FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Multimodal Interface for an Intelligent Wheelchair

Sérgio Miguel Fontes de Vasconcelos

Report of Project/Dissertation
Master in Informatics and Computing Engineering

Supervisor: Luís Paulo Reis (PhD)

Co-Supervisors: Brígida Mónica Faria (MsC), Rodrigo Braga (PhD)

17th January, 2011

Multimodal Interface for an Intelligent Wheelchair

Sérgio Miguel Fontes de Vasconcelos

Report of Project/Dissertation
Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: António Augusto de Sousa (PhD)
External Examiner: Pedro Miguel Moreira (PhD)
Supervisor: Luís Paulo Reis (PhD)

11th February, 2011

Abstract

Since the demographics of population, with respect to age, are continuously changing, politicians and scientists start to pay more attention to the needs of senior individuals. Additionally, the well-being and needs of disabled individuals are also becoming highly valued in the political and entrepreneurial society.

Recent advances in computer science, artificial intelligence, multimedia and robotics allow us to extend and evolve the areas and applications which are used to aid in these patients daily lives.

Multimodal systems process two or more combined user input modes, such as voice, gaze, gestures, as well as head and body movements, in a coordinated fashion with a multimedia system output. This class of systems represents a variation of the traditional interfaces paradigm.

This research work consisted in the project and development of a multimodal interface based on the integration of several input devices, using different recognition methodologies. The developed system was integrated in the previously designed IntellWheels platform used to control the *NIAD&R/LIACC* intelligent wheelchairs.

The main goal was to combine all the possible inputs in a truly multimodal interface that is able to accept and recognize high-level commands defined through user defined input sequences.

Our approach consisted of using three basic input modalities, namely speech recognition, detection of head movements, and a generic gamepad.

To validate the effectiveness of the developed prototype, two different experiments were made. A number of volunteers tested the system firstly by driving a simulated intelligent wheelchair along a defined route. In the second experiment, other group of individuals tested the system using a real intelligent wheelchair. All the participants were able to successfully complete the proposed routes using the different input modalities.

The developed work allowed us to conclude that multimodal systems are ideal to be used by people with physical disability, due to their interaction flexibility. Nevertheless, some further improvements to the developed prototype should be made in order to obtain a more complete level of multimodality. The next step will consist of having the whole system tested by individuals with severe physical disabilities, such as cerebral palsy and quadriplegia.

Resumo

Uma vez que a demografia da população, no que toca à idade, está em constante mudança, políticos e cientistas começam a prestar mais atenção às necessidades de indivíduos idosos. Adicionalmente, o bem-estar e as necessidades das pessoas portadoras de deficiência física são cada vez mais valorizadadas pela sociedade política e empresarial.

Avanços recentes em ciências da computação, inteligência artificial, multimédia e robótica, permitem-nos ampliar e desenvolver as áreas e aplicações que são usadas para ajudar estes pacientes no seu quotidiano.

Sistemas multimodais caracterizam-se por processar duas ou mais modalidades entrada, tais como voz, visão, gestos, assim como movimentos da cabeça e do corpo, de uma forma coordenada com uma saída de sistema multimédia. Esta classe de sistemas representa uma variação do paradigma tradicional de interfaces.

O trabalho proposto nesta dissertação consistiu em projectar e desenvolver uma interface multimodal, baseada na integração de diversos dispositivos de entrada, utilizando diferentes metodologias de reconhecimento. O sistema desenvolvido foi integrado na plataforma IntellWheels, utilizada para controlar os protótipos de cadeiras de rodas inteligentes previamente projectadas pelo *NIAD&R/LIACC*.

O principal objectivo foi integrar variadas formas de interacção numa interface multimodal capaz de aceitar e reconhecer comandos de alto nível definidos através de sequências de entrada definidos pelo utilizador.

A abordagem adoptada consistiu em utilizar três modalidades básicas de entrada, nomeadamente reconhecimento de voz, detecção de movimentos da cabeça, e um joystick tradicional.

Para validar a eficácia do protótipo desenvolvido, foram efectuadas duas experiências. Inicialmente, um conjunto de voluntários testou o sistema conduzindo uma cadeira de rodas simulada, ao longo de um trajecto previamente definido. Na segunda experiência, outro grupo de pessoas testou o sistema usando uma cadeira de rodas inteligente real. Todos os participantes foram capazes de completar com sucesso as rotas propostas utilizando as diferentes modalidades de entrada.

O trabalho desenvolvido permitiu-nos concluir que os sistemas multimodais são ideais para serem utilizados por pessoas com deficiência física, devido à sua flexibilidade de interacção. No entanto, algumas melhorias ao protótipo desenvolvido ainda necessitam ser efectuadas de forma a atingir uma maior robustez do sistema. O próximo passo consistirá em ter todo o sistema testado por pessoas com deficiências físicas graves, nomeadamente paralisia cerebral e tetraplegia.

Acknowledgements

I want to sincerely thank the people who helped to make this an interesting project and this semester at FEUP a memorable one. First of all, I would like to express my gratitude to my thesis supervisor Luís Paulo Reis for providing excellent guidance, thoughts and work reviews throughout the development of this dissertation.

Secondly, I would like to thank Rodrigo Braga and Marcelo Petry for their patience while explaining me all the details concerning the IntellWheels platform. I would also like to thank Mónica Faria for the help given at the preparation of both experiments, and the precious review of the results analysis.

To all the volunteers who participated on both experiments, thank you for your time and comments. All your feedback was of great value to improve my work.

A special word of gratitude goes to my family and girlfriend, for all the encouragement in moments of greater fatigue.

To all the usual suspects, thank you for all the cheerful tea-break discussions, friendship and support through the long days and late nights. It was my pleasure to share this fantastic journey with you.

This work was partially supported by the Artificial Intelligence and Computer Science Lab. of the University of Porto and by the Fundação para a Ciência e a Tecnologia through the project "INTELLWHEELS - Intelligent Wheelchair with Flexible Multimodal Interface" (grant ref. FCT/RIPD/ADA/109636/2009).

Sérgio Vasconcelos

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

API	Application Programming Interface
ECU	Environmental Control Unit
CFG	Context-Free Grammar
IW	Intelligent Wheelchair
IWP	IntellWheels Platform
IMI	IntellWheels Multimodal Interface
HMI	Human Machine Interaction
LIACC	Artificial Intelligence and Computer Science Laboratory
GUI	Graphical User Interface
SAPI	Speech Application Programming Interface
TTS	Text To Speech
XML	eXtensible Markup Language
WIMP	Window, Icon, Menu, Pointing device

Chapter 1

Introduction

1.1 Context

The last decades have provided a large set of improvements within all sectors of our society. As the quality of life increases, also the average of population life expectancy tends to augment. According to data provided by the United Nations and the World Health Organization, a portion of 650 million people (10% of world population) are afflicted with some sort of handicap, and 20% of them have physical disabilities. Moreover, these numbers have been continuously increasing since the world population is growing and ageing. Other factors include environment degradation, sub nutrition, and the appearance of chronic health issues. The main reasons for physical disability range from traffic accidents, war motives and land mines, to falls. Physical injuries could also be caused by medical conditions, like cerebral palsy, multiple sclerosis, respiratory and circulatory diseases, genetic diseases or chemical and drugs exposition [Day08].

Usually, the physical deficiency may lead to a lack of control of some muscles of the arms, legs or face. It is quite difficult to generalize physical deficiencies since each person has different symptoms and uses different strategies to deal with them. A good example is cerebral palsy, characterized by injuries in the connection of some brain areas, resulting in a group of permanent disorders in the development of movement and posture, causing activity limitation, such as incapacity of moving the arms, legs, or even talking. There is no cure for cerebral palsy, but the symptoms might change with age.

Due to its high proportion, the fraction of population with physical disabilities has earned more relevance and has attracted the attention of international health care organizations, universities and companies interested in developing and adapting new products. The actual tendency reflects the demand of an increase on health and rehabilitation services, in a way that senior and handicapped individuals, who have been at the margin of society for a long period, might become more and more independent performing quotidian tasks. Consequently, regardless the age, mobility is a fundamental characteristic for every human being. Children with disabilities are very often deprived of important opportunities and face serious disadvantages compared to other children. Adults who lose their independent means of locomotion become less self sufficient, raising a negative attitude towards themselves. The loss of mobility originates obstacles that reduce the personal and vocational objectives [Sim05]. Therefore is necessary to develop technologies that can aid this population group, in a way to assure the comfort and independence of the elderly and handicapped people. Wheelchairs are important locomotion devices for those individuals. Considering the previously explained motives, there is a growing demand for safer and more comfortable wheelchairs, and therefore, a new Intelligent Wheelchair (IW) concept was introduced. Like many other robotic systems, the main capabilities of an intelligent wheelchair should be: Autonomous navigation with safety, flexibility and capability of avoiding obstacles; intelligent interface with the user; communication with other devices (like automatic doors and other wheelchairs). However, most of the Intelligent Wheelchairs developed by distinct research laboratories [Sim05], have hardware and software architectures too specific for the wheelchair model used/project developed and are typically very difficult to configure in order for the user to start using them.

1.2 Motivation

The aim of this project is to contribute to the research and implementation of technologies that can improve the quality of life of disabled and senior individuals that depend on locomotion devices to attain mobility. In present day society, the available means on the market have proven to assist a considerable portion of these individuals in a very satisfactory way. Nevertheless, the market strategy tendency of entrepreneurs and companies is to develop general solutions that focus on the common issues concerning physical disability. Thus, many individuals with more severe physical disabilities see themselves incapable of using those solutions, due to additional constraints that limit their ability to control the commercial locomotion devices. More precisely, the traditional electric wheelchairs can only be manoeuvred using a manual joystick. Any person that does not have enough hand dexterity to control a joystick is unable to operate this kind of wheelchair, and is usually left without a viable alternative. This happens because specific designed devices are very expensive due to the complex technologies used, and the narrowness of its market.

The goal of this research proposal is to adapt the existing solutions in a way that the minorities consisting of severe handicapped patients become highly independent in everyday locomotion activities. The strategy to follow consists in making use of recent technology improvements in computer science and multimedia, recurring to popular and low-cost devices.

Moreover, it is intended to make possible to drive a wheelchair in an efficient and comfortable fashion, by developing an user friendly multimodal interface that allows the patient to control the wheelchair using configurable high level commands that can be input through facial expressions, voice control, head movements and so on.

1.3 Objectives

The main objectives of this work are related to the study and development of an intelligent wheelchair (IW), based on the adaptation of a commercial electric wheelchair with hardware flexibility. Sensorial and control system implementation, as well as navigational methods and multimodal interface design are the core research focus of this project. Another important objective is to minimize the aesthetic characteristics of the wheelchairs, in order to assure the well being of the patients while driving them. This research project is related to a multimodal interface (MMI) which will be integrated in the intelligent wheelchairs of the IntellWheels project under development at LIACC (Artificial Intelligence and Computer Science Laboratory) at the University of Porto [BPMR09]. Besides the design of the multimodal interface, further developments on new and existing input modules will also be studied. These modules may implement voice, video and other types of technology.

The multimodal interface must be able to:

- Provide user friendly characteristics.
- Allow the control of the IW through user-defined, configurable input sequences that should be associated with the desired output high-level commands.
- Be an integration of the developed input modules (speech, facial expressions, joystick, keyboard, head movements, etc.).
- Show environment characteristics to the user.

The research question to be answered is: "Is it possible do drive, in a safe and robust manner, an Intelligent Wheelchair using high-level commands processed by a multimodal interface?" In order to answer this research question, a complete multimodal interface for an IW will be projected and implemented. Also, a large number of experiments with distinct individuals, driving the IW in simulated and real environments, using different command options, will be performed.

1.4 Structure of the Document

This document is organized in nine different chapters. Chapter 1, the current chapter, describes the presented topics and the overall document structure. The second chapter gives an overview on different intelligent wheelchair projects over time, since pioneer to recent developed projects, and shows some techniques used in each one of them. Chapter 3 presents a brief introduction to human machine interaction, and describes some of the more relevant interaction modalities used in projects of this nature. These modalities focus on the specific context in which this project is contained, and some additional projects are also briefly described to serve as examples. The fourth chapter presents a specific type of human machine interface that can combine many different interaction modalities described in chapter 3. A theoretical explanation is given and basic design principles are explained. Following, some relevant projects related to multimodal interfaces are also mentioned. Chapter 5 gives a general overview of the IntellWheels project, describing its objectives, the achieved work, and the overall characteristics of the IntellWheels platform (IWP). In chapter 6, the specification of the proposed IntellWheels Multimodal Interface (IMI) is presented, explaining the main desired features and architectural details. In Chapter 7, we describe the most relevant details concerning the implementation of the IntellWheels Multimodal Interface. Chapter 8 contains the description of the different tests and experiments that were prepared in order to assess the quality of the developed work. Finally, Chapter 9, presents some conclusions about the results that were obtained by the developed work, and some leads for future work are also given.

Chapter 2

Intelligent Wheelchairs

2.1 Relevant Projects

A vast amount of projects related to intelligent wheelchairs has been published over time, many of them being still under development in the last years. Although the first publications regarding intelligent wheelchairs date back to 1982, it was only in the last decade that this research area started to grow and evolve. Figure 2.1 shows the existing number of projects publications related to intelligent wheelchairs, referenced in four search engines.

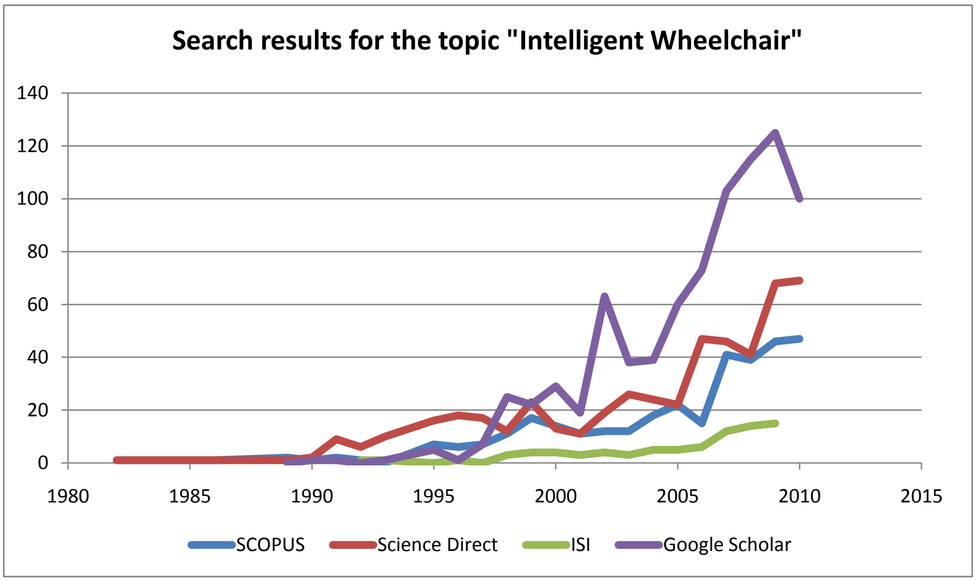


Figure 2.1: Search results for the topic "Intelligent Wheelchair"

One of the first prototypes was proposed by Madarasz in 1986 (Figure 2.2), which presented a wheelchair designed to transport a person to a desired room within an office building given only the destination room number [MHCM86]. The wheelchair has been equipped with an on-board computer, a digital camera, and a scanning ultrasonic range finder. Its main objective was to be operated without human intervention in populous environments without colliding with objects or people present in its surroundings.

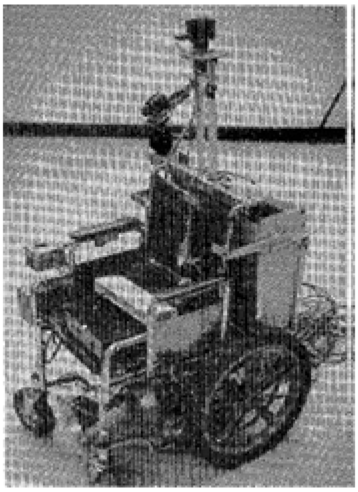


Figure 2.2: Protoype developed by Madarasz

Hoyer and Holper [BHB+98] presented the architecture of a modular control for an omni-directional wheelchair (Figure 2.3).

Figure 2.3: Omnidirectional wheelchair

NavChair is an intelligent wheelchair prototype started in 1991 (Figure 2.4), which main characteristics are obstacle avoidance, follow and pass through doors. A shared-control system was implemented in which the user plans the route, does some navigation and indicates direction and speed of travel. A voice control module was also adapted to this wheelchair [BBL+94] [SLB+98] [LBJ+99].

Miller and Slak [MS95, Mil98] presented Tin Man I in 1993, an intelligent wheelchair which can find its way through doorways, follow hallways, and do limited amounts of reckoning navigation.

Tin Man I (Figure 2.5) has three different operating modes. The first consists of having an human driver, providing an automatic shared control to avoid obstacles. Another

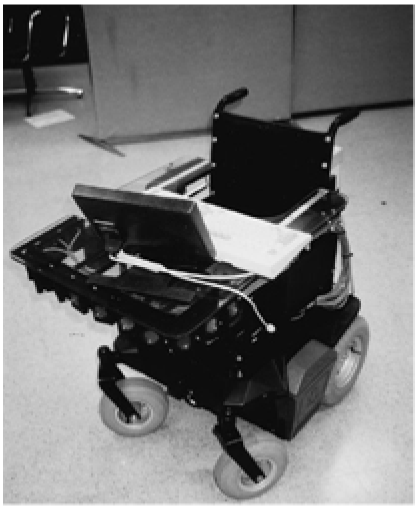


Figure 2.4: NavChair

operating mode consists of moving along a pre-defined trajectory. This wheelchair is also capable of moving to certain locations, (x, y), through the use of environment internal maps.

Figure 2.5: TinMan 1 wheelchair

Later, this projected evolved to Tin Man II, a cleaned up version of the original wheelchair which eliminated the mechanical joystick interface. Tin Man II has been shown to be effective in many indoor environments. New features were added to this prototype, since backtracking, backup, wall following, door way passing and docking.

Another method presented by researchers consists in controlling the wheelchair using an eye-gaze tracker to detect the relative position of the user's eyes to directly control the wheelchair. [Lak05] [RBG$^+$07].

One of those projects is Wheelesley (Figure 2.6), which was built by the KISS Institute for Practical Robotics. The base is a Vector Mobility powered wheelchair. In this project, interaction between the user and the wheelchair is investigated, as the user gives the high-level commands through a customized interface displayed in a Macintosh Powerbook.

The customizable interface allows different configurations for people with different access methods [Yan98].

Figure 2.6: Wheelesley wheelchair

There are three different operating modes: manual mode, joystick mode and interface mode.

In manual mode, the joystick commands are passed directly to the motor controller without sensor mediation. In joystick mode, the user's joystick commands are carried out using low-level control on the robot to avoid obstacles. In interface mode, the arrows are used to the direct the robot and consist of four directional arrows and a stop button.

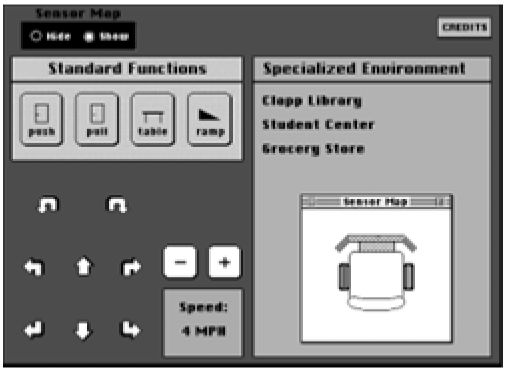


Figure 2.7: Wheesley wheelchair user interface

The original interface was customized for two access methods. The first is an eye tracking device called Eagle Eyes [GDCO96]. The second is a single switch scanning device. Another access method implemented in the Wheesley interface is named single switch scanning. The control panel scans through forward, left, right and backward, and the user clicks the single switch when the desired arrow is highlighted (Figure 2.7).

The semiautonomous robotic system FRIEND consists of an electric wheelchair with the robot arm MANUS [BBG99]. Both devices are controlled by speech recognition using a voice input module installed in a computer connected to the system (Figure 2.8).

Figure 2.8: FRIEND wheelchair project

Researchers at the Nara Institute of Science and Technology in Japan developed a real-time stereo face tracking and gaze measurement system to measure the head pose and gaze direction simultaneously. This system was then implemented in their WATSON smart wheelchair project (Figure 2.9) that can be manoeuvred by using gaze and head gestures, such as nodding and shaking [MIO01].

Figure 2.9: Overview of wheelchair WATSON

There are other projects that present solutions for individuals with quadriplegia, by using computer vision to detect and interpret facial expressions, such as NLPRWheelchair [Wei04] and OSAKA Wheelchair [NKSS99].

Recently, researchers at the Massachusetts Institute of Technology developed an intelligent wheelchair, named MIT Intelligent Wheelchair (Figure 2.10), equipped with an interactive interface for intelligent speech recognition. The user is able to spell full sentences that are understood by the system as high level commands. The interface interacts with the user, confirming the desired commands, by asking questions to assure the correctness of the resulting actions [RTR] [Roy09]. Furthermore, learning models were implemented in order to learn the topics and clusters of objects that co-occur together. An example of this interface behaviour is presented in Table 2.1.

	Non-learner		Learner
User:	Take me to the elevator.	**User:**	Take me to the elevator.
Robot:	Where did you want to go?	**Robot:**	Do you want to go to the Gates Tower?
User:	The The Gates elevator please	**User:**	Yes.
Robot:	Do you want to go to the Gates Tower?	**Robot:**	Going to Gates.
User:	Yes.		
Robot:	Going to Gates.		

Table 2.1: Multimodal user interaction with MIT Intelligent Wheelchair

Figure 2.10: MIT Intelligent Wheelchair

Another innovative project consists in a powered wheelchair that can be manoeuvred by individuals with high-level spinal cord injuries, using a recently developed device, the tongue mouse [Gho]. The device consists in a small magnet which it's attached to the user tongue with tissue adhesive. Movements of the magnetic tracer are detected by an array of magnetic field sensors mounted on wireless headphones worn by the subject. The sensor output signals are then wirelessly transmitted to a portable computer, which is carried on the wheelchair.

Figure 2.11: Tongue Drive System

2.2 Overview of Intelligent Wheelchair Projects

Table 2.2 presents an overview of the most relevant intelligent wheelchair projects, containing a brief description of the desired objectives and types of sensors used by each research group.

2.3 General requirements

Two major concerns regarding intelligent wheelchair projects are the user adaptability and the fulfilment of security requirements.

User adaptability

In order to be accepted by its user, an intelligent wheelchair must be able to adapt itself to each user characteristics. Specially in the context of supporting individuals with disabilities, the focus should be on how the user abilities can be complemented. As a consequence, researchers concentrate their efforts not on developing complete autonomous systems, but instead, semi-autonomous. These robots are able to execute certain tasks in an autonomous fashion, but on the other hand, some others require human experienced intervention. Therefore, an intelligent wheelchair is a highly interactive system, controlled by both human and software. This is why the design of a human machine interface is a key factor while developing an intelligent wheelchair.

Security requirements

The inaccurate behaviour of rehabilitation robots might cause severe damage to their users. For that reason, such robots must be considered as crucial security systems. For intelligent wheelchairs, this classification is even more acceptable, since in most of the cases the users depend on the total correctness of the system behaviour. Only a few research groups take into account how to project a safe smart wheelchair. One of them concerns the Rolland smart wheelchair [RL98], which uses formal methods techniques to perform risk analysis, and verification models to define the system security requirements, in order to proof that these requirements were accomplished and therefore process the shared control.

Functionality

The variety of necessary functionalities is as large as the amount of different existing disabilities. An intelligent wheelchair must behave in a safe and robust way in its natural environment. It is not acceptable that this environment should be completely reconstructed in a way that the only control mode is possible taking only into account the developer

Project	Sensors	Description
CPWNS [YBS02]	Vision, Odometry	Reproduces paths that are taught to the system through manual driving between two different positions
The Wheelchair [GBH+98]	Vision, Infrared, Sonar	Automatic navigation through the recognition of artificial markings on the floor
Intelligent Wheelchair System [Mur01]	Vision, Sonar, Facial expression recognition	The user is able to control the wheelchair through facial expression recognition. The wheelchair behavior adapts to the surrounding environment
INRO [SRLS98]	GPS, Sonar, Slope detection	Automatic navigation and wheelchair protection
LOUSON III [Zho00]	Vision, sonar, gyroscope, bar	User shared control and target following
OMNI [BHB+98]	Sonar, infrared, touch sensor, odometry	Has operational modes that solute specific tasks and obstacle avoidance
MAid [PSF01]	Sonar, infrared, laser scanner, odometry	Semi-autonomous behavior that allows for specific tasks solution, namely entering a bath-room. In automatic mode the wheelchair navigates until a destination determined by the user
RobChair [PN02]	Sonar, infrared, touch sensor	Obstacle avoidance
Rolland [RL98]	Vision, sonar, infrared, touch sensor, odometry	Evaluates and learns environment characteristics while navigates, in order to plan paths. Learns obstacle avoidance through training
SENARIO [KST+97]	Sonar, Odometry	User shared control and autonomous navigation
Sirus [oEEENY02]	Sonar, odometry	Obstacle avoidance and reproduction of pre-planned routes
TetraNauta [VAR+02]	Vision, sonar, infrared	Autonomous navigation by following artificial markings on the floor
VAHM [BHHP01]	Sonar, infrared, odometry	Autonomous navigation using internal maps and semi-autonomous navigation with obstacle avoidance and wall following
Wheelesley [Yan98]	Vision, sonar, infrared	Assisted navigation based in computer vision
A.W. [MHCM86]	Vision, laser scanner	Autonomous navigation using internal maps in dynamic environments
ACCoMo [HH05]	RFID Athena, infrared	Obstacle avoidance and autonomous navigation using environment internal maps. Collaborative behavior with the user and other wheelchairs
Tin Man II [Mil98]	Sonar, infrared, odometry, touch sensor, compass	Obstacle avoidance, line following and target reaching in (x, y) coordinates. Wall following, doorways passing, backup, docking, and return to initial point
SENA [MGG04]	Infrared, laser scanner, vision	Obstacle avoidance, voice control, visual control, autonomous navigation using internal maps

Table 2.2: Overview of Intelligent Wheelchair projects

intentions. Configuration and maintenance should be as intuitive as possible, since these tasks are performed by rehabilitation personnel and not robotic specialists. In the next section, a brief description of the existing technology used in intelligent wheelchairs will be given.

Sensors

The wheelchair displacement is always tracked through the processing of velocity and direction of its movement. This information is either provided by external encoders attached to the wheelchair's wheel axis, or by internal electronic devices. On the other hand, the use of proximity sensors might change significantly. Sonars are often used to perform this task. Usually, sonars are installed around the wheelchair (Senario, Rolland), and sometimes only on the front (NavChair, INRO). Infra-red sensors are also often used (RobChair, Wheelesley). Due to its high cost, laser scanners are rarely used (MAid).

Among passive sensors, touch or contact sensors are usually used. They provide a binary signal informing whether or not they are in contact with some object (Deictic, Wheelesley, TAO). Other passive sensors are video cameras, which can also be used to calculate distances between the wheelchair and surrounding obstacles (TAO, Deictic). Cameras are also used to detect holes or stairs (INRO, Senario).

Obstacle detection

The efficiency of obstacle avoidance depends on two major factors. The first is the quality of the used sensors, and the other concerns the techniques used to interpret, represent and process the data provided by the sensors.

Obstacle avoidance

To assure a safe locomotion, an intelligent wheelchair must be able to avoid obstacles in a robust way. However, among the amount of different existing projects, there are different comprehensions and interpretations regarding this matter.

Navigation

The basic requirement for navigation is a tracing technique that works correctly. Providing methods of automatic localization that can work correctly in different unknown environments is a challenge for research groups in this area. A popular approach to facilitate the wheelchair adaptation to different environments is learning through instruction. After being trained in a specific environment, the wheelchair can navigate and perform tasks inside that specific environment. During the training period, the system must build a map of the environment, which is later used to perform comparisons with the real world, using therefore autonomous navigation techniques. Some projects use topological maps (TAO), and others use a combination of topological and scale maps (Rolland, Senario). GPS can also be used for navigation in external environments. Some projects use GPS as a navigation technique (INRO, NavChair, TAO).

