their continued support, both emotional and financial. Without help from them this project would not have been possible.

Contents

1 Introduction **7**

1.1 Problem Statement 7

1.2 Research Goal 8

1.3 Document Structure 8

2 Related Research **9**

2.1 Introduction 9

2.2 Human Robot Interaction 9

2.2.1 Background 10

2.2.2 Current Work 10

2.3 Lego Mindstorms NXT 11

2.3.1 Technical Specifications 11

2.3.2 Sensors 12

2.4 Modalities 12

2.4.1 Web 12

2.4.2 Mobile 13

2.4.3 Game Controllers 14

2.4.3.1 Nintendo Wiimote 14

2.4.3.2 Microsoft Xbox 360 Controller 14

2.4.4 Speech 15

2.5 Bluetooth Communication 15

2.6 Applications of Work 16

2.7 Conclusion 16

3 System Design 18

3.1 Introduction 18

3.2 Overview 18

3.2.1 The Server 18

3.2.2 Game Controllers 19

3.2.2.1 Nintendo Wiimote 19

3.2.2.2 Microsoft Xbox 360 Controller 20

3.2.3 Mobile 22

3.2.4 Web 22

3.2.4.1 AJAX 22

3.2.4.2 Interaction 22

3.2.5 Speech 23

3.2.6 Interaction 23

4 System Implementation **24**

4.1 Introduction 24

4.2 Server 24

4.2.1 The Protocol 26

4.2.2 TCP Communication 27

4.3 Game Controllers 27

4.3.1 Nintendo Wiimote 27

4.3.2 Microsoft Xbox 360 Controller 27

4.4 Mobile 29

4.4.1 J2ME 30

4.4.2 Implementation 30

4.5 Web 31

4.5.1 The Google Web Toolkit 31

4.6 Speech 34

4.7 Summary 34

5 Evaluation and Results **35**

5.1 Introduction 35

5.2 User Evaluation 35

5.2.1 Speed and efficiency 35

5.2.2 Usability 36

5.2.3 Other Tests 38

5.3 Summary 38

6 Conclusion and Future Work **39**

6.1 Summary 39

6.2 Future Extensions 39

References **41**

A Project Poster **45**

List of Figures

3.1 System overview 19
3.2 Nintendo Wiimote 20
3.3 Xbox Controller Interface 21
3.4 Keyboard Interaction 23

4.1 The web interface 34

5.1 Speed and efficiency course 37

Listings

4.1 Connecting and Initialising the Robot 25

4.2 The “move” method 26

4.3 Protocol message example 27

4.4 Wiimote Accelerometer Desensitising 28

4.5 Getting data from the Xbox 360 controller 29

4.6 Mobile keyPressed handler 30

4.7 NxtWebService.java 32

4.8 NxtWebServiceAsync.java 32

4.9 NxtWebServiceImpl.java 33

Chapter 1

Introduction

The aim of this project is to develop a distributed system that will allow for effective multimodal communication and collaboration between humans and robots. The system should allow for easy adding and removing of modalities.

Human-robot interaction is a fairly new area of focus, but research into this area is growing rapidly due to the integration of robots and other electronic devices into our daily lives. This project aims to create multiple redundant interaction techniques to control a robot, including remote interactions.

1.1 Problem Statement

As the number and scope of applications for human-robot interaction expand, so too does the importance of developing efficient and effective techniques to manage and improve these interactions. For effective interaction between humans and robots, it is essential to provide a number of utilizable, redundant communication channels. Users should be able to focus on the task at hand, and not be concerned with how the interaction takes place. Personal robots should be accessible from anywhere, and therefore embedding them on the Internet is a logical step, but one that presents us with multiple challenges.

1.2 Research Goal

The goal of this project is to make use of various technologies to develop a modular, distributed system to enhance the way in which humans and robots interact. The system should handle multiple modalities and be easily extendable.

The goal is also to ascertain whether tangible user interfaces (TUIs) present any advantage over normal key and mouse based interfaces. The research will focus on the LEGO NXT robotics kit as a proof of concept, but this research could be extended to other robotics platforms as well.

1.3 Document Structure

The remainder of this dissertation is organized as follows. Chapter 2 presents previous and current work in the field of human-robot interaction. The LEGO NXT robotics kit is introduced and the technical specifications given. An overview of the proposed modalities are given and related work in human-robot interaction using these modalities is presented. Bluetooth technology is discussed and is found to be a suitable communication platform between a computer and LEGO NXT robot. Applications of work in the field of human-robot interaction is also presented.

Chapter 3 introduces the proposed design of the multimodal interface. The design and purpose of the server component of the system is discussed together with the design of each of the modalities.

The implementation of the system is described in Chapter 4 of the dissertation. The programming languages, development environments and libraries used are discussed. The main concepts and algorithms used are presented and discussed.

Chapter 5 describes the evaluation procedures and results of a user study. The two tests conducted were a speed and efficiency test, and a usability test. The results show that tangible user interfaces are more effective for human-robot interaction.

Chapter 6 concludes the dissertation and describes possible extensions and future work in this field of research.

Chapter 2

Related Research

2.1 Introduction

This chapter introduces the field of human-robot interaction by making use of a LEGO robotics kit to demonstrate the interactions, the different modalities or methods of interaction relevant to the project, and the communication channels and applications of this work. The first section introduces human-robot interaction and the current work being done in the area. The next section describes the LEGO NXT robotics kit and all its components. The third section discusses multimodal systems and the different modalities focussed on for human-robot interaction. The following section provides a brief outline of Bluetooth technology as this is the main communication channel between the LEGO NXT robot and a personal computer. The final section of the chapter discusses the uses of mobile robots and human-robot interaction.

2.2 Human Robot Interaction

Human-robot interaction (HRI) is a fairly new subfield of human-computer interaction (HCI). A great deal of effort and research in the robotics field has focused on the development of hardware and software to improve and extend robot functionality and intelligence. Relatively little research has been done on the techniques and controls people use to interact with and command robots, despite the fact that robots are continually being deployed in broader fields and more demanding areas of work where meaningful, efficient and effective interaction between humans and robots is essential [16].

2.2.1 Background

Robots are increasing in power and are able to do much more than ever before. However, integrating them into our daily lives and interacting with them in natural meaningful ways remains a challenge. The goal is the integration of classic interfaces such as graphical I/O devices, newer interfaces such as speech, and remote control and mobile interfaces.

