JRRD Volume 48, Number 9, 2011
Pages 1061–1076
Journal of Rehabilitation Research & Development

IntellWheels: Modular development platform for intelligent wheelchairs

Rodrigo Antonio Marques Braga, PhD;[1–2] Marcelo Petry, BEng;[1,3] Luis Paulo Reis, PhD;[1–2]* António Paulo Moreira, PhD[3–4]

[1]DEI/FEUP, Department of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal; [2]LIACC, Artificial Intelligence and Computer Science Laboratory of the University of Porto, Porto, Portugal; [3]INESC-P, Institute for Systems and Computer Engineering of Porto, Porto, Portugal; [4]DEEC/FEUP, Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Porto, Portugal

Abstract—Intelligent wheelchairs (IWs) can become an important solution to the challenge of assisting individuals who have disabilities and are thus unable to perform their daily activities using classic powered wheelchairs. This article describes the concept and design of IntellWheels, a modular platform to facilitate the development of IWs through a multiagent system paradigm. In fact, modularity is achieved not only in the software perspective, but also through a generic hardware framework that was designed to fit, in a straightforward manner, almost any commercial powered wheelchair. Experimental results demonstrate the successful integration of all modules in the platform, providing safe motion to the IW. Furthermore, the results achieved with a prototype running in autonomous mode in simulated and mixed-reality environments also demonstrate the potential of our approach. Although some future research is still necessary to fully accomplish our objectives, preliminary tests have shown that IntellWheels will effectively reduce users' limitations, offering them a much more independent life.

Key words: artificial intelligence, human-robot interfaces, independent mobility, intelligent robotics, intelligent wheelchair, mixed reality, multiagent systems, multimodal interface, service robots, simulation, voice control.

INTRODUCTION

Physical injuries are frequently caused by accidents, exposure to chemicals and drugs, and diseases like cerebral palsy and multiple sclerosis. Such medical conditions cause the patients to have limited control of some muscles of the arms, legs, and face and, thus, affect their mobility. A generalized approach to treating and assisting those with physical disabilities has not yet been achieved. Usually, each patient shows a different combination of symptoms, which calls for different strategies.

In response to numerous mobility problems, many intelligent wheelchair (IW) projects have been created over the last years [1]. According to the general concept presented in the relevant literature, we define an "intelligent wheelchair" as a robotic device built from an electric-powered wheelchair provided with a sensorial system, actuators, and processing capabilities. At the same time, it is assumed that IW may include at least some features, such as autonomous navigation, autonomous planning, extended human-machine interaction, semiautonomous

Abbreviations: ACL = agent communication language, FIPA = Foundation for Intelligent Physical Agents, GUI = graphical user interface, IW = intelligent wheelchair, JADE = Java Agent DEvelopment Framework, MAS = multiagent system, MMI = multimodal interface, MR = mixed reality, PDDL = planning domain definition language, PID = proportional-integral-derivative, UDP = user datagram protocol, USB = universal serial bus.

*Address all correspondence to Luis Paulo Reis, PhD; Faculty of Engineering, University of Porto, Rua Dr, Roberto Frias, s/n, 42 00-465, Porto, Portugal; +351-225081829, +351-919455251; fax: +351-225081443.
Email: lpreis1970@gmail.com

DOI:10.1682/JRRD.2010.08.0139

behavior with obstacle avoidance, and cooperative and collaborative behavior. Thus, IWs may be a good solution to the challenge of assisting people with severe disabilities who are unable to operate classic electric wheelchairs by themselves in their daily activities.

This article presents the concept and design of a platform for the development of IWs. The IntellWheels platform was developed according to the multiagent paradigm and a modular concept to increase the flexibility of the system (a generic framework that can be implemented with almost any commercial wheelchair and assist people with different impairments) and facilitate the development of new IWs. In addition, our research considers the final cost of the proposed approach in an effort to make it more accessible to the target population. Similarly, we have tried to keep the original aesthetics and ergonomics of ordinary powered wheelchairs so that the assemblage of the hardware framework does not interfere with the comfort and workability of the wheelchair in the execution of daily tasks. Furthermore, we also aim to extend the human-machine interaction to assist not only elderly people, but also people with severe mobility restrictions.

The rest of the article is subdivided as follows: the "Related Work" subsection provides an overview of the work concerning IW development; the "Methods" section discusses the platform, explaining in detail each module of the structure; the "Results" section reveals the experiments and results; the "Discussion" section deliberates on the current state of the project and its guidelines; and the "Conclusions" section presents the final conclusions and some future research topics.

RELATED WORK

In recent years, many IWs have been developed and a large number of scientific projects have been initiated in the area [1]. In 2009 alone, more than 90 publications related to IWs were found in the Institute of Electrical and Electronics Engineers Xplore Digital Library.

Madarasz et al. proposed one of the first autonomous wheelchairs for those with physical disabilities [2]. They presented a wheelchair equipped with a microcomputer, digital camera, and ultrasonic rangefinder. Their objective was to develop a vehicle capable of operating without human intervention in populated environments and with few or no collisions with objects or people. Hoyer and Hölper presented the architecture of a modular control for an omnidirectional wheelchair [3]. According to them, this structure takes advantage of the local intelligence of each unit to yield high independence from other modules and an open control system. NavChair is described in Levine et al. and has some interesting capabilities, such as wall following, automatic obstacle avoidance, and doorway passing [4].

Miller developed the Tin Man I [5]. Initially, this system had three modes of operation: human-guided with obstacle avoidance, movement forward along a heading, and movement to a specific point (*x*, *y*). Afterward, the project evolved into Tin Man II with the inclusion of new capabilities, such as backup, backtracking, wall following, doorway passing, and docking. By including some of Tin Man's capabilities, the Maid project is designed to navigate in two particularly difficult and tiresome situations, namely, narrow cluttered environments and wide crowded areas [6].

Wellman et al. proposed a hybrid wheelchair equipped with two legs in addition to its regular four wheels, enabling the wheelchair to climb over steps and move through rough terrain [7]. Some projects present solutions for people with tetraplegia by using the recognition of facial expressions as the main input guiding the wheelchair [8–10]. Others control IWs with user "thoughts." This technology typically uses sensors that measure the electromagnetic waves of the brain [11–12].

ACCoMo (Autonomous, Cooperative, COllaborative MObile) is an IW prototype that allows disabled individuals to move safely in indoor environments [13]. ACCoMo is an agent-based prototype with simple autonomous, cooperative, and collaborative behaviors. In addition, other important projects present solutions to most common issues faced by patients with physical injuries, such as the intelligent navigation system discussed in SENARIO (SENsor Aided intelligent wheelchaiR navigation) [14]; the autonomous and semiautonomous movements of VAHM (*Véhicule Autonome pour Handicapé Moteur* ["Autonomous Vehicle for People with Disabilities"]) [15]; the obstacle avoidance and shared-control system of Rolland [16]; SIAMO (*Sistema Integral de Ayuda a la Movilidad* ["Integral System for Assisted Mobility"]) [17] and its different alternatives for guidance, safety, and comfort through an innovative user-machine interface; and finally, the semiautonomous robotic system FRIEND (Functional Robot arm with user-frIENdly interface for Disabled people) [18] and its robot arm MANUS.

Although several prototypes have been developed and different approaches have been proposed for IWs [19], at the moment, no platform has been proposed that simultaneously enables—

- Easy development of low-cost IWs using traditional electric-powered wheelchairs with minor aesthetical and ergonomic modifications.
- Testing of new algorithms, new and/or better human-machine interfaces, and patient drive training through a virtual and/or mixed reality (MR) environment.

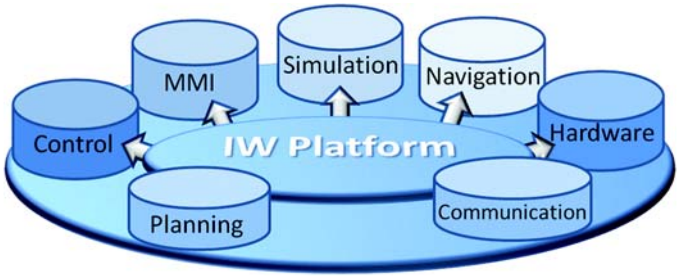


Figure 1.
IntellWheels project modules. IW = intelligent wheelchair, MMI = multimodal interface.

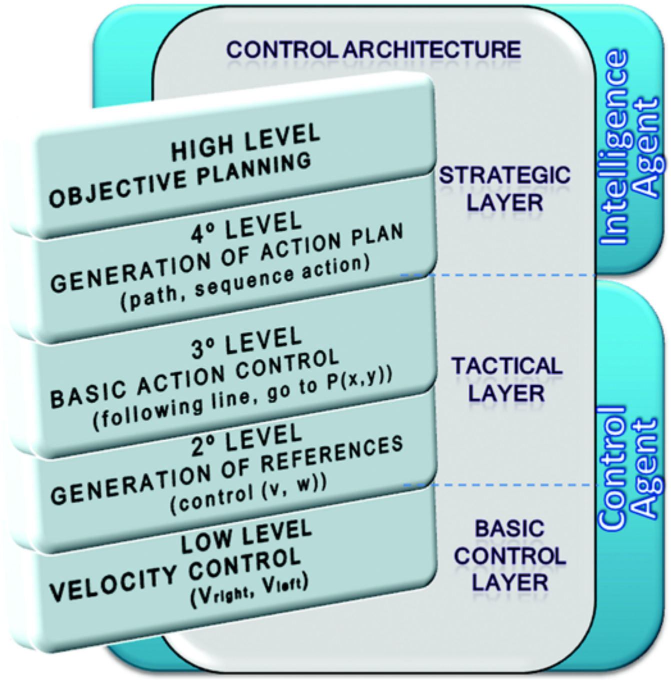


Figure 2.
IntellWheels multilevel control architecture. 2^{o} = 2nd, 3^{o} = 3rd, 4^{o} = 4th, P = point, v = linear velocity, w = angular velocity, V = velocity.