2.4 Summary

In this chapter, the main requirements to design and develop an intelligent wheelchair were described, as well as different approaches used by researchers to solve common problems. A table containing an overview of the most relevant projects was also shown, giving a short description of the used sensors, and each research focus.

It is valid to claim that there is not any project which can gather a different range of functionalities, namely hardware modularity to allow for an easy adaptation on different electric wheelchair models with minimal modifications, as well as a platform to permit the interaction with simulated environments and augmented reality. Another important factor is the concern in minimize the impact of the original wheelchair design.

Chapter 3

Human Machine Interaction

This chapter contains an overview of some low cost input devices that can be used in human computer interaction, as well as some recognition methods used to extract users intentions. The most popular mode of Human Machine Interaction (HMI) still relies on the keyboard and mouse. These devices have grown to be familiar but tend to restrict the information and command flow between the user and the computer system. This limitation has become even more apparent with the emergence of novel display technology such as virtual reality and wearable computers [SPH98]. The advances in signal processing, sensor technology and machine vision allow researchers to greatly expand the traditional user interfaces and devices.

3.1 Input Devices

A wide number of computer peripherals can be classified as input devices. An input device is something that is used to interact with, or provide data to the computer, as opposed to an output device that displays data for the user. The most common input devices are the mouse and keyboard. However, there are other input devices that can be used in human machine interaction, such as microphone, video camera, joystick, tongue mouse, accelerometer, gyroscope, data gloves, brain headsets, and others.

3.2 Recognition methods

There is a wide range of recognition methods using different input devices. In this section an overview of the most common techniques used in human machine interaction is given. Figure 3.1 shows an association between the human body and some of the existing recognition methods.

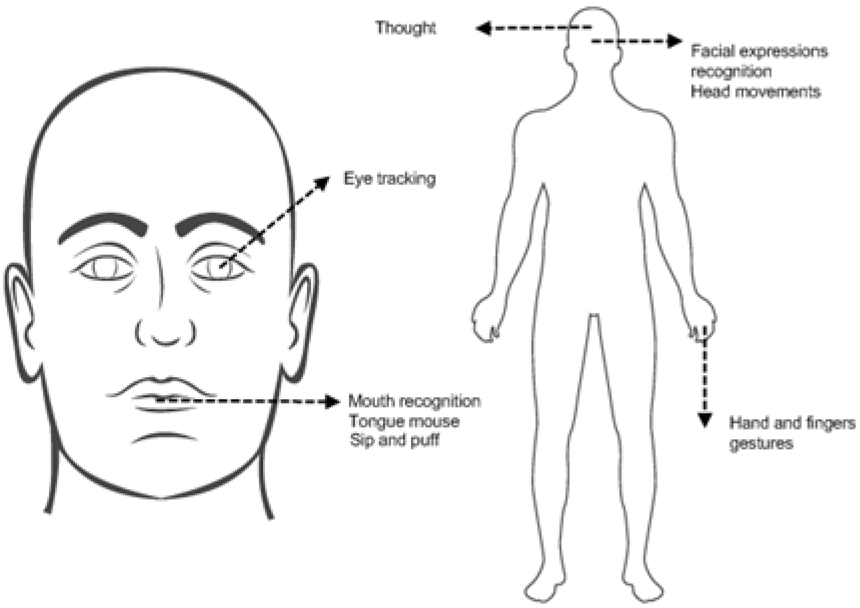


Figure 3.1: Human body and recognition methods

3.2.1 Video based systems

Video based systems can be used to recognize different types of human body motion. There are many publications related to video based recognizers, ranging from head movements, facial expressions, mouth recognition, and others. Some intelligent wheelchair projects use video based systems to capture the facial expressions, gaze and position and orientation of the patients' head to control the wheelchair.

3.2.1.1 Facial expression analysis

Human face-to-face communication is an ideal model for designing a multimodal human-computer-interface. Facial expressions are the facial changes in response to a person's internal emotional states, intentions, or social communications. Suwa et al. [SSF78] presented an early attempt to automatically analyse facial expressions by tracking the motion of twenty identified spots on an image sequence in 1978. After that, much progress has been made to build computer systems to help us understand and use this natural form of human communication.

Facial expression analysis refers to computer systems that attempt to automatically analyse and recognize facial motions and facial feature changes from visual information. Human interpretation is aided by context, body gesture, voice, individual differences, and cultural factors as well as by facial configuration and timing. Computer facial expression analysis systems need to analyse the facial actions regardless of context, culture, gender, and so on.

The facial action coding system (*FACS*) is the most widely used method to measure facial movement. *FACS* defines 32 action units, which are a contraction or relaxation of one or more muscles [SCW$^+$01].

Some recent intelligent wheelchair projects presented input methods for people with quadriplegia, by using the recognition of facial expressions as the main input to guide the wheelchair [Sil02].

A survey on recent techniques for detecting facial expressions using Artificial Neural Networks, Hidden Markov Models and Support Vector Machine can be consulted in [GW10].

3.2.1.2 Eye gaze tracking

The first methods used for eye tracking took advantage of Electro-oculography (EOG), which is a biological measurement technique used to determine the resting potential of the retina. An example of a system using this technique is Eagle Eyes [GDCO96], used in the previously referred Wheelesley wheelchair. The "Dark Pupil/Light Pupil" is another eye tracking technique using infra-red light. Under infra-red illumination, the pupil becomes very white, almost the exact opposite of its visual-spectrum appearance. By capturing both the dark and light pupil images, the high contrast (which is mostly localized to the pupil) can be used via image subtraction to evaluate the pupil location with very high accuracy [ZJ05].

With the growth of photography and video recording technology, far more reliable and less invasive means were developed to simply observe a user's eye motions during long periods of activity. [AHM09]

Eye tracking devices fall into two categories. The first is passive, and is focused on detecting the gaze of the user relative to the rest of the world and in particular what elements of the visible field are currently being focused upon. The second is more active, and considers the eye not as simply a means of observation, but a means of control as well. The second category may be interesting in the way that wheelchairs may be controlled hands-free, through gaze. Tall and Alapetite [TASA$^+$09] present a study about gaze-controlled driving using video-based systems to track the user's eyes gaze orientation.

3.2.1.3 Head movements

Some video based systems capture the orientation and position of the user's head [JHLY07], which can be used as an input to represent a specific desired output. This technique has been used in some wheelchair projects, namely at Osaka University [NKSS99]. In the WATSON wheelchair [MIO01], a video system is used to detect the user's gaze and face orientation.

3.2.1.4 Mouth recognition

Also mouth recognition is used to associate the users' intention to obtain desired outputs. This method uses pattern recognition techniques to detect certain shapes of the mouth which act as input tokens. Ju, Shin and Kim [JSK09] proposed a novel IW interface using face and mouth recognition for severely disabled individuals. For the recognition of the user's intention, the direction of the IW is determined according to the face inclination, while proceeding and stop are determined by the shape of the mouth. The proposed interface allows the user to control the wheelchair directly by changing their face inclination and mouth shape. If the user wants the wheelchair to move forward, they just say "*Go*". Conversely, to stop the wheelchair, the user just says "*Uhm*". The control commands using the shape of the mouth are only effective when the user is looking forward, thereby preventing over-recognition when the user is talking to someone.

3.2.2 Speech recognition

Speech recognition is the ability of a computer to break down human spoken words to binary code understandable by the computer. This emerging technology is also useful for disabled people. By extending the interface of the computer to the disabled, they are given the freedom to interact with their computer and allow them to do things with their computer that they could not have done with a keyboard or a mouse.

Speech recognition systems were first used by severely disabled individuals with normal speech. The goal was to promote independence whereby speech recognition was used to convert human speech signals into effective actions [Man]. Frequently, speech is the only remaining means of communication left for these individuals. The first voice activated wheelchair with an environmental control unit (ECU) was developed in the late 1970's at Rehabilitation Medicine in New York [Man]. The user could operate multiple items including the telephone, radio, fans, curtains, intercom, page-turner and more. A group of individuals with cerebral palsy rated the wheelchair as superior to breath control systems because it eliminated the need for scanning, allowing the user quicker access by directly selecting the desired function with voice [Man].

As the robustness of the available speech recognition systems was improved during the last years, the widespread availability of this low-cost technology was used in many other intelligent wheelchair projects such as NavChair, RobChair, Senario, Tetra-Nauta, MIT Wheelchair, etc.

Although today speech recognition can be used with satisfactory effectiveness in many situations, it still shows to be flawed when the surrounding environment is noisy. Other situations include cases where the user's voice does not match the training data, or when the user cannot achieve proper speaker adaptation.

Sasou and Kojima [SK09] proposed a noise robust speech recognition applied to a voice-driven wheelchair. Their approach consists of using an array of microphones installed on the wheelchair, unlike the common method of using a singular head microphone placed close to the user's mouth. The achieved accuracy is very similar to the headset microphone method, and has the advantage of avoiding situations where the headset microphone position must be adjusted by the user. For hand disabled individuals, who are one of the major users of this wheelchair, this represents an interesting approach.

3.2.3 Gestures recognition

Gesture recognition is the interpretation of a human gesture by a computing device. The goal of gesture recognition research is to develop systems which can identify specific human gestures and use them to send information for a device control.

Recognizing gestures as input allows computers to be more accessible for the physically-impaired and makes interaction more natural in a *3D* virtual world environment. Hand and body gestures can be amplified by a controller that contains accelerometers and gyroscopes to sense tilting, rotation and acceleration of movement. Alternatively, as explained in the previous sub-chapter, the computing device can be equipped with a camera so that specific software can recognize and interpret the gestures.

3.2.3.1 Data gloves

A data glove is an interactive device, consisting of a glove worn on the hand, which facilitates tactile sensing and motion control in robotics and virtual reality. Data gloves are one of several types of electromechanical devices used in haptics applications. Various sensor technologies are used to capture physical data such as bending of fingers. Often a motion tracker, such as a magnetic tracking device or inertial tracking device, is attached to capture the global position/rotation data of the glove. These movements are then interpreted by the software that accompanies the glove, so any one movement can mean any number of things. Gestures can then be categorized into useful information, such as to recognize Sign Language or other symbolic functions.

One example is the CyberGlove [Sys] (Figure 3.2), a glove that captures the position and movement of fingers and wrist. It has up to 22 sensors, including three bend sensors one each finger, four abduction sensors, plus sensors measuring thumb crossover, palm arch, wrist flexion and wrist abduction.

A recent technology, AcceleGlove [Ant], is an open-source data glove which can be programmed and adapted to a specific application, using a Java framework.

DRive is a gesture recognition system that allows a quadriplegic person to control a car interface using a data glove. More information regarding this project is available at [Kav08].

Figure 3.2: Cyberglove data glove

3.2.4 Thought recognition

A brain-computer interface (BCI) is a direct communication pathway between a brain and an external device. BCIs are often aimed at assisting, augmenting or repairing human cognitive or sensory-motor functions. As opposed to invasive and partially-invasive BCIs, which require brain implants, non-invasive BCIs do not require any brain intervention. The most studied potential non-invasive interface is electroencephalography (EEG). By using a headset equipped with a number of electrodes, the user can train thought patterns that will be associated to a certain output action. In spite of being in a premature state of development, this technology might be of good use for medical purposes, namely for severely handicapped individuals.

The idea of integrating brain computer interfaces in intelligent wheelchairs is already present in the literature. For example, the Maia project is an European project which aims at developing an EEG-based BCI for controlling an autonomous wheelchair [PMV+07]. Behaviours like obstacle avoidance and wall following are built in the autonomous control of the wheelchair, while the user just gives simple commands like "go straight", "go left".

The authors of [BCS+08] propose a slightly different approach for an user to drive an intelligent wheelchair, using a BCI. Instead of performing high-level commands, the user must continuously drive the wheelchair.

Another project under development at the National University of Singapore consists of an wheelchair controlled through a P300-based BCI [Reb06]. The wheelchair is autonomous, but its movements are constrained to predefined paths. The user selects a destination, and the wheelchair automatically calculates the trajectory to the desired place. If an unexpected situation occurs, the wheelchair stops and waits for further commands.

Unfortunately, there are still issues that cause problems when trying to use a BCI. There are large differences in the brain activity between people, and even within one person the brain activity changes quickly over time [MHD+10]. This makes it difficult to create a system that will understand what the user is trying to do, especially for a longer duration. Depending on the BCI paradigm used a lot of training may be required (ranging from minutes to months) before the user is able to generate the correct signal for the system [MHD+10].

Figure 3.3 shows the wheelchair prototype developed by the LURCH project [Pro]. It uses a non-invasive BCI that allows the user to drive the wheelchair.

Figure 3.3: Lurch wheelchair controlled by BCI

3.2.5 Sip and puff

Sip-and-puff technology is a method used to send signals to a device using air pressure by "sipping" and "puffing" on a device called a "straw" or "wand." It is primarily used by people who do not have the use of their hands. It is usually a type of wheelchair mechanism for people with quadriplegia with very high injury.

The mouth-controlled input provides people with quadriplegia a simple and effective way to control mouse movement. Movement and operation of this joystick is similar to that of a 'Mouthstick'. Mouse button clicking is accomplished with the help of sips or puffs function of the joystick.

A sip-and-puff input device combined with scanning software means that many keyboard-accessible programs can be used with this device.

Some wheelchairs use sip and puff technology to aid in the navigation (SIAMO). Figure 3.4 shows a wheelchair controlled by a sip and puff device.

Figure 3.4: Sip and puff powered wheelchair

3.3 Summary

Intelligent wheelchairs have been used to explore a variety of alternatives to the more traditional input methods associated with powered wheelchairs (e.g. joystick). Voice recognition has often been used for intelligent wheelchairs (e.g., NavChair, RobChair, SENARIO, Tetra-Nauta) because of the low cost and widespread availability of commercial voice recognition hardware and software.

More exotic input methods that have been implemented include detection of the wheelchair user's sight path (i.e., where the user is looking) through electro-oculographic (EOG) activity (e.g., Wheelesely, SIAMO) or the use of machine vision to calculate the position and orientation of the wheelchair user's head (e.g., Osaka University, Watson).

Some more recent projects presented input methods for people with quadriplegia, by using the recognition of facial expressions as the main input to guide the wheelchair. Recently, researchers at the Georgia Institute of Technology presented a prototype of a tongue mouse that may used by individuals with spinal injuries as a joystick to control motorized wheelchairs. Another solution suggested by a group of researchers consists of operating a intelligent wheelchair using the thought as an input method, by means of sensors that capture the user's brain electromagnetic waves.

Chapter 4

Multimodal Interfaces

Multimodal interfaces process two or more combined user input modes, such as speech, pen, touch, manual gestures and gaze, in a coordinated manner with multimedia system output [Ovi02].

They represent a new class of emerging systems that aim to interpret symbols of human language and behavior, incorporating technologies based on recognition. Multimodal interfaces represent a replacement to the traditional WIMP (Window, Icon, Menu, Pointer) graphical user interfaces. The main reason for the development of this kind of interfaces relies on the expressive, transparent, efficient and highly mobile form of human-computer interaction they offer. The interaction style provided by multimodal interfaces permits users to have a bigger flexibility to combine modalities of inputs, or to switch from one input to another that may be better suited for a particular task or setting. For handicapped individuals this interface paradigm can be a potential benefit.

Multimodal systems have been evolving at a fast rate in the last decade, with steady progress towards building more general and reliable systems [Ovi02]. Major improvements have occurred in the hardware and software necessary to support the key component technologies incorporated in multimodal systems and in techniques for fusing parallel input streams. To date, most current multimodal systems are bimodal, with the two most mature types involving speech and pen or touch input and audio-visual input (e.g., speech and lip movements) [Ovi02]. However, these systems have been diversified to include new modality combinations such as speech and manual gesturing, and gaze tracking and manual input. Multimodal applications also range from map-based and virtual reality systems for simulation and training, to multi-biometric person identification/verification systems for security purposes, to medical, educational, and web-based transaction [Ovi02].

4.1 Advantages

Multimodal interfaces were first seen as more efficient than unimodal interfaces. However, evaluations showed that multimodal interfaces only speed up task completion by 10% [Ovi97]. Hence, efficiency should not be considered that main advantage of multimodal interfaces. On the other hand, multimodal interfaces have been shown to improve error handling and reliability: users made 36% fewer errors with a multimodal interface than with a unimodal interface [Ovi97]. Multimodal interfaces also add greater expressive power, and greater potential precision in visual-spatial tasks. Finally, they provided improved support for user's preferred interaction style, since 95%-100% of users prefer multimodal interaction over unimodal interaction [Ovi97].

4.2 Features

Compared to other types of human machine interaction, multimodal interaction tries to offer users a more natural and expressive interaction, using speech, gestures, gaze, direction, etc. Multimodal interfaces are thus expected to offer easier and more intuitive ways to use computers. Multimodal systems have the potential to enhance human machine interaction in a number of ways:

- Enhanced robustness due to combining different partial information sources;
- Flexible personalization based on user and context;
- New functionality involving multi-user and mobile interaction.

When comparing multimodal user interfaces (MUI) with standard graphical user interfaces (GUI), it is possible to draw the differences contained in Table 4.1.

GUI	**MUI**
Single input stream	Multiple input streams
Atomic, deterministic	Continuous, probabilistic
Sequential processing	Parallel processing
Centralized architectures	Distributed time-sensitive architectures

Table 4.1: Differences between GUI and MUI [Ovi02]

In standard WIMP interaction style, only one physical input device is used to control the position of a cursor and present information arranged in windows and represented with icons. Differently, in multimodal interfaces various modalities can be used as input streams (voice, gestures, facial expressions, etc.). Furthermore, input from graphical user interfaces is generally deterministic, with either mouse position or characters typed on a keyboard used to control the computer. In multimodal interfaces, input streams have to be first interpreted by probabilistic recognizers [DLO09], and thus their results are weighted by a degree of uncertainty. Further, events are not always clearly temporally delimited and thus require a continuous interpretation. Since multiple recognizers are needed to interpret multimodal input, multimodal systems depend on synchronized parallel processing.

Moreover, the time sensitivity of multimodal systems is crucial to determine the order of processing multimodal commands in parallel or in sequence. Finally, multimodal systems often implement a distributed architecture, to deal out the computation and insure synchronization. Multimodal systems can be very resource demanding in some cases (e.g., speech/gesture recognition) [DLO09].

4.3 Design principles for multimodal systems

Classical design principles for human computer interfaces recommend conducting iterative evaluations at different stages in the design process [BM03]. However, evaluations and available guidelines mainly focus on output characteristics, because input devices are often a priori fixed as mouse and keyboard. In multimodal interfaces design, evaluations must deal with both input and output devices, and test reciprocal influences they have on each other [BM03].

In the course of the last decade, researchers have highlighted particular empirical findings that have guided the design of multimodal interfaces compared to other sorts of human machine interfaces [Ovi02]. Based on empirical findings, Oviatt [Ovi02] distilled implications for how more effective multimodal interfaces could be designed.

In more recent years, research has also focused in mainstreaming multimodal interfaces. In this trend, Reeves et al. [Ree04] defined the following "guidelines for multimodal interface design":

- Multimodal systems should be designed for the broadest range of users and contexts of use, since the availability of multiple modalities supports flexibility. For example, the same user may benefit from speech input in a car, but pen input in a noisy environment.

- Designers should take care to address privacy and security issues when creating multimodal systems: speech, for example, should not be used as a modality to convey private or personal information in public contexts.

- Modalities should be integrated in a manner compatible with user preferences and capabilities, for example, combining complementary audio and visual modes that users can co-process more easily.

- Multimodal systems should be designed to adapt easily to different contexts, user profiles and application needs.

- Error preventing and handling is a major advantage of multimodal interface design, for both user and system centered reasons. Specific guidelines include integrating complementary modalities to improve system robustness, and giving users better control over modality selection so they can avoid errors.

According to Petros Maragos [Mar08], some important design principles that should be followed when projecting a multimodal interface are presented in Table 4.2

Symmetric multimodality	The principle of "symmetric" multimodality requires that the same modalities are used for both input and output. Using the same modality for both input and output reduces the cognitive load and improves efficiency.
No representation without presentation	There should always be output presentation for internal system representations (system states) and vice versa, output should correspond to an internal semantic representation or communication act. This principle emphasizes consistency between the interface and data model of the system.
Efficiency and Synergy	Synergy is a design principle that applies to systems that more than one input or output modalities. A synergistic multimodal interface is more than the sum of its parts. To achieve high synergy it is important not only to use the appropriate modality for each part of the application, but also to follow for interplay between the modalities, e.g., speech misrecognitions should be resolved via the GUI interface.
Robustness	Novel "modalities" are based on new technologies and interaction paradigms that might be error prone, e.g., recognition errors for speech input. Each modality is more or less error-prone depending on the specific environment or situation it is being used. Also users tend to use the less error-prone modality at each context and switch modalities after system errors. These behaviors can be reinforced by appropriate user interfaces design, e.g., use facial expression recognition as a default modality for resolving ambiguities that arise from speech recognition errors.
Compositionality	The space of possible user interface configurations increases exponentially as the number of modes increases. Combining different input types can help to achieve a more robust and reliable interface.

Table 4.2: Design principles for multimodal interfaces [Mar08]

Overall, synergy, robustness, modularity, customizability and consistency are some important features of successful multimodal systems.

Inspired by Norman's action cycle [Nor88], and based on well accepted findings and taxonomies, the following model of multimodal human-machine communication can be drawn, together with the major concepts that should be considered when building a multimodal system: the fusion of multimodal inputs, and the multimodal fission to generate an adequate message to the user, according to the context of use, preferences and profile.

When a human interacts with a machine, his communication can be divided in four different states. The first state is a decision state, in which the communication message content is prepared consciously for an intention, or unconsciously for attentional content or emotions [DLO09]. The second state is the action state, where the communication means to transmit the message using different possible techniques, such as speech, gestures or facial expressions. The machine, in turn, will make use of a number of different modules to grasp the most information possible from a user, and will have similarly four main states [DLO09].

A recent survey on interactive design of multimodal interfaces can be found in [KRR10].

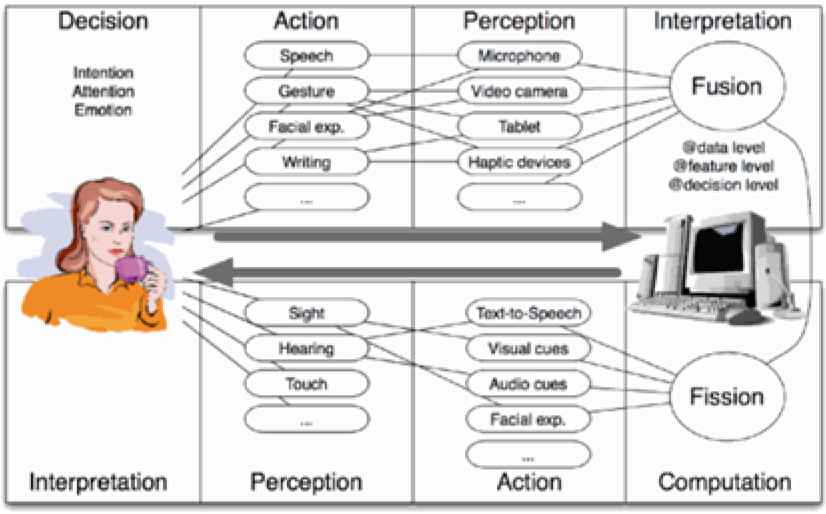


Figure 4.1: A representation of multimodal human machine interaction loop [Ovi02]

At first, the messages are interpreted in the perception state, where the multimodal system receives information from one or multiple sensors, at one or multiple levels of expression. In the interpretation state, the multimodal system will try to give some meaning to the different information it collected in the perception state. This is typically the place where fusion of multimodal messages takes place. Further, in the computational state, action is taken following the business logic and dialogue manager rules defined by the developer. Depending on the meaning extracted in the interpretation state, an answer is generated and transmitted in the action state, in which a fission engine will determine the most relevant modalities to return the message, depending on the context of use (e.g. in the car, office, etc.) and the profile the user (disabled, elderly, etc.) [DLO09].

4.4 Multimodal Interfaces Projects

Media Room

A classical example of a multimodal interface is the Media Room demonstration system. This system allowed one person to move an object into a new location on a map on the screen by saying "Put that there" while pointing to the object itself, and then pointing to the desired destination [Bol80].

MATCH

MATCH is a working city guide and navigation system that currently enables mobile users to access restaurant and subway information for New York City. Users interact with a graphical interface displaying restaurant listings and a dynamic map showing locations and street information. They are free to provide input using speech, by drawing on the display with a stylus, or by using synchronous multimodal combinations of the two modes. The output is also flexible, being a combination of a graphical call-out on the display, synchronized with a text-to-speech prompt of the desired information [JB04].

QuickSet

QuickSet is one of the more widely known and older map-based applications that make use of speech and pen gesture input [JCM$^+$97]. QuickSet is a military-training application that allows users to use one of the two modalities or both simultaneously to express a full command. For instance, users may simply draw out with a pen a predefined symbol for platoons at a given location on the map to create a new platoon in that location. Alternatively, users could use speech to specify their intent on creating a new platoon and could specify vocally the coordinates in which to place the platoon. Lastly, users could express vocally their intent on making a new platoon while making a pointing gesture with a pen to specify the location of the new platoon [JCM$^+$97].

Real Hunter

A more recent multimodal map-based application is Real Hunter. It is a real-estate interface that expects users to select objects or regions with touch input while making queries using speech. For instance, the user can ask "How much is this?" while pointing to a house on the map [CHZ04].

ICANDO

ICANDO (Intellectual Computer AssistaNt for Disabled Operators) is a multimodal interface intended mainly for the assistance of persons without hands or with disabilities of their hands and arms. It combines one module for automatic recognition of voice commands in English, French and Russian, as well as an head tracking module. ICANDO interface was applied for hands-free work with a graphical user interface of a personal computer in tasks such as Internet communication and work with graphical and text documents [KR07].

Xbox Kinect

Kinect for Xbox 360, or simply Kinect, is a controller-free gaming and entertainment experience developed by Microsoft for the Xbox 360 video game platform. It enables users to control and interact with the Xbox 360 without the need to to touch a game controller, through a natural user interface using gestures and voice commands [Mic].

OpenInterface

OpenInterface is the name of an European organization aimed to expand, coordinate and integrate the efforts of several research labs in the field of multimodal interfaces. This organization is responsible for developing a software platform to allow a faster prototyping and development of multimodal interfaces based on the reutilization of components [Ope].

4.5 Summary

In this chapter was introduced the concept of multimodal interface. The combination of different input modalities to design human machine interfaces has evolved in the last years, especially due to the advance of the available interaction devices and technologies. This type of interaction offers an interesting alternative to the traditional WIMP paradigm, and might be of great assistance to individuals with little body movement aptitude. The design principles to follow while designing a multimodal interface were also explained, as well as the multimodal interaction loop. Finally, some relevant projects on multimodal interfaces were briefly described.

Chapter 5

IntellWheels Platform

This chapter presents the concept, design and implementation of the platform for development of intelligent wheelchairs, named IntellWheels. Since the aim of this work was to design a flexible tool that could facilitate the development of new technologies that could be applied to intelligent wheelchairs (IW), it may be considered as a comprehensive framework to enable the development of intelligent wheelchairs. The system was designed with modular features and the paradigm of multi-agent systems was followed.

5.1 Concept

The IntellWheels project main objective is the creation of a development platform for intelligent wheelchairs [BPMR08a], entitled IntellWheels Platform (IWP). The project main focus is the research and design of a multi-agent platform, enabling easy integration of different sensors, actuators, devices for extended interaction with the user [LBSM09], navigation methods and planning techniques and methodologies for intelligent cooperation to resolve problems associated with intelligent wheelchairs [BPMR08b].