To develop human-machine interfaces that allow for intuitive human interaction and control is the goal of the leitproject MORPHA, an acronym for Interaction and Communication between Humans and Anthropomorphic Robot Assistants [32]. Communication, collaboration and interaction between humans and robot assistants is the central idea behind MORPHA. The communications between humans and robots should be human-friendly, in other words, easy for humans to interpret [22]. MORPHA is made up of a consortium of research institutions based in Germany, industry, and small and medium enterprises. MORPHA is intended to "pave the way for new types of assistance systems in the fields of industrial production, housekeeping, and home care." [22].

The goal of effective interaction between humans and robots makes it essential to provide a number of utilizable, redundant communication channels. The challenge is to integrate these interactions to enable the robot to perform useful tasks [22].

2.2.2 Current Work

Currently there is a lot of research and work on equipping robots with the capacity to interact with humans in a natural way, where both share the same interaction channels [19]. A big area of research at the moment is in space robotics [9]. The interactions between humans and robots during extended space missions will be unlike anything that NASA has designed and implemented to date. Robot teams may at times be operated from ground control, while surface astronauts may communicate with robots using voice-commands, gestures, and wireless mobile communications [9].

2.3 Lego Mindstorms NXT

The LEGO Mindstorms NXT is a programmable robotics kit, released by LEGO in July 2006 [23]. There has recently been much interest in achieving educational and research objectives through the use of low-cost robot kits such as the NXT [15]. The main component of the NXT set is a brick shaped device known as the NXT Brick. It can accept input from up to four sensors and control three servo motors.

2.3.1 Technical Specifications

The technical specifications of the NXT Brick are as follows:

- 32-bit AT91SAM7S256 (ARM7TDMI) main microprocessor @ 48 MHz (256 KB flash memory, 64 KB RAM)
- 8-bit ATmega48 microcontroller @ 4 MHz (4 KB flash memory, 512 Bytes RAM)
- CSR BlueCore 4 Bluetooth controller @ 26 MHz (8 MBit external flash memory, 47 KB RAM)
- 100×64 pixel LCD matrix display
- Can be programmed using Windows or Mac OS (NBC/NXC supports Linux as well)
- Users create programs with new software, powered by LabVIEW from National Instruments
- A single USB 1.1 port full speed (12 Mbit/s)
- Bluetooth (Class II) wireless connectivity, to transfer programs to the NXT wirelessly or offer ways to control robots remotely (through mobile phones and possibly PDA's)
- 4 input ports, 6-wire cable digital platform (One port includes an IEC 61158 Fieldbus Type 4/EN 50 170 (P-NET) compliant expansion port for future use)
- 3 output ports, 6-wire cable digital platform
- Digital Wire Interface, allowing for third-party development of external devices [34]

2.3.2 Sensors

The LEGO NXT kit contains the following parts:

- Three servo motors that can sense their rotations to within one degree of accuracy.
- A touch sensor that can detect whether it is currently being pressed, released or has been bumped.
- A light sensor that can detect the light level on a scale of 0 to 100, with 100 being very bright and 0 being very dark.
- A sound sensor that can detect a sound's amplitude on a scale of 0 to 100, with 100 being very loud and 0 being no sound.
- An ultrasonic sensor that can measure the distance from an object. It can measure up to a distance of 233cm with a precision of 3cm. [34]

2.4 Modalities

With an integrated system, users need not be concerned with how to communicate with the robot, leaving them free to focus on the task at hand. By integrating the modalities to create a multimodal system, a user can choose any combination of the modalities [31]. The advantage of working with a multimodal system lies in redundancy.

Human-robot interaction becomes more natural and intuitive as the level of flexibility and redundancy of the human interface increases [18]. Moreover, a multimodal system is able to function more robustly and is less error prone than using an individual technology [30].

2.4.1 Web

One of the earliest implementations of Internet robots was the 1994 Mercury Project [13]. This project was set up as a feasibility study to allow a robot to be manipulated and controlled via the World Wide Web. The primary goal was to develop a system that would be reliable enough to operate 24 hours a day. It was the first project of its kind to be undertaken, and over 50 000 unique hosts accessed the interface during its availability period between September 1994 and March 1995.

The World Wide Web provides an easy, low-cost and widely accessible interface to a mobile robot. However, direct robot control is not suitable for Internet based mobile robot operation because of the high-latency associated with the Internet communications [26]. For Internet robots, it is common to use supervisory control by giving the robot more local intelligence [26]. Thus, an Internet controlled robot uses the Internet mainly as a command transmission and sensor feedback medium [17].

There are 3 major issues faced by Internet Robots [17]:

1. High latency, lag and packet loss means that data transmission cannot be guaranteed.
2. The Internet allows inexperienced users to guide the robot. The user may not be familiar with the technology and could potentially harm the robot.
3. The web interface has to be easy to understand.

2.4.2 Mobile

Although there has been unprecedented growth in wireless services and technologies in recent years, very little research has focused on the possible uses of these systems for remote mobile robot interaction [6]. An existing interface for remotely controlling a mobile robot makes use of a Mobile Information Device Profile (MIDP)-enabled mobile phone [6]. This interface was designed at the University of Madrid to send direct control commands to the robot. This mobile method of interaction opens up many new possibilities to businesses and industries to provide innovative security services.

In 2002 Fujitsu [12] announced the release of a mobile phone controlled robot, the MARON-1. The robot can be remotely controlled via a mobile phone to operate home electric appliances or to monitor household security. With the remote operation the robot is able to take pictures of its surroundings and relay these to the mobile phone.

2.4.3 Game Controllers

Tangible user interfaces (TUIs) take advantage of embodied interaction by coupling physical objects with their computerized counterparts [16]. This provides users with simple, natural physical interaction metaphors. TUIs make efficient and effective use of affordances [16], which provide clues as to how things work or are operated. For example, it would make logical sense for an up arrow button on a TUI to represent the forward movement of a mobile robot. TUIs allow users to “perceive and act at the same place and at the same time” and to be more attentive and focus on the current task [16]. “Tangible user interfaces (TUIs) exploit embodied interaction, coupling physical objects with computerized qualities, and ideally empowering users with simple and natural physical interaction metaphors.” [16].

2.4.3.1 Nintendo Wiimote

The Wiimote is a rectangular hand-held device included with the Nintendo Wii console, but which can also be purchased separately. The primary features of the Wiimote are the three-axis accelerometer, seven buttons including a four-way directional keypad and Bluetooth connectivity [8]. The Wiimote can be seen as a generic 3D TUI as well as a very successful 2D TUI. The Wiimote can also be seen as a gestural interface, representing a gestural/TUI duality [16].