METHODS

The IntellWheels project focuses on creating a platform to develop IWs. It is mainly concerned with the research and design of a multiagent system (MAS) that will enable easy integration of distinct sensors, actuators, user input devices, navigation methodologies, intelligent planning techniques, and cooperation methodologies. This platform will facilitate the development and testing of new methodologies and techniques and then be integrated with minor modifications into most commercially available electric wheelchairs.

We believe that this platform can bring real capabilities to the wheelchairs, such as intelligent planning and autonomous and semiautonomous navigation. Development of the platform is achieved through an advanced control system, which progresses from a simple shared control (obstacle avoidance during manual navigation) to complex high-level orders (achieved through the combination of a more interactive user-machine interface, autonomous driving, mapping, and strategy definitions). Thus, we propose a solution where such complex problems are broken down into several modules (**Figure 1**). Each module (planning, control, multimodal interface [MMI], simulation, navigation, hardware, and communication) will be fully described in the following subsections.

IntellWheels has a multilevel control architecture subdivided into three layers: strategic, tactical, and basic control (**Figure 2**). These three layers are distributed by two agents: intelligence and control. Our platform is modeled with the MAS approach in order to easily integrate new features (agents with new abilities). Advantages of such an approach are that agents can show self-organization and that complex behavior can emerge through simple individual strategies. **Figure 3** depicts the software architecture, with the different agents modeled in the platform. Note that communication between agents is performed through the agent communication language (ACL).

The IntellWheels MAS architecture was designed to follow the standards of the Foundation for Intelligent Physical Agents (FIPA) [20] in order to promote the interoperation of heterogeneous agents and the services that they can represent (**Figure 4**). The main agents, those which are embedded in the wheelchair, are briefly described below:

1. Intelligence agent. This agent implements the planning module and is responsible for the strategy layer, where

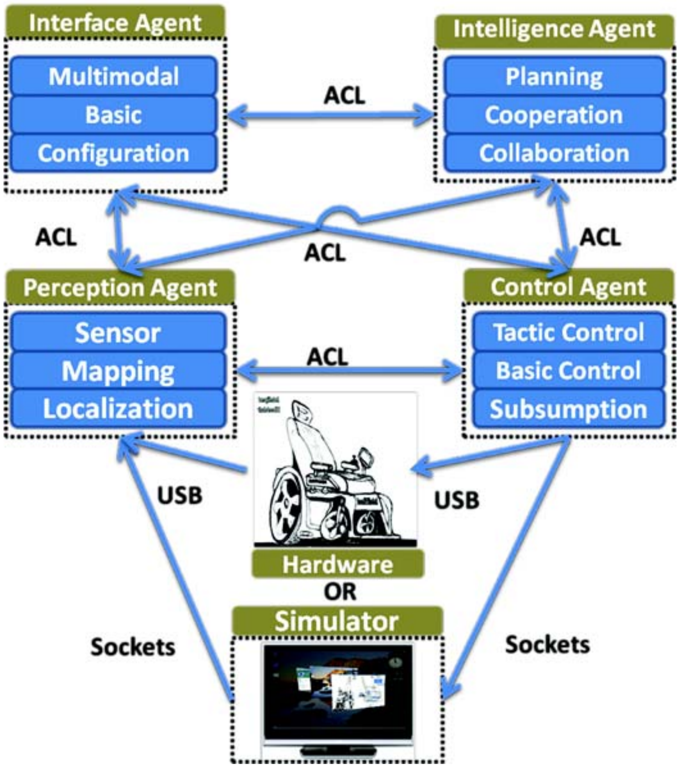


Figure 3.
IntellWheels software architecture. ACL = agent communication language, USB = universal serial bus.

high-level decision are made, such as continuous planning, runtime monitoring, and cooperation with other agents. The high-level strategy plan is responsible for creating a sequence of high-level actions required to achieve a global goal (based on a planning algorithm). Furthermore, this agent is also responsible for generating action plans with sequences of basic actions (path-planning algorithm).

2. Control agent. This agent implements the tactical layer that includes a basic action control (e.g., follow line, spin, follow wall, go to point) and a generator of references, which computes the linear and angular speeds of the IW. This agent also implements the basic control layer, which is responsible for deriving the low-level speed control (proportional-integral-derivative [PID] controller of the wheel's speed).
3. Interface agent. This agent collects user inputs (through the MMI module) and displays the most relevant information (e.g., sensor readings, speed, position) through a graphical user interface (GUI). In addition, it manages the interaction between the user and the other system agents, forwarding user orders to the most appropriate agent.
4. Perception agent. This agent represents the perception system of mobile robots. Its objectives include reading the appropriate sensor and updating the internal world representation, mapping, and localization.

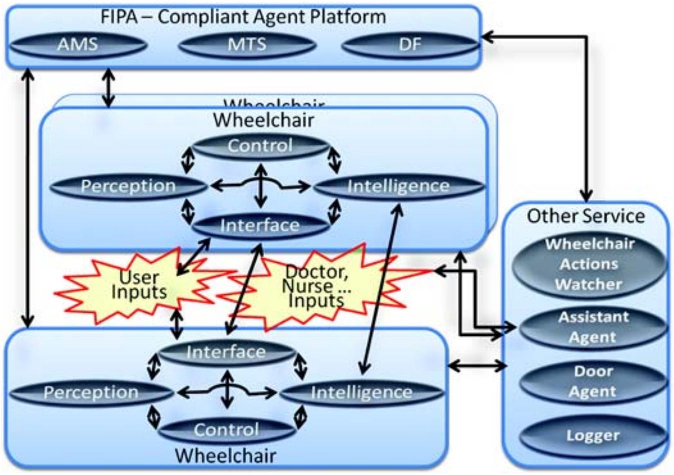


Figure 4.
IntellWheels multiagent architecture. AMS = agent manager system, DF = directory facility, FIPA = Foundation for Intelligent Physical Agents, MTS = message transport service.

Other agents complete our set of platform agents, designated as services agents. Several agents were created to help the IW system achieve its global goals. These agents can cooperate and collaborate with the agents embedded in the mobile robot. The door agent is responsible for controlling the doors and gates in the IW environment, opening and closing doors to allow or inhibit access in restricted areas. The logger agent is responsible for creating permanent log files about the messages exchanged between agents in order to assist the debugging process and system analysis. The wheelchair actions watcher agent is responsible for centralizing the control of all traffic in the IW environment, thus avoiding traffic conflicts. The role of this agent is to monitor all activities and actions when necessary so as to avoid potential conflicts and to solve possible deadlocks. The assistant agent is responsible for systemwide human interaction, as well as for receiving and handling global goals. This agent is the interface between nurses, doctors, therapists, and assistants with the IW system.

In this IntellWheels system, an IW can assume bodily form in three different modes: real, virtual reality, and MR. To instantiate the body robot, the hardware for the real robot, the simulator for the virtual robot, or both must be used for the MR (**Figure 5**). Therefore, one of the most innovative features of the platform is that it allows interactions between real and virtual IWs. These interactions enable high-complexity tests, with a substantial number of objects, devices, and other wheelchairs. Furthermore, it implies a large reduction in project costs, because building a large number of real IWs is not necessary to perform interaction tests [21].

IntellWheels Hardware Module

To be intelligent, an electric-powered wheelchair needs to sense its surroundings; plan its next actions; and react according to environment changes, user commands, and goals. Thus, to provide these capabilities to electric-powered wheelchairs, we developed a generic hardware framework (designed to be flexible enough to fit most commercial wheelchairs) [22]. This framework contains a set of devices that can be classified according to their functionality into three blocks: user inputs (traditional joystick, universal serial bus [USB] joystick, head gestures, keyboard, facial expression, and voice), IW sensors (sonar, encoder, webcam, infrared), and other hardware devices (control/data acquisition board, power module, personal computer notebook). **Figure 6** depicts the resulting proposed architecture of the IntellWheels hardware framework.

User Inputs

To allow people with distinct disabilities to drive the IW, the platform currently presents six inputs. Thus, we aim to give options to the patients and let them choose the most comfortable and suitable input. The current input devices progress from traditional joysticks to facial-expression recognition and are detailed below:

- Traditional joystick. Although this input device is the most common way to drive wheelchairs, it may not be suitable for people with severe injuries like tetraplegia and restricted arm movements.
- USB joystick. This kind of joystick has the advantage of including many configurable buttons that can be customized to execute high-level actions.
- Head gesture. Such an input device is a friendly human-machine interface that allows elderly and disabled people to steer the IW based on their head movements (**Video 1**).
- Keyboard and touch screen. These devices can be used to configure the IW parameters and also as alternatives to control the wheelchair.
- Facial expressions. By using an ordinary webcam, this input device recognizes some simple facial expressions, using them as inputs to execute basic commands (e.g., go forward, right, and left) through high-level commands (e.g., go to the dining hall, go to the bedroom).