It is expected that this platform may facilitate the development and testing of new methodologies and techniques relating to intelligent wheelchairs. Within this concept, an attempt has been made at creating a set of tools (software and hardware) that can be easily integrated into any powered wheelchair, commercially available, with minor modifications. The project also takes into consideration issues such as low cost maintenance, and the preservation of the patient comfort, as well as the original wheelchair ergonomics.

Some relevant capabilities of a intelligent wheelchair can be enumerated: intelligent planning, autonomous navigation, and semi-autonomous navigation using commands given by the user (patient) with a high level language. These capabilities can be achieved through an advanced control system, ranging from a simple shared control (in which the system can prevent crashes when the user is manually controlling the wheelchair) to the control of more complex tasks using high level commands (in this case, commands interpreted by a multimodal interface, automatic planning, autonomous driving and environment mapping). The system was designed with six basic modules, illustrated in Figure 5.1, namely: planning, interface, simulation, communication, navigation and hardware.

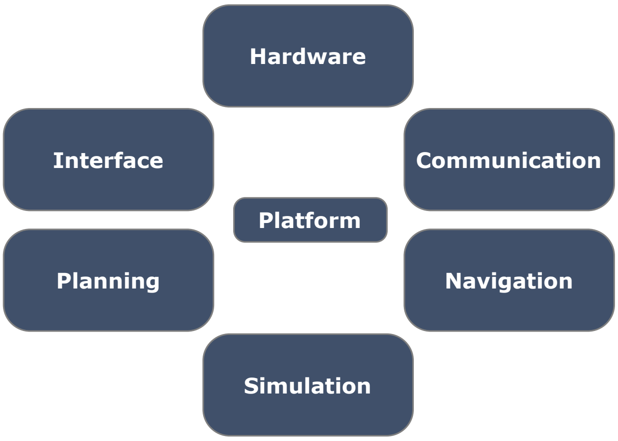


Figure 5.1: Basic modules of IntellWheels Platform

5.2 Software Architecture

Figure 5.2 represents the software architecture proposed in this project. This architecture uses the paradigm of Multi-Agent Systems (MAS). The decision of using the MAS paradigm was due to its flexibility, in order to subserve the addition of new modules and the interaction between the wheelchairs, and other intelligent devices. In this architecture one can observe four basic agents, which compose a single wheelchair:

- **Interface Agent**: responsible for the interaction between the patient and the wheelchair
- **Intelligence Agent**: responsible for the planning actions of the wheelchair
- **Perception Agent**: responsible for reading the adequate sensor values for each context, location and environment mapping
- **Control Agent**: responsible for the control activities of basic actions, wheel control, and obstacle avoidance

These agents are heterogeneous and can collaborate with other agents of another wheelchair. It can be observed in the architecture, that the previously described agents can share the control of both a real wheelchair, or a simulated wheelchair.

One of the most innovative features of the platform is the use of the mixed reality concept, which allows the interaction between a real intelligent wheelchair with virtual objects and virtual wheelchairs. This possibility of interaction makes it possible for high

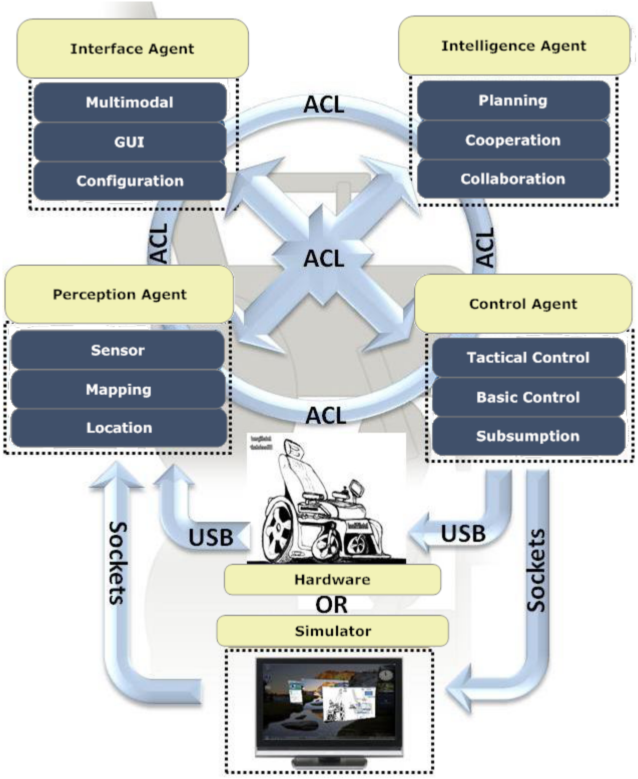


Figure 5.2: Software Architecture of IntellWheels Platform

complexity testing using large sets of devices and wheelchairs, decreasing the costs associated with the development of new technologies, since there is no need to build a large set of real intelligent wheelchairs with an expensive and complex infrastructure. With this feature, it is possible to analyse and evaluate the interaction between a large number of IW, by setting an environment composed by a real wheelchair and a number of virtual wheelchairs.

The interaction between the agents in this scenario would result in the same interaction of a scenario composed by real wheelchairs only, once the involved agents are the same. The only basic difference would be the robot body presence.

Figure 5.3 illustrates the combination of the possible different operation modes provided by the platform.

The Intellwheels platform allows the system to work in real mode (the IW has a real body), simulated (the body of the wheelchair is virtual) or mixed reality (real IW with perception of real and virtual objects). In real mode it is necessary to connect the system (software) to the IW hardware. In the simulated mode, the software is connected to the IWP simulator. In the mixed reality mode, the system is connected to both (hardware and simulator).

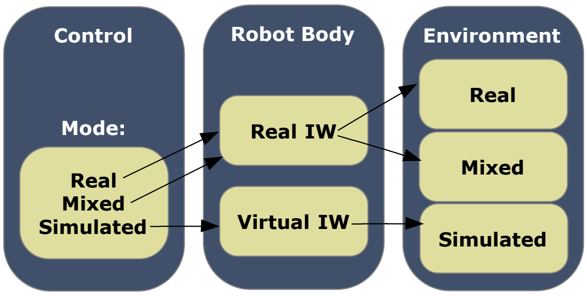


Figure 5.3: Operation modes of IntellWheels Platform

5.3 Platform Hardware

For a wheelchair to be considered intelligent, it should be able to understand the surrounding environment, plan their actions, react to environmental changes and provide an expanded interface for optimized user interaction. The interface should provide means of recognizing the users intentions in a flexible and configurable fashion, and turn them into navigation commands that are subsequently processed by the control. To meet these characteristics, a hardware kit has been developed, composed by low cost devices that can be easily adapted to commercial wheelchairs, and at the same time, fulfil the desired system requirements. Figure 5.4 illustrates the developed hardware architecture.

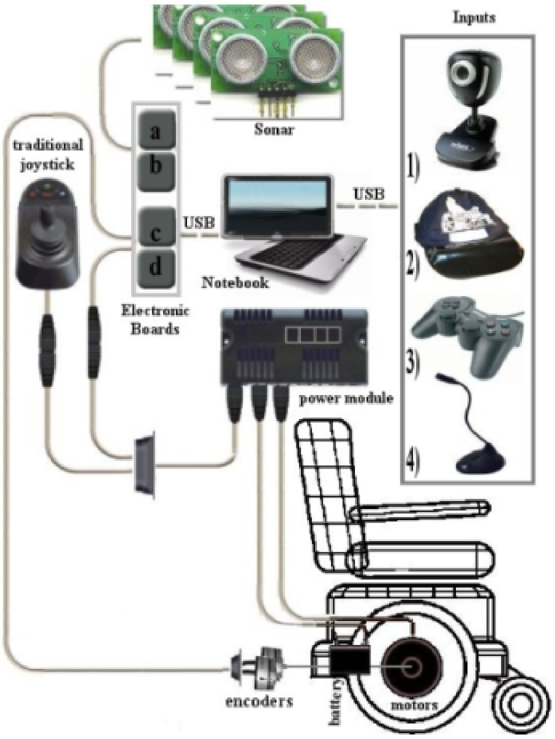


Figure 5.4: IntellWheels Platform hardware architecture

Input Devices

Several types of input devices were used in this project to allow people with different disabilities to be able to drive the IW. The intention is to offer the patient the freedom to choose the device they find most comfortable and safe to drive the wheelchair.

Moreover, these multiple inputs for interaction with the IW can be integrated with a control system responsible for the decision of enabling or disabling any kind of input, in case of any observed conflict or dangerous situation. For example, in a very noisy environment, in which voice recognition might not have an accurate behaviour, this type of input can be temporarily disabled. These devices range from traditional joysticks, accelerometers, to commands expressed by speech or facial expressions.

A detailed description of the input devices and recognition methods that were used for developing the interface for the patient can be found in Chapter 7.

Sensors

To compose the necessary set of hardware to provide the wheelchair's ability to avoid obstacles, follow walls, map the environment and see the holes and unevenness in the ground, an U-shaped bar was designed, containing a set of eight ultrasonic sensors and twelve infra-red sensors.

To refine the odometry, two encoders were also included, and coupled to the wheels (for distance, velocity and position measurements). Additionally, to recognize artificial landmarks, a web-cam was also installed. It is important to remember that the low cost and maintenance of ergonomics and comfort of the original commercial wheelchair were regarded as some of the project requirements. Thus, the use of high-cost sensors such as laser range-finders or *3D* video cameras was ruled out.

Other hardware devices

To complement the hardware platform, some other devices were used, namely:

- Control and data acquisition board: This board is used to collect data from the sensors and send the motors the reference control for the power module. The board (ArduinoMega12 (Pardue, 2010)) is connected to the computer platform via a *USB* connection.

- Power module: It converts the control signal in power for the motors and offers overcurrent protection (business module produced by PG Drives Technology, model VR2).

- Notebook: To run the platform software used, a laptop (HP Pavilion tx1270EP, AMD Turion 64 X2 TI60) is used. However, other computers with equivalent or superior performance might be used.

5.4 Simulator

This system module, named IntellWheels Simulator [BMR10], allows the creation of a virtual world where one can simulate the environment of an enclosure (e.g. a floor of a hospital), as well as wheelchairs and generic objects (tables, doors and other objects). The purpose of this simulator is essentially to support the testing of algorithms, analyse and test the modules of the platform and safely train users of the IW in a simulated environment. Another purpose of the simulator is related to the testing scenarios involving a large number of intelligent wheelchairs, which would be impossible with real IW due to its cost and complexity.

Since every small modification in the algorithms or hardware would imply, in real environments, a high increase of time and monetary costs, having a simulator is desirable and of great importance. Additionally, its use preserves the safety of the user, avoiding potentially dangerous situations during the test phases. The IntellWheels Simulator is an adaptation of the open-source simulator named *Ciber-Rato* [LPM+02]. Figures 5.5, 5.6 and 5.7 show examples of the IntellWheels simulator viewer, which provides different visualization modes.

Figure 5.5: IntellWheels Simulator *2D* Viewer

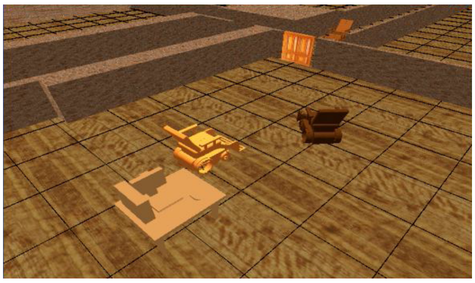


Figure 5.6: IntellWheels Simulator *3D* Viewer, Free View

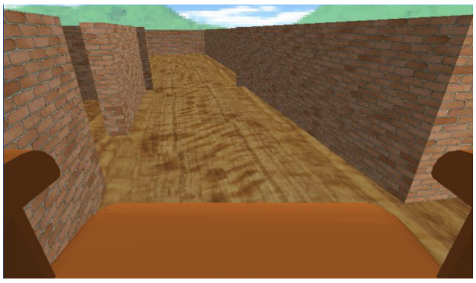


Figure 5.7: IntellWheels Simulator *3D* Viewer, First Person View

5.5 Control and Navigation

The navigation module of the IWP includes a set of algorithms responsible for performing the treatment of the values of sensors on the wheelchair, in order to trace its location and map the environment [Bra10]. The set of functions related to this module is implemented in a distributed manner among the agents Perception, Control and Intelligence.

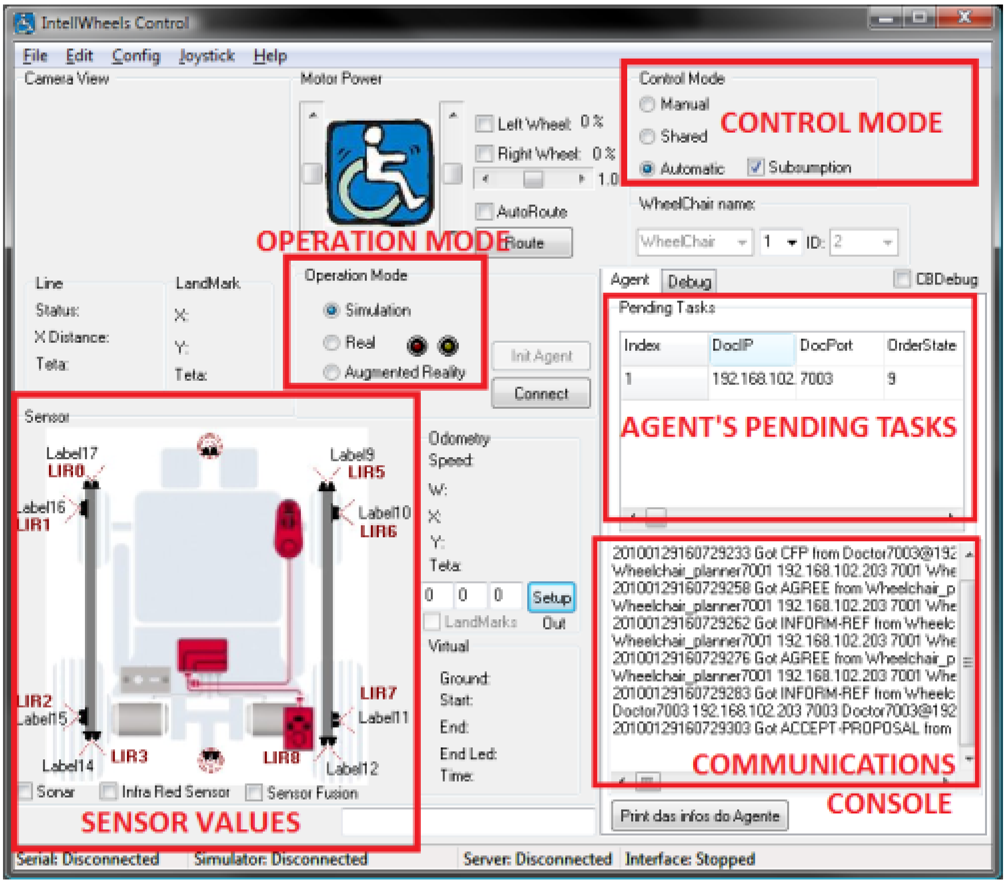
The IWP presents a multi-level control architecture, subdivided into three layers: strategic layer, tactical layer and basic control layer [Bra10].

Some simple algorithms were implemented for the control of basic actions, such as: follow line, go to point *x,y* and turn to angle *theta*. These algorithms were based on the Cartesian position of the wheelchair. Other algorithms were based on the information of distance sensors (ultrasonic and infra-red sensors) were also implemented, such as following walls and avoiding obstacles. The implemented actions are depicted in Table 5.1.

Action name	**Description**
Go to X,Y	Go to a specific position of a known environment
Spin Theta	Spin an user-defined angle
Follow Line	Move along the nearest landmarks
Follow Right Wall	Continuously follow the nearest obstacle on the right side
Follow Left Wall	Continuously follow the nearest obstacle on the left side
Go Forward	Move forward at constant pre-defined speed
Go Back	Backtrack at constant pre-defined speed
Right Spin	Continuously spin to the right side, at constant pre-defined speed
Left Spin	Continuously spin to the left side, at constant pre-defined speed
Right Turn	Turn right at pre-defined constant linear/angular speeds
Left Turn	Turn left at pre-defined constant linear/angular speeds

Table 5.1: High-Level output actions implemented by the IWP control module

Figure 5.8 shows the IWP control module, responsible for implementing the basic control algorithms. It is also for enabling the connection with the simulator, acting as an intermediate player between the multimodal interface and the simulator. As can be seen in the figure, provides a number of information. The agent's pending tasks provides a registry for the tasks that are still under consideration to be executed. The communications' console acts as a debugger, displaying all the incoming messages. On the left side of the application, it is possible to consult the values transmitted by the infra-red and sonar values. Being the core component of the IWP, the control module enables the wheelchair's configuration parameters, such as the operating mode (real, simulated or mixed), and the control mode to be used (manual, automatic or shared).



Left Wheel: 0 %

Right Wheel: 0 %

1.0

Control Mode

○ Manual
○ Shared
● Automatic ☑ Subsumption

WheelChair name: WheelChair 1 ID: 2

Operation Mode

● Simulation
○ Real
○ Augmented Reality

0 0 0

Pending Tasks

Index	DocIP	DocPort	OrderState
1	192.168.102.	7003	9

```
20100129160729233 Got CFP from Doctor7003@192
Wheelchair_planner7001 192.168.102.203 7001 Whe
20100129160729258 Got AGREE from Wheelchair_p
Wheelchair_planner7001 192.168.102.203 7001 Whe
20100129160729262 Got INFORM-REF from Wheelc
Wheelchair_planner7001 192.168.102.203 7001 Whe
20100129160729276 Got AGREE from Wheelchair_p
Wheelchair_planner7001 192.168.102.203 7001 Whe
20100129160729283 Got INFORM-REF from Wheelc
Doctor7003 192.168.102.203 7003 Doctor7003@192
20100129160729303 Got ACCEPT-PROPOSAL from
```

Serial: Disconnected | Simulator: Disconnected | Server: Disconnected | Interface: Stopped

Figure 5.8: IntellWheels Platform control module

5.6 Summary

In this chapter we have presented an overview of the concept of a platform for developing intelligent wheelchairs (IWP), softly describing the most relevant modules and some aspects of its implementation. This platform is based on a multi-agent system with a focus to facilitate the integration of different sensors, actuators, flexible interface, navigation methodologies, planning techniques, etc.

Chapter 6

IntellWheels Multimodal Interface

The problem of designing specific interfaces for individuals with severe physical disabilities presents many challenges to which many authors have tried to answer over time.

There are several publications in the literature of projects related to this issue, which tried different approaches to implement viable solutions to address this problem. However, as was shown previously in this document, the vast majority of these projects present limited solutions concerning the accessibility and recognition methods offered to the user to drive a particular wheelchair.

It is common to find solutions that only provide voice recognition, while others focus merely on facial expressions recognition. Since the physical disability is very wide and specific to each individual, it becomes important to provide the greatest possible number of recognition methods to try to cover the largest possible number of individuals with different characteristics.

The proposed multimodal interface offers three basic methods of recognition: speech recognition, recognition of head movements and the use of a generic gamepad. In addition, we propose an architecture that makes the interface extensible at this level, making it possible to add new devices and recognition methods in a transparent fashion. It also presents a flexible paradigm that allows the user to define the sequences of inputs to assign to each action, allowing for an easy and optimized configuration for each user.

The first two sections present the established use cases and main features the IntellWheels Multimodal Interface (IMI) should provide. Following, the proposed IMI global architecture is presented and explained.

6.1 Input Sequence

The concept behind an input sequence is very important to understand the interaction style we want to adopt. As explained in Chapter 4, some multimodal systems combine different input modalities in order to issue multimodal commands. Although the merging of different inputs can provide a very attractive way of human machine interaction, it is not ideal for people with physical disabilities. The objective of the IntellWheels multimodal interface is to provide the user a large variety of different possible inputs, that can be expressed by individuals with different physical capabilities. What we propose is a simple mechanism to allow the user to interact with the multimodal interface. It consists of intercepting the user inputs in a sequential fashion, without fusion. Since we are dealing with a system that aims to be used by disabled people, avoiding errors is of extreme importance. Consequently, having singular input tokens associated to the request of an output action is not advisable. For example, if we associate a voice input which consists of pronouncing the word "Go" to the output action that makes the wheelchair to start moving forward, this may origin mishaps in the way that a false recognition by the speech recognition module would originate a not desired event. What we propose is the possibility of creating input sequences, composed by simple different input tokens, that may be easily performed by the user.

Figure 6.1 contains a graphical illustration of an input sequence. It would consist of pronouncing the expression "This is an example", followed by pressing the button 5 of the gamepad, and finally blinking the left eye. This is a simple example to help better understand the concept behind an input sequence.

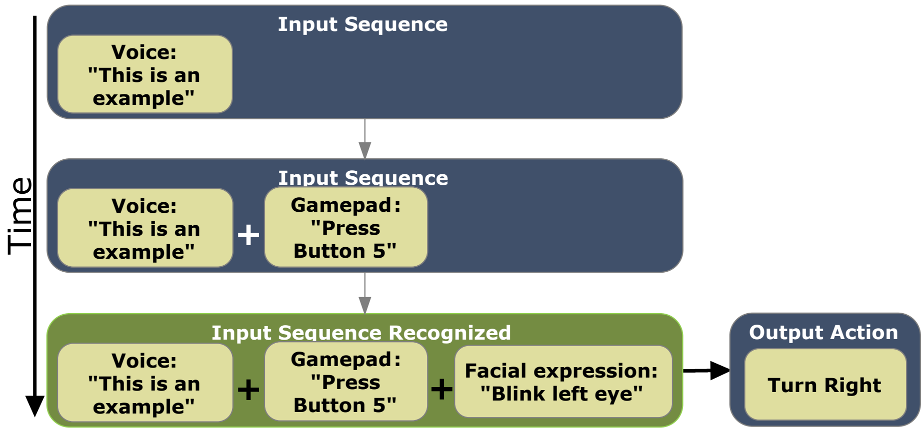


Figure 6.1: Concept of input sequence

6.2 Use Cases

This section presents the main use cases of the IntellWheels Multimodal Interface. As can be seen in figure 6.2, they were grouped in four different packages: Navigation Mode, Sequence Manager, Device Manager and User Profile. Since one of the main goals of this proposal is to provide the user with a full control of the interface, only one actor can be identified.

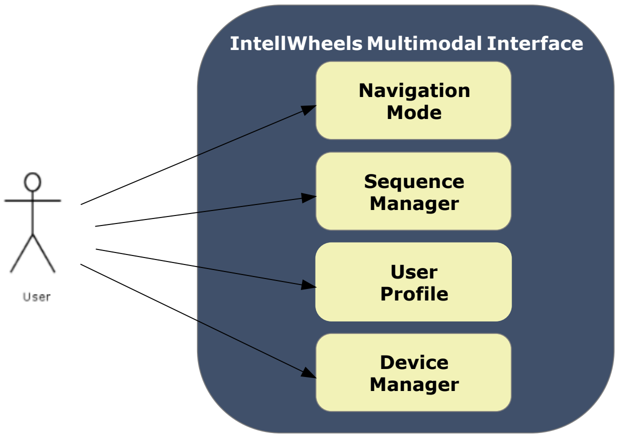


Figure 6.2: Use Cases Overview

Navigation Mode

The package Navigation Mode, presented in figure 6.3, contains a single use case related to the use of the saved associations between input sequences and output actions. This use case represents the most important feature of the multimodal interface, which is to request a certain action, via input sequence, using a previously created association.

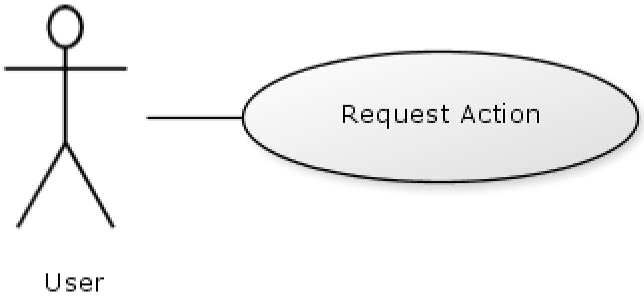


Figure 6.3: Navigation Mode Use Cases

Sequence Manager

The package Sequence Manager, which can be observed in Figure 6.4, contains the use cases related to the management of the saved associations between input sequences and output actions. The user can create a new association by performing the input of the desired sequence and map it to the desired action. The other use case is related to consulting associations created in the past, and the possibility of deleting them.

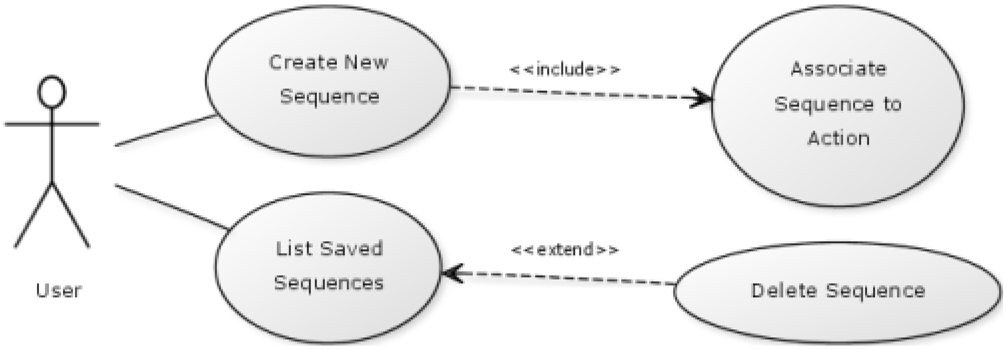


Figure 6.4: Sequence Manager Use Cases

Device Manager

The package Device Manager, shown in figure 6.5, includes the use case that allows the user to consult a small viewer which shows information regarding the state of the three basic input devices. This viewer is intended to test the correct functioning of the input devices. The user should be also able to adjust some minor configurations for these devices, to attain an optimized experience while using the multimodal interface. Configurations may consist of defining a minimum speech recognition trust-level, maximum speed for the gamepad joysticks, etc.

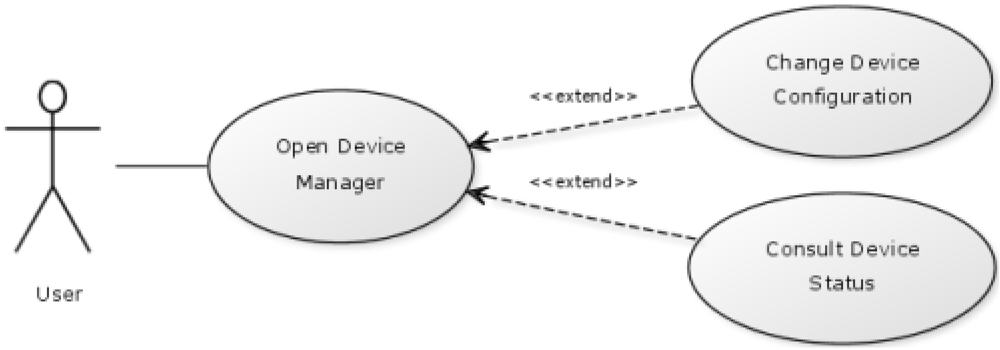


Figure 6.5: Device Manager Use Cases

User Profile

The last package, named User Profile, can be consulted in figure 6.6. It includes the use cases related to the creation or update of an user profile. During this process, the user can separately train the use of the basic input modalities offered by the IntellWheels multimodal interface.