Guo and Sharlin [16] successfully used the Wiimote as a robotic interface to control a Sony AIBO robotic dog, and came to the conclusion that a gestural input scheme with a tangible user interface outperforms traditional input devices, in this case a keypad, in terms of speed and accuracy.

2.4.3.2 Microsoft Xbox 360 Controller

The Xbox 360 controller [27] is obtainable in both wired and wireless varieties and comes bundled with the Microsoft Xbox 360 console or can be purchased separately. To connect the wireless controller to a PC, a special Microsoft wireless receiver must be used. The primary features of the Xbox 360 controller are the two analogue sticks and an 8-directional keypad.

Ballsun-Stanton and Schull [2] successfully used an Xbox 360 controller to control a cable-array robotic sculpture in the form of a Manta. Fox News [10] also has a news article depicting the use of an Xbox 360 controller to control an unmanned vigilante robot helicopter known as the Autonomous Rotorcraft Sniper System (ARSS).

2.4.4 Speech

To perform speech recognition, a signal is acquired, digitally processed and then analyzed to extract the features of the speech. The design of a speech recognition grammar is very important because human spoken language is particularly ambiguous [1]. A number of projects have succeeded in integrating speech recognition as part of a multimodal robotic interface. Perzanowski et al. [31] implemented a multimodal interface consisting of speech and gesture recognition on a team of Nomad 200 and RWI ATRV-Jr. robots. The purpose of this was to be able to control one or all of the robots. by addressing them individually or as a team.

2.5 Bluetooth Communication

Bluetooth is a short-range, low-power wireless technology for data exchange and communication. It enables the connecting of electronic devices to form Personal Area Networks (PANs) as well as ad hoc networks [7]. Since the inception of Bluetooth in 1998, it has been adopted and is rapidly being developed by large influential companies such as Ericsson, IBM, Toshiba, Nokia, Intel and Microsoft [11].

Bluetooth technology was originally developed with the purpose of replacing cabling between devices [11]. Bluetooth operates on the unlicenced ISM (Industry Scientific Medical) band at 2.4 GHz. Bluetooth class 1 has a range of about 10 metres, while Bluetooth class 2 has a range of up to 100 metres. Due to ISM being an open band, Bluetooth is susceptible to sources of interference such as microwaves and 802.11 wireless networks [7]. To minimize such a risk, Bluetooth technology makes use of a Frequency Hopping Spread Spectrum (FHSS). This allows multiple Bluetooth networks and devices to operate concurrently without interfering with each other [7].

With wireless communication capabilities, mobile robots can connect to other mobile robots or computers. Therefore mobile robots can be controlled and monitored wirelessly using a computer. Bluetooth is well suited to this task due to its low cost, low power consumption and size. The computer acts as the master with the mobile robot being the slave device. Full-duplex Bluetooth communication enables the robot to send and receive messages to and from the computer [3].

2.6 Applications of Work

Mobile robots are often found in industry, military and security environments, as well as consumer products for entertainment or various household tasks such as a robotic vacuum cleaner.

Many researchers have reported work in the field of HRI with elderly people with dementia and children with autism [5]. A museum guidebot has been developed with the emphasis on friendly human-robot interaction through non-verbal behaviours [21].

Mobile robots have also been used as EOD (explosive ordnance disposal) robots for safely removing explosives from areas difficult for humans to reach [4].

Mobile robots are being used in the military for unmanned exploration and even for defence and active engagement [28, 29].

Personal mobile robots can be used in homes for monitoring security, performing certain household tasks and to relay information back to the user.

2.7 Conclusion

Current and previous work in the area of human-robot interaction has been explored. It has been established that human-robot interaction research is important and necessary for effective interaction between humans and robots. Interactions should be natural and simple for human users to learn and understand so that they can focus on the tasks at hand rather than focusing on how to interact with the robot.

The LEGO NXT robotics kit was introduced and the capabilities and limitations of the sensors discussed.

Different modalities for a multimodal human-robotics interface were introduced and where relevant, previous work using a particular modality was presented.

Bluetooth communication technology was introduced and found to be a suitable channel of communication between a mobile robot and computer due to its wireless capabilities, low cost, and low power consumption.

Applications of mobile robots and interactions with humans were discussed and numerous scenarios of mobile robots in the real world were shown.

Chapter 3

System Design

3.1 Introduction

The design goal of this project is to develop a system that accepts inputs from multiple modalities, and then merges and translates these to signals that a robot can understand. This chapter describes the conceptual design of such a system.

3.2 Overview

The system is designed as a client-server architecture, in which the different modalities are implemented as clients. Using a client-server architecture, makes the task of adding more clients (modalities) trivial. The clients communicate with the server, and the server processes the data and sends the instructions to the Lego NXT robot over a Bluetooth connection. This system architecture is illustrated in Fig. 3.1.

3.2.1 The Server

The server is the central part of the system. It manages the data being sent from the different modalities, merging and translating the data and handling the communication with the Lego

