A Cooperative Communications Platform for Safety Critical Robotics: An Experimental Evaluation

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

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) and motivated by Intelligent Wheelchair systems. It uses a message-oriented paradigm as a mean for fault-tolerant communications in open transmission systems as well as a facilitator for entity collaboration. It provides an overview of the related work in the area, the background and the main constraints to system development, contextualized an IW development project. The results achieved enable us to conclude on the effectiveness and adequacy of the proposed communication model to the field of mobile robots, as well as to conclude that it may outperform JADE in several test scenarios.

1 Introduction

The World Health Organization (WHO), estimates that around 2% of world population (130 million people) live with physical handicaps. The most common aid for

---

Frederico M. Cunha
Artificial Intelligence and Computer Science Laboratory - LIACC, Faculty of Eng. of University of Porto - FEUP, Rua Dr. Roberto Frias s/n, Porto, Portugal, e-mail: fredericom.cunha@gmail.com

Rodrigo A. M. Braga
Artificial Intelligence and Computer Science Laboratory - LIACC, Faculty of Eng. of University of Porto - FEUP, Rua Dr. Roberto Frias s/n, Porto, Portugal, e-mail: rodrigo.braga@fe.up.pt

Luis P. Reis
Artificial Intelligence and Computer Science Laboratory - LIACC, Faculty of Eng. of University of Porto - FEUP, Rua Dr. Roberto Frias s/n, Porto, Portugal, e-mail: lpreis@fe.up.pt

this kind of mobility problem is the wheelchair, specifically the electric wheelchair. Numerous Intelligent Wheelchair (IW) related projects have been announced, and are under development in the last years. The increase study of this field, led to a globally accepted view of the main functional requirements for such systems. According to [10] and further developed in [2], the Communication systems is one of the main functions of an IW, as it enables interaction with other devices like other IWs, intelligent robots and remotely operated control software for medical staff. Although many IW projects exist, the communication system used is rarely described 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 [11]. The SENA robot, presented in [7], is one of the few IW projects that approaches the communication system. Currently, it uses a custom Multi-agent System (MAS) to implement communications [6], thus taking advantage of MAS maturity, robustness, scalability, easy management of information and agents.

This article will focus in the field of mobile robotics, specifically IW communications and will: identify the main requirements to a safe and secure communication system, propose solutions that can address the found constraints, as well as, test the proposed solutions and compare the results with commonly used platforms. The rest of this article is divided into 3 additional sections. Section 2 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 3 describes the test environment, the applied test methodology and presents the test results. Section 4 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 System Description

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. Currently it implements a multi-layer control architecture, described in [1] and [2], distributed through 4 agents. The user interacts with the system through a multi-modal interface, that allows different input sources to be used [2]. Also a simulator is available, enabling a mix of different realities to be used when testing the platform. The following communication system's requirements were identified: multiple agent support, compatibility with other communication languages and communication systems, system's reconfiguration according to changes in the physical and networked environment, applicable to open transmission systems, use of defensive methodologies to protect system communication.

Normally, a multi-agent platform, such as Jade, would be used to enable communications, organize and manage the different agents. However with common multi-agent frameworks it is not possible to easily customize or enhance system functionality in order to adapt the system to a specific reality. Moreover, to implement fault-tolerant methods, it is required to use federated communication models with support on centralized databases, as resolve the issues related with firewall traversal and dependability on proprietary extensions. Thus, in Intellwhells case, the solution was to develop a new communication system, based on MAS methodologies that could address the system's requirements.

Taking the requirements into account, the new system was based on the FIPA standard [5], and uses the FIPA-ACL communicative acts as well as FIPA-SL. It was developed to work in two separate environments. The first, a global network applied to all agents connected to a network, and the second applied to the agents present in the same pc. Central in this architecture is the election of a Container entity, similar to Jade [9], and the distribution of a Local Agents List (LAL), as well as a Global Agent List (GAL), 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 no acceptable for these changes to cause a system's malfunction.

2.1 Architecture

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 agents. 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 perform 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. 1 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 [3] and recommended by [8].

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 network level. It also prevents the appli-

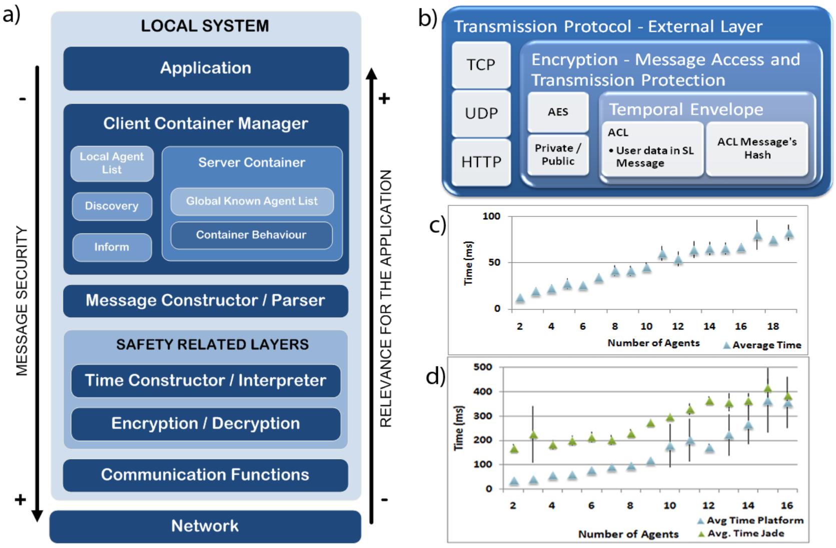

Fig. 1 a) Platform's layered structure; b) Resulting message envelope; c) Results in the second test; d) Results in the third test.

cation from receiving messages with a size larger than the one specified by the user during agent implementation. When not specified the default messages' maximum size is 16kBytes for TCP and 8kBytes for UDP.