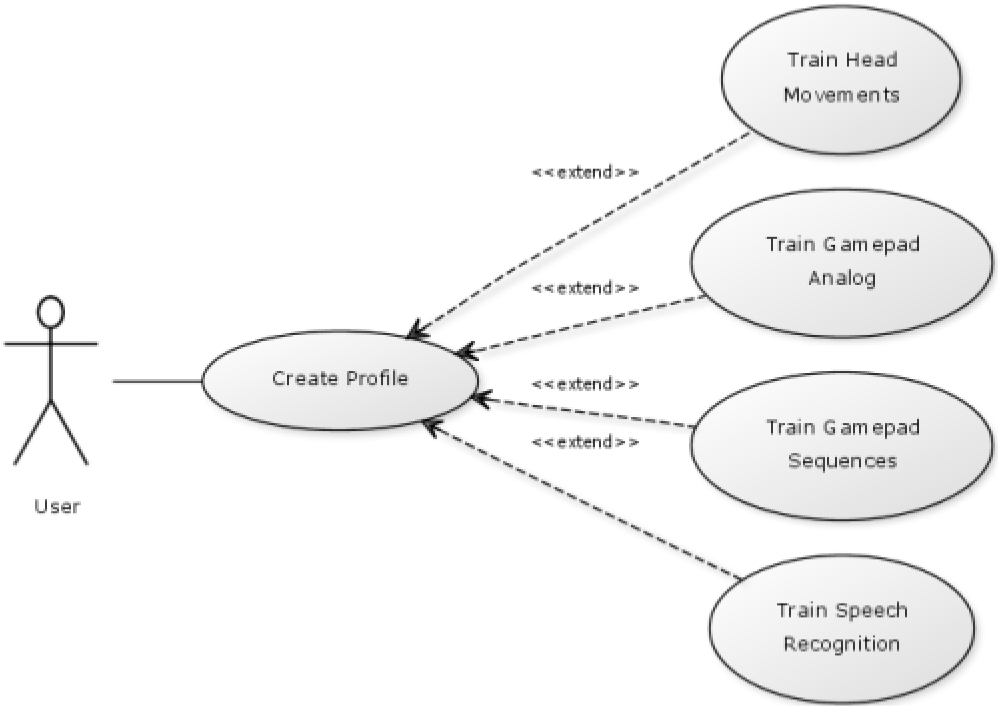


Figure 6.6: User Profile Use Cases

6.3 Requirements

Output Actions

Since the PIW control module is a distinct agent from the multimodal interface, and has itself the role of establishing the connection with the wheelchair robot and sending the low-level orders directly to the motors, a protocol should be created for these two agents to communicate and exchange information between them. Thus, the control module should notify the multimodal interface with the available output actions, previously described in 5.5 (e.g. *go forward*, *left turn*, *manual mode*, etc.).

With this in mind, an abstract structure for defining an output action should be created to make the multimodal interface independent and unaware of the context of its use.

Figure 6.7: Exchange of actions commands description

Input Devices

Since a multimodal interface is a human-machine interface which combines several communications between the user and the machine, one of the requirements of the IMI is the ability to accept inputs from a number of input devices. Therefore, the interface should be aware of the available input modalities, by having a core component capable of interpreting the inputs sent by all the input devices, in a generic and abstract way. These inputs should have their origin in internal (gamepad, speech recognition, head movements) or external input devices (e.g. a facial expression recognizer). To allow the interaction between the IMI and external input devices, a simple communication protocol should be created.

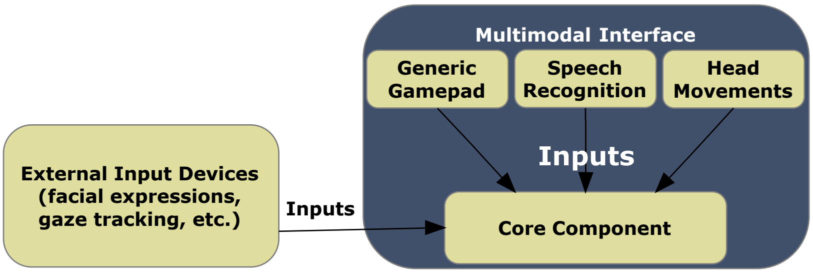


Figure 6.8: Input Devices and the IMI

Custom Configuration

One of the most important goals of the IMI is to allow the end user to easily define the associations between a set of input sequences and output actions. Hence, the IMI should provide a flexible way of creating a customized input sequence that could be freely associated to a certain available action.

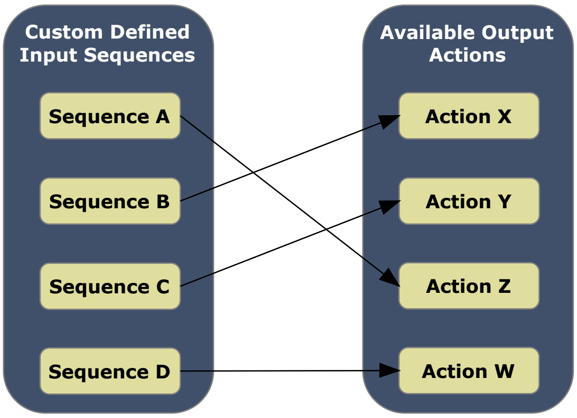


Figure 6.9: Associations between input sequences and output actions

Interface Actions

To offer the user a complete multimodal experience while using the IMI, another feature is proposed. Using the previously described features, it should be possible to associate input sequences to actions that are fully related with the interface itself. These actions may consist, for example, of moving the mouse pointer using head movements, or consulting the existing associations by pronouncing a voice command.

The goal is to avoid the WIMP interaction style while performing GUI related operations.

In spite of being of a distinct type, interface actions should follow the same abstract structure created for the output actions. At the same time, associations related to interface actions should be equally saved to a database.

Database

To avoid the user having to configure the interface according to their preferences each time the IMI is used, all the information regarding the output actions, custom input sequences and associations should be saved to a database. This external database should be available and accessible by the IMI at the beginning of the execution, and during the run-time process. Therefore, every time the IMI is launched, all the past information should be loaded to become available during each execution. Similarly, the IMI should save all the new information generated either by the control module that could send new output actions, or by the user, while creating new associations.

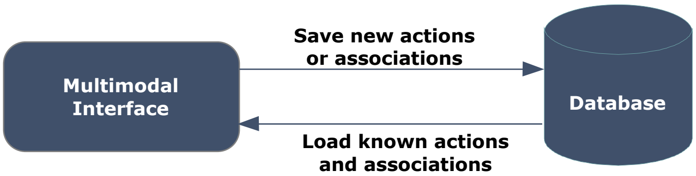


Figure 6.10: IMI Database

User Profile

In addition to the wide variety of disabilities that cause different types of limited mobility, each person has specific characteristics, which may be related to physical and cognitive factors. Thus, individuals with similar symptomatology, may have significant differences.

It is fair to say that characteristics such as moving the head, ability to pronounce words, ability to move the hand and fingers, can vary substantially from individual to individual. Similarly, the time of learning and proficiency in using assistive devices may also vary greatly.

Tracing an user diagnostic can be very useful to adjust certain settings allowing for an optimized configuration and improved interaction between the user and the multimodal interface.

Accordingly, the IMI should contain a module capable of performing series of training sessions, composed of small tests for each input modality. These tests may consist, for example, of asking the user to press a certain sequence of buttons on the gamepad, or to move one of the gamepads' analog switches to a certain position. Another test may be asking the user to pronounce a set of voice commands, or to perform a specific head movement. These tests should be performed sequentially and should have an increasing difficulty. Additionally, the tests should be reconfigurable and extensible. Finally, the set of sets should be saved on a database, accessible by the IMI.

With this in mind, the following user characteristics should be extracted. These characteristics are separated into three different types: quantitative and qualitative.

The quantitative measures consist of:

- The average time taken to perform a full button sequence;
- The average time between pressing two buttons;
- The average time to position the head on a certain position;
- The average trust level of speech recognition;
- Maximum amplitude achieved with the gamepad analog switches in different directions;
- Maximum amplitude achieved with the head in different directions;
- Number of errors made using the gamepad;
- Number of errors made using speech recognition.

Using the quantitative measures, the following qualitative measures should be estimated:

- User ability using the gamepad buttons;
- User ability to perform head movements;
- User ability to pronounce voice commands.

At the end of the training session, the tracked user information should be saved to an external database, containing the users' profile. The user profile can be used to improve security, by defining, for each user, a global trust level for each input modality. The trust level can be used to advice the user of each modality to use, at the creation of a new association. Also, it could be useful to activate confirmation events whenever an user requests a certain output action using an input level with a low trust level.

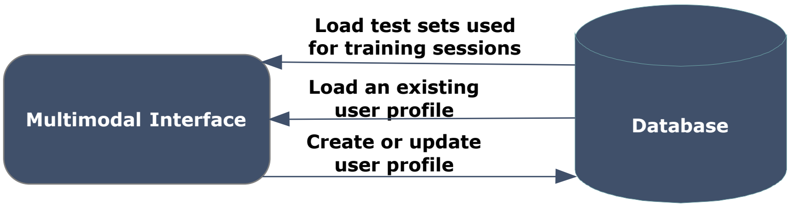


Figure 6.11: User profile Database

Graphical User Interface

Although the IMI is intended to be used by any person on variable contexts, one should remember that the main objective is to provide an easy, flexible and intuitive way of driving an IW running an instance of the IWP.

Consequently, a special effort should be put into the design of this component, since the end users may have severe disabilities and may extensively depend on the usability of the GUI.

Therefore, the following characteristics were proposed for the IMI GUI. These features are, in one way or another, related to some of the design principles that should be followed for multimodal interfaces, described in 4.3.

- Display the status of each input device (Connected/Disconnected);
- Display the current interface mode (Navigation, Sequence Creator, User Profile);
- Display visual cues for real-time sequence inputs;
- Show visual cues of the wheelchair trajectory;
- Show visual information of the training sets used for tracing the user profile.

Additionally, to improve the comfort and usability given by the IMI, another feature is proposed. It consists of an interactive navigation assistant character, offering advice and information regarding distinct events. This character should interact with the user by giving audio cues, using a TTS technology. At the same time, the cues should also be presented on the GUI, contained inside a textual area reserved for the effect.

The following list presents an enumeration of the cues, hints and advices that should be offered by the navigation assistant:

- Show the current output action being performed;
- Ask for confirmation of an output action request;
- Inform the user whenever an invalid sequence is given;
- Give information regarding control module and input device connection handlers;

- Propose the creation of a new association, for output actions without associations;
- Inform the user whenever an attempt to associate an existing input sequence is made;
- Inform the user of the desired inputs for the user profile training session.

6.4 Global Architecture

Considering all the proposed features, the overall architecture of the system is presented in Figure 6.12. This subsection gives an overview of all the system components roles.

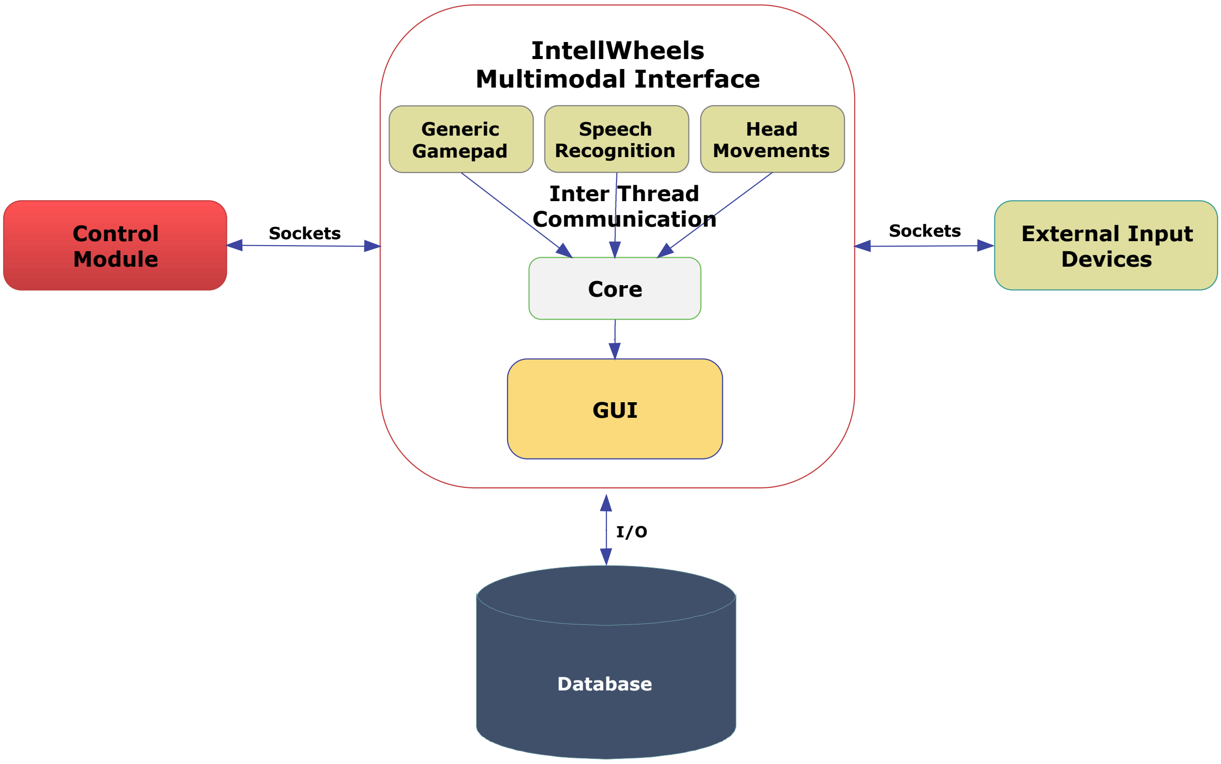


Figure 6.12: IntellWheels Multimodal Interface Global Architecture

The IMI is composed by five main components:

- **Generic Gamepad** - Represents the thread responsible for implementing a driver to access a generic gamepad. It handles the device connection, and the state of each button: pressed or released. It also reads the position of the two analog switches that compose a generic gamepad. Its task consists also of implementing a set of events used to notify the core component every time there is a change on the gamepad status, or a request by the component regarding analog positions is made.
- **Speech Recognition** - This component is responsible for accessing a microphone and handle the voice commands given by the user. It notifies the core component every time a known voice command is recognized.

- **Head Movements** - This is the component responsible for accessing the device that tracks the users' head movements. One of its functions is detecting specific head movements and notify the core component on every detection. Another function is to continuously send, upon request, the users' head relative position to the core component.
- **Core** - This is the main component of the IMI. One of its functions is to access the database to store and load all the necessary data. It is also responsible for receiving inputs of all the input devices available in the system (internal or external), and implement the algorithms responsible for matching input sequences with desired output actions. Furthermore, it manages the GUI state by sending it information regarding systems events. Finally, it implements the User Profile tracker.
- **GUI** - The GUI role is to make a graphical presentation of all the information that might be useful for the user.

The external components are:

- **Control Module** - This component interacts with the IMI by initially sending information concerning the actions it can perform. It is responsible for receiving IMI requests, and generate the desired system output.
- **External Input Devices** - This component is composed by a variable number of input devices that might connect to the IMI to extend its multimodality. Each external input device should have a similar behaviour compared to the internal input devices. The only difference is that they are not embedded in the IMI.
- **Database** - The database task is to store all the information regarding available actions, existing associations and inputs. It also stores the users profiles and the description of the test sets used by the User Profile tracker.

6.5 Summary

This chapter presented the proposed architecture for the IntellWheels Multimodal Interface. A description of the main desired use cases, features and requirements was given. It is important to refer that the design principles described in 4.3 were considered in the IntellWheels Multimodal Interface project. The next Chapter describes the actual system implementation.

Chapter 7

System Implementation

Using the specification of the previous chapter as guideline, a prototype of the Intell-Wheels Multimodal Interface was implemented. The development took advantage of Embarcadero Delphi 2010 [Emb]. It is an integrated software development environment that allows visual, event-oriented programming through Pascal programming language.

This chapter presents the most relevant implementation details. It includes the description of the basic input modalities that were implemented, the data representation structures that were used, and the interaction between all the system's components. Furthermore, the used approach to analyse and validate input sequences is also documented. Finally, the methodologies used to track the user's profile are also fully explained.

7.1 Basic Input Modalities

This section presents the three basic input modalities that were embedded to the IMI: Generic Gamepad, Speech Recognition and Head Movements. Besides the use of a generic gamepad, a head set with an incorporated microphone was used for the speech recognition module.

To implement the component responsible for detecting the head movements, we used a Nintendo Wii Remote [Nin].

Since the main objective of this experiment was to try to prove the concept proposed in the last chapter, low cost devices that could be easily programmable were used.

The following subsections present a more detailed description concerning the implementation of the IMI embedded input recognition modalities.

Generic Gamepad

USB Generic Gamepads are a bit more sophisticated compared to regular joysticks. This type of gamepad is widely used in video games once it offers a big number of buttons and analog axis that can be easily programmed to provided an attractive way of navigation.

Figure 7.1 shows an example of a generic gamepad that can be used with the IWI. These gamepads provide ten programmable buttons and two analog joysticks. Each analog switch contains two axis.

Figure 7.1: Generic Gamepad

Speech Recognition

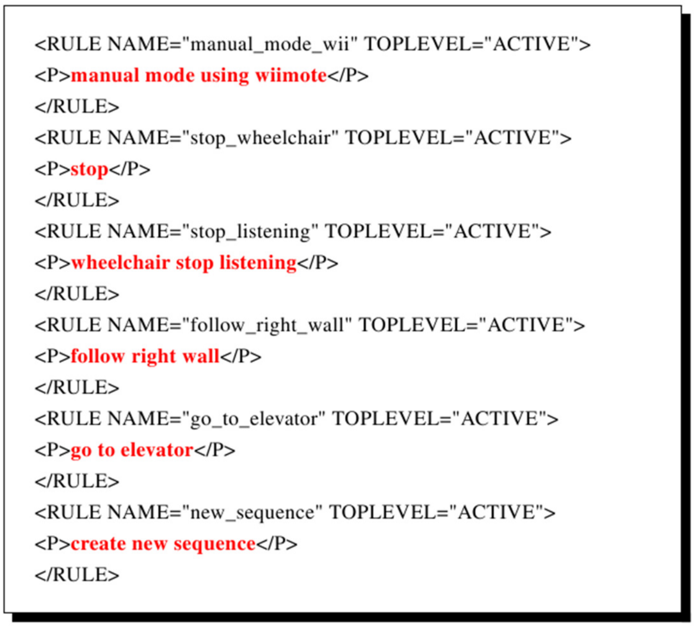
This feature uses a headset microphone (Figure 7.3) to capture the sound of the users speech, which is then interpreted by a speech recognition engine. The speech recognition module used in the IMI takes advantage of the Microsoft Speech Recognition Engine. This engine provides an *Application Programming Interface* (API), named *Speech Application Programming Interface* (SAPI) [SAP]. SAPI allows the use of speech recognition and speech synthesis within Windows applications.

SAPI was chosen since it is a freely redistributable component which can be shipped with any Windows application that wishes to use speech technology. Although many versions of the available speech recognition and speech synthesis engines are also freely redistributable, SAPI offers a standard set of interfaces accessible from a variety of programming languages.

Therefore, it was possible to implement this feature without the need of installing and accessing external software. Besides, it proved to be a very efficient choice since the accuracy of recognition was extremely satisfactory, even without the user having to previously train the voice before the first use.

One interesting feature of the Microsoft Windows SAPI is the possibility of designing grammar rules that can be used for specific contexts. This feature makes it possible to limit the set of recognition hypothesis to a custom set of voice commands, substantially increasing the recognition match accuracy.

To implement the desired functionality for the IMI, a context-free grammar (CFG) was used to store the possible voice commands to be transformed in input tokens. This grammar follows the SAPI grammar format, and is saved into a XML grammar file. Figure 7.2 shows an example of a small SAPI XML grammar:



```
<RULE NAME="manual_mode_wii" TOPLEVEL="ACTIVE">
<P>manual mode using wiimote</P>
</RULE>
<RULE NAME="stop_wheelchair" TOPLEVEL="ACTIVE">
<P>stop</P>
</RULE>
<RULE NAME="stop_listening" TOPLEVEL="ACTIVE">
<P>wheelchair stop listening</P>
</RULE>
<RULE NAME="follow_right_wall" TOPLEVEL="ACTIVE">
<P>follow right wall</P>
</RULE>
<RULE NAME="go_to_elevator" TOPLEVEL="ACTIVE">
<P>go to elevator</P>
</RULE>
<RULE NAME="new_sequence" TOPLEVEL="ACTIVE">
<P>create new sequence</P>
</RULE>
```

Figure 7.2: Example of SAPI grammar

The red coloured expressions represent the voice commands that may be spelled by the user, and recognized by the recognition engine. Limiting the voice commands to a pre-defined grammar showed to be very efficient. However, to extend the set of recognizable expressions it is necessary to manually add new entries to the XML grammar.

Figure 7.3: Headset with Microphone

Head Movements

The Wii Remote, sometimes unofficially nicknamed "Wiimote" (Figure 7.4), is the primary controller for Nintendo's Wii console. A main feature of the Wii Remote is its motion sensing capability, which allows the user to interact with a computer via gesture recognition through the use of a three-axis accelerometer [Nin].

In our approach, besides using the Wii Remote accelerometer values to directly control the wheelchair's trajectory, we implemented methods to detect four distinct head movements:

Figure 7.4: Nintendo Wii Remote

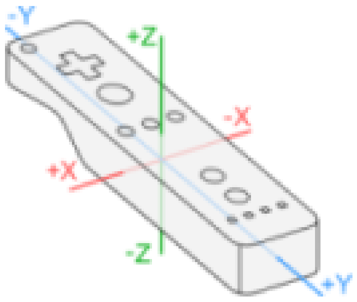


Figure 7.5: Wiimote's Axis

- leaning the head forward;
- leaning the head back;
- leaning the head right;
- leaning the head left.

Figure 7.6 presents an illustration of the head movements to be detected:

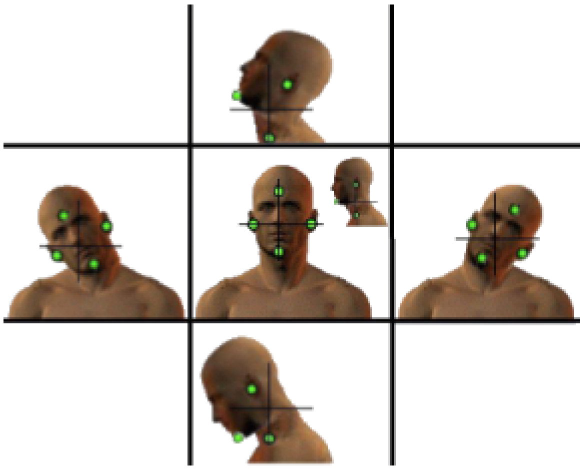


Figure 7.6: Head Movements

The approach is to define an amplitude value that starts a small timer. The timer is started if the user leans the head to an amplitude higher than the one that is defined. After the timer event is triggered, the head movement module checks if the head position (accelerometer value) is back to the original position. Both the amplitude and the interval between the triggered event are configurable.

These head movements can be used to compose input sequences, having a similar utility of a voice command or a button on the gamepad.

7.2 External Input Modalities

To accept inputs from external devices, a very simple protocol was implemented. The IMI acts as a server for receiving external connections. An interaction between an external device has three distinct phases. On the connection phase, the external application should identify itself by sending a command with its name.

This information is used to, via navigation assistant, notify the user that a new input device is available.

After this phase, the IMI is ready to receive inputs from a new input device. However, these inputs are only useful for defining input sequences and not to act as a parameter source, as the gamepad and wiimote analog switches and accelerometer.

On disconnect, the external application should notify the IMI, sending a disconnect command.

Table 7.1 shows the defined syntax for the possible events.

Event	**Command**	**Example**
On Connect	<CLIENT_ON identifier>	*<CLIENT_ON webcam>*
Send Input	device_id.input	*webcam.blink_left_eye*
On Disconnect	<CLIENT_OFF identifier>	*<CLIENT_OFF webcam>*

Table 7.1: External Devices Communication Protocol

7.3 Data Representation Structures

This sub-section presents the structures used to represent the output actions, input sequences and associations.

Actions

The output actions provided by the IWP control module, shown in 5.5, differ slightly in type and nature and can be divided into four types:

1. **High-level** - the trajectory of the wheelchair's movement is calculated solely by the control module of the platform (e.g., go forward, go back, turn right, follow right wall), using pre-defined values for the speed of the wheels.
2. **Medium-level** - identical to the *high level* type, whereas the user is responsible for setting the desired speed/angle for the movement.
3. **Manual-mode** - the user is responsible for controlling the wheelchair's trajectory. If the shared control option is activated, the control module of the platform may help only by detecting obstacles and avoiding collisions.

4. **Stop** - the wheelchair is stopped and enters a stand-by mode until further request.

To design a generic structure to represent them on the multimodal interface side, the following descriptors were used:

- **name** - The name of the action;
- **type** - The action type;
- **parameters** - The kind of parameter to be passed;
- **hint** - The hint text for the TTS navigation assistant.

Table 7.2 shows the relation between each action type and the number of parameters sent, which will generate a certain output.

Action type	**Sent parameters**	**Parameter type**
High-Level	0	n/a
Mid-Level	1	Linear Speed or Rotation Angle
Manual-Mode	2	Linear Speed and Angular Speed
Stop	0	n/a

Table 7.2: Action type parameters

The possible sources for extracting these parameters are the gamepad's analog switches and the wiimote's accelerometers, previously shown.

Depending on the parameter, one may need one or two axis to extract the desired output. For sending linear and angular speeds, the value of each axis is directly sent to the control module. A rotation angle is calculated based on the direction of the resulting vector. To represent the number of axis needed for each parameter type, the following descriptors were used: *analog* (1 axis) or *vector* (2 axis).

Figure 7.7 contains an illustration of how an action type and its parameter descriptor relates to the wheelchair's movement output.

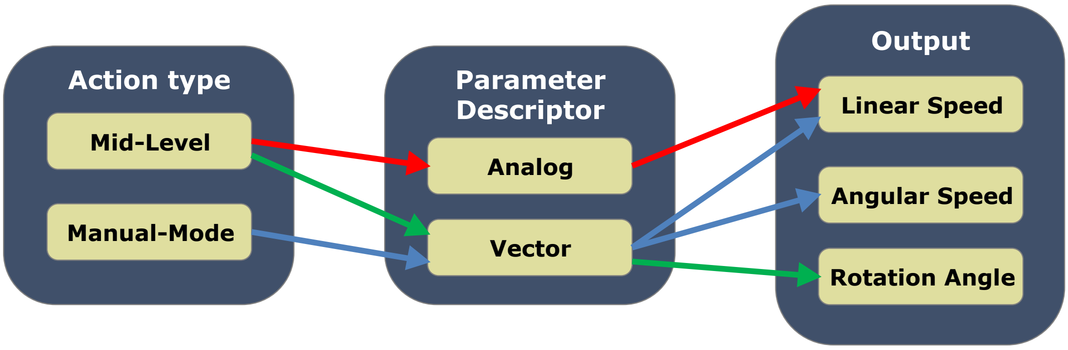


Figure 7.7: Mid-Level Rotation Angle Parameter

In the mid-level type, the parameter to be passed might be related to the wheelchair's linear speed (e.g. go forward at speed 30%) or to the rotation angle (e.g. spin 90^o), depending on the action nature. Since the linear speed requires only one axis, and the rotation angle requires two axis, the parameter descriptor for a mid-level may vary (*analog* or *vector*).

Figure 7.8 shows an example of how a rotation angle is calculated according to the two axis that are used. In the example shown, the angle would be calculated using the Wiimote, so the user would lean the head to express the desired rotation angle. For a rotation angle, the *vector* descriptor is used.

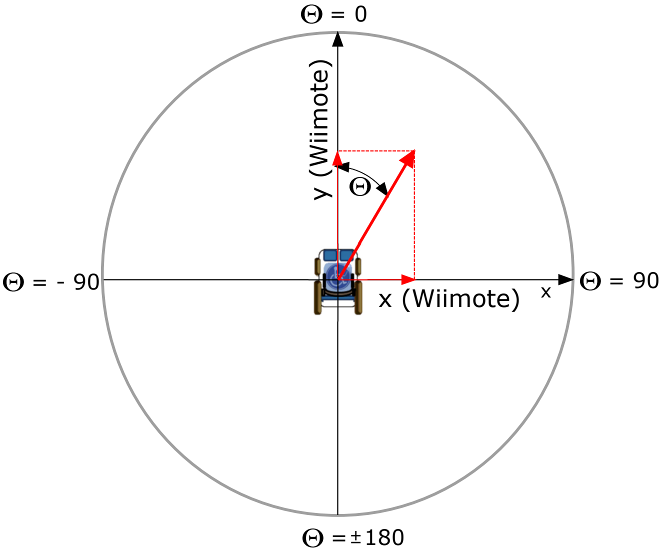


Figure 7.8: Mid-Level Rotation Angle Parameter

To extract a linear speed, the *analog* descriptor is used, since it is only necessary to read the value of one axis. For example, to configure a parameter for an action named "Go Forward at Desired Speed", the user could chose the gamepad's left analog switch 'y' axis to express the desired speed. When requesting this action, the user should press the analog switch to a certain position, in order to set the desired speed. Figure 7.9 illustrates the example.