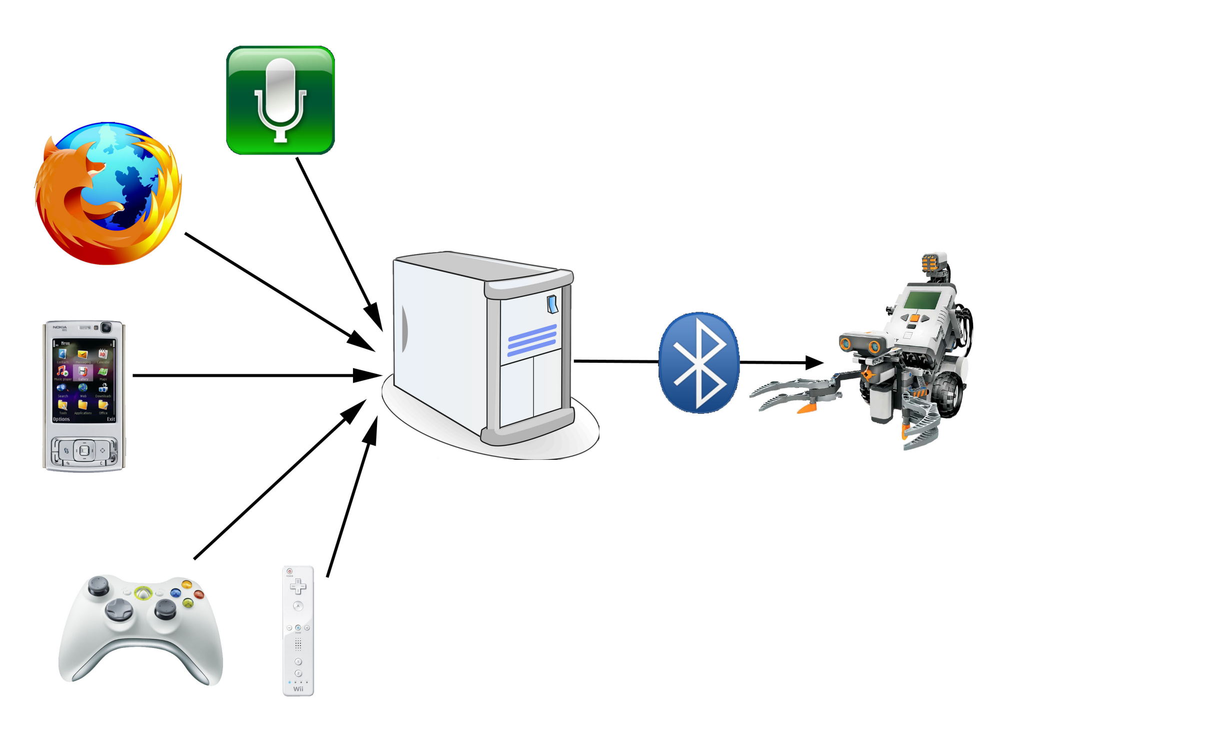

Figure 3.1: System overview

NXT robot. The server consists primarily of two parts with the first part handling communication between the server and the robot, and the second part handling communication between the clients and the server.

Communicating with the Lego NXT robot is done using a channel created between the computer and the robot by means of a Bluetooth COM port. This enables message communication to take place between the robot and server.

3.2.2 Game Controllers

3.2.2.1 Nintendo Wiimote

The Nintendo Wiimote connects to a computer over Bluetooth. The computer picks it up as a Bluetooth HID (Human Interface Device Profile) device. Bluetooth HID is a wrapper of the Human Interface Device protocol defined for USB, and provides support for interaction devices such as mice, joysticks and keyboards.

Figure 3.2: Nintendo Wiimote

Due to the built in accelerometer sensors, the Nintendo Wiimote is able to pick up changes in motion, and can therefore act as an effective gestural interface. "Using gestures to interact with robots is not a new idea. A significant amount of work has been done using either vision based or glove based mechanisms to capture human arm and hand gestures." [16].

Although the Wiimote is a 3-dimensional gestural interface, in this project it is be used to control only two dimensions: forward-backward robot movement and left-right robot movement. Tilting the Wiimote downward will move the robot forward, tilting it upward will move the robot backward, tilting it left will move the robot to the left and tilting it right will move the robot to the right. Figure 3.2 shows the Nintendo Wiimote.

3.2.2.2 Microsoft Xbox 360 Controller

Microsoft's Xbox 360 Controller connects to a computer using a Microsoft wireless receiver. This enables wireless communication to take place between the computer and the controller. The controller is a suitable tangible user interface, that can be used to control two dimensions: forward-backward and left-right. This allows for easy natural mapping to the Xbox 360 controller analogue sticks. The left analogue stick controls the forward-backward movement while the right analogue stick controls the left-right movement. Figure 3.3 illustrates this mapping.

Figure 3.3: Xbox Controller Interface

3.2.3 Mobile

Mobile wireless devices are affordable and abundant in today's world and are therefore a great way to interact with a robot remotely. Almost all mobile phones are able to run small Java applications which then connect to a server using a data connection. When successfully connected, the user will be able to manipulate the robot's movements using the device's directional pad or the numeric keypad.

3.2.4 Web

The Web has developed at a rapid pace, but until recently, developing highly interactive applications and interfaces on the web was a great challenge. Web 2.0 is the next generation web which aims to revolutionise methods of interaction, styles of development and content [24]. The idea of Web 2.0 is to let users run applications entirely through a browser without any local machine installation. These new feature rich programs are generally achieved with AJAX (Asynchronous JavaScript And XML) [20].

3.2.4.1 AJAX

AJAX is a technology that enables web applications to communicate with the web server in the background, thus not needing the page to reload every time an action is performed. This increases the level of interaction and adds enhanced functionality. This is the Asynchronous part of the AJAX acronym.

JavaScript is what allows a web page to contain a program. A script can be embedded into an HTML page, that allows the web page to connect to a server.

Extensible Markup Language (XML) is used to pass and share information. XML is the protocol for exchanging data between the web browser and the service. [20].

3.2.4.2 Interaction

Interaction with the robot is done using the arrow keys on the keyboard, as illustrated in Fig. 3.4. A keyboard listener picks up any keypress events and sends them to the web server.

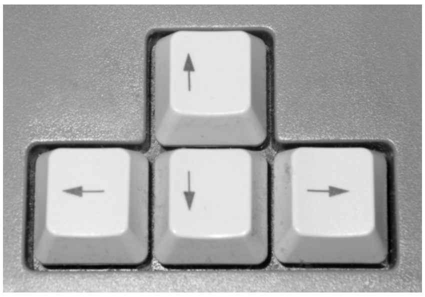

Figure 3.4: Keyboard Interaction

3.2.5 Speech

Speech is one of the most natural methods of interaction. The idea of using speech as an interaction technique is not new. Command and control based speech recognition enables a developer to provide a limited set of possible word combinations and the speech recognition engine matches the words spoken by the user to this limited list. The command and control method of speech recognition results in higher accuracy of recognition, thus leading to better results [33].

3.2.6 Interaction

To interact with the robot, the user would use simple commands such as “forward” to go forward, “back” to go backward, “forward-left” to turn left while moving forward, “forward-right” to turn right while moving forward, “back-left” to turn left while going backward, and “back-right” to turn right while going backward.

Chapter 4

System Implementation

4.1 Introduction

This chapter describes how the system was implemented. Discussions of the various programming languages, concepts and techniques used are presented. The implementation of the system follows the design described in the previous chapter.

4.2 Server

The server application was developed in C# using Microsoft Visual Studio 2008 and the .NET Framework 3.5. Bluetooth communication and translation is managed by an external, open source library, namely the NKH.Mindsqualls which provides a unified and simple API to communicate with and control the Lego NXT robot. It supports all the standard LEGO sensors as well as a few of the HiTechnic sensors such as the colour, compass and tilt sensors.

The first challenge was to establish a connection between the server and the robot. The robot is first paired with Windows so that it can be seen as a device, and is assigned a COM port. The user types this COM port number into the server GUI and clicks on the “connect” button.

Code listing 4.1 shows the code that is executed when the user clicks the “connect” button.