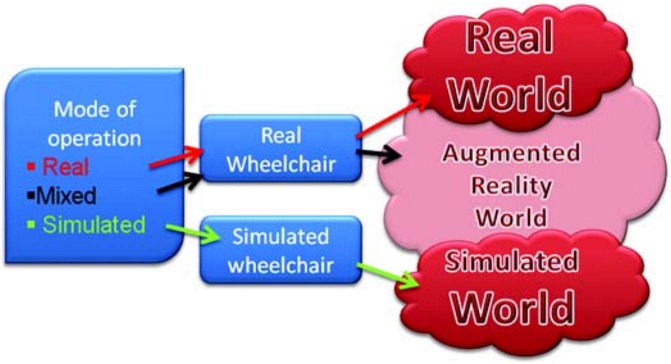


Figure 5.
IntellWheels modes of operation.

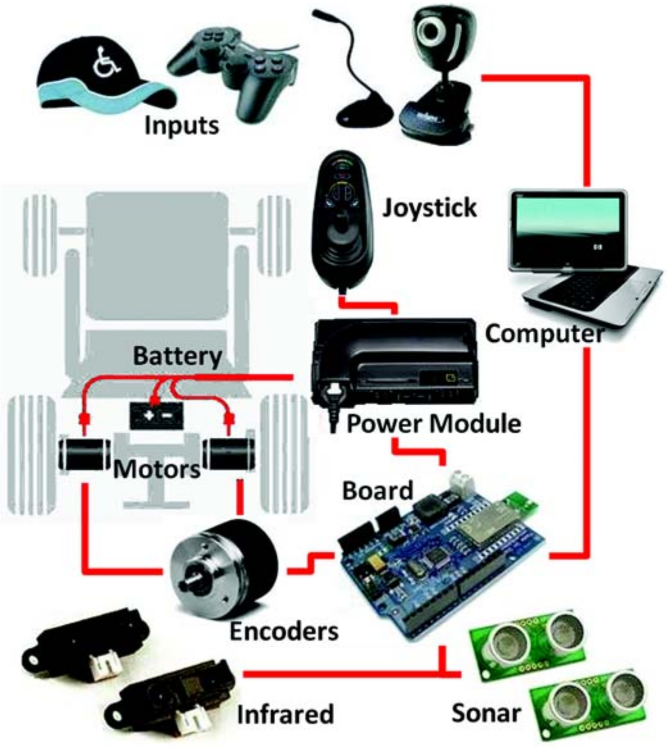


Figure 6.
Architecture of IntellWheels hardware framework.

- Voice. Using commercial speech-recognition software [23], we developed the necessary conditions and applications to command the wheelchair by using the voice as an input. The system uses a standard microphone to capture and analyze the sound using the speech-recognition module (**Video 2**).

Sensors

To compose the IW hardware framework and endow the wheelchair with the ability to avoid obstacles, follow walls, and perceive unevenness in the ground, we designed one U-shaped bar with a set of 8 ultrasound sensors and 12 infrared sensors . The hardware framework also includes two encoders assembled on the wheels (allowing the tools to measure distance, speed, and position) and a webcam to read artificial landmarks and refine the odometry.

Other Hardware Devices

The other devices present in the hardware framework are—

- Control/data acquisition board. The interface board is used to gather sensor information and send the reference to the power module to control both motors. This board connects the platform to the computer host via USB.
- Power module. This converts the control command into a power signal that drives the wheelchair.
- Commercial notebook. To run the platform, we used a notebook computer (HP Pavilion tx1270EP, AMD Turion 64 X2 TI60, Hewlitt-Packard Company; Palo Alto, California).

Simulation Module

The IntellWheels simulator is a customization of the "Cyber-Mouse" simulator [24]. The Cyber-Mouse simulator presents several useful characteristics for IW simulation, such as the simulation of different environments, differential robots with two wheels, and some sensors (e.g., compass and proximity sensors, GPS [global positioning system]). In addition to the simulation server, it also contains a two-dimensional simulation viewer specific to the Cyber-Mouse competition [25].

The IntellWheels simulation module preserves the Cyber-Mouse conceptual architecture but significantly adjusts the robot model and the collision-detection policies. This module implements a simulator that creates a virtual world to safely, easily, and inexpensively run experiments with IWs. Furthermore, the simulator's involvement in the project is even greater as the notion of MR is introduced. **Figure 7** depicts the possible connections with the simulation server, e.g., real wheelchair agents, virtual wheelchair agents, virtual door agents, viewer agents, and medical agents. Such types of interactions between the real and virtual worlds create an MR environment.

The MR support stretches the IntellWheels simulator's capabilities beyond merely testing algorithms. Thus, we can evaluate the reaction of a real IW in a more dynamic scenario—with moving obstacles, complex maps, and other intelligent agents moving around. In other words, a real IW connected to the simulator is capable of interacting with virtual objects. The perception agent uses the data gathered from the real encoders to compute and then send the wheelchair's position to the simulation server. Once the data are received, the simulator places the IW virtual body into its respective position and returns the perception of the virtual proximity sensor's perception to the real wheelchair agent. Next, the real wheelchair agent combines the data from real and virtual proximity sensors, computes the motor power, and sends it to the real wheelchair . The IntellWheels simulator is fully described by Braga et al. [26], who analyze its constraints regarding the simulation of IW.

Visualization is an important aspect of human understanding, since human beings process graphic information preconsciously [27–28]. Keeping that in mind, the IntellWheels viewer was developed to represent the IWs and the environment (e.g., map with its walls, doors,

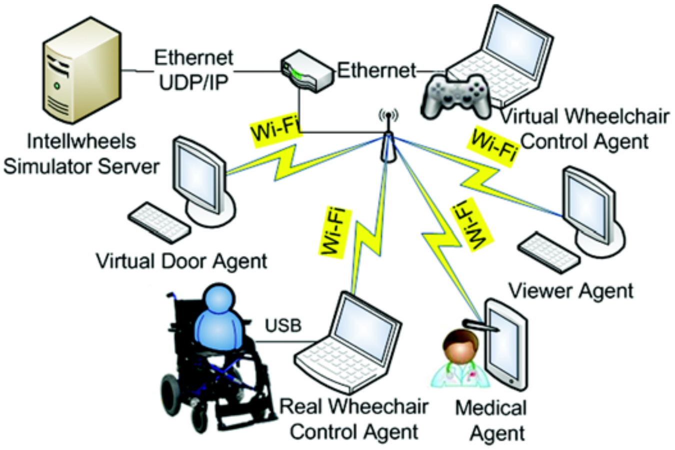


Figure 7.
IntellWheels simulator architecture. UDP/IP = user datagram protocol over internet protocol network, USB = universal serial bus.

objects). The viewer is connected to the simulator through the user datagram protocol (UDP) to exchange XML (extensible markup language) messages. In every simulator step, new world state data, including robot information, are sent to the viewer to update its graphical representation. Robots and environments can be visualized in two and three dimensions. The three-dimensional visualization uses OpenGL technology [29], allowing a first-person view (similar to how a real wheelchair driver would see the world) or a free-camera view.

Planning Module

The planning module is responsible for creating a sequence of high-level actions that are required to achieve the global goal. It comprehends the strategic layer of the control strategy and is implemented by the intelligent agent of the MAS.

The planner used in this work was first implemented based on the Stanford Research Institute Problem Solver algorithm, but it is currently being replaced by a planning graph methodology with planning domain definition language (PDDL) [30]. The planning graph is a powerful data structure that encodes information about which states may be reachable; in other words, it consists of a sequence of levels that correspond to a set of state and/or actions. The PDDL has been used to describe our problem and domain.

Another duty of this module is to generate a path to achieve the objectives proposed by the planner, considering information from the world model. To find a path from a given initial point to a given goal point, the system includes an adapted A* Algorithm [22,31].

Navigation Module

The navigation module encloses a wide set of algorithms responsible for performing the wheelchair's sensors treatment, localization, and mapping (**Video 3**). The suite of functions pertaining to this module is currently implemented by the perception agent, explained in detail in Braga et al. [22].

Control Module

The IntellWheels control module is offset by tactical and basic control layers. The tactical layer is responsible for subdividing the path calculated by the planning module into basic forms (lines, circles, and points) and for computing the wheelchair's linear and angular speeds to put the wheelchair into motion [31].

The basic control layer, the lowest level of control, essentially consists of computing the speed reference for each wheel. These references are then transferred by serial communication to the interface board and controlled by a digital PID controller implemented in the control/data acquisition board.

Communication Module

Safe communications in open transmission systems, safe navigation, and obstacle avoidance are some of the constraints applicable to mobile robots and IWs. With the proliferation of Wi-Fi technologies and devices, the current way in which communications occur is evolving. While these new technologies present advantages, they also have some disadvantages, specifically in the field of safety-related systems or safety-critical systems (a system that in the event of a failure can damage individuals, properties, or the environment) [32–33].

If a mobile robot is a safety-related system or part of one, the communication system must prevent failures and prove to be safe for unauthorized access while maintaining the desired level of compatibility with the system's available physical media transmission layers. To address and solve these issues, one must follow the standard [34], which describes the known threats to communications and their defensive methods applicable for safety-critical systems that use open-transmission media layers.

Usually, a multiagent platform such as the Java Agent DEvelopment Framework (JADE) [35] would be used to enable communications and organize the different agents. However, with common multiagent platforms, customizing and enhancing functionalities to better adapt the system to safety-critical problems is not possible. Our solution to this problem was to develop new methods in a new multiagent platform.