In the manual-mode type the *vector* descriptor is used, since two axis are needed to directly control the wheelchair. Therefore, the linear and angular speeds are sent, since the wheelchair used in this experimental validation automatically calculates the speed of both wheels according to these parameters [Bra10].

The maximum value for both linear and angular speeds accepted by this wheelchair model is 100. To obtain a correct movement while controlling the wheelchair in manual mode, using the gamepad or the wiimote, it was necessary to parametrize the maximum values of each of both devices axis to 100.

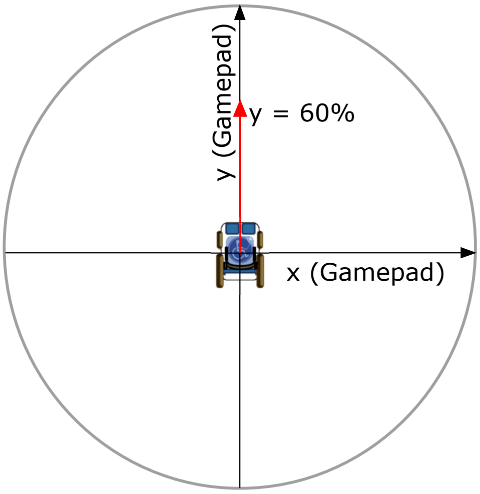


Figure 7.9: Mid-Level Linear Speed Parameter

Additionally, after some initial experiments, we noticed a very abrupt variation on the wheelchair's movement every time we tried to perform curvilinear trajectories. After some tests, we discovered that in order to obtain a smooth variation of the wheelchair's direction while curving, it is necessary to truncate the linear and angular speeds to a limited space. The limited space is the one contained on the circumference of radius 100.

To truncate the speeds, the algorithm depicted in Figure 7.10 was designed.

```
$x \leftarrow angular\_speed$
$y \leftarrow linear\_speed$
$hypothenuse \leftarrow \sqrt{x^2+y^2}$
if $hypothenuse > 100$ then
    $\theta \leftarrow \arctan(\frac{y}{x})$
    $x \leftarrow 100 \times \cos\theta$
    $y \leftarrow 100 \times \sin\theta$
end if
```

Figure 7.10: Linear and angular speeds truncation algorithm

This is a portion of the full algorithm, used to truncate the vector in the first quadrant. For the remaining quadrants, it is necessary to invert the sign of the linear speed, or the angular speed, depending on the quadrant.

Figure 7.11 shows a graphical explanation of this process.

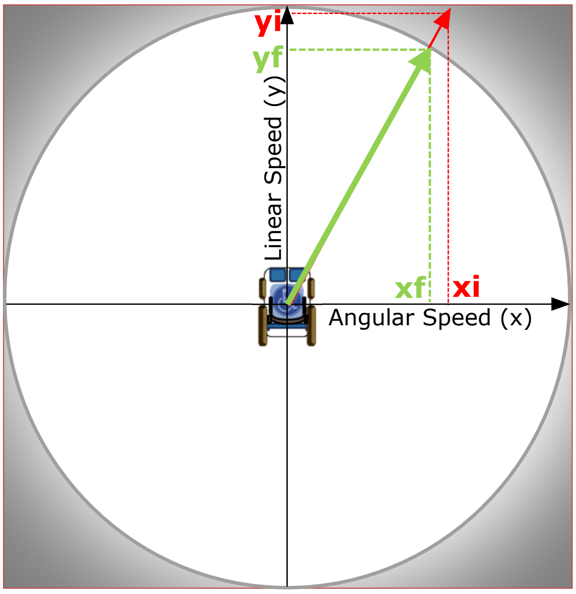


Figure 7.11: Truncation of linear and angular speeds

Input Sequences

An input sequence is composed by one more input tokens. An input token is formed by two parts: device descriptor and input descriptor, assuming the following format: *device_id.input_id*. This way, an input sequence must be formed by one or more input tokens.

Some examples of valid input sequences are present in Table 7.3.

Input Sequence	**Description**
<joystick.2 wiimote.right>	Pressing the button *2* of the gamepad, followed by leaning the head to the right
<microphone.turn_right webcam.open_mouth	Pronouncing the expression "Turn right", followed by opening the mouth>

Table 7.3: Examples of IMI input sequences

Associations

To represent an association between an action and an input sequence, we used the following structure:

- **inputs** - the sequence of input tokens that will trigger the action's request
- **action name** - the output action to request
- **parameters** - the source of the parameters

7.4 IMI Actions

Since the IMI offers a different set of features, it was important to think of a way to make them easily accessible by people with physical disabilities. Depending on the level of physical disability, controlling a mouse can represent an impossible task for many individuals. For example, to consult the list of available output actions or saved associations, it is necessary to perform two clicks on the interface. With this in mind, we decided to adopt the same style of interaction, not only to request output actions of a certain control module, but also for controlling actions related to the IMI. This way, we made it possible to associate an input sequence to a different type of action: interface action. The set of implemented interface actions, embedded in the IMI, can be consulted in Table 7.4.

Interface Action	**Description**
Mouse left	Click left mouse button
Double Mouse left	Double click left mouse button
Mouse right	Click right mouse button
View / Hide Output Actions	Open / Close the output actions viewer
View / Hide Associations	Open / Close the list of saved associations
View / Hide Device Manager	Open / Close the input devices' tabs panel
New sequence	Start the sequence creator mode
Save sequence	Associate sequence to action and save
Calibrate Wiimote	Reset the wiimote controler accelerometer values to 0
Enable/Disable Speech	Turn the speech recognition module on or off
Enable/Disable Assistant	Turn the TTS navigation assistant feature on or off
Mouse POV	Enable the control of the mouse using the gamepad's POV
Mouse Wiimote	Enable the control of the mouse using head movements
Minimize/Maximize	Show or hide the application
Profiler	Start the user profile tracker module

Table 7.4: List of IMI interface actions

In spite of being a different action type, the structure used to save it is the one used for the output actions. Only the *type* field changes.

7.5 XML Database

To store all the information related to output actions, input sequences and associations, we used a XML file. For it is a very simple and legible markup language, it proved to be a good choice to implement the database feature proposed in the last chapter. The XML Schema, i.e., the grammar used to describe each one of the information structures was based on the tags/descriptors previously shown in the last subsection.

Figure 7.12 contains an example of an IMI database, containing one output action, and one association.

```xml
<ACTIONS>
<item>
<name>Manual</name>
<type>manual-mode</type>
<parameters>vector</parameters>
<hint>Entering manual mode</hint>
</item>
</ACTIONS>
<ASSOCIATIONS>
<item>
<inputs>microphone.manual_mode_wiimote</inputs>
<action>Manual</action>
<parameters>wiimote.x wiimote.y</parameters>
</item>
</ASSOCIATIONS>
```

Figure 7.12: Example of IMI XML Database

7.6 Multimodal Interaction Loop

7.6.1 Control module vs IMI

To allow the interaction between the IMI and the IWP control module, the latter had to be adapted in order to implement the action structure described in 7.3. After the connection process, the control module must send the list of the available actions.

Additionally, it should also periodically send the availability of each action. This feature was implemented because in order to perform some actions, the control module may depend on external software or hardware, which might fail. Figures 7.13 and 7.14 show the global flow for both the IMI and the control module.

7.6.2 Sequence Analysis

The input sequence analysis represents the most important feature of the IMI. Each time an input token is perceived by an input device recognizer, a notification is sent to the core component, which is responsible for checking if the current input sequence matches any of the existing associations.

The sequence analyser implementation takes advantage of the method used to represent an input sequence. As it was explained before, an input token consists of the pair formed by the device descriptor and the input descriptor.

Our approach consists of keeping sorted all the input sequences that are associated to output actions. In this way, it is possible to apply a binary search every time a new input

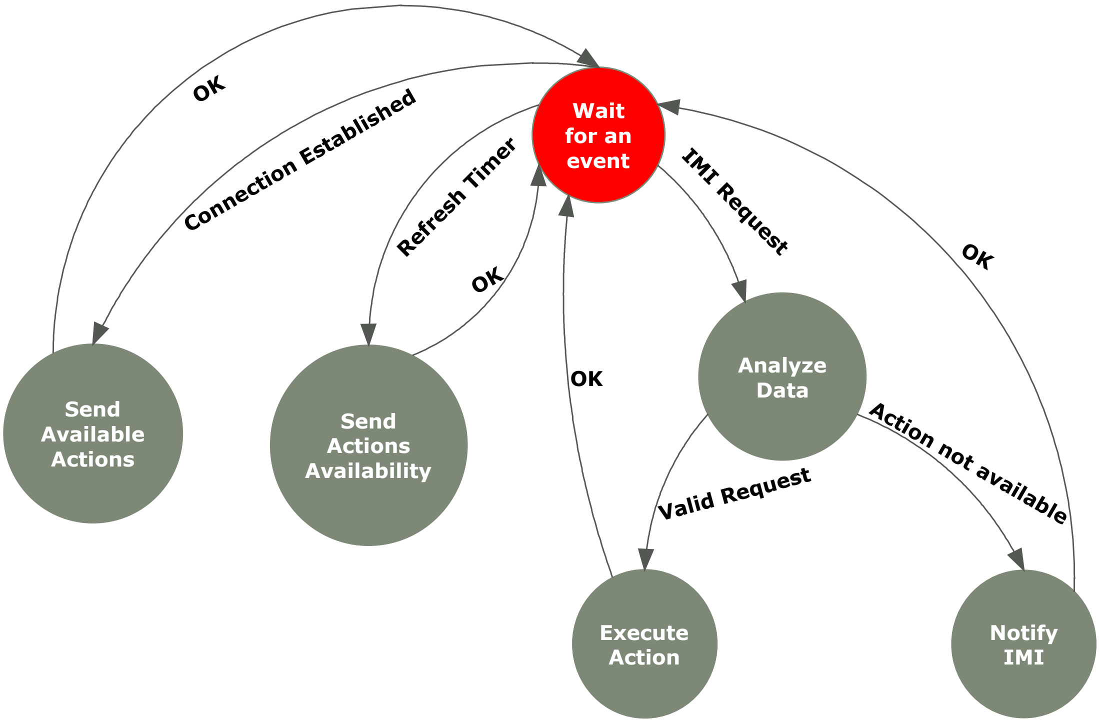


Figure 7.13: Control module data flow

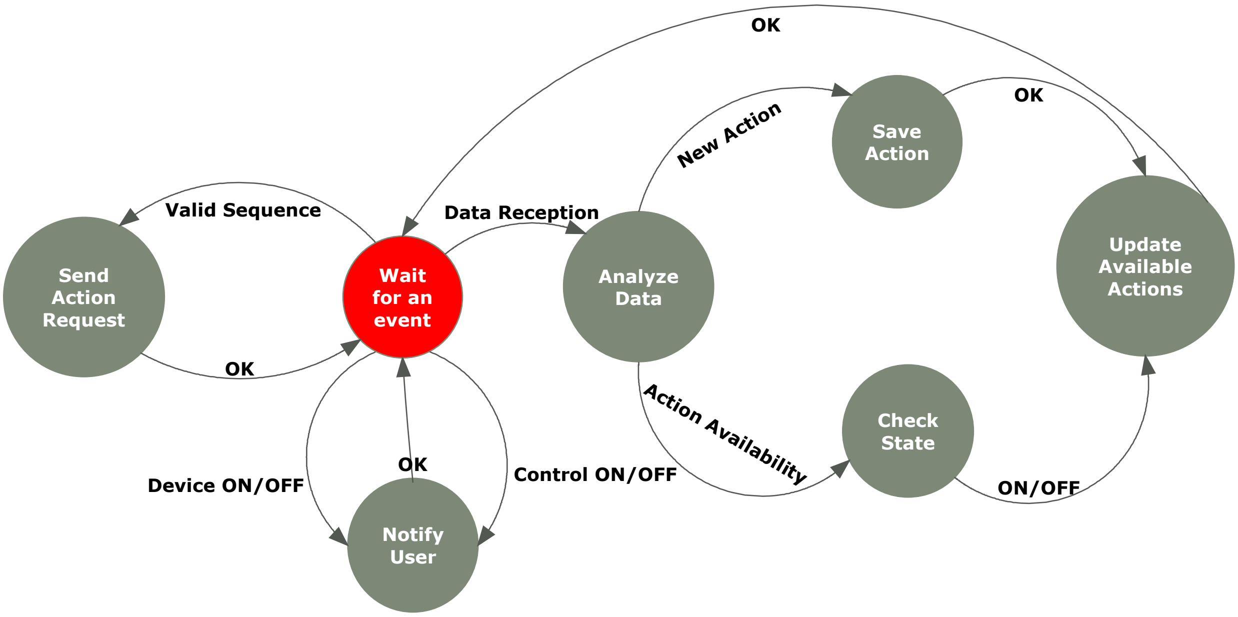


Figure 7.14: IMI global flow

token is received. The Binary Search algorithm [Knu98] has a fast performance and it is ideal to this situation, since an input sequence is sequentially growing. Therefore, it is necessary to constantly compare the current input sequence with fragments of the stored associations. If at any time, the user's input sequence does not match any fragment of the same size, the sequence is immediately discarded, and the user is notified.

Inversely, if the current input sequence is a fragment of one or more associations, a timer is activated, in order to wait for further inputs that may transform the current input sequence into a valid match.

When a match happens, it is necessary to verify if there is any other association composed by the current input sequence and one or more input tokens. In this situation, a different timer is activated. Meanwhile, the IMI waits for further inputs. If no input token has been given by the user when the timer event is triggered, the associated output action is requested to the control module. Otherwise, the timer is turned off and the process takes its normal flow.

Finally, if there is a perfect match, which means that the current input sequence is mapped to an output action, and there is not any other input sequence partially equal, the associated output action is immediately requested.

Figure 7.15 illustrates the flow of the implemented version of Binary Search algorithm.

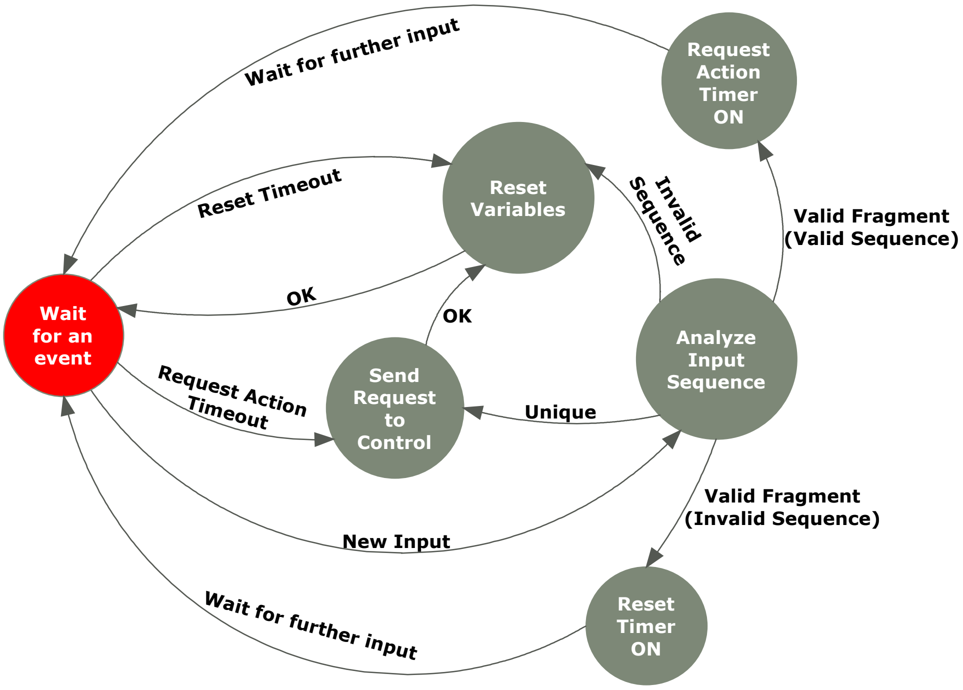


Figure 7.15: Sequence Analysis Flow

7.6.3 Sequence Creator

In order to enable the creation of a new input sequence, and consequent association to an output action, the interface has switch to a different flow mode. When this process is started, the navigation mode is temporarily halted, by turning off the sequence analysis. Instead, the input sequence is sequentially updated and kept until the user decides to associate it with a desired output action, or, alternatively, cancel the creation of a new association. At any stage of this process, the user may exit the sequence manager option and return to the navigation mode.

Figure 7.16 shows the IMI flow for the association of a new input sequence to a desired output action.

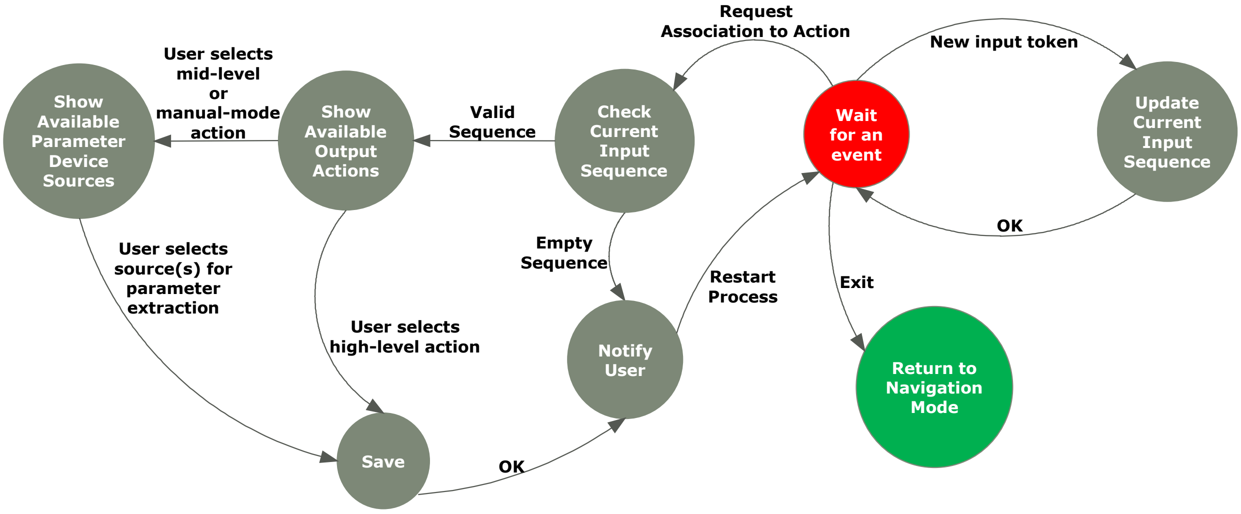


Figure 7.16: Sequence Creator Flow

7.7 User Profile

This subsection presents the attempt made to implement the proposed User Profile feature presented in Chapter 6.

Firstly, we explain the approach followed to define the format of the test sets to be loaded by the module responsible for tracking the user's profile. Secondly, we show the simple profiling methods that were implemented to create an user classification. Following, it is explained how the extracted information was used to adjust certain settings of the interface. Finally, a demonstration of how the profile is stored to enable future use is also made.

7.7.1 Definition of test sets

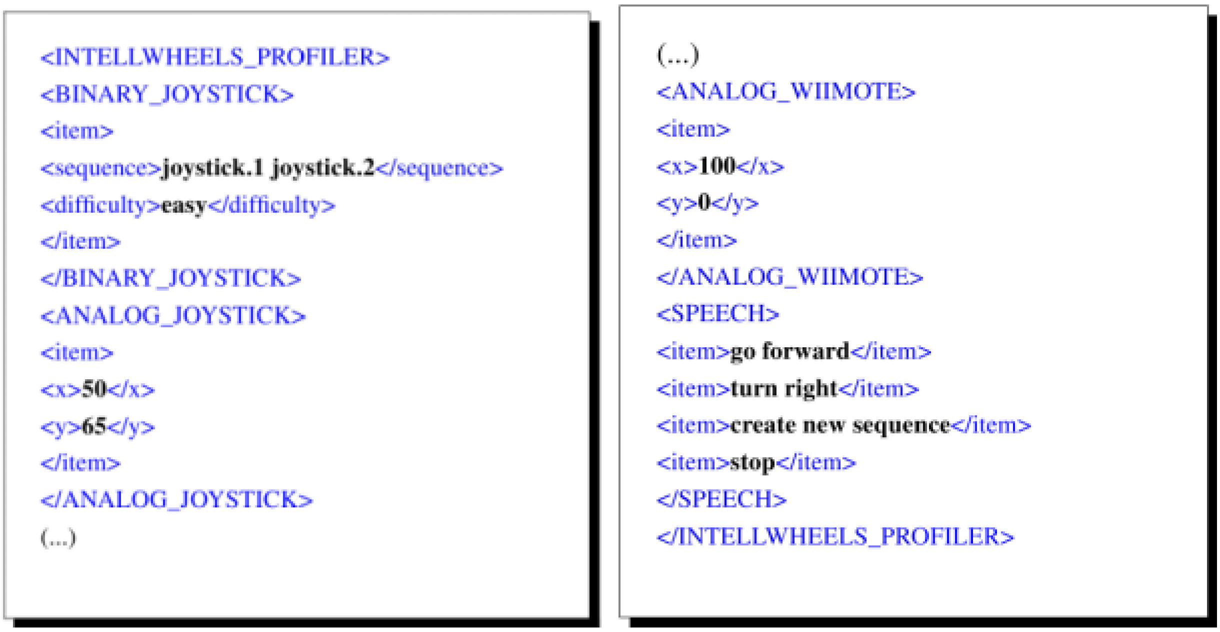
In addition to the wide variety of disabilities that cause different types of limited mobility, each person has specific characteristics. Thus, individuals with similar symptomatology, may have significant differences in either the physical or the cognitive level.

It is fair to say that characteristics such as moving your head, ability to pronounce words, ability to move the hand and fingers, can vary substantially from individual to individual. Similarly, the time of learning and proficiency in using assistive devices may also vary greatly.

Tracing an user diagnostic can be very useful to adjust certain settings allowing for an optimized configuration and improved interaction between the user and the multimodal interface.

To perform the measures described in 6, a simple XML grammar was defined. It implements four configurable distinct test types:

1. Sequences of gamepad buttons;
2. Voice commands;
3. Positions for both joysticks;
4. Positions for head.



```
<INTELLWHEELS_PROFILER>
<BINARY_JOYSTICK>
<item>
<sequence>joystick.1 joystick.2</sequence>
<difficulty>easy</difficulty>
</item>
</BINARY_JOYSTICK>
<ANALOG_JOYSTICK>
<item>
<x>50</x>
<y>65</y>
</item>
</ANALOG_JOYSTICK>
(...)
```

```
(...)
<ANALOG_WIIMOTE>
<item>
<x>100</x>
<y>0</y>
</item>
</ANALOG_WIIMOTE>
<SPEECH>
<item>go forward</item>
<item>turn right</item>
<item>create new sequence</item>
<item>stop</item>
</SPEECH>
</INTELLWHEELS_PROFILER>
```

Figure 7.17: Example of XML containing user profiler test set

The proposed XML grammar makes it possible for an external operator to configure a test set that they may find appropriate for a specific context or user. When an user starts the training session, the four different types of tests are iterated. In order to attain a consistent classification of the user, the defined grammar should be sufficiently extensive.

To define the user proficiency in using the gamepad buttons, a very simple method was implemented. Each sequence defined on the grammar should have an associated difficulty level (easy, medium or hard). The difficulty type of a sequence may be related to its size, and to the physical distance between the buttons on the gamepad. Since the layout of a generic gamepad may change depending on the model, defining whether or not a sequence is of easy, medium or difficulty level is left to the operator.

When the user completes the gamepad sequences training part, an error rate is calculated for each of the difficulty levels. If these rates are higher than a minimum acceptable configurable value, the user classification is defined. This classification is then used to turn on the security feature, which is characterized by a confirmation event performed by the navigation assistant. An example follows:

Difficulty	**Max. Accepted Error Rate**
Easy	20%
Medium	30%
Hard	50%

Table 7.5: User classification

For a grammar with 5 sequences of difficulty type *easy*, the maximum number of accepted errors would be 1 (1/5 = 20%). If the user fails more that one sequence, the confirmation event is triggered for any input sequence, of any difficulty type, and the gamepad training session is terminated. If the error rate for the *easy* type is less than *20%*, the training with the sub-set composed by the sequences of *medium* difficulty is initiated. At the end, a similar method is applied. If the error rate for the medium level is higher than *30%*, the confirmation is triggered for the *medium* and *hard* levels of difficulty, and the training session is terminated. Finally, if the user makes it to the last level of difficulty, the training for the *hard* sequences sub-set is started. If the error rate is higher than *50%*, the confirmation event is triggered only for sequences with a *hard* difficulty level. The best scenario takes place when the user is able to surpass the maximum accepted error rates for all the difficulty levels. In this situation, the confirmation event is turned off, and an output request is immediately triggered for any kind of input sequence composed only by gamepad buttons.

Defining the ideal maximum acceptable error rates is not easy. With this in mind, we made it possible to also configure these values in the XML grammar.

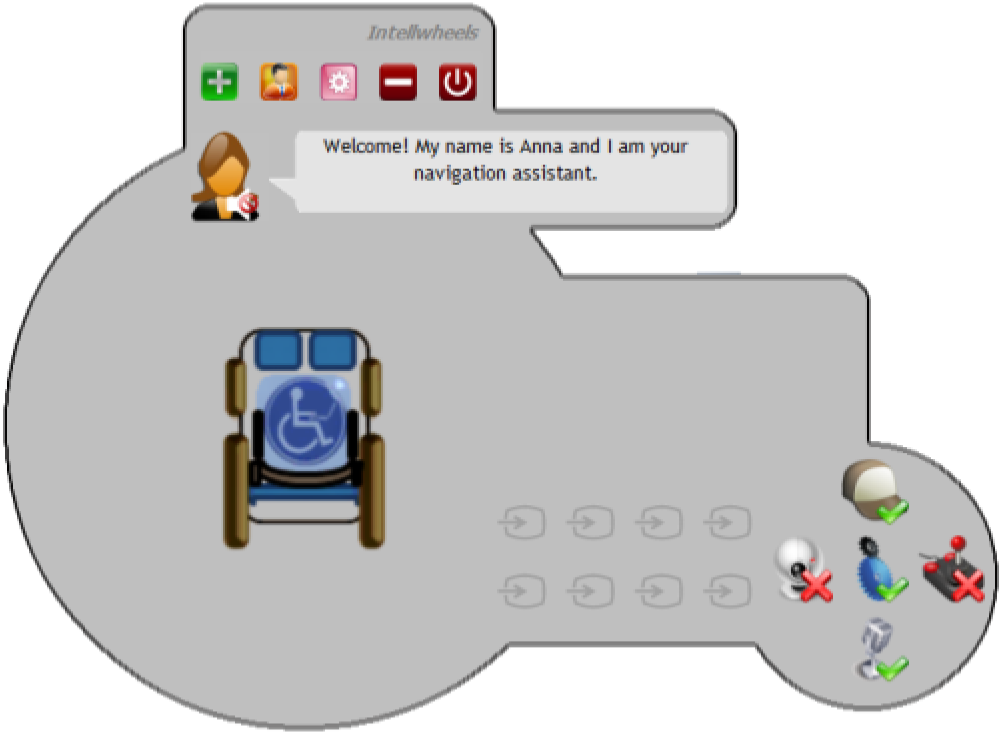
The joystick phase of the training session can be used to calculate the maximum amplitude achieved by the user. This value can then used to parametrize the maximum speed value. Imagining an user that can only push the joystick to 50% of maximum amplitude, the speed can be calculated by multiplying the axis value by two. This feature was not implemented. However, all the background preparation to implement it was set.