Listing 4.1: Connecting and Initialising the Robot

```
private void button1_Click(object sender, EventArgs e)
        {
          try
          {
                byte comPort = byte.Parse(this.txtComPort.Text)
                   ;
                brick = new NxtBrick(comPort);
                brick.MotorB = new NxtMotor();{
                brick.MotorC = new NxtMotor();
                motorPair = new NxtMotorSync(brick.MotorB,
                   brick.MotorC);
                brick.Connect();
                motorPair.ResetMotorPosition(false);
                txtLog.AppendText("Connected: " + brick.Name +
                   "\n");
                myServ = new Server(this);
            }
            catch
            {
                Disconnect();
            }
        }
```

Listing 4.2: The “move” method

```
public void move(sbyte power, sbyte turn)
        {
            if (motorPair != null)
            {
                if (!((power == lastSpeed) && (turn == lastTurn
                   )))
                {
                    motorPair.ResetMotorPosition(true);
                    motorPair.Run(power, 0, turn);
                    lastSpeed = power;
                    lastTurn = turn;
                }

            }
            else
                MessageBox.Show("Please connect the NXT!");
        }
```

A new instance of NxtBrick is created, the motors are initialized, a motor pair is created and the program establishes a connection with the NXT brick. Once the brick is connected, the TCP server is initialized and starts listening for connections.

The purpose of the “motorPair” object is to synchronise the two wheels of the robot, so that when a turn ratio is specified, the library knows how much power to assign to each motor so that the robot operates correctly.

Code listing 4.2 shows the “move” algorithm for the robot.

The “move” method accepts the power and turn-ratio as input parameters. The motorPair is reset when the move method is called. This is done so that the new turn ratio is not relative to the previous turn ratio. The method has a check in place such that if the new power and turn ratio is equal to the previously sent power and turn ratio, the program will not resend the commands to the NXT robot. The move method is called when the server receives a message from one of the clients.

4.2.1 The Protocol

Messages are sent to and accepted by the server in the following format:

<modality>;<power>;<turnRatio> where power is an integer value between -100 and 100, and turnRatio is an integer value between -10 and 10. Listing 4.3 shows an example of a received message. In this case the modality is an Xbox 360 controller, the power value is 60 and the turn ratio is 7. The server displays the currently connected modalities.

Listing 4.3: Protocol message example

```
xbox360;60;7
```

4.2.2 TCP Communication

The server creates a listening socket on port 3000 to handle incoming connections from clients. The server starts listening for connections once a connection to the Lego NXT robot has been established. The server runs on a separate thread to the main application to prevent blocking the main thread. This also allows the server to handle multiple incoming connections. A new thread is spawned on each incoming connection.

4.3 Game Controllers

4.3.1 Nintendo Wiimote

The Nintendo Wiimote client application was developed in C# using Microsoft Visual Studio 2008 and the .NET Framework 3.5. An external library, WiimoteLib was used to handle the Bluetooth communication between the Wiimote and the application. This is an excellent API and provides an enumeration of the Wiimote.

The acceleration state values are returned as values between -1 and 1. To make this compatible with the server protocol, these values were multiplied by 100. As the accelerometer is extremely sensitive, some desensitising of the input was done by specifying exact values when the values received from the Wiimote were in a particular range. Listing 4.4 shows this implementation.

4.3.2 Microsoft Xbox 360 Controller

The Microsoft Xbox 360 Controller client application was developed in C# using Microsoft Visual Studio 2008, the .NET Framework 3.5 and the Microsoft XNA Framework. The Microsoft

Listing 4.4: Wiimote Accelerometer Desensitising

```
        ws = args.WiimoteState;
    Double power = (ws.AccelState.Values.Y) * 100;
    Double turn = (ws.AccelState.Values.X) * 100;

    if ((power >= 50) && (power < 80))
        power = 65;
    else {
        if (power >= 80)
            power = 90;
        else {
            if ((power < 50) && (power > 20))
                power = 35;
            else {
                if ((power <= 20) && (power > 0))
                    power = 10;
            }
        }
    }
    if ((power > -20) && (power < 0))
        power = -10;
    else {
        if ((power > -50) && (power <= -20))
            power = -35;
        else {
            if ((power > -80) && (power <= -50))
                power = -65;
            else {
                if (power <= -80)
                    power = -90;
            }
        }
    }
    if (turn > 20)
        turn = 20;
    else {
        if ((turn <= 20) && (turn > 0))
            turn = 0;
        else {
            if ((turn < 0) && (turn >= -20))
                turn = 0;
            else {
                if (turn < -20)
                    turn = -20;
            }
        }
    }
    if (power < 0)
        turn = -turn;
```

Listing 4.5: Getting data from the Xbox 360 controller

```
protected override void Update(GameTime gameTime)
{
    // Allows the application to exit
    if (GamePad.GetState(PlayerIndex.One).Buttons.Back
       == ButtonState.Pressed)
        this.Exit();

    GamePadState gps =
                GamePad.GetState(PlayerIndex.One);

    float speed = gps.ThumbSticks.Left.Y;
    float x = gps.ThumbSticks.Right.X;

    String data = "xbox360;" + (Math.Round(speed * 100)
       ) + ";" + (Math.Round(x * 10));
    theCon.sendData(data); // send the data to the
       server

                base.Update(gameTime);
}
```

XNA Framework is aimed at game development for the Windows and Xbox 360 platforms. This framework simplifies integrating game controllers into a C# application.

Obtaining the x and y coordinates of the analogue thumbsticks is trivial. The left thumbstick controls the speed while the right thumbstick controls the turn ratio. The speed value is multiplied by 100 and the turn ratio value is multiplied by 10 to match the server's protocol. Listing 4.5 shows how data is received from the Xbox 360 controller. The "update" method is called at specific intervals, determined by the GameTime which is specified in milliseconds.

4.4 Mobile

The mobile application was developed in Java using the Eclipse IDE and the Java 2 Micro Edition Framework (J2ME).

Listing 4.6: Mobile keyPressed handler

```
protected void keyPressed(int keyCode) {
    String dataSend;
    switch(getGameAction(keyCode)){
        case UP: dataSend = "mobile;80;0"; break;
        case DOWN : dataSend = "mobile;-80;0"; break;
        case LEFT : dataSend = "mobile;50;-8"; break;
        case RIGHT : dataSend = "mobile;50;8"; break;
        default: dataSend = "mobile;0;0";
        }
    theServer.sendCommand(dataSend);
}
```

4.4.1 J2ME

J2ME is divided into three distinct parts: configurations, profiles and optional APIs. A configuration is designed for a specific kind of device based on its processing capabilities and available memory. The configuration specifies a strict subset of the Java 2 Platform and a Java Virtual Machine (JVM). There are two types of configurations: the Connected Device Configuration (CDC) and the Connected, Limitied Device Configuration (CDLC) [25]. The only one of interest to this project is the CDLC as it encompasses mobile phones.