The IntellWheels communication system was implemented in Object-Pascal, following the FIPA guidelines for ACL, and a set of services, such as an agent management system, a message transport system, and a directory facilitator. "The system's architecture was designed as five separate layers, with their respective receiving and sending handling methods, and interfaces running in parallel (**Figure 8**). This way, it becomes possible for the user to choose which layers should be applied to the application, without compromising the agent's functionality" [36] while applying the fault-tolerant methods and adhering to the Open Systems Interconnection Reference Model detailed in Malm et al. [32] and CENELEC (European Committee for Electrotechnical Standardization) [34].

"Crucial to this architecture is the election of a Container entity, similar to JADE, and the distribution of a Local Agents List, as well as a Global Agent List, using a message-oriented paradigm. These lists contain the applications' configurations that enable communications and distribution of the public encryption key between agents. The Container was designed to be responsible for the lists maintenance operations that include creation, update, and deletion. However, and contrary to other systems, the Container was not designed as a separate entity or as the base for agents ' creation and their activity. The idea behind this is that it is admissible and probable for a wheelchair to lose network connectivity or to change its network configuration, but it is not acceptable for these changes to cause a system malfunction.

"The Communications layer is responsible for receiving and sending messages from and to the message transport layer. It allows the user to choose between TCP/IP, UDP or even HTTP messages. This layer also prevents the interpretation of repeated messages, present in the physical media, and enables the retransmission of messages, thus preventing packet loss at the network level. It also prevents the application from receiving messages with a size that is larger than the one specified by the user during agent implementation.

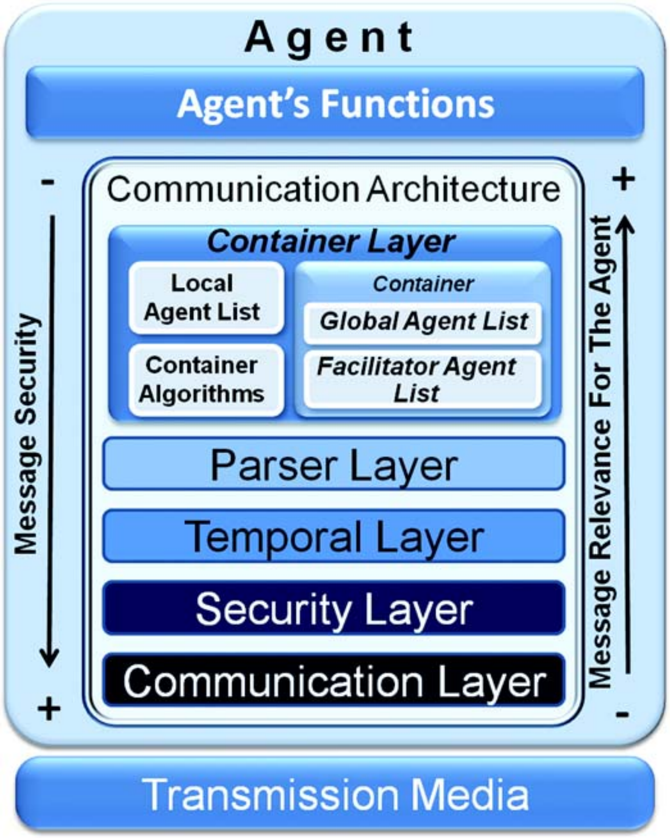


Figure 8.
Local platform structure.

"The Security layer is responsible for the message's security, preventing the interception and modification of messages. The Encryption method is chosen according to the message's destination and the platforms' knowledge at that moment. The possible encryption methods involve the use of a private and public key pair or an AES preshared key. It also performs message integrity checking by crossreferencing the message with the transmitted message's hash" [36].

The Temporal layer is responsible for adding time restrictions to the messages. These restrictions can be seen as a defensive measure. Adding a timestamp to the messages' data enables filtrating of outdated messages.

Finally, the "Parser layer is responsible for the construction of the message according to the FIPA-ACL standard and represented using the normative constant FIPA-SL. It also selects the messages that are accepted by the application according to their correct structure configuration and to the sender's presence in the platform, thus stopping any communication from an unauthenticated application" [36].

Multimodal Interface Module

An interface is an element that establishes boundaries between two entities. Currently, most traditional human-machine interfaces are based on a single and not customizable input/output correlation. An evolution of this paradigm and a way to create a more natural interaction with the user is to establish a multimodal interaction, which contemplates a broader range of modes and channels of communication, such as video, voice, and pen. According to Oviatt, an MMI "processes two or more user input modes—such as speech, pen, touch, manual gesture s, gaze, and head and body movements—in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to naturally recognize occurring forms of human languages or behaviors, and that incorporate one or more recognition-based technologies (e.g., speech, pen, vision)" [37].

The IntellWheels MMI module is designed to allow several input devices to be connected simultaneously (voice, facial expressions, head gestures, keyb oard, touch-screen, and joystick). Thus, it allows the wheelchair to be controlled through input sequences from the same channel of communication or even from the combination of distinct channels (input devices). Through this module, users can create the most suitable input se quence according to their limitations—w hich may be associated with one or more output commands. Furthermore, this application can provide an interaction between the environment and the input methods, so that, at any instance, the input information can be analyzed and checked for reliability to ensure user safety.

The interaction between the MMI module and the input device driver is based on a client/server architecture, in which the MMI module acts as a server and the input device drivers as clients. During the connection, the MMI requests information to the in put device driv er regarding its characteristics (e.g., name, kind and number of inputs). Then, once the connection is established, the input device driver triggers user actions and sends the new state to the MMI module. **Figure 9** depicts the connection between input device drivers and the MMI module.

RESULTS

This section presents the imple mentation and the experiments used to evaluate s ome modules of the platform and its operation as a whole. The following results show the IW prototype as semblage; analysis of the shared control and of the autonomous planning and navigation algorithms; and tests of the interface agent, MR, and multiagent interaction.

IntellWheels Prototype

The first result is the assemblage of a real IW prototype. Following the IntellWhe els guidelines, the prototype was developed based on a c ommercial wheelchair (model Evolution Electr onics, Vassilli; Padova, Italy, http://www.vassilli.it/). The Evolution wheelchair has the following features: two dif ferential-drive rear wheels, two front passive castors, two 12 V batteries (45 Ah) and one traditional joystick. **Figure 10** shows the conventional wheelchair with an integrated IntellWheels hardware module.

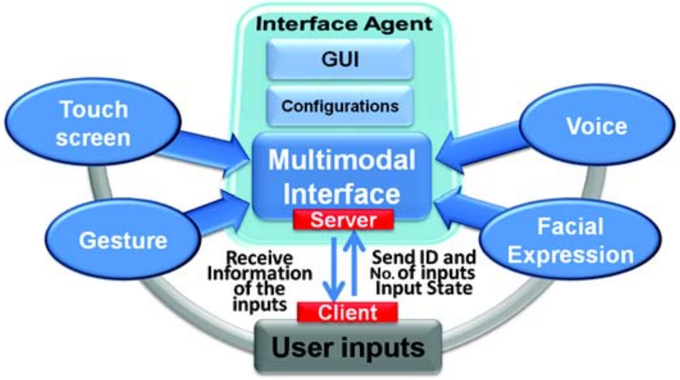


Figure 9.
Multimodal interface module. GUI = g raphical user interface, ID = identification.

Interface Agent

So far, the inte rface agent was developed to help engineers evaluate the wheelchair behavior during the tests. Thus, its current GUI (**Figure 11**) consists of several groups of information. In the upper left corner, a panel contains a camera view and the localization resulting from the landmark recognition. In the bottom left corner, a schematic of a wheelchair shows the distance to nearby objects measured by each sonar and infrared sensor. In the center, a panel shows the information provided by the odometry, the speed of each wheel, and the buttons to choose the operation mode. Finally, the right side of the window contains the information regarding the MMI module (e.g., list of actions , list of inputs, li st of sequences).

The next experiment was designed to test the MMI. This test consisted of a user manually steering the rea l

