Evaluation of a Communication Platform for Safety Critical Robotics

Frederico M. Cunha, Rodrigo A. M. Braga, and Luis P. Reis

Artificial Intelligence and Computer Science Laboratory - LIACC,
Faculty of Engineering of University of Porto - FEUP,
Rua Dr. Roberto Frias s/n, Porto, Portugal
fredericom.cunha@gmail.com,rodrigo.braga@fe.up.pt,lpreis@fe.up.pt

Abstract. As the number of handicapped people increases worldwide, Intelligent Wheelchairs (IW) are becoming the solution to enable a higher degree of independence for wheelchair users. In addition, IW Projects relevance is increasing, mainly in the fields of robotics and safety-related systems due to their inherent and still unresolved problems related with environment uncertainty, safe communications and collaboration methodologies. This paper describes the development of new communication system, based on multi-agent systems (MAS) methodologies and motivated by Intelligent Wheelchair systems, as a mean to enable fault-tolerant communications in open transmission systems and as a agent collaboration enabler. It provides an overview of the related work, the background and the main constraints to system development. It proposes and discusses a new communication model, based on messages, for multi-agent systems, that tackles the problems that exist in a dynamic and uncertain environment in the field of mobile robotics. The achieved results enable us to conclude on the effectiveness of the proposed comunication model and its adequacy to the field of mobile robots in dynamic environments while establishing a comparison with a commonly used MAS, JADE.

Key words: Agent-based system development, Agent development techniques, tools and environments, Development environments, Agent Communication, Communication protocols, Safety Critical Systems.

1 Introduction

The World Health Organization (WHO), estimates that around 2% of world population (130 million people) live with physical handicaps. This increasing number is due to several physical disabilities caused by congenital deficiencies, diseases, accidents, wars, and the aging of population as consequence of the increase in life expectancy, [1].

The most common aid for this kind of mobility problem is the wheelchair, specifically the electric wheelchair. However, it does not provide the autonomous life that most users want. To address this problem, numerous Intelligent Whellchair (IW) related projects have been announced, and are under development in the last years. The increase study of this problem, led to a globally accepted view of the main functional requirements for such systems. According to [8] and further developed by, [3], the main functions of an IW, can be categorized as the following:

- **Interaction with the user** this includes hand based control(such as joystick, keyboard, mouse, touch screen), voice based control, vision based control and other sensor based control;
- **Autonomous navigation** this must provide safety, flexibility and robust obstacle avoidance;
- **Communication systems** to enable and provide interaction with other devices, such as other IWs, automatic doors, remotely operated control software for medical staff.

Although many IW projects exist, the majority tends to concentrate their efforts in the interface with the user or in the navigational system. The communication system is rarely described in these projects and scarcely treated as an important and vital piece of an Intelligent Wheelchair. A common solution seen for the communication system is the use of CORBA based systems, or other technologies that enable communication through object sharing techniques, as seen in [10]. These allow an easier and fast development but are rather limited when addressing the problems of mobile robotics.

The SENA robot, presented in [6], is one of the few IW projects that approaches the communication system. One of the first references to it was the proposal of an architecture for this type of system, ACHRIN. This system although started as a rigid communication platform, it latter evolved into a Multiagent Systems (MAS) based communications, [5], thus taking advantage of MAS maturity, robustness, scalability and easy management of information and entities.

This work's main objectives are: to identify in the field of mobile robotics, with emphasis on IW cooperative communication, the main constraints to a safe and secure communication system, propose and test the solutions that can address the found constraints and compare the test results to previously used methods in a adequate environment.