A profile is based on the configuration and provides additional APIs such as storage and the user interface. The profile that will be used is the Mobile Information Device Profile (MIDP). Optional APIs define any additional functionality [25].

4.4.2 Implementation

When a user launches the application, they are asked to enter a server IP to connect to. This IP should be the address of the machine running the NXT server. Upon successful connection, the user is informed to use the keypad to control the robot. A key listener was implemented and listing 4.6 shows this implementation.

4.5 Web

The Web application was developed in Java using the Eclipse IDE and Google Web Toolkit (GWT). To interact with the server, a client-server architecture is needed. The GWT is a good choice for this implementation as it enables the rapid development of AJAX applications.

4.5.1 The Google Web Toolkit

The Google Web Toolkit (GWT) is a Java development framework for Web programmers, first released in May 2006. It helps Web developers develop highly efficient and interactive AJAX applications. The GWT enables the Web developer to code the application entirely in Java, which is then cross-compiled into optimized JavaScript code that runs in the client's web browser. The GWT also takes care of converting widgets such as text boxes, check boxes etc into basic HTML controls [20].

Communication with the server takes place using GWT Remote Procedure Calls (RPC). To make RPCs, you create an interface that specifies the remote methods you wish to call. When the browser calls the method, the GWT RPC serializes the arguments and invokes the proper method on the server [14].

On the client side we have the interface for the remote procedure call. The file NxtWebService.java is given as listing 4.7. This file specifies that there are two available methods, namely the "connect" and "doAction" methods. The RPC also requires an asynchronous method. This is illustrated by the file NxtWebServiceAsync.java, given in code listing 4.8.

The actual implementation of the web service takes place on the server side. The service creates the connection to the NXT Server application and sends commands to the application. The implemented web service is presented as listing 4.9.

The final web interface contains a text box where the user enters the NXT server IP and a "connect" button. When the user enters a valid IP and a connection is established, two panels appear: one for the keyboard listener and the other showing the output of the wireless camera attached to the robot. This interface is illustrated in Fig. 4.1.

Listing 4.7: NxtWebService.java

```
package nxtWeb.client;
import com.google.gwt.user.client.rpc.RemoteService;
import com.google.gwt.user.client.rpc.RemoteServiceRelativePath;

@RemoteServiceRelativePath("greet")
public interface NxtWebService extends RemoteService {
        String connect(String addr);
        String doAction(String action);
}
```

Listing 4.8: NxtWebServiceAsync.java

```
package nxtWeb.client;
import com.google.gwt.user.client.rpc.AsyncCallback;

public interface NxtWebServiceAsync {
        void connect(String addr, AsyncCallback<String>
           callback);
        void doAction(String action, AsyncCallback<String>
           callback);
}
```

Listing 4.9: NxtWebServiceImpl.java

```
package nxtWeb.server;
import nxtWeb.client.NxtWebService;
import com.google.gwt.user.server.rpc.RemoteServiceServlet;

@SuppressWarnings("serial")
public class NxtWebServiceImpl extends RemoteServiceServlet
   implements NxtWebService {

        NxtCom com;

        public String connect(String addr)
        {
                if (com == null)
                {
                   try{
                     com = new NxtCom(addr);
                     com.connect();
                   }
                   catch(Exception e)
                   {
                     return e.toString();
                   }
                }
                return "connected";
        }

        public String doAction(String action)
        {
           if(com != null)
           {
             try
             {
               com.doAction(action);
               return "success";
             }
             catch(Exception e)
             {
               return "failed";
             }
           }
           return "failed";
        }
}
```

Figure 4.1: The web interface

4.6 Speech

Due to time constraints, the speech interface has not yet been implemented.

4.7 Summary

The implementation of the system was discussed. The programming languages, development environments and tools were described. These were chosen for their suitability to the rapid application development approach used.

Chapter 5

Evaluation and Results

5.1 Introduction

In the previous chapter, the implementation of the system was discussed. This chapter details the evaluation procedure of the multimodal human robot interaction system. The results from a user study are analysed and discussed.

5.2 User Evaluation

For testing purposes, users were given a chance to use each of the modalities to control the NXT robot. A course was laid out so that timings could be made for each run. Users were asked about their experiences with each modality and to rate each modality out of 5, with 5 being highly user friendly and efficient, and 0 being unusable. This study had 6 participants.

Each user started with a different modality and the order in which they used them was random so as to remove any learning bias that may influence the study.

5.2.1 Speed and efficiency

The speed test involved the participants having to navigate the robot around a laid out course with a series of obstacles. The time it took for a participant to navigate the robot through the course was recorded for each modality.

The course was laid out in a figure of eight, as illustrated in Fig. 5.1. The participants were asked to navigate around these obstacles in the path shown.

The average times to complete the course using each modality are listed in Table 5.1.

Table 5.1: Average time of modalities

Modality	1	2	3	4	5	6	Average Time (seconds)
Web	52.2	46.1	48.5	51.35	50.3	47.3	49.3s
Xbox controller	34.6	42.2	37.3	33.3	35.7	36.2	36.5s
Wiimote	45.1	45.8	42.3	42.9	44.1	42.2	43.7s
Mobile	57.3	55.3	51.1	52.7	49.8	52.7	53.2s

As can be seen from the table, the tangible user interfaces performed significantly better than the remote access methods. Although all the interfaces require the participants to think about the abstract mappings, the TUIs had an advantage. The users need not focus on their hands, the interaction is natural. These findings are consistent with findings presented by Guo and Sharlin [16].

Using the remote methods of navigation, the participants focus more on how to actually control the robot rather than focusing on the task at hand, therefore completing the task would take more time.

5.2.2 Usability

Usability can be defined as the ease of use of a specific modality. Usability is composed of learnability, efficiency and satisfaction .The participants were asked to rate the usability of each of the modalities. They were given time to play around with each modality and rate their experiences. The average rating of each modality is listed in Table 5.2.