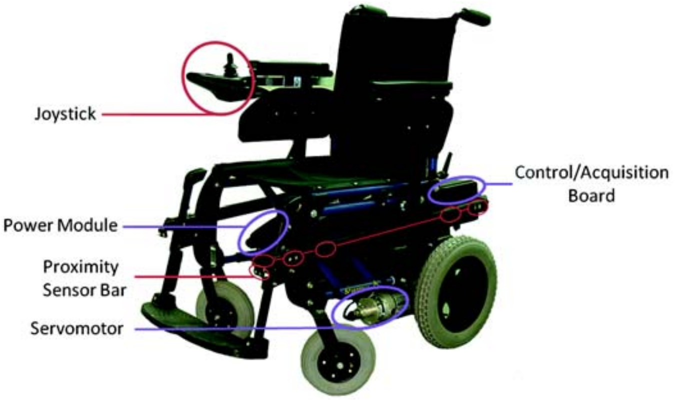
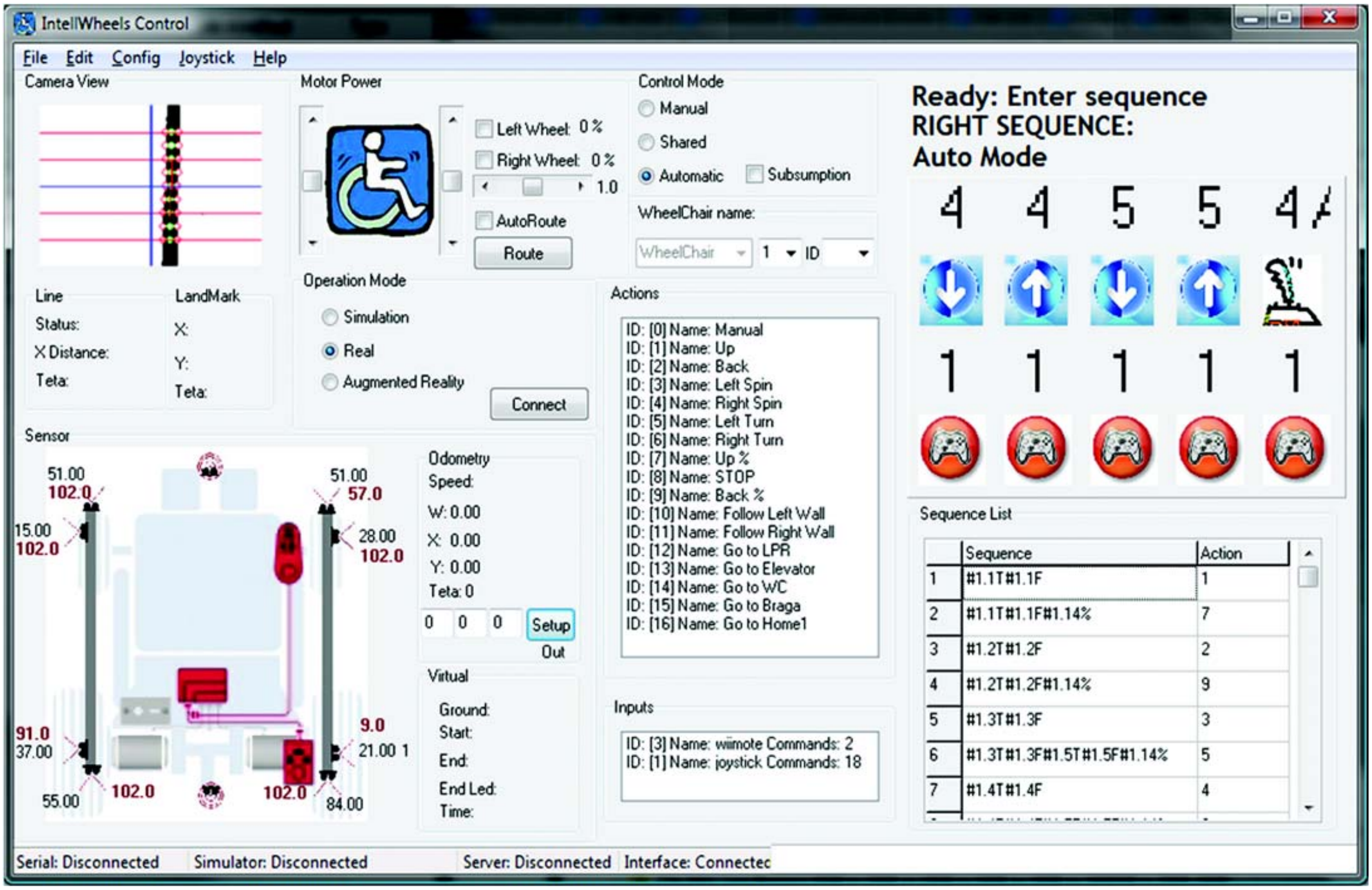


Figure 10.
IntellWheels prototype.



□ Left Wheel: 0 %
□ Right Wheel: 0 %
1.0
□ AutoRoute

○ Manual
○ Shared
● Automatic □ Subsumption

WheelChair name: WheelChair 1 ID

Ready: Enter sequence
RIGHT SEQUENCE:
Auto Mode

Line
Status:
X Distance:
Teta:

LandMark
X:
Y:
Teta:

Operation Mode
○ Simulation
● Real
○ Augmented Reality

Actions

ID: [0] Name: Manual
ID: [1] Name: Up
ID: [2] Name: Back
ID: [3] Name: Left Spin
ID: [4] Name: Right Spin
ID: [5] Name: Left Turn
ID: [6] Name: Right Turn
ID: [7] Name: Up %
ID: [8] Name: STOP
ID: [9] Name: Back %
ID: [10] Name: Follow Left Wall
ID: [11] Name: Follow Right Wall
ID: [12] Name: Go to LPR
ID: [13] Name: Go to Elevator
ID: [14] Name: Go to WC
ID: [15] Name: Go to Braga
ID: [16] Name: Go to Home1

Odometry
Speed:
W: 0.00
X: 0.00
Y: 0.00
Teta: 0
0 0 0
Out

Virtual
Ground:
Start:
End:
End Led:
Time:

Inputs

ID: [3] Name: wiimote Commands: 2
ID: [1] Name: joystick Commands: 18

Sequence List

	Sequence	Action
1	#1.1T#1.1F	1
2	#1.1T#1.1F#1.14%	7
3	#1.2T#1.2F	2
4	#1.2T#1.2F#1.14%	9
5	#1.3T#1.3F	3
6	#1.3T#1.3F#1.5T#1.5F#1.14%	5
7	#1.4T#1.4F	4

Serial: Disconnected | Simulator: Disconnected | Server: Disconnected | Interface: Connectec

Figure 11.
IntellWheels interface agent.

wheelchair along a narrow corridor with obstacles by simultaneously using the voice and head gestures to control the wheelchair (**Figure 12**). Specifically in this test, the control through head gestures remained disabled until the MMI received the voice command "Manual." Then, the user drove the wheelchair through the corridor by using head gestures to move forward and voice commands ("Right spin") to turn the wheelchair.

Shared Control

To evaluate the efficiency of shared control algorithms, eight volunteers each performed one set of four driving tests. Each set comprised four laps in a cluttered environment (**Figure 13**): two laps in the simulated scenario (one with and one without the assistance of the shared control algorithm) and two laps in the real scenario (one with and one without the assistance of the shared wheelchair control). All participants were between 26 and 39 years old and spent around 40 minutes running the experiments.

Volunteers were asked to drive the wheelchair by using a special human-machine interface based on their head gestures. Although all participants were nondisabled, their difficulty in controlling the wheelchair tends to be similar to the difficulty of people with disabilities who have restricted limb movements, since this control method is not usual in their daily tasks.

We analyzed the data collected during the shared control experiments within subjects rather than testing the performance of individuals against each other. This allowed us to estimate whether providing assistance actually helped each individual. By comparing the number of collisions each trial (with and without assistance), we could evaluate the performance of the shared control algorithm in the simulated and the real environments (**Figure 14**).

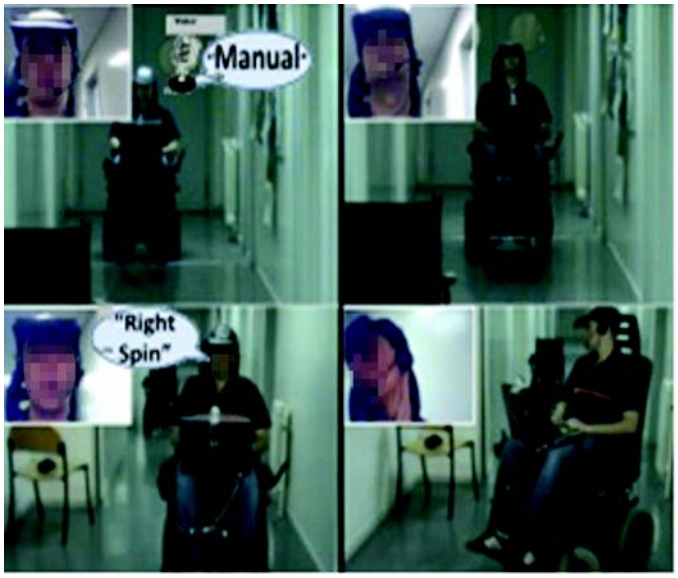


Figure 12.
Real wheelchair movement in corridor with obstacles with use of head gestures and voice control.

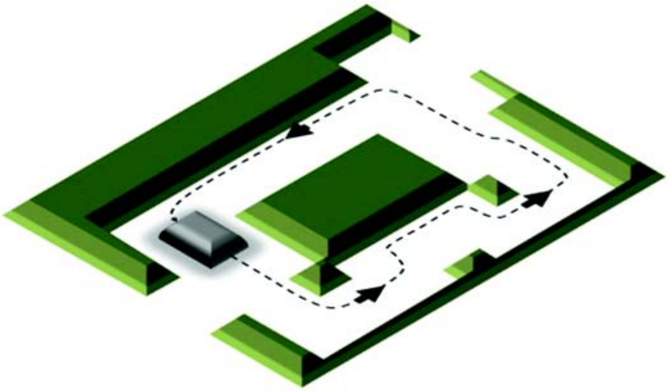


Figure 13.
Closed circuit in which experiments were conducted.

Therefore, experimental data were subjected to a nonparametric test for paired samples (Wilcoxon signed rank one-tailed test) with a confidence level of 95 percent ($p < 0.05$). Thus, in the real environment, the results of the Wilcoxon indicate a significant difference between the number of collisions with and without the shared control paradigm ($T = 0.00$, $n = 8$, $p = 0.01$). Furthermore, in the simulated environment, results also indicate a significant difference between the number of collisions with and without the shared control paradigm ($T = 0.00$, $n = 8$, $p = 0.009$). Therefore, our analysis provides evidence that the shared control paradigm is providing assistance that may reduce the number of collisions.