The speech component of the training session was used to define the average recognition trust level. This value is used to set the minimum average recognition level for the recognition module. An improvement of this feature can be done by setting a minimum average recognition level for each particular voice command.

Finally, the head movement phase of the training session has a similar purpose to the joystick's phase. Additionally, the maximum amplitude for each direction can be used to determine the range that will trigger each one of the leaning inputs, described at the beginning of this chapter.

7.8 Graphical User Interface

Figure 7.18 shows the developed graphical user interface for the IMI.



Welcome! My name is Anna and I am your navigation assistant.

Figure 7.18: IMI Graphical User Interface

We tried to design a simple graphical user interface, that could attractively present all the necessary information to the user.

When the application is launched, the navigation assistant, presented at the top of the GUI, starts by greeting the user. The name Anna was chosen since to implement the TTS feature, we used the Microsoft Anna Text to Speech Voice. Anna is responsible of interacting with the user by synthesizing the text presented in the talking balloon that stands on its right side. The text on the balloon changes every time a new event is triggered, in order to keep the user informed of the system's changes. The TTS feature can be switched on or off, either by clicking on the image chosen to represent the navigation assistant, or by associating an input sequence to the interface action responsible for the same setting.

On the top of the interface there is a set of five buttons, which implement the features related with the sequence manager, user profiler and device manager. The two buttons on the rightmost side are used to, respectively, minimize or close the application.

To keep the user informed of the availability of each one of the input devices, a set of five icons was placed at the right side of the interface. Every time an input device connects or disconnects from the multimodal interface, its icon changes. Since part of the future work for the IMI is implementing a facial expression recognizer, an icon to represent an webcam was also prepared. Additionally, an icon to represent the control module is also present. In the example presented in Figure 7.18, the wiimote controller, microphone and control module were connected to the multimodal interface. Inversely, the gamepad and the external input application concerning the facial expressions recognizer were not available.

Other information provided by the IMI relates to the wheelchair's trajectory. This is done by presenting a set of directional arrows that surround the wheelchair image placed on the left center of the interface.

Finally, the small set of eight icons at the center of the interface is reserved for the information regarding the input sequences. Every time the user gives a new input token, the respective image for that input token is shown.

A more detailed description of the IMI graphical user interface can be found in Appendix B. It presents some examples of different use cases of the IMI, and the way the information is graphically presented for each one.

7.9 Summary

This chapter explained some relevant implementation details of the IMI developed prototype. All the main features proposed in Chapter 6 have been implemented. Although the module responsible for tracking the user profile has been partially implemented, there are still some features that have to be implemented. It is expected that it might considerably increase the efficiency of the IMI. Being an interesting feature to explore, it represents part of the future work to be developed. Appendix B contains a description of other features, showing some illustrated examples of the developed application.

Chapter 8

Tests and Results

This chapter presents an evaluation of the results achieved by the system specification and by the implemented prototype. In order to test the integration of the multimodal interface with the previously developed IntellWheels Platform, two different test scenarios were prepared. The first test purpose was to evaluate the performance of the multimodal interface using the IWP simulator. Another test scenario was prepared, this time using the IntellWheels wheelchair prototype on a real environment. Other test consisted of using the IMI on a different context.

8.1 Experiment in a Simulated Environment

The first experiment involved students of the degree in Physiotherapy in the School of Health Technology at Porto (Escola Superior de Tecnologia da Saúde do Porto).

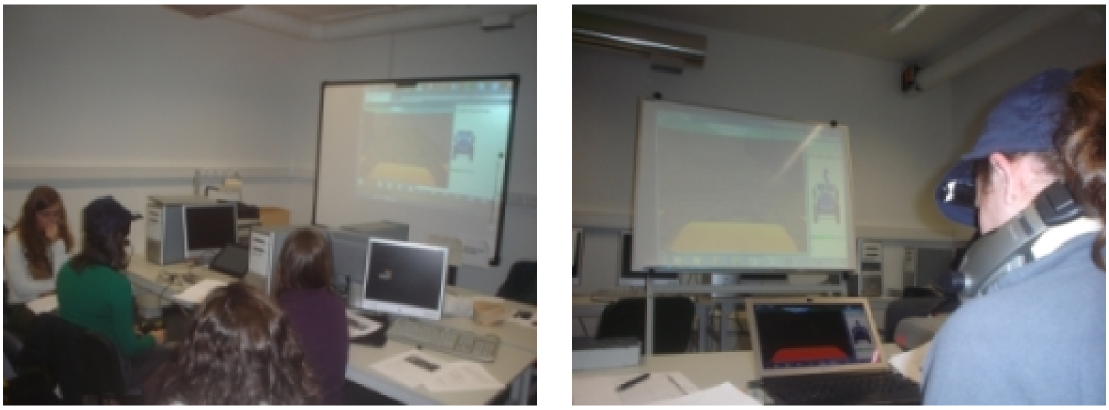


Figure 8.1: Students at ESTSP trying IMI with IWP simulator

There, a simulated scenario of part of the school was recreated, and a specific route was traced (using the IWP simulator). Figure 8.2 illustrates the *2D* representation of the route. The results evaluation was based on empirical research. In fact, this kind of analysis is very important since this type of population is more sensitive, due to their background, to the questions regarding Intelligent Wheelchairs' usability. Their feedback could be of great value to improve not only the already developed work, but also for proposing improvements and future features that might better assist disabled individuals.

The main goal of this pilot test was to attain the performance and efficiency of the developed basic input modalities, and the overall behaviour of the IMI. The intention was to list the existence of possible bugs, faults and inconsistencies of the IMI, based on the students' feedback.

The methodology applied was the gathering of opinions using a questionnaire that incorporates the System Usability Scale (SUS) [Bro96] which is a simple ten-item *Likert* scale [Mar07] giving a global view of individual assessments of usability [Bro96]. The questionnaire was divided into 4 parts (Available in Appendix A):

1. **User identification**: Several questions about the gender, weight, height, level of education and experience of using video games and appropriate commands;
2. **Usability and Safety**: It was applied the SUS, and questions related with safety and control;
3. **Controls of the IW**: It was asked the level of satisfaction with the different existing commands of the IW, such as joystick in manual and high level mode, voice commands and head movements, as well as the integration of all kind of modalities;
4. **Multimodal Interface**: It was asked the level of satisfaction about the information provided by the Multimodal Interface.

The type of study can be classified as quasi-experimental design in which the kind of sampling used was deterministic.

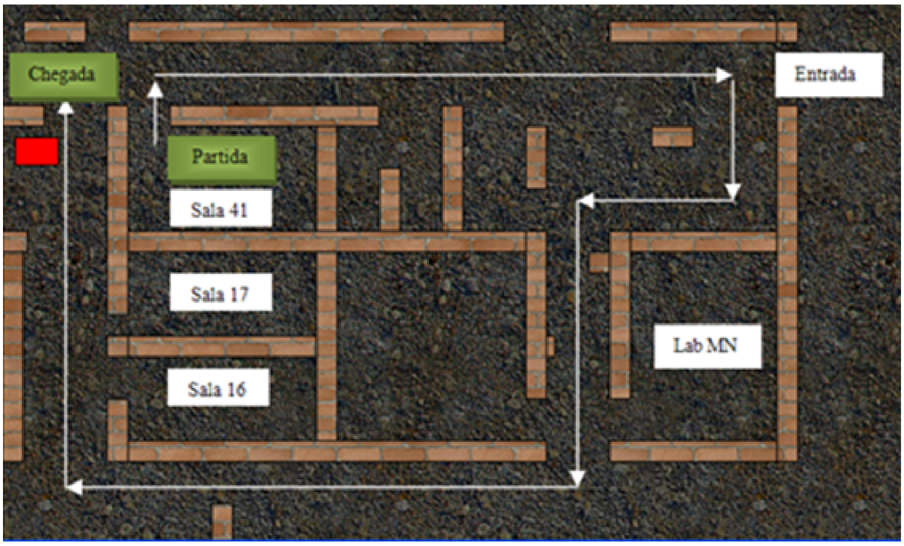


Figure 8.2: Map of the prepared route for the simulated experience

To take this route, several different scenarios were defined:

1. Driving the IW using the gamepad joystick in manual mode;
2. Driving the IW using the gamepad buttons in high-level mode;
3. Driving the IW with head movements (Wii controller) – With/without cervical collar;
4. Driving the IW with voice commands;
5. Driving the IW having the freedom to chose any type of input (gamepad, voice, head movements).

Since none of the students had any type of previous contact with the IntellWheels project, the first step was to provide an explanation of the main characteristics of the IMI, the type of output actions provided by the IWP control module, and the global goals of the experiment. Also, we had to explain the set of different input sequences that were prepared for this experiment. Table 8.1 contains the defined input sequences. To simplify the reading of the table, the following abbreviations are used:

- **GB** - Gamepad Button;
- **WB** - Wiimote Button;
- **GJ** - Gamepad joystick;
- **WM** - Wiimote head movements.

Output Action	Button Input Sequence	Voice Commands
Go Forward	GB 1	"Go Forward"
Go Back	GB 3	"Go Back"
Right Turn	GB 2	"Turn Right"
Left Turn	GB 4	"Turn Left"
Right Spin	GB 6	"Right Spin"
Left Spin	GB 5	"Left Spin"
Stop	GB 7 or GB 8	"STOP"
Manual mode using GJ	GB 9	"Manual mode joystick"
Manual mode using WM	WB 'A'	"Manual mode wiimote"

Table 8.1: Input Sequences used for the experiment using the simulated IW

The dimension of the first experience sample is 46 individuals, with a mean age of 21 years and standard deviation of 2.2. The sample is also characterized by having a larger number of female participants: 42 females for only 4 males. All of the individuals are students of the 3^{rd} year of Physiotherapy at ESTSP.

It was important to know beforehand the experience of the students using video games and gamepads, since the IW multimodal interface offers this type of control devices. The results can be observed in Figure 8.3.

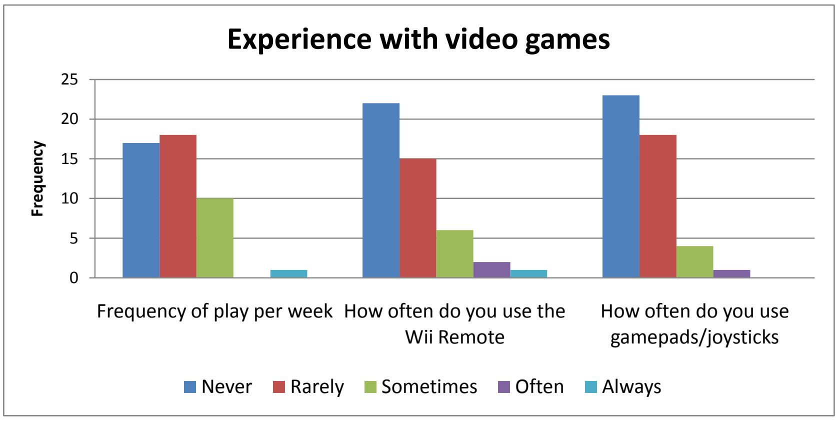


Figure 8.3: Experience with video games

Most of the answers focus on *Never* or *Rarely*, showing the students' lack of experience with this kind of devices.

In terms of usability the distribution of answers to the *SUS* can be consulted in Figure 8.4.

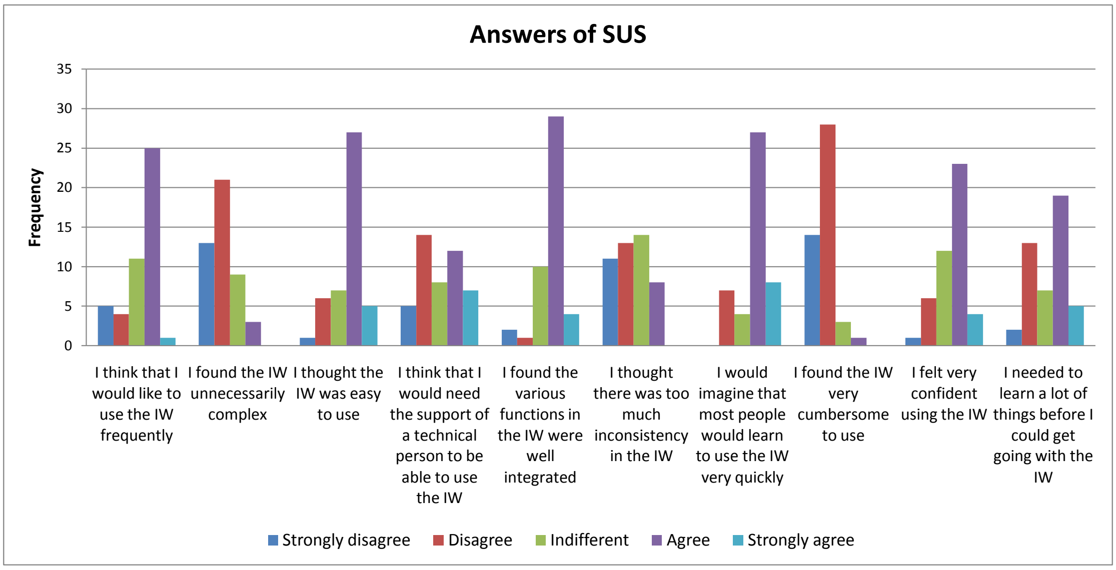


Figure 8.4: Distribution of answers to the SUS in the simulated experience

The SUS scale score is a measure that can be easily interpreted. The value varies from 0 to 100 and if the value is near 100, it means that the individual considered the system extremely effective, efficient and satisfactory. In this experiment the result of the total score has a mean of 63.3 and a standard deviation of 15.5. The median is 66.3, the mode 75.0, the minimum is 20.0 and the maximum is 87.5.

Some questions about safety and control were also made. The results for this experience can be observed in Figure 8.5.

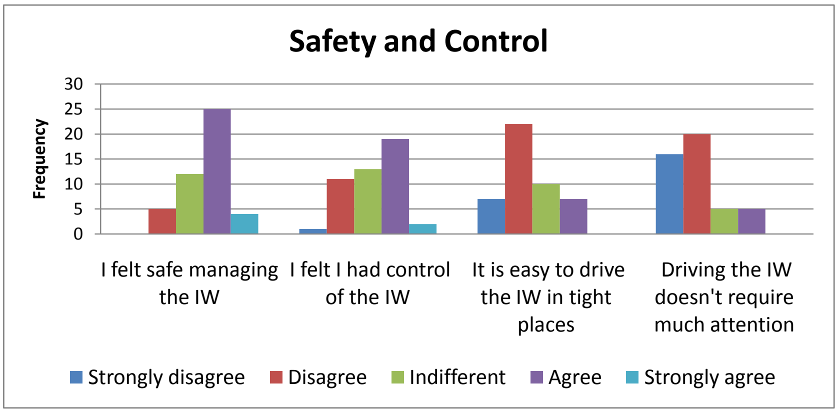


Figure 8.5: Safety and control results

The performance achieved by using the different modes for driving the IW were assessed by the time, number of collisions and total average deviation error from the given "ideal" trajectory relatively to the desired trajectory. Tables 8.2, 8.3 and 8.4 contain the dimension *n* of users that performed the experience in different conditions, the measures of central tendency (mean and median) and dispersion (standard deviation, minimum and maximum) of the experience using the simulator.

Table 8.2 refers to the time spent in order to complete the circuit. The time was measured using a chronometer. There was a total of 27 students that made the experiment using all the simple IW driving modalities: joystick in manual mode, gamepad in high-level mode, head movements and voice commands. However, only a total of 16 students used the IW having the freedom to use any of the available commands. Three students had some deviations from the original route and got lost in the circuit. So, their results had to be excluded from the statistical analysis.

Driving Mode Summary of Statistical Measures	Joystick in manual mode	Gamepad in high-level mode	Head movements	Voice commands	Freedom to drive with all commands
n	27	27	27	27	16
Mean	3,88	4,07	3,80	6,70	4,07
Minimum	2,42	2,88	2,32	3,08	3,12
Maximum	6,93	11,65	8,35	12,87	6,28

Table 8.2: Number of elapsed minutes to complete the circuit, using distinct control modes

As can be observed in Table 8.2, users which performed the experiment with voice commands spent, in average, an higher time (6.70 min) to complete a tour, compared to other command modes. The smallest time average was achieved using head movements (3.80 min). However, the smallest for the median was achieved using the gamepad with high-level commands (3.27). This last command mode presents the highest dispersion with a standard deviation of 1.29 min, a minimum of 2.88 and a maximum of 11.65 min. It seems to be a very effective mode for driving the IW for users that are able to use it correctly but is also a very ineffective mode for a small number of users.

The performance in terms of time having the freedom to drive with all the commands did present values near to others methods of manoeuvring the IW. The standard deviation in this case was the smallest with a value of 0.97. Also the maximum achieved for this mode has the smallest of all.

Another performance measure taken from the experiment was the number of collisions for each driving mode, for each one of the participants. The values of central tendency and dispersion can be observed in Table 8.3.

Driving Mode Summary of Statistical Measures	Joystick in manual mode	Gamepad in high-level mode	Head movements	Voice commands	Freedom to drive with all commands
n	27	27	27	27	16
Mean	10,86	3,33	6,67	31,83	9,07
Minimum	2	0	0	4	0
Maximum	33	8	22	73	38

Table 8.3: Number of collisions achieved for completing the circuit, using distinct command modes

Analysing the best performance in terms of collisions, the gamepad in high-level mode clearly presented the best average (3.33 collisions) and the voice commands also presented the worst average of collisions (31.83) confirming the worst result achieved for the time measure using this command mode.

Table 8.4 contains the values of the average error of deviation from the "ideal" trajectory which was asked to students to follow (Figure 8.2). The average error of deviation was calculated by the average distance from the point of the real trajectory to the "ideal" trajectory. The distance from this trajectory was given using a simple algorithm to calculate the distance from a point to a line segment [Alg10]. Figure 8.6 shows a graphical example of this process.

The average error of deviation from the trajectory does not present major differences between the different modes for driving the IW. However, it is important to remark the measures of dispersion in the case in which the students had the freedom to choose any

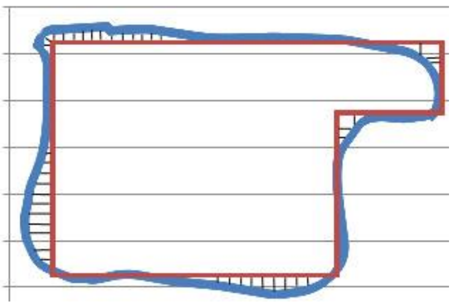


Figure 8.6: Calculation of total deviation from the ideal trajectory

type of command, since the standard deviation has the highest values and the difference between the minimum and maximum are also the highest. This reveals that in this mode, the individuals were more distant from the "ideal" trajectory, comparing to the other command modes.

Considering the values presented in Table 8.2, it is possible to affirm that there is, in terms of elapsed time, a similar efficiency between the first three driving modalities (joystick in manual mode, gamepad buttons using high-level commands, and head movements to directly control the wheelchair). The fourth option (voice commands) proved to be more difficult for most of the students. The reasons behind these results can be justified by different motives, and were based on observation.

Interestingly, as opposed to what was initially expected, the effectiveness of the voice recognition module was not the main factor. Although the tone of voice and pronunciation accent of some students influenced the recognition accuracy, the overall successful recognition percentage was very high. This was a good achievement since none of the participants performed the speech recognition training session.

It was common to see attendants pronouncing the desired voice command whenever they wanted a certain output action to be triggered. However, using speech recognition is a bit different compared to other interaction modalities, since there is a small delay caused by the pronunciation time. The trick to achieve a good performance consisted of

Driving Mode Summary of Statistical Measures	Joystick in manual mode	Gamepad in high-level mode	Head movements	Voice commands	Freedom to drive with all commands
n	27	27	27	27	16
Mean	0,23	0,24	0,32	0,32	0,29
Minimum	0,15	0,15	0,16	0,18	0,16
Maximum	0,28	0,38	0,53	0,42	0,82

Table 8.4: Average deviation error from the ideal trajectory using distinct command modes

pronouncing the voice commands a bit earlier for the action to be performed at the right time. Also, we noticed that some students did not feel very confident at pronouncing voice commands in the English language. The eventual lack of conviction and loudness increased the number of mishaps while controlling the wheelchair with voice commands.

Consulting Table 8.3, one may easily confirm the disparity between the number of collisions for each modality. This was mainly due to the difficulty of controlling the wheelchair in narrow places of the route. Sometimes it was necessary to combine several voice commands to perform a single curve. A single failed attempt to perform a certain action would easily originate a collision. Figures 8.7, 8.8, 8.9 and 8.10 show the trajectories obtained by one of the students for each lap, using the different driving modalities.

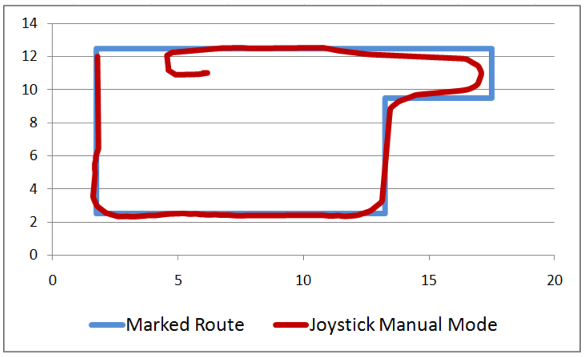


Figure 8.7: Trajectory using the joystick

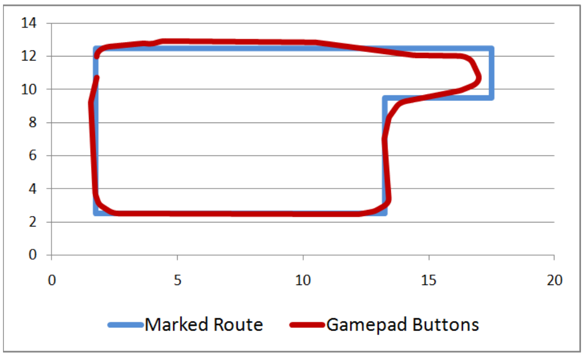


Figure 8.8: Trajectory using the gamepad buttons

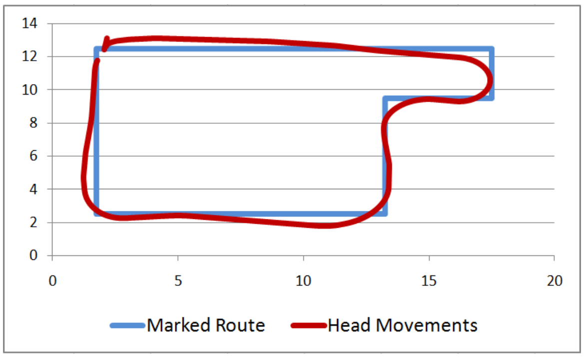


Figure 8.9: Trajectory using head movements

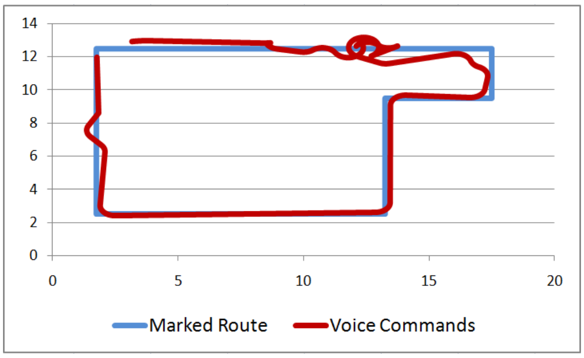


Figure 8.10: Trajectory using voice commands

As can be seen, there is a certain consistency in the trajectories obtained for the first three modalities. In this particular case, the attendant had some difficulties to control the wheelchair at a certain stage of the route, using voice commands. In spite of the different performances and trajectories obtained by each of the students, this example illustrates the global tendency.

Considering the students' feedback, we decided to make some minor adjustments to the IMI. The first is related to the head movements module. In this experience, in order to increase the speed of the wheelchair, one should lean the head to the front. Depending on the initial inclination of the head, the visibility could be reduced by the cap's flap. With this in mind, we inverted the linear speed axis for this module. Also, we attenuated the maximum curving speeds for the joystick and head movement control (manual-mode), by implementing the algorithm explained in Chapter 7.

8.2 Experiment in a Real Environment

The second experiment took place at Faculdade de Engenharia da Universidade do Porto. The goal was to evaluate the performance of the IMI using a real wheelchair. This experiment included the participation of 12 volunteers.

Figure 8.11: Volunteers at FEUP trying IMI with real IW

There were three individuals that participated in both experiences and the total score of SUS increased, except in one case. In fact, the result of the experience was slightly conditioned by a flaw on the wheelchair's firmware. Depending on the weight of the volunteer, issues related with the motors' power caused the wheelchair to turn off sometimes. This prevented the majority of volunteers from taking the established route uninterruptedly, since every time the wheelchair stopped it was necessary to quickly restart it.

The safety and control questions for the experience with the real IW can be observed in Figure 8.12.

From these results it is very interesting to conclude that the results were very similar for the simulated and real environments. Applying the Mann-Whitney test to compare the distribution of the answers about safety and control, it is possible to conclude that there are not statistical evidences to affirm that the distributions are not identical. In fact, the p values are all clearly higher than the significance level of 0.05. Table 8.5 contains the p values for each of the safety and control questions.

The next step was to check if there were differences between the results of usability in the experience with the simulated IW and with the real IW. To answer this question, the normality test Kolmogorov-Smirnov was applied, and the p value (0.162 in the case of the

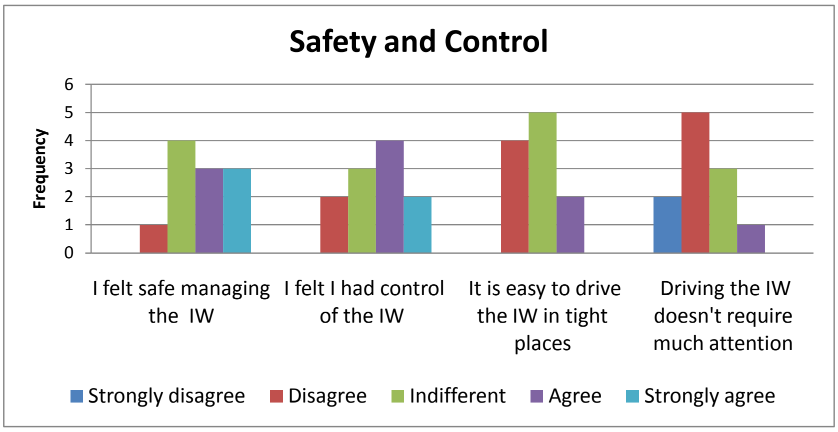


Figure 8.12: Safety and control results for the real IW

group that used the real IW) was bigger than the level of significance (0.05). Therefore, the normality of data can be assumed. Consequently, the suitable test to perform this analysis was the two independent sample *t* test (in order to check if there were significant differences in the means in those two cases. The *p* value of Levene's test for equality of variances [Lev] was 0.537 (higher than the level of significance). For that reason, the variances can be assumed equal.

Finally, the *p* value of the two independent samples *t* test was 0.007 (less than the significance level of 0.05). This allows us to conclude that there are statistical evidences to claim that the mean of the total score of SUS is different in the experiment with the real IW. This way, we may claim that the mean of the total score of SUS is higher in the experiment with the real IW compared to the experiment using the simulated IW.

Other interesting results relate to the number of collisions that were observed in both experiments. In the simulated environment, this number was considerably higher compared to the experience using the real Intelligent Wheelchair. This is related with the nature of each experience. We noticed a stronger effort was put by the participants in order to avoid collisions. Also, the minor adjustments made to the IMI also allowed for a better control of the wheelchair.