Table 5.2: Average rating of modalities

Modality	1	2	3	4	5	6	Rating (out of 5)
Web	4	5	4	4	4	4	4.2
Xbox controller	5	4	5	5	5	5	4.7
Wiimote	4	4	5	5	4	5	4.5
Mobile	3	3	4	4	5	4	3.8

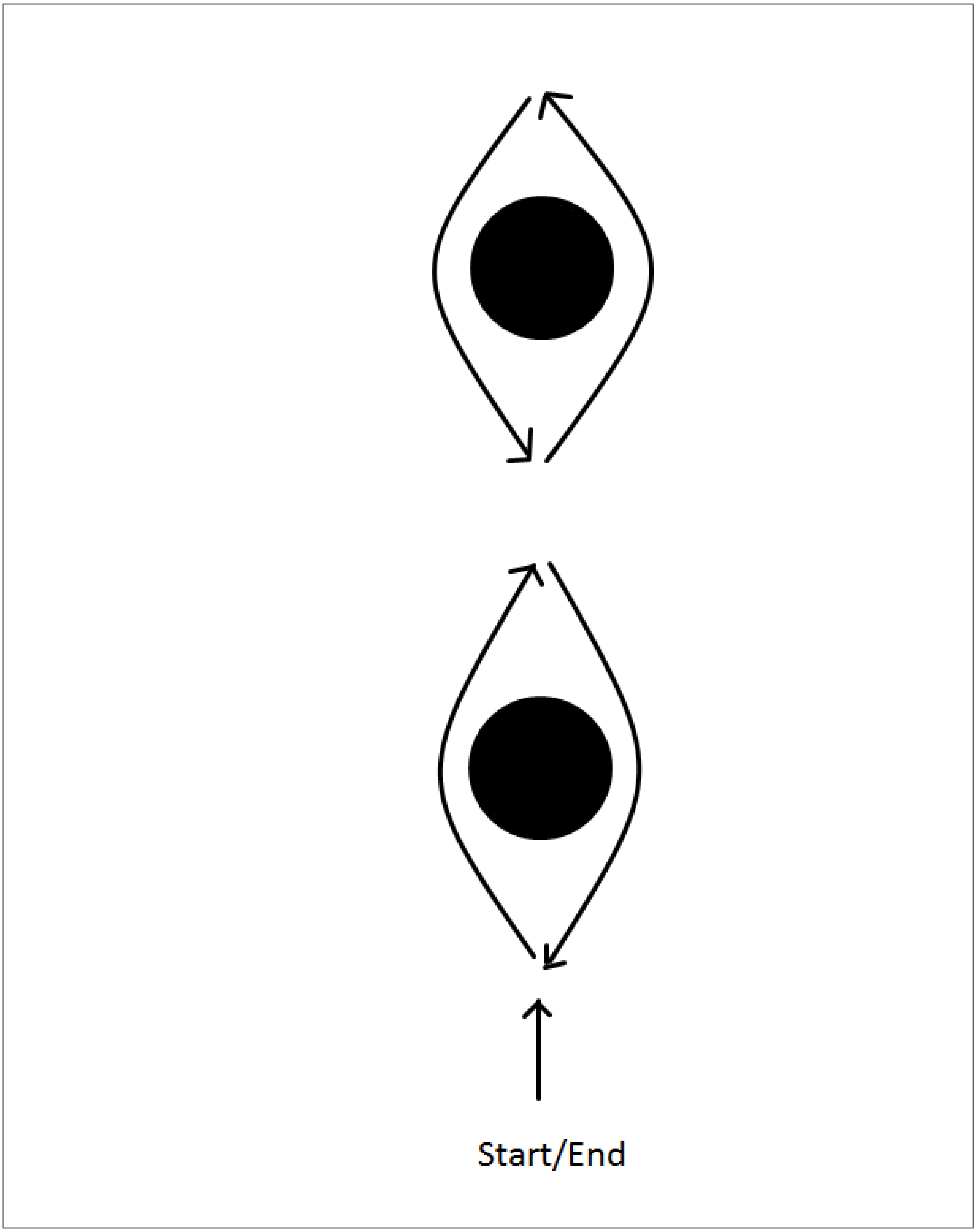

Figure 5.1: Speed and efficiency course

By interpreting the above ratings, we come to the same conclusion as in the speed and efficiency tests. The tangible user interfaces fare better than the traditional remote methods of interaction.

The remote interfaces still have their uses. By harnessing the power of the Internet, the robot could potentially be controlled from anywhere in the world with an Internet connection, either using a web browser or making use of the mobile implementation.

5.2.3 Other Tests

Another test that was done was to combine remote interaction with a tangible user interface. This was done by making use of the web interface such that the robot's webcam could be accessed, but making use of the Microsoft Xbox 360 controller to navigate the robot. The combination of these two interfaces proved to be very effective because you have the advantage of being able to control the robot remotely with the advantage of using a tangible user interface.

5.3 Summary

By making use of a user study, it was found that for speed and efficiency, controlling a robot with a tangible user interface (TUI) is more effective due to the more natural interaction with the robot. This allows the user to focus more on the task at hand rather than thinking about how to interact with the robot.

The usability study also found that tangible user interfaces are more effective. They allow for more efficient human robot interaction.

Chapter 6

Conclusion and Future Work

6.1 Summary

The goal of the project was to design and implement a functional multimodal interface for human-robot interaction. This process involved researching current developments in this area, and working out the challenges faced when creating such a system.

The first component of the system that had to be designed and implemented was the server. This is the core of the system and involves receiving input from multiple modalities, interpreting the data, and sending commands to the NXT robot over a Bluetooth connection.

The proposed modalities were discussed and the design and implementation of each of these modalities was presented. A user study was conducted and it was found that the tangible user interfaces such as the Microsft Xbox 360 controller and Nintendo Wii controller were the most effective means of robot control.

6.2 Future Extensions

Future extensions on this project include adding more interaction modalities, focusing on more natural interaction techniques including speech, vision and gesture. It would also be useful to integrate a degree of autonomy, especially when using remote methods of interaction. The reason for this is possible delays involved over long distance network communications.

It would be useful for the robot to be spatially aware of its surroundings and be tolerant of objects around it. Advanced sensor feedback from the robot to the user would also improve the collaboration between humans and robots.

References

[1] AYRES, T., AND NOLAN, B. Voice activated command and control with Java-enabled speech recognition over Wifi. In *PPPJ '04: Proceedings of the 3rd International Symposium on Principles and Practice of Programming in Java* (2004), Trinity College Dublin, pp. 114–119.

[2] BALLSUN-STANTON, B., AND SCHULL, J. Flying a Manta with Gesture and Controller: An Exploration of Certain Interfaces in Human-Robot Interaction. [Online; Available from: http://rit.academia.edu/documents/0010/6529/Ballsun-StantonSchullGW2007.pdf ,accessed 14 May 2009].