Multiagent Interaction

This experiment tested the cooperation of heterogeneous agents in a simulated hospital scenario. Using a robotic agent, we connected a virtual wheelchair to the simulator and executed the test of automatic door opening. **Figure 15** shows a series of print screens of the IntellWheels three-dimensional viewer during this automatic door test. The IW agent communicates with the door agent and controls the chair's movement forward, regardless of what its own proximity sensors detect. Otherwise, after the negotiation, the door agent waits for the wheelchair to be detected to open the door and closes it only when the sensors stop detecting it.

Planning and Autonomous Navigation

The goal in this subsection is to present the results of the wheelchair's autonomous planning and navigation. Using the planner module and the IntellWheels simulator to simulate the wheelchair's displacement, this test consisted of planning and transporting the patient between two different rooms. The sequence of actions required to achieve this simple goal includes picking up patient 1 in room 1, carrying this patient to room 2, and finishing the wheelchair journey in the hall. In this example, our final objective was configured at the planner as On(P1,Room2) ^ WithoutW(P1) ^ On(W1,Hall), meaning that in the final state,

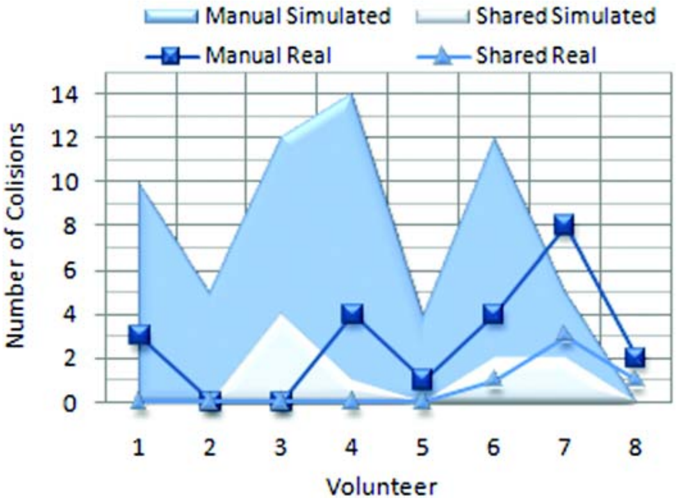


Figure 14.
Number of collisions per volunteer in simulated and real scenarios.

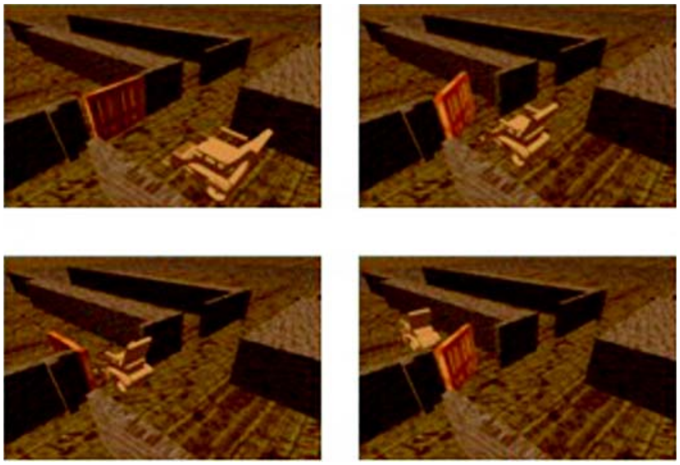


Figure 15.
Automatic opening of door.

patient 1 should be in room 2 (the bedroom), patient 1 should be without the wheelchair, and the wheelchair should be at room hall (the hall). The world state before the action, the resulting plan, and the following state are represented in **Figure 16**. The final plan achieved for this task is represented in the actions panel of **Figure 16**. This plan consists of the following action sequence: get (W1,P1,Room1), carry (W1 ,P1,Room1,Room2), leave (W1,P1,Room2), move (W1,Room2,Hall). The wheelchair starts by getting patient 1 at room 1, carrying him from room 1 to room 2, leaving him at room 2, and finally moving itself from room 2 to the hall. The planner developed is capable of planning any type of high-level plan with multiple wheelchairs and patients.

The final route and the travelled path, based on the previously mentioned plan, can be observed in **Figure 17**. This route contains the four basic movements that the IW needed to perform to achieve the final objective: from the initial point in room 1 (to pick up patient 1), to bedroom 2, and then after leaving patient 1, going (empty) to the hall.

Mixed Reality

The first MR experiment was designed to evaluate the interaction of a real IW with a virtual environment. MR experiments allow testing of the real IW in several scenarios (e.g., narrow corridors, crowded places, moving objects) safely (free of collisions with real objects, reducing the risk of damaging the equipment) and inexpensively (reduced time demanded to create scenarios, minimum infrastructure cost). For this test, the wheelchair was operated in the autonomous mode with obstacle avoidance assistance and the simulator loaded with a map modeled to represent the real test environment. To start the experiment, we set up the wheelchair in the MR mode and positioned it in the middle of the corridor. Real proximity sensors were disconnected and their perception data replaced by their simulated counterparts. After that, the IW was asked to move straightforward through the corridor.

Figure 18 depicts the results of such interaction. In this image sequence, one can observe the real IW avoiding a virtual box perceived by the virtual sensors; note that the real objects (like the box and the walls) of the real environment could not be sensed by the IW once its proximity sensors remained disconnected during the experiments.

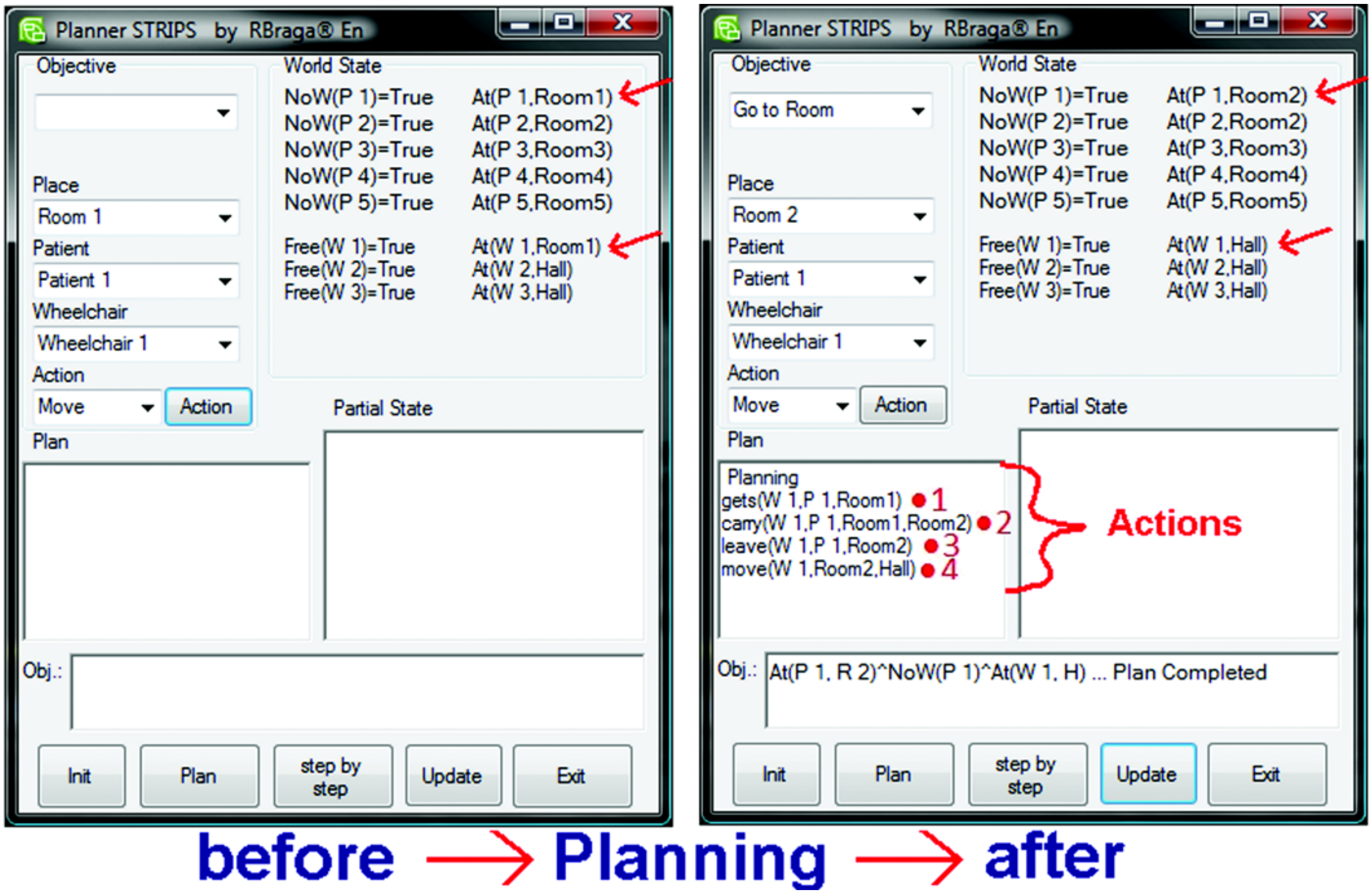
Finally, a simple MR environment was developed to help patients improve their ability to steer the wheelchair. Drills of patients with real wheelchairs in virtual scenarios can be performed with some realism, eliminating the risk of injuries and reducing the stress of steering the wheelchair in a real environment (**Figure 19**).