However, once again, controlling the wheelchair using voice commands represented the most difficult part of the challenge. After this experiment, we conclude that the length of the voice commands should be as short as possible. For example, instead of having

Question	P Value
I felt safe managing the IW	0.723
I felt I had control of the IW	0.409
It is easy to drive the IW in tight places	0.107
Driving the IW doesn't require much attention	0.357

Table 8.5: P Values of Mann-Whitney tests

Driving Mode Summary of Statistical Measures	Joystick in manual mode	Gamepad in high-level mode	Head movements	Voice commands
n	12	12	12	12
Mean	0,50	0,33	0,83	3,33
Minimum	0	0	0	0
Maximum	2	2	3	8

Table 8.6: Number of collisions in the real environment

the voice command *"Turn Right"*, we could use an alternative expression: "*Right*". This would decrease the delay time between the intention of turning right, and the actual output action "Right Turn" triggering. In any case, we conclude that using voice commands is more appropriated for wide spaces. Considering the size of the room where this experiment was performed, and the characteristics of the proposed route, which was difficult and narrow, the obtained results are satisfactory. Moreover, it is important to refer the lack of experience and training level of most of the volunteers, and the fact that the wheelchair's obstacle avoidance feature was not used. We believe that combining a good user training with the wheelchair's capacity to avoid obstacles and follow walls is enough to provide a safe and reliable experience.

By individually comparing the efficiency of each input module, we discovered that the majority of students preferred using the gamepad buttons, using high-level commands. However, some preferred other types of interaction, such as using the head to control the wheelchair. Although substantially smaller, a number of participants elected the voice commands as being a trustworthy style of driving the IW. Figure 8.13 shows the attendants' feedback concerning the different IMI modalities.

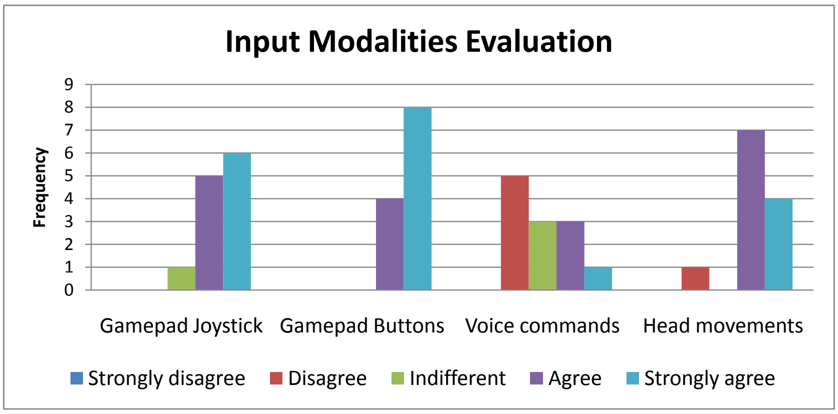


Figure 8.13: Evaluation of the different input modalities

Finally, Figure 8.14 shows the distribution of answers for the SUS questionnaire, for this experience.

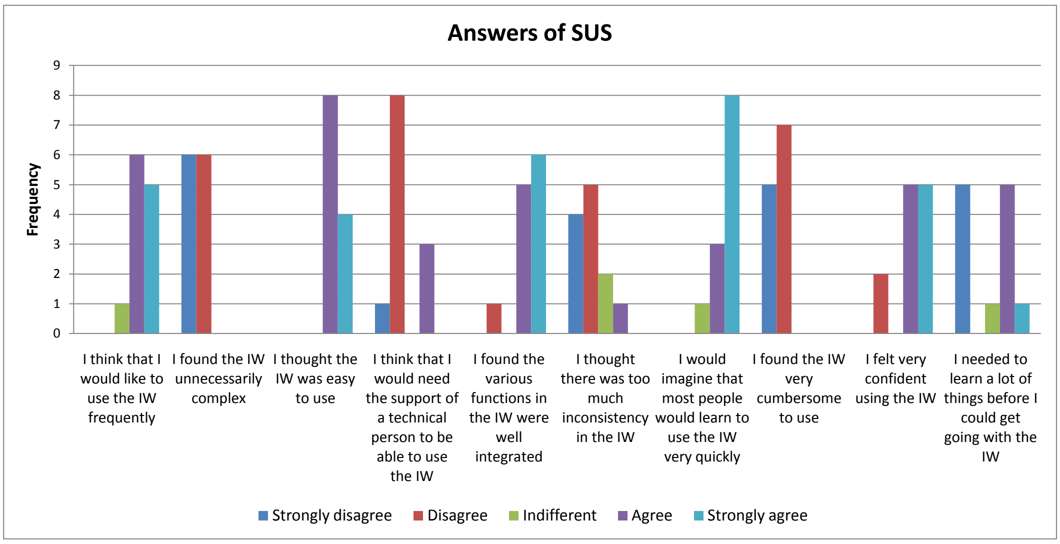


Figure 8.14: Distribution of answers to the SUS with the real IW

8.3 IMI used in different contexts

As proposed in Chapter 6, the IMI should be unaware of the context of its use. The algorithms used for implementing output actions should be contained inside a control module, and a communication protocol should be defined to allow the interaction between the two applications.

To validate this feature, a simple experiment was made. Part of the research interest of the authors of [RMR10b] consists of exploring the design of serious games for physical and cognitive rehabilitation of patients with impairments and disabilities. Rehabilitation is defined as a dynamic process of planned adaptive change in lifestyle in response to unplanned change imposed on the individual by disease or traumatic incident [GLB05]. In collaboration with the authors, we decided to have the IMI controlling one of the serious games presented in [RMR10a].

The serious game consists of a Flash application which objective is to test the user's memory performance. It starts by showing a set of random words, inviting the user to memorize such words. After this step, the application shows another set of random words, containing the words that should have been memorized. For each word that is showed on this second phase, the user should select it on the screen by clicking it if they think the word was one of those to be memorized. Other ways of selecting a word consist of making some noise (e.g. clapping to a microphone) or waving to a webcam. At the end, the application shows the user's performance, presenting the total number of correct and incorrect words.

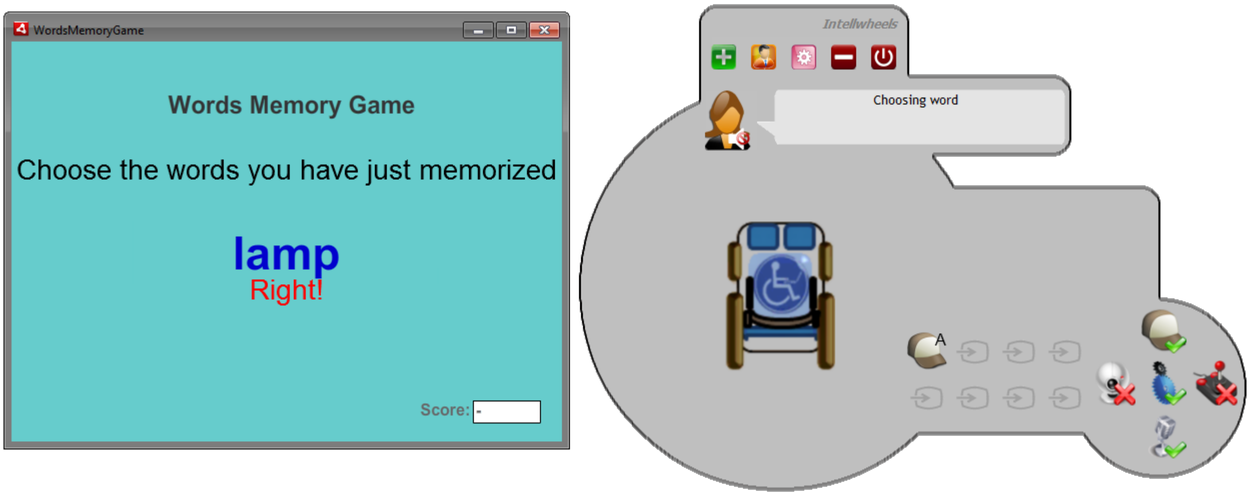
To enable the interaction between this application and the IMI, the game application had to be slightly adapted. Besides the obvious implementation of the server side socket, the action depicted in Table 8.7 was defined, in order to be sent to the IMI upon connection.

name	**type**	**hint**
Word confirmation	high-level	Choosing word

Table 8.7: Serious Game output action definition

Additionally, a simple event handler was written to invoke the already developed game's feature: selecting a word. In this way, the user could define input sequences on the IMI, and associate them to the serious game output action: *Word confirmation*.

Figure 8.15 shows the IMI being used to play the described game.



WordsMemoryGame

Words Memory Game

Choose the words you have just memorized

lamp

Right!

Score: -

Choosing word

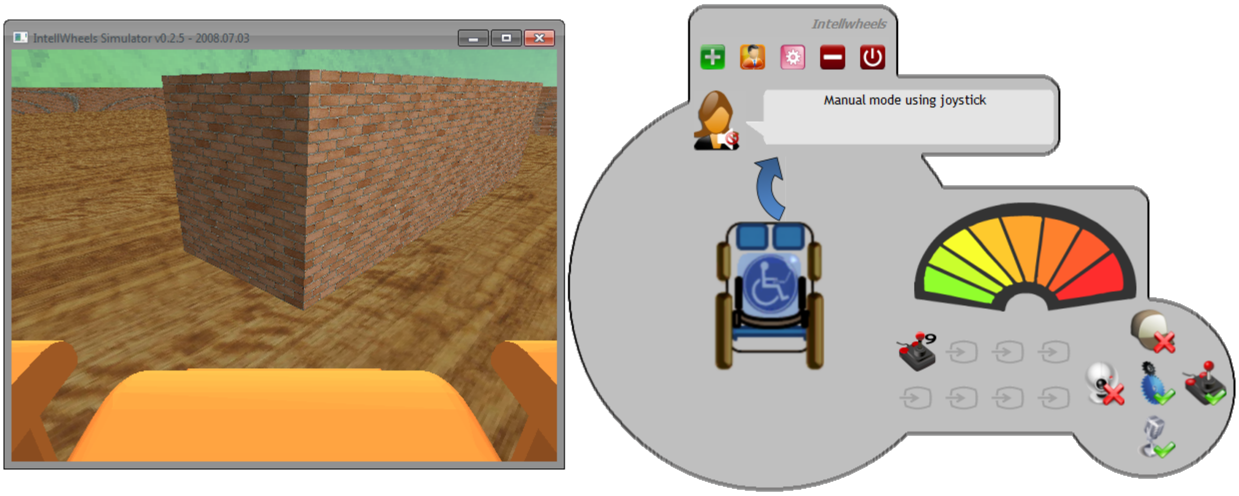
Figure 8.15: IMI controlling serious game

In this experiment, we defined two input sequences associations for this output action:

1. Pressing the button 'A' of the Wii Remote;
2. Pronouncing the voice command "*Check*".

Using the two associations, it was possible to finish the game using only the IMI.

Figure 8.16 shows the IMI being used in the context of the IWP simulator.



Manual mode using joystick

Figure 8.16: IMI controlling simulated IW

8.4 Summary

This chapter presented a description of the different experiments that were made, during the implementation stage of the IntellWheels multimodal interface. Since the usability quality of the IMI was an important variable that should be continuously validated, an initial experiment was performed, in collaboration with students of Physiotherapy of ESTSP. This experiment allowed us to learn, through the students' feedback, the strengths and weaknesses of the developed prototype. Their opinions and comments facilitated the second phase of development, for we tackled some usability details in order to obtain a more mature system.

The second experiment with volunteers at FEUP gave us the opportunity to test the IntellWheels Platform as a whole, and was useful not only to evaluate the performance evolution of the IMI, but also for collecting feedback regarding the overall system effectiveness. The results were very satisfactory. However, a number of improvements still needs to be made to compass the desired level of quality.

The last part of the chapter showed the IMI being used for two different contexts, taking not only advantage of the IWP modularity, but also of the well defined protocols to allow the interaction between the IMI and an external control module.

Chapter 9

Conclusions and Future Work

9.1 Work Summary

The goal of this dissertation was to specify and develop a prototype of a multimodal interface to be integrated in the IntellWheels platform. The final objective is to offer a multimodal style of driving the wheelchairs developed at NIAD&R/LIACC, to be used by elderly or disabled individuals. In order to achieve the proposed goals, a study of a number of Intelligent Wheelchair projects was made, to better understand the concepts behind this specific type of robot. Also, a revision of the existing input devices and recognition methods liable to be used by any kind of individual was also undertaken. The final step was to analyse the concept of multimodal interface, its advantages, features and design principles that should be followed while implementing a system of this nature.

The next step was the specification of a new multimodal interface (IMI). The multimodal interface should offer three basic input modalities: generic gamepad, speech recognition and head movements (using a Nintendo Wii Remote). This modalities were chosen due to the very low cost of its implementation. Additionally, the proposed multimodal interface should act as a server for external input modalities, to make it possible to easily integrate and test recognition modules developed in the future. Other main feature consisted in offering a flexible configuration, by allowing user defined input sequences to be freely associated to available output actions.

The IWP modular architecture allowed the IMI to be designed without the need of embedding the algorithms responsible for directly controlling an wheelchair, since this part of the system is implemented on the control module side. Instead, generic structures for representing output actions were created, as well as a simple communication protocol. This approach enabled the interaction between the IMI and the IWP control module. This peculiarity of the system architecture allows the IMI to act as an input server for different control modules, that can be applied to different contexts. A good example may consist of rapidly prototyping new video games without the need of addressing the implementation related with input devices and recognition methods.

Other features consisted in designing a friendly user interface, that could show useful information regarding the overall system state.

Having the complete system specification, a prototype of the IntellWheels multimodal interface was developed. Most of the specified features were implemented. The only exception has to be with the development of the module responsible for tracking the user profile. Due to feature prioritization, the completeness of this module was left for further development.

Finally, the results achieved by the project were assessed. Two distinct experiments were performed to test the IMI integration with the IntellWheels Platform components. The first experiment was made at ESTSP with a group of 46 students of the degree in Phisioterapy. The goal was to validate the first IMI prototype, using the IntellWheels simulator. The students' feedback was important to detect bugs and inconsistencies, and to collect opinions regarding the system usability.

The second experiment was performed at FEUP, and involved the presence of 12 volunteers. In this test, the latest wheelchair prototype developed at LIACC was used. Although the feedback of the volunteers varied, the overall satisfaction of the attendants was very positive. In any case, one may say that in order to take full advantage of the developed multimodal interface, a training session is advised. Moreover, some settings regarding each one of the input modalities should be configured taking into account the profile of each person.

9.2 Future Work

Despite having a complete system specification and a functional prototype that implements most of the proposed features, there are still further steps that should be followed to obtain a more mature system:

- Finish the implementation concerning the user profile tracking module. The idea behind this module is to automatically adjust a set of interface settings by asking the user to perform a series of configurable tests. These tests should extract information such as capacity of leaning the head, spelling ability, average time taken to perform different input sequences, etc.

- Develop a robust facial expression recognizer that could be embedded to the IntellWheels multimodal interface. This feature will consolidate the range of available inputs that can be used by people with disabilities.

- Extend the developed protocol for external input devices. It would be interesting to have an external application responsible for sending the parameter values for the actions of type *manual-mode* and *mid-level*. An example might consist of having an eye gaze tracker that could send the relative position of the eyes to directly control the wheelchair.

- Explore the recent developments in brain-computer interfaces. Although this type of modality is still under strong research, it is expected that one day it might break any physical disability barrier.

References

[AHM09] I. Ashwash, W. Hu, and G. Marcotte. Eye gestures recognition: A mechanism for Hands-Free computer control. 2009.

[Alg10] How to get this mathematical equation into a function? Available at: http://www.excelforum.com/excel-programming/754701-how-to-get-this-mathematical-equation-into-a-function.html, Consulted on January 2011, 2010.

[Ant] AnthroTronix. The AcceleGlove - capturing hand gestures in virtual reality. Available at: http://www.gizmag.com/go/2134/, Consulted on January 2011.

[BBG99] C. Martens N. Ruchel B. Borgerding, O. Ivlev and A. Gräser. Friend: Functional robot arm with user friendly interface for disabled people. *5th European Conference for the Advancement of Assistive Technology*, pages 57–65, 1999.

[BBL+94] D.A. Bell, J. Borenstein, S.P. Levine, Y. Koren, and J. Jaros. An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2018 –2022 vol.3, May 1994.

[BCS+08] R. Blatt, S. Ceriani, B. Dal Seno, G. Fontana, M. Matteucci, and D. Migliore. Brain control of a smart wheelchair. In *10th International Conference on Intelligent Autonomous Systems*, 2008.

[BHB+98] U. Borgolte, H. Hoyer, C. Bühler, H. Heck, and R. Hoelper. Architectural concepts of a semi-autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22:233–253, July 1998.

[BHHP01] G. Bourhis, O. Horn, O. Habert, and A. Pruski. An autonomous vehicle for people with motor disabilities. *Robotics & Automation Magazine, IEEE*, 8:20–28, 2001.

[BM03] S. Buisine and J. C Martin. Design principles for cooperation between modalities in bi-directional multimodal interfaces. In *Proceedings of the CHI 2003 workshop on Principles for multimodal user interface design, Ft. Lauderdale, Florida*, pages 72–75, 2003.

[BMR10] R. Braga, P. Malheiro, and L.P. Reis. Development of a realistic simulator for robotic intelligent wheelchairs in a hospital environment. In *RoboCup*

2009: Robot Soccer World Cup XIII, volume 5949 of *Lecture Notes in Computer Science*, pages 23–34. Springer Berlin / Heidelberg, 2010.

[Bol80] Richard A. Bolt. "put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, New York, NY, USA, 1980.

[BPMR08a] R. Braga, M. Petry, A. Moreira, and L.P Reis. A development platform for intelligent wheelchairs for disabled people. *In 5th International Conference on Informatics in Control, Automation and Robotics*, 1:115–121, 2008.

[BPMR08b] R. Braga, M. Petry, A. Moreira, and L.P Reis. Platform for intelligent wheelchairs using multi-level control and probabilistic motion model. *In 8th Portuguese Conference on Automatic Control*, pages 833–838, 2008.

[BPMR09] R. A.M Braga, M. Petry, A. P Moreira, and L. P Reis. Concept and design of the intellwheels platform for developing intelligent wheelchairs. *Informatics in Control, Automation and Robotics*, page 191–203, 2009.

[Bra10] R. A. M. Braga. *A Development Platform for Intelligent Wheelchairs*. PhD thesis, Universidade do Porto, 2010.

[Bro96] J. Brooke. SUS: A quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996.

[CHZ04] Joyce Y. Chai, Pengyu Hong, and Michelle X. Zhou. A probabilistic approach to reference resolution in multimodal user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*, IUI '04, pages 70–77, New York, NY, USA, 2004. ACM.

[Day08] H. Dayal. *Management of rehabilitation personnel within the context of the National Rehabilitation Policy*. PhD thesis, Faculty of Health Sciences, University of the Witwatersrand, 2008.

[DLO09] Bruno Dumas, Denis Lalanne, and Sharon Oviatt. Human machine interaction. volume 5440, chapter Multimodal Interfaces: A Survey of Principles, Models and Frameworks, pages 3–26. Springer-Verlag, Berlin, Heidelberg, 2009.

[Emb] Embarcadero. Available at: `http://www.embarcadero.com/products/delphi`, Consulted on January 2011.

[GBH+98] William S. Gribble, Robert L. Browning, Micheal Hewett, Emilio Remolina, and Benjamin Kuipers. Integrating vision and spatial reasoning for assistive navigation. In *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, pages 179–193, London, UK, 1998. Springer-Verlag.

[GDCO96] James Gips, Philip DiMattia, Francis X. Curran, and Peter Olivieri. Using eagleeyes - an electrodes based device for controlling the computer with your eyes to help people with special needs. In *Proceedings of the 5th International conference on Computers helping people with special needs. Part I*, pages 77–83, Munich, Germany, 1996.

[Gho] M. Ghovanloo. Quadriplegics can operate powered wheelchair with tongue drive system. Available at: `http://www.sciencedaily.com/releases/2009/07/090706112906.htm`, Consulted on January 2011.

[GLB05] W. Gunasekera, S. Lal, and J. Bendall. Rehabilitation of neurologically injured patients. In *Neurosurgery*, Springer Specialist Surgery Series, pages 407–421. Springer London, 2005.

[GW10] C. Gavankar and C. Warnekar. Automated system for interpreting non-verbal communication in video conferencing. *International Journal on Computer Science and Engineering (IJCSE)*, 2:22–27, 2010.

[HH05] T. Hamagami and H. Hirata. Development of intelligent wheelchair acquiring autonomous, cooperative, and collaborative behavior. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, page 3525–3530, 2005.

[JB04] M. Johnston and S. Bangalore. Matchkiosk: a multimodal interactive city guide. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Barcelona, Spain, Article No. 33, 2004.

[JCM$^+$97] Michael Johnston, Philip R. Cohen, David McGee, Sharon L. Oviatt, James A. Pittman, and Ira Smith. Unification-based multimodal integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL-35, pages 281–288, Morristown, NJ, USA, 1997.

[JHLY07] Pei Jia, Huosheng H. Hu, Tao Lu, and Kui Yuan. Head Gesture Recognition for Hands-free Control of an Intelligent Wheelchair. *Industrial Robot: An International Journal*, 34(1):60–68, 2007.

[JSK09] S. Ju, Y. Shin, and Y. Kim. Intelligent wheelchair (iw) interface using face and mouth recognition. In *Proceedings of the 13th international conference on Intelligent user interfaces*, IUI '09, pages 307–314. ACM, 2009.

[Kav08] Manolya Kavakli. Gesture recognition in virtual reality. *International Journal of Arts and Technology*, 1(2):215–229, 2008.

[Knu98] Donald E. Knuth. *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[KR07] Alexey Karpov and Andrey Ronzhin. ICANDO: Low cost multimodal interface for hand disabled people. *Journal on Multimodal User Interfaces*, 1(2):21–29, 2007.

[KRR10] Werner König, Roman Rädle, and Harald Reiterer. Interactive design of multimodal user interfaces. *Journal on Multimodal User Interfaces*, 3:197–213, 2010.

[KST$^+$97] N. I. Katevas, N. M. Sgouros, S. G. Tzafestas, G. Papakonstantinou, P. Beattie, J. M. Bishop, P. Tsanakas, and D. Koutsouris. The autonomous mobile robot SENARIO: a sensor aided intelligent navigation system for powered wheelchairs. *Robotics & Automation Magazine, IEEE*, 4(4):60–70, 1997.

[Lak05] H. Lakany. Steering a wheelchair by thought. *IEE Seminar Digests*, 2005(11059):v2–199–v2–199, 2005.

[LBJ$^+$99] S. P Levine, D. A Bell, L. A Jaros, R. C Simpson, Y. Koren, and J. Borenstein. The navchair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, 7(4):443–451, 1999.

[LBSM09] Reis. L.P., R. Braga, M. Sousa, and A. Moreira. Intellwheels mmi: A flexible interface for an intelligent wheelchair. In Jacky Baltes, Michail G. Lagoudakis, Tadashi Naruse, and Saeed Shiry Ghidary, editors, *RoboCup*, volume 5949 of *Lecture Notes in Computer Science*, pages 296–307. Springer, 2009.

[Lev] Levene test for equality of variances. Available at: `http://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm`, Consulted on January 2011.

[LPM$^+$02] N. Lau, A. Pereira, A. Melo, A. Neves, and J. Figueiredo. Ciber-rato: Um ambiente de simulação de robots móveis e autónomos. *Revista do DETUA*, 3(7):647–650, 2002.

[Man] P. Manasse. Speech recognition. Available at: `http://aac.unl.edu/Speech_Recognition.html`, Consulted on January 2011.

[Mar07] J. Maroco. *Análise Estatística – com utilização do SPSS*. Edições Sílabo, Lda, 2007.

[Mar08] Petros Maragos. *Multimodal processing and interaction : audio, video, text*. Springer, New York, 2008.

[MGG04] J. Madrigal, C. Galindo, and J. González. Assistive navigation of a robotic wheelchair using a multi hierarchical model of the environment. *Integrated Computer-Aided Engineering*, 11:309–322, December 2004.

[MHCM86] R. Madarasz, L. Heiny, R. Cromp, and N. Mazur. The design of an autonomous vehicle for the disabled. *IEEE Journal of Robotics and Automation*, 2(3):117 – 126, September 1986.

[MHD+10] C. Mahl, G. Hayrettin, P. Danny, E. Marieke, S. Lasse, D. Matthieu, and A. Alexandra. Bacteria hunt: Evaluating multi-paradigm bci interaction. *Journal on Multimodal User Interfaces*, 4(1):11–25, 2010.

[Mic] Microsoft. Kinect for xbox 360. Available at: `http://www.xbox.com/pt-PT/kinect`, Consulted on January 2011.

[Mil98] David P. Miller. Assistive robotics: An overview. In *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, pages 126–136, London, UK, 1998. Springer-Verlag.

[MIO01] Y. Matsumoto, T. Ino, and T. Ogasawara. Development of Intelligent Wheelchair System with Face and Gaze Based Interface. In *Proceedings of the 10th IEEE International Workshop on Robot and Human Communication*, pages 262–267, 2001.

[MS95] David P Miller and Marc G Slack. Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*, 2:77–88, 1995.

[Mur01] Y. Kuno N. Shimada e Y. Shirai. Murakami, Y. Collision avoidance by observing pedestrians' faces for intelligent wheelchairs. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 2018–2023, 2001.

[Nin] Nintendo. Wii controlers. Available at: `http://www.nintendo.com/wii/console/controllers`, Consulted on January 2011.

[NKSS99] S. Nakanishi, Y. Kuno, N. Shimada, and Y. Shirai. Robotic wheelchair based on observations of both user and environment. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 2, pages 912–917, 1999.

[Nor88] Donald Norman. *The psychology of everyday things*. Basic Books, New York, 1988.

[oEEENY02] Institute of Electrical and NY) Electronics Engineers (New York, NY);IEEE Control Systems Society (New York. *Proceedings of the 2002 IEEE International Conference on Control Applications : September 18-20, 2002, Scottish Exhibition and Conference Centre, Glasgow, Scotland, U.K.* IEEE Service Center, Piscataway N.J, 2002.

[Ope] OpenInterface. Openinterface platform. Available at: `http://www.openinterface.org/platform/`, Consulted on January 2011.

[Ovi97] Sharon Oviatt. Multimodal interactive maps: designing for human performance. *Human Computer Interaction*, 12:93–129, March 1997.

[Ovi02] Sharon Oviatt. *A Handbook of Human-Computer Interaction*. Jacko J. and Sears (Eds.), New Jersey, 2002.

[PMV+07] J. Philips, J. Millan, G. Vanacker, E. Lew, F. Galán, P. Ferrez, H. Van Brussel, and M. Nuttin. Adaptive shared control of a brain-actuated simulated wheelchair. In *Proceedings of the 10th IEEE International Conference on Rehabilitation Robotics*, page 408–414, Noordwijk, The Netherlands, 6 2007.

[PN02] G. Pires and U. Nunes. A wheelchair steered through voice commands and assisted by a reactive Fuzzy-Logic controller. *Journal of Intelligent and Robotic Systems*, 34:301–314, 2002.

[Pro] LURCH Project. Lurch - the autonomous wheelchair. Available at: `http://airwiki.ws.dei.polimi.it/index.php/LURCH_-_The_autonomous_wheelchair`, Consulted on January 2011.

[PSF01] E. Prassler, J. Scholz, and P. Fiorini. A robotics wheelchair for crowded public environment. *Robotics Automation Magazine, IEEE*, 8(1):38 –45, March 2001.

[RBG+07] Brice Rebsamen, Etienne Burdet, Cuntai Guan, Haihong Zhang, Chee Leong Teo, Qiang Zeng, Christian Laugier, and Marcelo H. Ang Jr. Controlling a wheelchair indoors using thought. *IEEE Intelligent Systems*, 22:18–24, March 2007.

[Reb06] *A Brain-Controlled Wheelchair Based on P300 and Path Guidance*, 2006.

[Ree04] Lai J. Larson J. Oviatt S. Balaji T. Buisine S. Collings P. Reeves, L. Guidelines for multimodal interface design. *Communications of the ACM*, 47(1):57–59, 2004.

[RL98] Thomas Röfer and Axel Lankenau. Architecture and applications of the bremen autonomous wheelchair. In *Proceedings of the Fourth Joint Conference on Information Systems*, pages 365–368, 1998.

[RMR10a] P. Rego, P. Moreira, and LP. Reis. New forms of interaction in serious games for rehabilitation (submitted to m. m. cruz-cunha (ed.). In *Handbook of Research on Serious Games as Educational, Business, and Research Tools: Development and Design: IGI Global*, 2010.