[3] CHOO, S. H., AMIN, S. H. M., FISAL, N., YEONG, C. F., AND BAKAR, J. A. Bluetooth Transceivers for Full Duplex Communications in Mobile Robots. *Jurnal Teknologi 37* (2002), 19–20.

[4] COSTO, S., CEPOLINA, F., ZOPPI, M., AND MOLFINO, R. AirEOD: a Robot for Onboard Airplanes Security. In *Proceedings of the 8th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2005)* (2005), pp. 995–1002.

[5] DAUTENHAHN, K. Roles and functions of robots in human society: implications from research in autism therapy. *Robotica 21*, 4 (2003), 443–452.

[6] DENKET, M. A. *Frontiers in Robotics Research.* Nova Publishers, 2006.

[7] DIDELES, M. Bluetooth: a technical overview. *Crossroads 9*, 4 (2003), 11–18.

[8] FERNANDEZ, D., AND PEEK, B. *Coding4Fun - 10 .NET Programming Projects for Wiimote, YouTube, World of Warcraft, and More.* O'Reilly, 2008.

[9] FONG, T., AND NOURBAKHSH, I. Interaction challenges in human-robot space exploration. *Interactions 12*, 2 (2005), 42–45.

[10] FOX. U.S. Army Test Flying Robot Sniper. News Report, April 2009. [Online; Available from: http://www.foxnews.com/story/0,2933,517481,00.html ,accessed 10 June 2009].

[11] FRANCIA, G. A., KILARU, A., PHUONG, L., AND VASHI, M. An empirical study of Bluetooth performance. In *MSCCC '04: Proceedings of the 2nd annual conference on Mid-south college computing* (Little Rock, Arkansas, United States, 2004), Mid-South College Computing Conference, pp. 81–93.

[12] FUJITSU. Fujitsu Develops Mobile Phone-Controlled Robot for the Home. Press Release, October 2002. [Online; Available from: http://pr.fujitsu.com/en/news/2002/10/7.html ,accessed 10 June 2009].

[13] GOLDBERG, K., GENTNER, S., SUTTER, C., WIEGLEY, J., AND FARZIN, B. The Mercury Project: a Feasibility study for online robots. *Beyond Webcams: An Introduction to Online Robots* (2002), 17–36.

[14] GOOGLE. Product Overview - Google Web Toolkit, 2009. [Online; Available at: http://code.google.com/webtoolkit/overview.html ,accessed 2-November-2009].

[15] GREENWALD, L. KOPENA, J. Mobile robot labs. In *Robotics & Automation Magazine, IEEE* (June 2003), vol. 10, pp. 25–32.

[16] GUO, C., AND SHARLIN, E. Exploring the use of Tangible User Interfaces for Human-Robot Interaction: a comparative study. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), ACM, pp. 121–130.

[17] HAZRY, D., SUGISAKA, M., AND YUJI, T. Human-Robot Interface over the Web Based Intelligent System. *American Journal of Applied Sciences 3* (2006), 1634–1639.

[18] IBA, S. *Interactive Multi-Modal Robot Programming*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2004.

[19] IOSSIFIDIS, I., AND STEINHAGE, A. Dynamical systems: A framework for man machine interaction. In *Proc. of the International Conference on Automation and Information: Theory and Applications (AITA 2001)* (2001).

[20] KEREKI, F. Web 2.0 development with the Google Web Toolkit. *Linux J. 2009*, 178 (2009), 2.

[21] KUNO, Y., SADAZUKA, K., KAWASHIMA, M., YAMAZAKI, K., YAMAZAKI, A., AND KUZUOKA, H. Museum guide robot based on sociological interaction analysis. In *CHI*

'07: Proceedings of the SIGCHI conference on Human factors in computing systems (New York, NY, USA, 2007), ACM, pp. 1191–1194.

[22] LAY, K. PRASSLER, E., DILLMANN, R. GRUNWALD, G., HÄGELE, M. LAWITZKY, G., STOPP, A., AND VON SEELEN, W. MORPHA: Communication and Interaction with Intelligent, Anthropomorphic Robot Assistants. *Tagungsband Statustage Leitprojekte Mensch-Technik-Interaktion in der Wissensgesellschaft* (2001).

[23] LEGO®. What's NXT? LEGO Group Unveils LEGO® MINDSTORMS NXT Robotics Toolset at Consumer Electronics Show. Press Release, January 2006. [Online; Accessible at: http://www.lego.com/eng/info/default.asp?page=pressdetail& contentid=17278&countrycode=2057&yearcode=&archive=false ,accessed 5 June 2009].

[24] LEWIS, D. What is Web 2.0? *Crossroads 13*, 1 (2006), 3–3.

[25] LI, S., AND KNUDSEN, J. *Beginning J2ME: from novice to professional*. Apress, 2005.

[26] MCNEILL, M. D. J., AND HUTTON, A. Interactive control of robots on the Internet. *Robotics and Autonomous Systems* (2001), 127–139.

[27] MICROSOFT. Description of the Wired and Wireless Xbox 360 Controllers. Microsoft Help and Support Knowledge Base, September 2007.

[28] ORON-GILAD, T. *Interfaces for ground and air military robots: workshop summary*. National Academies Press, 2005.

[29] OSBORN, K. Army Robots: Will Humans Still Be in Control? *TIME Magazine March* (2009).

[30] OVIATT, S. Taming recognition errors with a multimodal interface. *Commun. ACM 43*, 9 (2000), 45–51.

[31] PERZANOWSKI, D. SCHULTZ, A., ADAMS, W. MARSH, E., AND BUGAJSKA, M. Building a multimodal human-robot interface. *Intelligent Systems, IEEE 16 Issue 1* (2001), 16–21.

[32] PRASSLER, E., LAWITZKY, G., STOPP, A., GRUNWALD, G., HÄGELE, M., DILLMANN, R., AND IOSSIFIDIS, I. *Advances in Human Robot Interaction*, vol. 14/2004 of *Springer Tracts in Advanced Robotics STAR*. Springer Press, 2004.

[33] SRINIVAS, S. Speech Recognition using Csharp. [Online; Available from: http://www.c-sharpcorner.com/UploadFile/ssrinivas/SpeeechRecognitionusingCSharp

11222005054918AM/SpeeechRecognitionusingCSharp.aspx ,accessed 18-October-2009].

[34] WIKIPEDIA. Lego Mindstorms NXT — Wikipedia, The Free Encyclopedia. [Online; Available from: http://en.wikipedia.org/wiki/Lego_Mindstorms_NXT ,accessed 10-June-2009].

Appendix A

Project Poster