DISCUSSION

The IntellWheels platform was designed to be a framework for research and development of IWs. Thus, this project does not intend to deliver a prototype for people with a specific injury or reduced dexterity but rather to create a generic development tool. With this in mind, the platform was designed as an MAS containing several modules, such as planning, control, MMI, simulation, navigation, hardware framework, and communication framework. At first, we were not concerned with the individual performance of each module (algorithms), but with its integration in the system. At the same time, we did not focus on the intelligence level of each agent, but on the intelligence level that may emerge from the system as a whole.

Although several concepts have been proposed through the IntellWheels platform, we have not yet practiced or tested all of them. Thus, one may sort the proposed features into three stages: implemented and tested, implemented but not tested, and planned but not implemented.

The first case, implemented and tested, includes those features that are closely related to the overall idea of the platform and were evaluated in the "Results" section. Thus, with the IW prototype, we verified the compatibility of the IntellWheels hardware framework with



Before:

Objective:
Place: Room 1
Patient: Patient 1
Wheelchair: Wheelchair 1
Action: Move

World State:
NoW(P 1)=True At(P 1,Room1)
NoW(P 2)=True At(P 2,Room2)
NoW(P 3)=True At(P 3,Room3)
NoW(P 4)=True At(P 4,Room4)
NoW(P 5)=True At(P 5,Room5)
Free(W 1)=True At(W 1,Room1)
Free(W 2)=True At(W 2,Hall)
Free(W 3)=True At(W 3,Hall)

After:

Objective: Go to Room
Place: Room 2
Patient: Patient 1
Wheelchair: Wheelchair 1
Action: Move

World State:
NoW(P 1)=True At(P 1,Room2)
NoW(P 2)=True At(P 2,Room2)
NoW(P 3)=True At(P 3,Room3)
NoW(P 4)=True At(P 4,Room4)
NoW(P 5)=True At(P 5,Room5)
Free(W 1)=True At(W 1,Hall)
Free(W 2)=True At(W 2,Hall)
Free(W 3)=True At(W 3,Hall)

Plan:
Planning
gets(W 1,P 1,Room1) ● 1
carry(W 1,P 1,Room1,Room2) ● 2
leave(W 1,P 1,Room2) ● 3
move(W 1,Room2,Hall) ● 4

Obj.: At(P 1, R 2)^NoW(P 1)^At(W 1, H) ... Plan Completed

Figure 16.
Planning experiment.

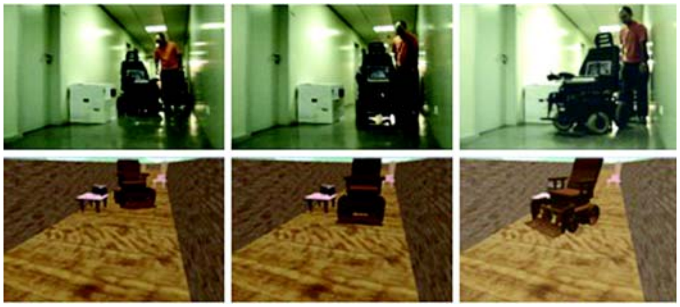


Figure 17.
Mixed-reality test: Real intelligent wheelchairs (upper images) interacting with virtual objects (lower images).

common powered wheelchairs. In addition, the visual characteristics and ergonomics of the wheelchair were not affected by the assemblage of the hardware devices, achieving another proposed goal. Furthermore, we must emphasize that the objective of building a low-cost IW was also achieved, with its hardware cost kept under US$4,000 ($2,400 for the powered wheelchair and $1,500 for sensors and other hardware devices). Nevertheless, the platform does not prevent the increment of new or additional sensors to the hardware framework, but on the contrary, it is open for the addition of other sensors as soon as their impact on the final cost of the prototype is reduced (e.g., laser range finders should continue to come down in price in the coming years).

The second case, implemented but not tested, has to do with the skills (communication, localization, and facial expressions input) that were implemented and whose results were not shown in this article. As mentioned before, in this article we aimed to introduce the platform as a whole and not to test each IW skill specifically. However, each of these skills has already been evaluated and their results published in previous conferences. The communication module was described in

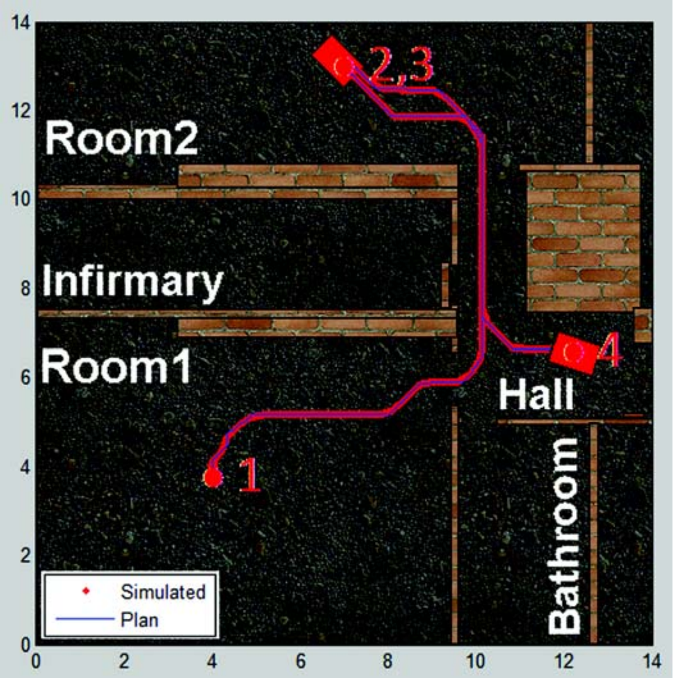


Figure 18.
Planned and executed routes of simulated intelligent wheelchair in autonomous operation mode.

Figure 19.
Mixed-reality environment developed to assist in rehabilitation of patients.

Cunha et al. [38], which details the development of the communication system as a means to enable fault-tolerant communications in open transmission systems and as a facilitator for entity collaboration. The results presented in Cunha et al. [38] establish a comparison with JADE and demonstrate the effectiveness and adequacy of the proposed communication model to mobile robots in dynamic environments. In Braga et al. [22], a probabilistic odometry motion model for an active localization system was discussed and tested. One solution to reduce the localization error was to compute the uncertainty (variance) of the odometry. Thus, whenever the uncertainty overcomes a given threshold, the wheelchair's path is replanned and forced to pass through the nearest landmark—resetting the localization error. The facial expression input makes use of image-processing algorithms to detect features, such as color segmentation and edge detection, followed by the application of a neural network to detect the user's desire. The results shown in Faria et al. [39] provide evidence that comfortably driving an IW with the use of facial expressions is possible. However, such input still demonstrates some limitations regarding color segmentation (much sensitivity to large light variations and slight color shifts) and shape extraction (improve precision without increasing the processing time). Finally, the last case, planned but not implemented, will be discussed in the "Conclusions" section as future work.

CONCLUSIONS

This article presented the design and implementation of the IntellWheels development platform for IWs. The project is based on three main innovative ideas. First, the IntellWheels project is based on a generic IW framework that enables easy development of new IWs and control algorithms. The framework is flexible enough to enable the easy transformation of commercial wheelchairs into IWs with minor hardware changes. Furthermore, it facilitates the introduction of new modules and algorithms in the IW system.

Second, the IW interaction's methodology is based on a flexible MMI. MMI experiments were performed to verify the module efficiency and the wheelchair controllability with the use of several input devices. The results achieved confirmed the MMI capabilities—except for the voice module, which has demonstrated a lack of robustness in noisy backgrounds. Therefore, we verified that the MMI allows the user to control the wheelchair through sequences and combinations of inputs (e.g., buttons, voice commands, facial expressions, stick direction) from the same or different input devices.

Finally, the third contribution is related to the MR scenarios provided by the IntellWheels simulator. The simulator has demonstrated that it is capable not only of simulating environments and wheelchairs but also of creating a scenario that enables interaction between real and virtual objects (e.g., wheelchairs, tables, walls, obstacles).

Some future directions include the development of an intelligent-input decision control and its integration with the MMI module. Such intelligent-input decision control may be responsible for establishing confidence levels and for managing inputs according to its perception—avoiding conflicts, noise, or other dangerous situations. Equally important are improvements in the robustness of facial-expression recognition, text-to-speech output, and some kind of virtual user assistant to improve the user integration process. Moreover, the creation of an intuitive and friendly GUI designed for people who are elderly or have severe disabilities is necessary. Other improvements concern the localization module, including new methodologies to improve uncertainty about the wheelchair's position and orientation. New approaches may include map matching, visual odometry, and global localization systems to reduce localization errors.

ACKNOWLEDGMENTS

Author Contributions:
Study concept and design: R. A. M. Braga, M. Petry, L. P. Reis, A. P. Moreira.
Acquisition of data: R. A. M. Braga, M. Petry.
Analysis and interpretation of data: R. A. M. Braga.
Drafting of manuscript: R. A. M. Braga, M. Petry.
Critical revision of manuscript for important intellectual content: L. P. Reis, A. P. Moreira.
Obtained funding: L. P. Reis.
Study supervision: L. P. Reis, A. P. Moreira.
Financial Disclosures: The authors have declared that no competing interests exist.
Funding/Support: This material was based on work partially supported by the Artificial Intelligence and Computer Science Laboratory 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 FCT/RIPD/ADA/109636/2009).
Institutional Review: Human subjects approval was not required. However, all subjects were informed about exact characteristics of test and gave informed oral consent before participating.
Additional Contributions: The authors would like to thank the volunteers who participated in this study. The first and the second authors also acknowledge CAPES-Brazil (grant 4142-05-5) and FCT (grant SFRH/BD/60727/2009) for their PhD scholarship funding.