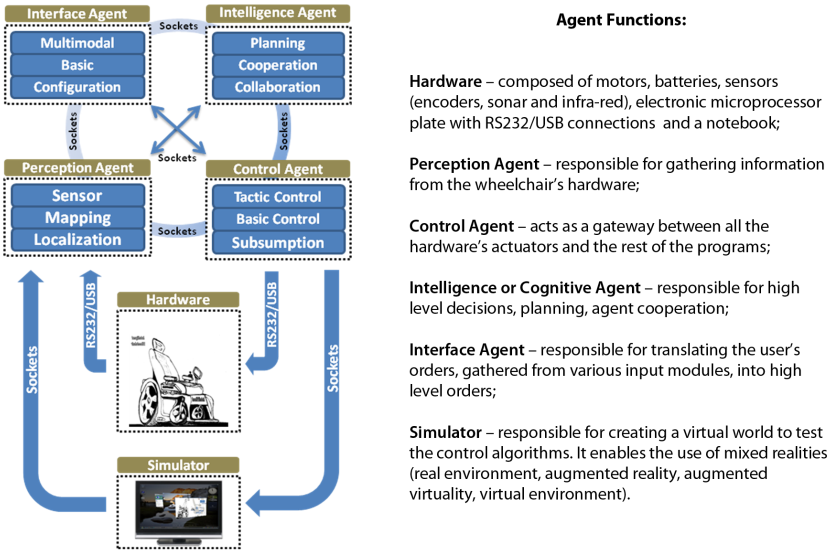
The rest of this article is divided into 4 additional sections. Section 2 provides an overview of the IntellWheels project and its different modules as the motivation for this study. Section 3 provides a description of the applicable constrains, linking them to the described project and to the field of mobile robotics. A detailed description of the proposed solutions is also given in this section. Section 4 describes the test environment, the applied test methodology and presents the test results. Section 5 discusses the relevance of the test results and the applicability of the proposed methods to the field of mobile robots and robots in dynamic and safety critical environments.

2 Intellwheels Project

The IntellWheels' main objective is to create a complete intelligent system, hardware and software, which can be easily integrated into any commercially available electric wheelchair with minor modifications to its structure. It aims to contribute to the advance in this field by providing solutions for some of the problems. IntellWheels' target user ranges from the wheelchair users with small locomotion disabilities to those with severe handicaps.

With the collaboration of different persons and integrating their work in the project, the architecture evolved along the time, from a static and rigid control layer architecture. The current architecture is composed of several logical and coherent modules organized as different micro-agents within a macro agent, the Intelligent Wheelchair, and can be seen in Fig. 1.

While these software modules are parts of the wheelchair agent, they are themselves, autonomous agents demanding data from and serving data to other software agents. This modular organization, enables a higher degree of upgradability for the architecture, and opens new paths for collaboration between IW. An example is the possibility for an IW to share or run a planning service for another IW, or for an IW agent to share its control interface with another IW.



Agent Functions:

Hardware – composed of motors, batteries, sensors (encoders, sonar and infra-red), electronic microprocessor plate with RS232/USB connections and a notebook;

Perception Agent – responsible for gathering information from the wheelchair's hardware;

Control Agent – acts as a gateway between all the hardware's actuators and the rest of the programs;

Intelligence or Cognitive Agent – responsible for high level decisions, planning, agent cooperation;

Interface Agent – responsible for translating the user's orders, gathered from various input modules, into high level orders;

Simulator – responsible for creating a virtual world to test the control algorithms. It enables the use of mixed realities (real environment, augmented reality, augmented virtuality, virtual environment).

Fig. 1. Current Intellwheels Architecture diagram and description of the different agents

The new architecture, its organization and the environment constraints, served as motivation for a new communication system, described in this article. This new system was designed to enable the easy development and integration of new functional modules as also to facilitate the use of new cooperative and communication methodologies while maintaining a high degree of reliability. Thus enabling the transmission of messages, the dynamic configuration of the different agents and their rapid adaptation to a new communication environment.

3 System Description

Normally, a multi-agent platform, such as Jade, would be used to enable communications as well as organize and manage the different agents. However with common multi-agent platforms it is not possible to customize or enhance system functionality in order to adapt the systen to a specific reality.

In Intellwhells case, the solution was to develop a new communication system, based on MAS methodologies, that could address the following constraints: multiple agent support, compatibility with other communication languages and communication systems, an easy to use and to configure systems interface, system's reconfiguration according to changes in the physical and networked environment, aplicable to open transmission systems and an easy to use interface to facilitate knowledge sharing amongts agents.