The **Decryption and Encryption layer** is 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 at the moment. 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.

The **Time layer** is responsible for adding a time stamp to the message to be sent, and for organizing the received messages according to the time stamp. It also performs 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.

The **Message Constructor and Parser layer** is responsible for the construction of the message according to the FIPA-ACL standard and represented using 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.

The **Client Container Manager** manages the application's organization and integration in the local and networked environments, 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 LAL and GAL.

3 Tests: Scenario, Protocol and Results

For validation purposes, 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. For all tests the application Process Explorer from Sysinternals, was used as a resource monitor. The tests were made in a PC with an Intel Core 2 Duo 2,53GHz cpu, 4GB of RAM, with Windows Vista and using a 802.11n wireless network card. 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 platform, three type of tests categories were established and implemented.

The first tests' objectives were to evaluate the effectiveness, performance and to determine the platform's limits, when submitted to stress conditions. It used messages sent in short intervals with a increasing size. A two agent interaction was implemented, one serving as a initiator and the other a message repeater. The initiator would compare the sent message with the one received and then increase the message's size with 100 new bytes. The test results, seen in Fig. 2, prove that the platform was able to transmit all messages until the maximum size limits, for 4h and with small time deviations between rounds.

The second test's objective was to evaluate 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. The test was repeated using the maximum number of 19 agent instantiations. The results, seen in Fig. 1c), show that the platform was able to deliver all messages in the correct order without errors, and that it scales linearly with the amount of agents present.

The third test's objective was to compare the platform's performance and scalability with Jade's, using a Book Seller Book Buyer agent protocol, described in [4]. In 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 for a successful interaction was measured and can be seen in Fig. 1d). These show that Jade is slower than the proposed platform with a small number of agents and both with small variations. However when the number of agents increased the proposed platform's results intervals became wider.

4 Conclusions

In the field of IW, object interaction is absolutely needed, to enable cooperation between distinct robots. The results achieved using the proposed architecture, allow us to conclude that the platform can in fact provide safe communications between a large number of agents. Additionally, the implemented defensive procedures, proved

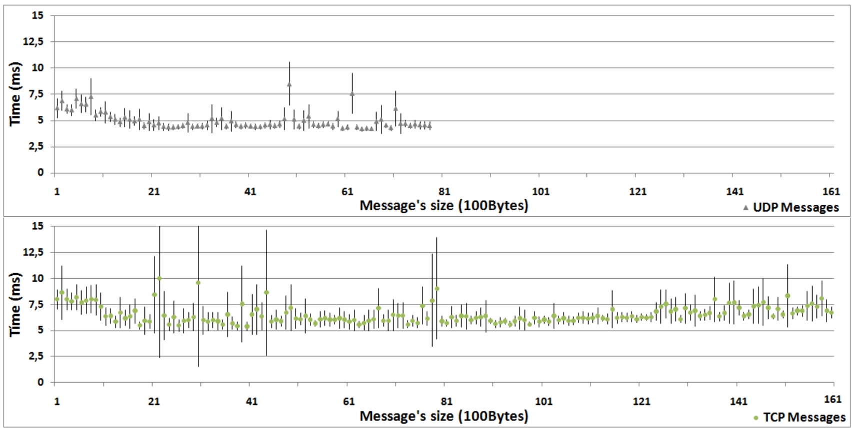

Fig. 2 Results achieved in the first test's scenario.

to be effective when applied to this field. Moreover, the collected data shows that the framework is able to outperform JADE in several test scenarios.

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 Proc. of the 5th Int. Conf. on Informatics in Control, Aut. and Robotics. Vol I. pp.115-121. May, 2008.
2. Braga R. A. M., Petry M., Moreira A. P., Reis L. P. (2009) Concept and Design of the Intellwheels Platform for Developing Intelligent Wheelchairs. Informatics in Control, Aut. and Robotics. Vol 37. pp. 191-203. April, 2009.
3. EN 50159-2, Mar. 2001. Railway applications Communication, signaling and processing systems, Part 2: Safety related communication in open transmission systems. In Vol. Europ. Commitee for Electr. Standardization.
4. Fabio L. Bellifemine, Giovanni Caire, and Dominic Greenwood. Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology). Wiley, April 2007.
5. FIPA, Found. for Intelligent Physical Agents, http://www.fipa.org. October, 2009.
6. Galindo C., Cruz-Martin A., Blanco J.L., Fernndez-Madrigal J.A., and Gonzalez J (2006) A Multi-Agent Control Architecture for a Robotic Wheelchair. In App. Bionics and Biomechanics, vol III, pp. 179-189.
7. Galindo C., Gonzalez J., Fernandez-Madrigal J.A. (2006) A Control Architecture for Human-Robot Integration. Application to a Robotic Wheelchair. In IEEE Trans Systems, Man Cybernet-Part B 2006, vol. 36.
8. IEC 61508. Functional safety of electrical/electronic/programmable electronic safety-related systems. In Vol. European Committee for Electrotechnical Standardization.
9. JADE, Java Agent Dev. Framework, http://jade.tilab.com. October, 2009.
10. 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).
11. Prenzel O., Feuser J., Graser A. (2005) Rehabilitation robot in intelligent home environment - software architecture and implementation of a distributed system. In Rehab. Robotics, 2005. ICORR 2005. 9th Int. Conf.(28Jun-1Juy 2005), pp. 530-535.