REFERENCES

1. Simpson RC. Smart wheelchairs: A literature review. J Rehabil Res Dev. 2005;42(4):423–36. [PMID: 16320139] DOI:10.1682/JRRD.2004.08.0101
2. Madarasz R, Heiny L, Cromp R, Mazur NM. The design of an autonomous vehicle for the disabled. IEEE J Robot Autom. 1986;2(3):117–26.
3. Hoyer H, Hölper R. Open control architecture for an intelligent omnidirectional wheelchair. Proceedings of the 1st TIDE Congress; 1993 Apr 6–7; Brussels, Belgium. Amsterdam (the Netherlands): IOS Press; 1993. p. 93–97.
4. Levine SP, Bell DA, Jaros LA, Simpson RC, Koren Y, Borenstein J. The NavChair Assistive Wheelchair Navigation System. IEEE Trans Rehabil Eng. 1999;7(4):443–51. [PMID: 10609632] DOI:10.1109/86.808948
5. Miller D. Assistive robotics: An overview. Lect Notes Comput Sci. 1998;1458:126–36. DOI:10.1007/BFb0055975
6. Prassler E, Scholz J, Fiorini P. A robotic wheelchair for crowded public environment. IEEE Robot Autom. 2001; 8(1):38–45. DOI:10.1109/100.924358
7. Wellman P, Krovi V, Kumar V. An adaptive mobility system for the disabled. Proceedings of the IEEE International Conferences on Robotics and Automation; 1994 May 8–13; San Diego, CA. Los Alamitos (CA): IEEE; 1994. p. 2006–11.
8. Jia P, Hu HH, Lu T, Yuan K. Head gesture recognition for hands-free control of an intelligent wheelchair. J Ind Robot. 2007;34(1):60–68. DOI:10.1108/01439910710718469
9. Ng PC, De Silva LC. Head gestures recognition. Proceedings of the International Conference on Image Processing; 2001 Oct 7–10; Thessaloniki, Greece. Los Alamitos (CA): IEEE; 2001. p. 266–69.
10. Adachi Y, Kuno Y, Shimada N, Shirai Y. Intelligent wheelchair using visual information on human faces. Proceedings of the International Conference in Intelligent Robots and Systems; 1998 Oct 13–17; Victoria, Canada. Los Alamitos (CA): IEEE; 1998. p. 354–59.
11. Lakany H. Steering a wheelchair by thought. IEEE International Workshop on Intelligent Building Environments; 2005; Colchester, UK. Glasgow (UK): University of Strathclyde; 2009. p. 199–202.
12. Rebsamen B, Burdet E, Guan C, Zhang H, Teo CL, Zeng Q, Laugier C, Ang MH Jr. Controlling a wheelchair indoors using thought. IEEE Intell Syst. 2007;22(2):18–24. DOI:10.1109/MIS.2007.26
13. Hamagami T, Hirata H. Development of intelligent wheelchair acquiring autonomous, cooperative, and collaborative behavior. Conf Proc IEEE Int Conf Syst Man Cybern; 2004 Oct 10–13. Los Alamitos (CA); 2004. p. 3525–30.
14. Katevas NL, Sgouros NM, Tzafestas SG, Papakonstantinou G, Beattie P, Bishop JM, Tsanakas P, Koutsouris D. The autonomous mobile robot SENARIO: A sensor aided intelligent navigation system for powered wheelchairs. IEEE Robot Autom. 1997;4(4):60–70. DOI:10.1109/100.637806
15. Bourhis G, Horn O, Habert O, Pruski A. An autonomous vehicle for people with motor disabilities. IEEE Robot Autom. 2001;8(1):20–28. DOI:10.1109/100.924353
16. Lankenau A, Rofer T. A versatile and safe mobility assistant. IEEE Robot Autom. 2001;8(1):29–37. DOI:10.1109/100.924355

17. Mazo M. An integral system for assisted mobility. IEEE Robot Autom. 2001;8(1):46–56. DOI:10.1109/100.924361
18. Martens C, Ruchel N, Lang O, Ivlev O, Graser A. A FRIEND for assisting handicapped people. IEEE Robot Autom. 2001;8(1):57–65. DOI:10.1109/100.924364
19. Simpson R, LoPresti E, Hayashi S, Nourbakhsh I, Miller D. The smart wheelchair component system. J Rehabil Res Dev. 2004;41(3B):429–42. [PMID: 15543461] DOI:10.1682/JRRD.2003.03.0032
20. Foundation for Intelligent Physical Agents [Internet]. 2010. Available from: http://www.fipa.org
21. Braga RA, Petry M, Moreira AP, Reis LP. INTELLWHEELS—A development platform for intelligent wheelchairs for disabled people. Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics; 2008; Funchal, Madeira, Portugal: ICINCO. p. 115–21.
22. Braga RA, Petry MR, Moreira AP, Reis LP. Concept and design of the IntellWheels development platform for intelligent wheelchairs. Lect Notes Electr Eng/Informa Control Autom Robot. 2009;37:191–203.
23. Embedded ViaVoice [Internet]. Armonk (NY): IBM; 2009. Available from: http://www-306.ibm.com/software/pervasive/embedded_viavoice/
24. Lau N, Pereira A, Melo A, Neves A, Figueiredo J. Ciber-Rato: Um ambiente de simulação de robots móveis e autónomos. [Cyber-Mouse: A simulated environment for autonomous mobile robots.] DETUA. 2002;3(7). Portuguese.
25. Lau N, Pereira A, Melo A, Neves A, Figueiredo J. Ciber-Rato: Uma competição robótica num ambiente virtual. [Cyber-Mouse: Robotics competition in a virtual environment.] DETUA. 2002;3(7):647–50. Portuguese.
26. Braga RA, Malheiro P, Reis LP. Development of a realistic simulator for robotic intelligent wheelchairs in a hospital environment. Proceedings of the RoboCup 2009 Symposium; 2009; Graz, Austria.
27. Lau N, Pereira A, Melo A, Neves A, Figueiredo J. O visualizador do ambiente de simulação Ciber-Rato. [Viewer for Cyber-Mouse simulated environment.] DETUA. 2002;3(7):651–54. Portuguese.
28. Rohrer MR. Seeing is believing: The importance of visualization in manufacturing simulation. Proceedings of the 32nd Conference on Winter Simulation; 2000. San Diego (CA): Society for Computer Simulation International; 2000. p. 1211–16.
29. Woo M, Neider J, Davis T. OpenGL programming guide: The official guide to learning OpenGL, version 1.2. 3rd ed. Reading (MA): Addison-Wesley; 1999.
30. Fox M, Long D. PDDL2.1: An extension to PDDL for expressing temporal planning domains. J Artif Intell Res. 2003;20:61–124.
31. Braga RA, Petry MR, Reis LP, Oliveira EC. Multi-level control of an intelligent wheelchair in a hospital environment using a Cyber-Mouse simulation system. Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics; 2008; Funchal, Madeira, Portugal: ICINCO. p. 179–82.
32. Malm T, Hérard J, Bøegh J, Kivipuro M. Validation of safety-related wireless machine control systems. Oslo (Norway): Nordic Innovation Centre; 2007.
33. Fowler K. Mission-critical and safety-critical development. IEEE Instrum Meas Mag. 2004;7(4):52–59. DOI:10.1109/MIM.2004.1383466
34. European Committee for Electrotechnical Standardization (CENELEC). EN 50159-2 Railway applications—Communication, signalling and processing systems—Part 2: Safety related communication in open transmission systems. Brussels (Belgium): CENELEC; 2001.
35. Bellifemine FL, Caire G, Greenwood D. Developing multi-agent systems with JADE. Hoboken (NJ): Wiley; 2007. DOI:10.1002/9780470058411
36. Cunha FM, Braga RA, Reis LR. Evaluation of a communication platform for safety critical robotics. Lecture Notes in Computer Science 6114. Artificial Intelligence and Soft Computing 10th International Conference, ICAISC; 2010 Jun 13–17; Zakopane, Poland. New York (NY): Springer 2010. p. 239–46.
37. Oviatt S. Multimodal interfaces. In: Sears A, Jacko JA, editors. The human-computer interaction handbook. New York (NY): Lawrence Erlbaum; 2002. p. 286–304.
38. Cunha FM, Braga RA, Reis LP. A cooperative communications platform for safety critical robotics: An experimental evaluation. Adv Intell Soft Comput. 2010;70:151–56.
39. Faria PM, Braga RA, Valgôde E, Reis LP. Interface framework to drive an intelligent wheelchair using facial expressions. Proceedings of the IEEE International Symposium on Industrial Electronics; 2007 Jun 4–7; Vigo, Spain. Los Alamitos (CA): IEEE; 2007. p. 1791–96.

Submitted for publication August 9, 2010. Accepted in revised form December 30, 2010.

This article and any supplementary material should be cited as follows:
Braga RA, Petry M, Reis LP, Moreira AP. IntellWheels: Modular development platform for intelligent wheelchairs. J Rehabil Res Dev. 2011;48(9):1061–76.
DOI:10.1682/JRRD.2010.08.0139

ResearcherID: Luis Paulo Reis, PhD: E-9707-2011