Taking these constraints into account, and the unique characteristic that all software for Intellwheels behaves as individual agents within the wheelchair, the new system was developed as two separate platforms. The first, a global platform, applied to all software within the same network environment and based on the FIPA standard, [4]. The second, a local platform, responsible for the management of all local services needed to run an IW.

Central in this architecture is the election of a Container entity and the distribution of a Local Agents List (LAL), as well as a Global Agent List (GAL), while using message-oriented paradigm as opposed to object-oriented. The use of object sharing platforms was discarded due to their higher architecture complexity, firewall traversal and dependibility on proprietary extensions. In these lists are the applications' configurations that enable communications (distribution of the public encryption key) between agents.

The Container, named after the Jade Container entity, [7], was designed to be responsible for the LAL's maintenance. These maintenance operations include the following: creation, update, deletion. However and contrary to the Jade system, 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 probabal for a whellchair to lose network connectivity or to change its network configuration but it is no acceptable for these changes to cause a system's malfunction.

Common to both architectures is the use of the FIPA-ACL communicative acts for the messages' structure, as well as FIPA-SL for language representation, thus providing a strong base for compatibility with other objects or systems.

3.1 Micro Architecture, the Local Platform

As above mentioned, the local platform is organized and maintained by a Container entity. It is not however, a separate software. The system was designed for the container algorithms to remain inside the communication's structure, thus making it a part of all applications. This way it is also possible to start an agent and for it not to depend directly on the communication system's configuration, state machine and resources to performe its function.

The system's architecture was designed as five separate layers with their respective receiving and sending handling methods and interfaces, as seen in Fig. 2 a), running in parallel. This way, it becomes possible for the user to choose which layers should be applied to the application, without compromising the agent's functionality while following the OSI Reference Model and implementing fault tolerant methods described in [9] and [2].

Another of the features of this structure is the full customization of the specific messages' handling functions to be applied within each layer, giving the user total control over the resulting message envelope, seen in Fig. 2 b).

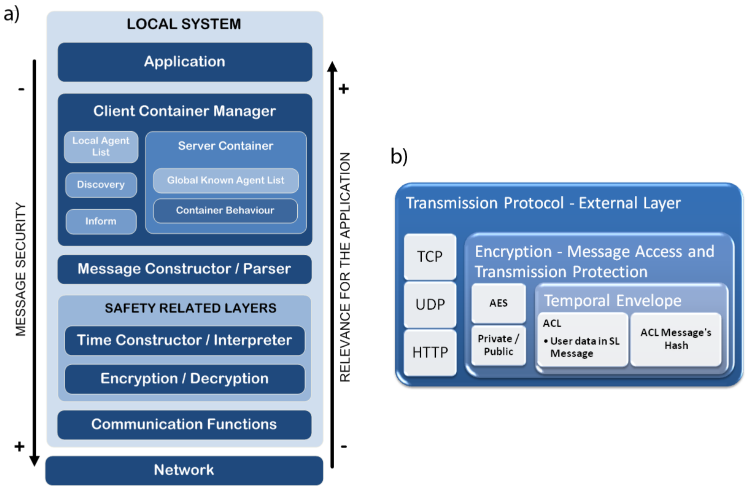


Fig. 2. Local platform structure, a). The software is organized into 5 independent layers that can be used to construct the application's communication system. The resulting default message envelope can be seen in b).

The layer's functions are the following:

- **Communications layer** it is responsible for receiving and sending messages from and to the message transport layer. Enables 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 in the media layer. It also prevents the application from receiving messages with a size larger than the one specified by the user. When not specified the default messages' sizes are 8kBytes for UDP and 16kBytes for TCP;
- **Decryption and Encryption layer** responsible for the message's security, preventing the interception and modification of messages. The Encryption method is chosen accordingly to the message's destination and the knowledge requested, in the case of platform messages exchange. The user's message encryption can be chosen by the user, and treated accordingly. The possible encryption methods involve the use of a private and public key pair or an AES pre-shared key. It also performs

message integrity checking by cross-referencing the message with the transmitted message's hash;
- **Time layer** responsible for adding a time stamp to the message to send, for organizing the received messages according to the time stamp and for the detection and elimination of injected packets. Another function that it performs is the configuration and synchronization of the local system's clock with a networked NTP clock, if available. This configuration is done automatically and only if the application is the local Container;
- **Message Constructor / Parser layer** responsible for the construction of the message according to the FIPA-ACL standard and represented using FIPA-SL. It is also responsible for selecting the messages that should be 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 outside application;
- **Client Container Manager** responsible for managing the application's organization and integration in the local and networked platforms, implementing methods like replication, fault detection, recovery and discovery. It also implements the user interface with the communication system by enabling direct access to the Local Agent List and Global Agent List.

An additional platform component was designed, the Log. Linked to all of the previously presented layers by a buffer, it is responsible for logging the activity that occurs within the application, either messages sent by the user or by the platform's maintenance functions. It's interface enables the user to choose the layers to be logged as well as the functions that apply to the log. By default every line presented in the log file is identifed by a timestamp and by a custom message to identify the function that generated it. However when encryption is enabled, no log is performed to the encrypted messages as a security measure, to prevent encription key discovery. It's also possible for the user to use the Log component in the agent functions to record their state.

4 Tests: Scenario, Protocol and Results

This section presents the testbed, the test scenarios and the achieved results.

In order to validate the proposed solutions, the above described architecture, was implemented in Pascal, using Borland Delphi Professional v7.2 IDE. Jade v3.7 and the Eclipse Platform v3.4.2 were used to develop and test the Jade agents described. Also, as an support for the testing, an automated application laucher and data logger, was developed. For all tests the application Process Explorer from Sysinternals, was used as a resource monitor.

All tests were repeated 20 times with the same conditions and all data was analysed and represented with a confidence interval of 95%.

To validate the local platform two test scenarios were established, each with different objectives.

The first test's objective was to measure the performance, effectiveness and scalibity of the communication platform. It consisted on a agent sending a message, with a fixed size of 500 Bytes, that would be redistributed to all agents in a serialized manner. The time that the message took to be passed between all agents and return to the initiator, was measured and analysed. To gather this test's data, one agent was implemented to follow a protocol, that can be seen in Fig. 3 a), when instantiated more than once. In this picture, x represents a integer number used as a control mechanism for message differentiation, and the numbers in between round brackets the interaction sequence. This test was repeated using the maximum number of 19 agent instantiations. The gathered results not only demonstrate that the platform was able to deliver all messages in the correct order without errors, but also that the platform scales linearly with the amount of agents present.

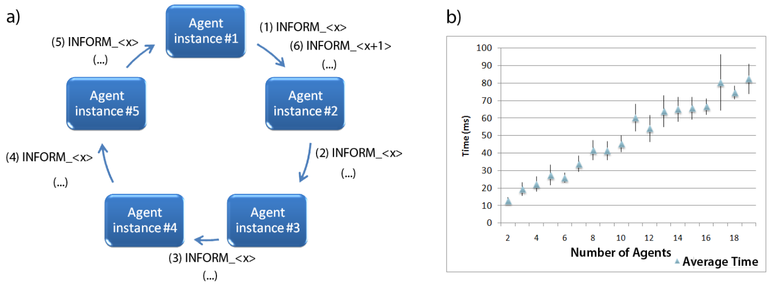


Fig. 3. Implemented protocol in the first test's scenario, a). Test's results, b).

The second test's scenario objective was to evaluate the performance and scalability of the platform in a well known agent simulation case, and to compare the results with the ones obtained, in the same scenario, using the Jade platform. The implemented scenario followed the Book Seller Book Buyer protocol, seen in Fig. 4 a) and b), for the example of one buyer and two sellers. For the test, only one buyer agent was instantiated, while the number of seller agents increased in each test round to the maximum of 15 agents. The time needed by the buyer agent to successfully accomplish the book's request and acquisition, was logged for both the Jade platform and the proposed architecture. The results in Fig. 4 c), allow a comparison to be established between the two platforms. While Jade is slower with a smaller number of agents, it's test's results are closer together, with small variations between them. On the other side, the proposed platform is able to communicate very fast with a small number of agents, but when this number increases, the result's interval increases also becoming wider.