[RMR10b] P. Rego, P. Moreira, and LP. Reis. Serious games for rehabilitation: A survey and a classification towards a taxonomy. *5^a Conferência Ibérica de Sistemas e Tecnologias de Informação*, 1:349–354, 2010.

[Roy09] Teller S. Reimer B. Roy, N. Personal digital interfaces for intelligent wheelchairs, 2009.

[RTR] N. Roy, S. Teller, and B. Reimber. The intelligent wheelchair project. Available at: `http://www.csail.mit.edu/videoarchive/research/robo/intelligent-wheelchair`, Consulted on January 2011.

[SAP] Microsoft speech api (sapi) 5.4. Available at: `http://msdn.microsoft.com/en-us/library/ee125663(v=vs.85).aspx`, Consulted on January 2011.

[SCW+01] Michael A Sayette, Jeffrey F Cohn, Joan M Wertz, Michael A Perrott, and Dominic J Parrott. A psychometric evaluation of the facial action coding system for assessing spontaneous expression. *Journal of Nonverbal Behavior*, 25:167–185, 2001.

[Sil02] L. Silva. Head gestures recognition. In *Proceedings of the International Conference on Image Processing*, volume 3, page 266–269, 2002.

[Sim05] Richard C. Simpson. Smart wheelchairs: A literature review. *Journal of Rehabilitation Research and Development*, 42(4):423–436, 2005.

[SK09] Akira Sasou and Hiroaki Kojima. Noise robust speech recognition applied to voice-driven wheelchair. *EURASIP Journal on Advances in Signal Processing*, 2009:41:1–41:1, January 2009.

[SLB+98] R. C Simpson, S. P Levine, D. A Bell, L. A Jaros, Y. Koren, and J. Borenstein. Navchair: An assistive wheelchair navigation system with automatic adaptation. *Assistive Technology and Artificial Intelligence*, pages 235–255, 1998.

[SPH98] R. Sharma, V.I. Pavlovic, and T.S. Huang. Toward multimodal human-computer interface. *Proceedings of the IEEE*, 86(5):853 –869, May 1998.

[SRLS98] K. Schilling, H. Roth, R. Lieb, and H. Stutzle. Sensors to improve the safety for wheelchair users. *Proceedings of the 3rd Annual TIDE-Technology for Inclusive Design and Equality-Congress*, pages 327–331, 1998.

[SSF78] M. Suwa, N. Sugie, and K. Fujimora. A preliminary note on pattern recognition of human emotional expression. *4th International Joint Conference on Pattern Recognition*, (2):408—-410, 1978.

[Sys] CyberGlove Systems. Cyberglove ii. Available at: `http://www.cyberglovesystems.com/products/cyberglove-ii/overview`, Consulted on January 2011.

[TASA+09] Martin Tall, Alexandre Alapetite, Javier San Agustin, Henrik H.T Skovsgaard, John Paulin Hansen, Dan Witzner Hansen, and Emilie Møllenbach. Gaze-controlled driving. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI '09, pages 4387–4392, New York, NY, USA, 2009. ACM.

[VAR+02] S. Vicente, C. Amaya, F. Rio, A. Balcells, and D. Muniz. Tetranauta: a intelligent wheelchair for users with very severe mobility restrictions. In *Control Applications, 2002. Proceedings of the 2002 International Conference on*, volume 2, pages 778–783, 2002.

[Wei04] Y. Wei. *Vision-based Human-robot Interaction and Navigation of Intelligent Service Robots.* PhD thesis, Institute of Automation, Chinese Academy of Sciences, Beijing, China, 2004.

[Yan98] Holly A. Yanco. Wheelesley: A robotic wheelchair system: Indoor navigation and user interface. In *Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing*, pages 256–268, London, UK, 1998. Springer-Verlag.

[YBS02] J. Yoder, T. Baumgartner, and S. Skaar. Initial results in the development of a guidance system for a powered wheelchair. *IEEE Transactions on Rehabilitation Engineering*, 4(3):143–151, 2002.

[Zho00] D. Zhongguo. *Proceedings of the 3rd World Congress on Intelligent Control and Automation*. Press of University of Science and Technology of China, Institute of Electrical and Electronics Engineers, 2000.

[ZJ05] Z. Zhu and Q. Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98:124–154, April 2005.

Appendix A

Survey

ESTSP POLITÉCNICO DO PORTO

Questionário I

Questionário – Usabilidade da Cadeira de Rodas Inteligente (CRI) (Simulador)

Através deste questionário pretende-se obter informações sobre a utilização da Cadeira de Rodas Inteligente (IntellWheels). A recolha destes dados permitirá que intervenções futuras na plataforma melhorem as condições de utilização. Os dados fornecidos serão tratados de forma agregada bem como será mantida a confidencialidade. Agradecemos a disponibilidade e a colaboração de todos.

1. Identificação do Utilizador.

1.1. Idade

1.2. Sexo

1.3. Peso

1.4. Altura

1.5. Tipo de diagnóstico/dificuldade

1.6. Assinale o seu nível de escolaridade (ou que frequenta):

1º Ciclo		Bacharelato	
2º Ciclo		Licenciatura	
3º Ciclo		Mestrado	
Secundário		Doutoramento	

1.7. Qual a frequência de utilização de Jogos de Vídeo?
Instruções: Cada opção de resposta expressa uma atitude numa escala de 1 a 5, onde:
1 = Nunca, 2 = Raramente, 3 = Às vezes, 4 = Muitas vezes, 5 = Sempre
Para cada questão deve tentar comparar a sua opinião com cada uma das opções de resposta, marcando com um x na opção mais exacta.

	1	2	3	4	5
1.7.1. Costumo jogar jogos de vídeo.					
1.7.2. Costumo utilizar comandos da Wii.					
1.7.3. Costumo utilizar o joystick.					

2. Usabilidade e Segurança da Cadeira de Rodas Inteligente (Simulador).
Instruções: Cada opção de resposta expressa uma atitude numa escala de 1 a 5, onde:
1 = Discordo totalmente, 2 = Discordo, 3 = Indiferente, 4 = Concordo, 5 = Concordo totalmente
Para cada questão deve tentar comparar a sua opinião com cada uma das opções de resposta, marcando com um x na opção mais exacta.

Tradução da Escala SUS *(System Usability Scale)*	1	2	3	4	5
2.1. Penso que gostaria de utilizar a CRI com frequência.					
2.2. Achei que a CRI é desnecessariamente complexa.					
2.3. Penso que a CRI é de fácil utilização.					
2.4. Penso que iria necessitar de apoio de um técnico para ser capaz de utilizar a CRI em pleno.					
2.5. Penso que as diversas funções da CRI foram bem integradas.					
2.6. Penso que a CRI tem muitas incoerências.					
2.7. Penso que a maioria dos utilizadores aprenderia facilmente a utilizar a CRI.					
2.8. Penso que a CRI é muito complicada de utilizar.					
2.9. Senti-me confiante na utilização da CRI.					
2.10. Penso que ainda necessitaria de aprender muitas funcionalidades da CRI.					

Segurança	1	2	3	4	5
2.11. Senti-me seguro na condução da CRI.					
2.12. Senti que tive controlo da CRI.					
2.13. É fácil conduzir a CRI em espaços estreitos.					
2.14. Conduzir a CRI não necessita muita atenção.					
2.15. A CRI tem o mesmo comportamento no ambiente simulado e real.					

3. Utilização dos Meios de Controlo da Cadeira de Rodas Inteligente (Simulador).
Instruções: Cada opção de resposta expressa o nível de satisfação numa escala de 1 a 5, onde:
1 = Muito Insatisfeito, 2 = Insatisfeito, 3 = Indiferente, 4 = Satisfeito, 5 = Muito Satisfeito
Para cada questão deve tentar comparar a sua opinião com cada uma das opções de resposta, marcando com um x na opção mais exacta.

3.1. Indique o nível de satisfação relativamente ao tipo de controlo da CRI.

	1	2	3	4	5
3.1.1. Utilização do joystick no modo manual.					
3.1.2. Utilização do joystick no modo de alto nível.					
3.1.3. Utilização dos comandos de voz no modo de alto nível.					
3.1.4. Utilização dos movimentos de cabeça no modo manual.					
3.1.5. Utilização de modo integrada de todos os comandos.					

3.2. Indique quais as falhas/incoerências que encontrou durante a realização da experiência com CRI:

3.3. Indique quais as principais dificuldades que sentiu durante a realização da experiência com CRI:

3.4. Indique possíveis alterações de melhoria do Simulador:

4. Interface Multimodal.
Instruções: Cada opção de resposta expressa o nível de satisfação numa escala de 1 a 5, onde:
1 = Discordo totalmente, 2 = Discordo, 3 = Indiferente, 4 = Concordo, 5 = Concordo totalmente
Para cada questão deve tentar comparar a sua opinião com cada uma das opções de resposta, marcando com um x na opção mais exacta.

	1	2	3	4	5
4.1. A Interface Multimodal facilitou a informação sobre o tipo de comandos que estava a utilizar.					

4.2. Indique possíveis alterações de melhoria da Interface Multimodal:

Survey

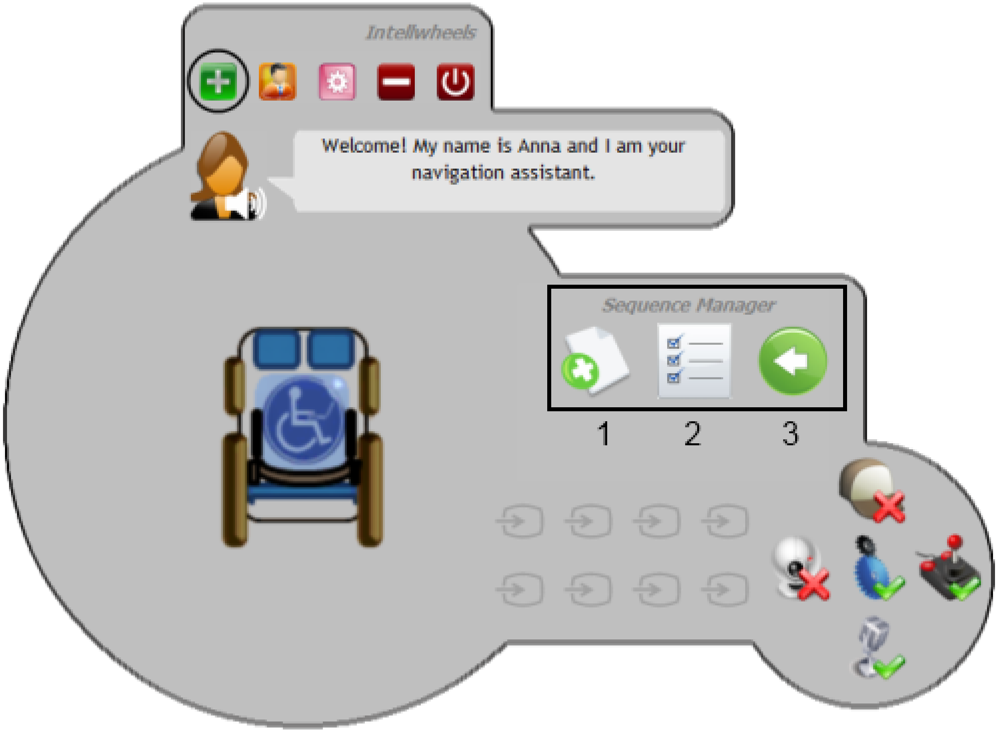
Appendix B

User Guide

B.1 Sequence Manager

This subsection presents the IMI being used to consult, create and delete associations between input sequences and output actions. To access the sequence manager features, one can select the first button present at the top of the graphical user interface. This will turn visible the features related with the sequence manager:

1. Create new sequence
2. Consult list of existing associations
3. Return to navigation mode



Welcome! My name is Anna and I am your navigation assistant.

Figure B.1: Accessing sequence manager

Associations

Clicking the second button on the sequence manager panel, the user can consult the list of saved associations. A panel is shown, graphically presenting the input associations associated to each output action. Figure B.2 contains an illustration of this feature.

Welcome! My name is Anna and I am your navigation assistant.

Figure B.2: IMI associations panel

New association

To create a new input sequence, the user should click the first icon, or pronounce "Create new sequence". This will activate the sequence creator mode. Figure B.3 shows the wizard being used to create a new input sequence. When the wizard is activated, three extra options are shown. The user can save the new input sequence, and restart or cancel the creation process.

This input sequence shown on the example consists of pressing the button 2 of the gamepad, followed by leaning the head to the right. When the user selects the option *Save*, a panel presenting the available actions is shown. The buttons were made large to make it possible for the user to select an output action using the touch screen feature of the wheelchair's monitor.

If the user selects an high-level action, the association is immediately created. If, instead, a mid-level or manual-mode action is selected, the wizard shows an extra panel containing the desired source for the action parameters. Figures **??** contains an illustration of the panel shown for the parameter extraction.

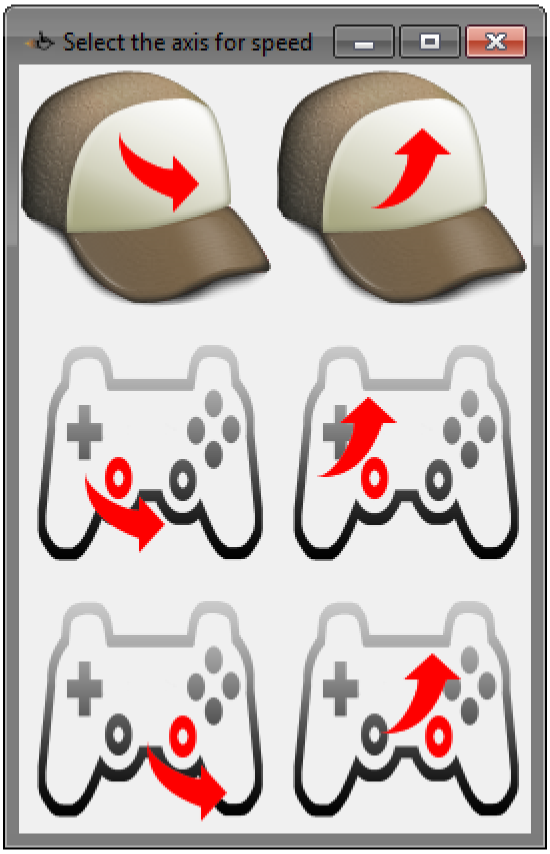
In this example, we associated the input sequence to the interface action of clicking the left mouse button. Figure B.6 shows the associations panel after the creation of the new input sequence.

Reading sequence...Please say "save new sequence" whenever you are finished.

Figure B.3: New input sequence wizard

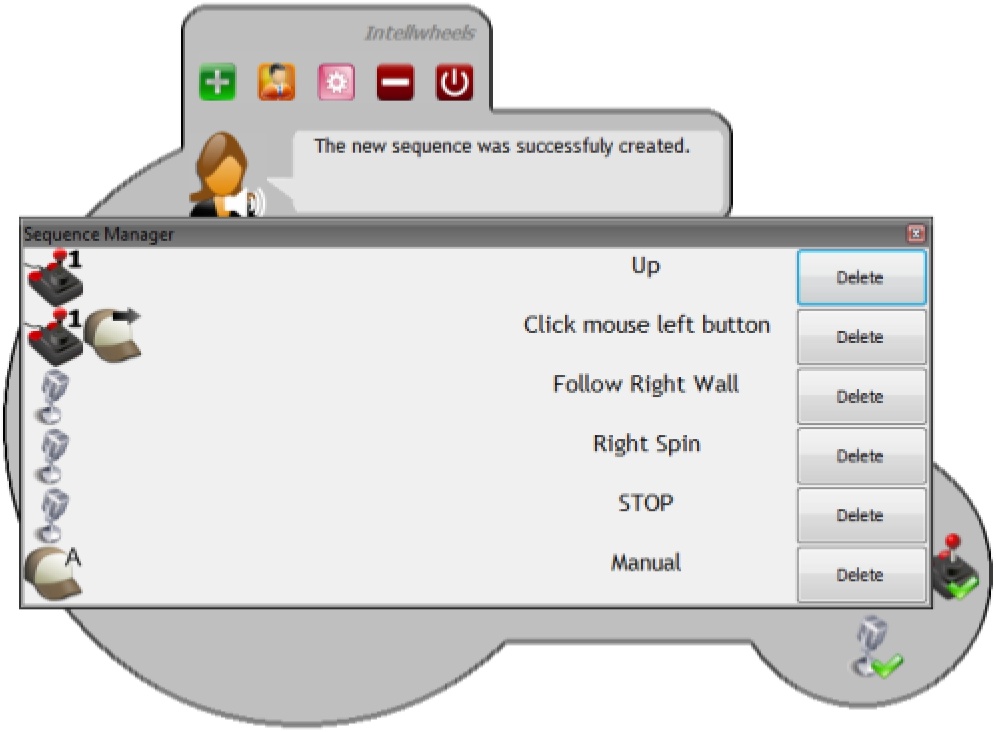
What would be the desired action?

Figure B.4: Selecting an output action



Select the axis for speed

Figure B.5: Selecting action parameter sources



The new sequence was successfuly created.

Sequence Manager

Up	Delete
Click mouse left button	Delete
Follow Right Wall	Delete
Right Spin	Delete
STOP	Delete
Manual	Delete

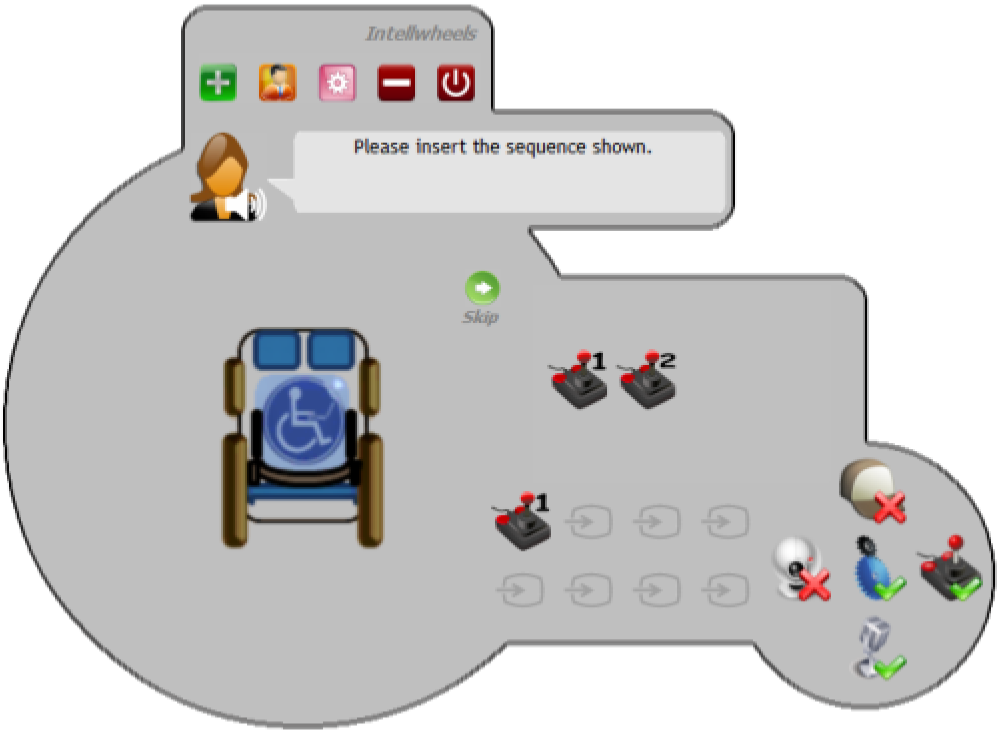
Figure B.6: Associations panel after creation of new sequence

B.2 User Profiler

To start the User Profiler feature, the user has to click the second button on the top of the IMI (Figure B.7).

Figure B.7: Starting user profiler module

As it was explained in Chapter 7, the test set present on the XML file is iteratively shown to the user. It starts by asking the user to perform the gamepad button sequences. Figure B.8 shows an example.

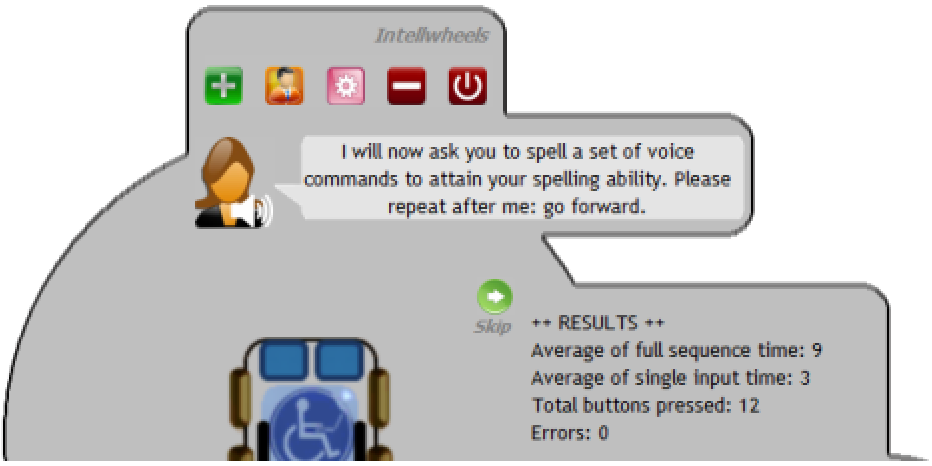


Please insert the sequence shown.

Figure B.8: User profiler gamepad test

When the users finishes the first component of the user profiler module, the navigation assistant asks the user to pronounce the voice commands present on the XML. Also, the quantitative results for the gamepad buttons test is presented. Figure B.9 shows an example.

The last part of the user profiler test is shown in Figure B.10. The user is invited to place the gamepad's joystick into certain positions. A similar approach is used for the head movements test.



I will now ask you to spell a set of voice commands to attain your spelling ability. Please repeat after me: go forward.

++ RESULTS ++
Average of full sequence time: 9
Average of single input time: 3
Total buttons pressed: 12
Errors: 0

Figure B.9: User profiler voice test

Thank you. Now, try to place the red dot on top of the green dot, using the left analog switch of your joystick.

++ Results ++
Average recognition confidence level: 0.0%
Errors: 0

Figure B.10: User profiler joystick test

B.3 Device Manager

To access the Device Manager feature, the user should click the third button present at the top of the IMI (Figure B.11). By clicking this button, a tabs panel is open. Each tab is used for debugging the recognition modules for the gamepad, speech recognition, and head movements, respectively.

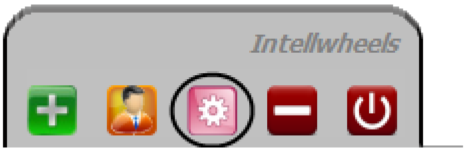
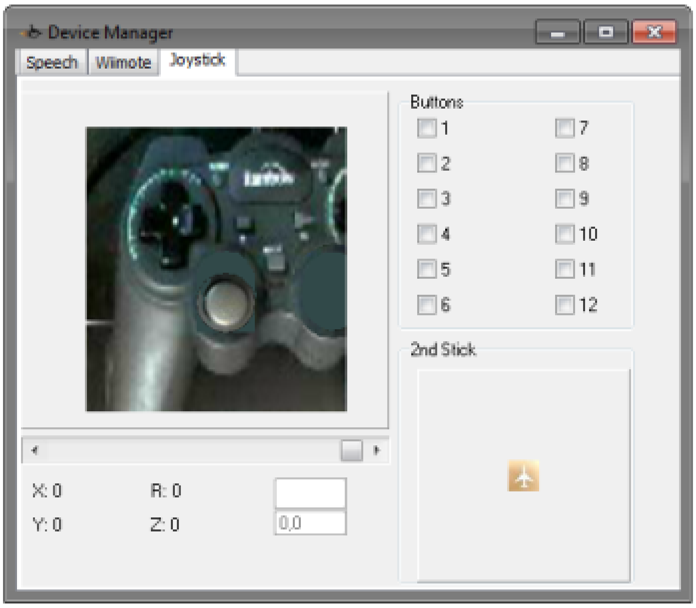


Figure B.11: IMI Device Manager

Gamepad

The gamepad tab shows the values for each analog switch (joystick) of the gamepad. It also shows which buttons are being pressed. The scroll bar is used to limit the maximum speed for both joysticks.

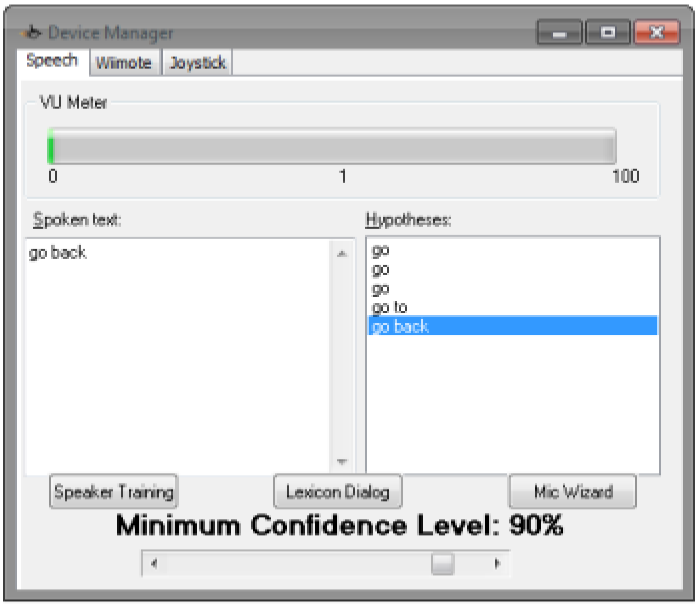


X: 0
Y: 0
R: 0
Z: 0
0,0

Figure B.12: Device Manager: Gamepad

Speech Recognition

The speech recognition tab is used to consult the status of the speech recognition module. It shows all the recognized hypothesis on the right side. On the left side, a text box shows all the voice commands that were accepted and sent to the IMI core component. The scroll bar is useful to configure the minimum trust level used to filter the recognized voice commands.



Spoken text:
go back

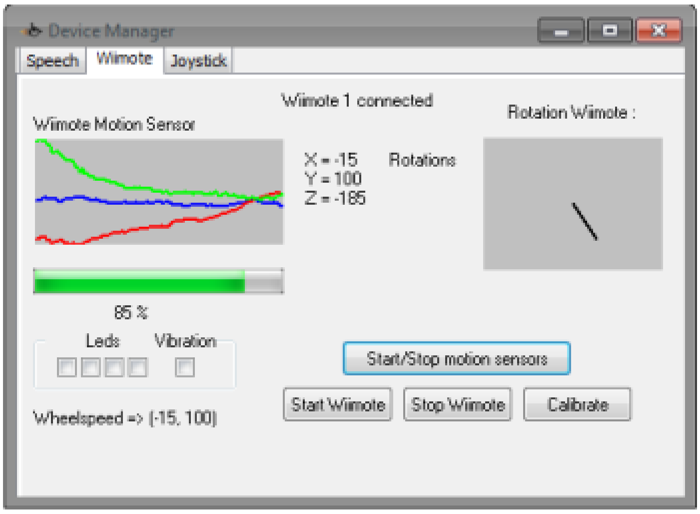
Hypotheses:
go
go
go
go to
go back

Minimum Confidence Level: 90%

Figure B.13: Device Manager: Speech Recognition

Wiimote

In the wiimote controller tab, it is possible to consult the information related to the accelerometer values, as well as the parametrization of these values, that are sent to directly control the wheelchair.



Wiimote 1 connected

X = -15
Y = 100
Z = -185

85 %

Wheelspeed => (-15, 100)

Figure B.14: Device Manager: Wiimote

B.4 External Input STUB

In order to exemplify the interaction between the IMI and an external input device, a simple STUB was implemented. The STUB represents a fictional facial expression recognizer, that sends two distinct inputs: eye blink and open mouth. Figure B.15 shows an example of the STUB being used to create a new input sequence using facial expression inputs.

Reading sequence...Please say "save new sequence" whenever you are finished.

Figure B.15: Facial expression recognizer STUB