5 Conclusions

With the rapid and growing number of Intelligent Wheelchairs under development, more attention must be paid to the development of communication systems. In this field, object interaction is absolutely needed, in order to enable cooperation between objects. This paper's main objective was to propose and describe new solutions for agents' communication, applicable to the field

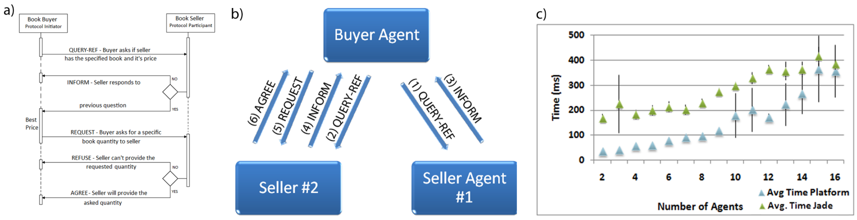


Fig. 4. Book buyer and book seller agents' protocol, a). An example of message exchange in this test scenario with seller #2 providing the book's best price, b). Test's results for Jade and the described platorm, c).

of mobile robots in dynamic environments. The tests results achieved using the proposed architecture, show that the platform can in fact provide cooperative communications between agents, and provide a new solution for agent-based systems. Moreover, the defensive methods implemented, proved to be able to protect data transmission and system functionality. However the collected data also shows that there is still work to be done on the platform's scalability.

References

1. Braga, R. A. M., Petry, M, Moreira, A. P., Reis, L. P., 2008. Intellwheels: A Development PlatForm for Intelligent Wheelchairs for Disabled People. In Proceeding of the 5th Int. Conf. on Informatics in Control, Automation and Robotics (Funchal, Madeira, Portugal. May 1115). Vol I. pp.115-121.
2. EN 50159-2, Mar. 2001. Railway applications Communication, signaling and processing systems, Part 2: Safety related communication in open transmission systems. In Vol. European Commitee for Electr. Standardization.
3. Faria, P. M., Braga, R A. M., Valgde, E., Reis L. P., 2007. Platform to Drive an IW using Facial Expressions. In Proc. 9th Int. Conf. on Enterprise Information Systems, HCI (Funchal, Madeira, Portugal.), ICEIS2007. pp. 164-169.
4. FIPA, Foundation for Intelligent Physical Agents, `http://www.fipa.org`.
5. Galindo C., Cruz-Martin A., Blanco J.L., Fernndez-Madrigal J.A., and Gonzalez J, 2006b. A Multi-Agent Control Architecture for a Robotic Wheelchair. In Applied Bionics and Biomechanics, vol III, pp. 179-189.
6. Galindo C., Gonzalez J., and Fernandez-Madrigal J.A., 2006a. A Control Architecture for Human-Robot Integration. Application to a Robotic Wheelchair. In IEEE Trans Systems, Man Cybernet-Part B 2006, vol. 36.
7. JADE, Java Agent Development Framework, `http://jade.tilab.com`.
8. Jia, P. and Hu, H., 2005. Head Gesture based Control of an Intelligent Wheelchair. In CACSUK-11th Ann. Conf. Chinese Aut. Comp. Society in the UK (Sheffield, UK).
9. Malm, T., Hrard, J., Boegh, J. and Kivipuro, M. , 2007. Validation of Safety-Related Wireless Machine Control Systems, NT TechReport, Nordic Innov. Centre, Norway.
10. Prenzel, O.; Feuser, J.; Graser, A., 2005. Rehabilitation robot in intelligent home environment - software architecture and implementation of a distributed system. In Rehabilitation Robotics, 2005. ICORR 2005. 9th Int. Conf.(28 Jun-1 Juy 2005), pp. 530-535.