Teleoperation mechanisms in a Multi-Agent System

Vasco Santos
R&D Division - IntRoSys, S.A.
Campus da FCT-UNL,
2829-516-Portugal
vps@uninova.pt

Pedro Santana, Luís Correia
LabMAg - University of Lisbon
Campo Grande,
1749-016 - Portugal
{Pedro.Santana,
Luis.Correia} @di.fc.ul.pt

José Barata
UNINOVA
Campus da FCT-UNL,
2829-516-Portugal
jab@uninova.pt

Abstract

Human judgement is an integral part of the teleoperation process, heavily influenced by the rate in which commands are given to and the telemetry data is received from the robot. In this paper a multi-agent system for robot teleoperation is proposed. Several entities (e.g. robot or human) can easily be added and removed. In addition, this approach fosters a shared notion of reality to every entity present in the network, through a mechanism similar to the distributed blackboard architecture. This method provided an exchange of information regarding the teleoperation scenario, i.e. perceptual clues. This promoted a study on how perceptual clues influenced the operators' judgement and performance. The experimental results in a physical suggest that the system is able to guarantee a close interaction between users and robots.

1 Introduction

In teleoperation, as in many system involving humans and machines, there is a clear need to map concepts between different representations. This can be done by using a shared ontology to provide a common language among team members, thus allowing the ability for the entities to understand each other. For instance, robot's sensor data must be mapped to a human readable set of symbols. Conversely, the human desired robot heading must be mapped into robot understandable symbols.

In teleoperation, if the operator has not proper situation awareness, i.e. a real perception of the robots' internal state (e.g. posture or position), the operator can not decide where to lead the robot. From the humans point of view there are several ways of providing perceptual clues (e.g. maps, video, etc.) for enhanced situation awareness. In some work, the robots' posture has been included in the video feed [14], also the ability of providing a camera with autonomous control has proved to improve the operators' awareness [7]. The joint use of video and maps has been proven useful by the work of Nielsen and Goodrich [8], where several combinations have been tested.

The operator must also be able to perform manoeuvres in real time fashion, since any delay may render that manoeuvre useless, or even dangerous. This limits the type and amount of information to be mapped between human and robot. Some multi-agent teleoperation approaches [10] have considered the load of telemetry and command messages on the network excessive thus placing them on a dedicated channel.

Contrary, in this paper teleoperation commands and telemetry messages have no dedicated channel. Having all teleoperation related information in the main MAS network, an approach where all this information is shared on the network as taken. In this multi-agent system network each entity is considered a knowledge source, fostering a shared representation of each entities beliefs, desires and intentions (BDIs), resembling the distributed blackboard architecture [9]. This approach enables a shared representation of all teleoperation orders and telemetry data through the system.

Having all teleoperation information flowing through the MAS network causes a higher load on the network, forcing the information exchange to be optimised. A study on this subject is also shown, as to see if this shared reality approach promotes an improved operators situation awareness, as well as, of other entities added to the system.

Section 2 presents the teleoperation model, exposing the different components and entities present in the system, design approaches and assumptions. Then, the computational model, where the different interactions semantics used by the agents during their interaction are explained, is presented in Section 3. Then, in Section 4, it is presented the physical instantiation of the teleoperation model with all of its' entities, as well as, a series of tools and mechanisms designed to provide the operator with better awareness of the robot's status and location. Afterwards, the experimental results are presented in Section 5. Finally, conclusions and future work directions are presented in Section 6.

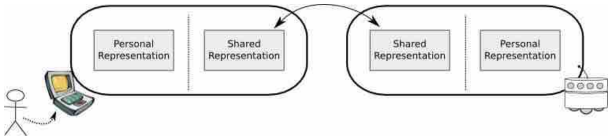


Figure 1. Generic System Overview

2 System Overview

Considering that each entity has its own perception on the reality, a shared representation must be imposed, in order to allow a common way of interaction, i.e. an ontological commitment [5], [3], [6] (see Figure 1).

This dual reality representation is implemented as part of an architecture described in previous work [12], [13], supported by the multi-agent systems' tool JADE [1] and C++. JADE is built under FIPA[1] rules, as to allow proper interaction mechanisms. These two reality representations are needed in a similar way as humans use it, when humans interact with each other they use language, while interacting with themselves hormones and other stimuli may be used. This allows the entity to have real-time responses concerning itself and lesser time dependent responses while interacting with others. The proposed system was designed based on a network inhabited by cooperative agents, i.e. all contribute for the global good. This assumption allows for a simplification on the negotiation mechanisms, since there is no need to have elaborated trust mechanisms or partner selection mechanisms. It is also assumed that when an entity subscribes a service it has all the required abilities to accomplish that service.

Considering the teleoperation scenario, there is a clear dependence of time. This means that every information to be shared must be time tagged. This works similarly to the concept of blackboard, meaning that each knowledge source, entity, updates its perception of reality within the community in a time dependent fashion. The shared representation update is conditional on the type of information. This idea will described later in this section and in detail on section 3.1.

Each robot, human, or any autonomous mechanism (e.g. camera) is represented by an agent. The entities' agent makes the bridge between the entities' self perception and the shared reality network. Figure 2 depicts an instantiated systems' overview, where several *entities* can be seen interacting amongst themselves through their associated agents. An Agent Communication Language (ACL) method is used to support agents interactions. Each *entity* supporting agent is composed by three main components: (1) the *Multi-Agent System Interaction Mechanism* (MAS-IM), (2) the *Knowledge Base* and (3) the *Physical Entity interface* (P.E. interface). The MAS-IM is the component that enables the agent to interact with other agents in the multi-agent community. In order to exploit its middleware services (e.g. yellow pages service). This module is built over the JADE platform. The Knowledge Base aggregates the knowledge of the entity, and partial shared knowledge about the other entities which the agent is interacting with. The physical entity module abstracts the physical entity, i.e. providing an access to its control system and telemetry data in the robots' case.

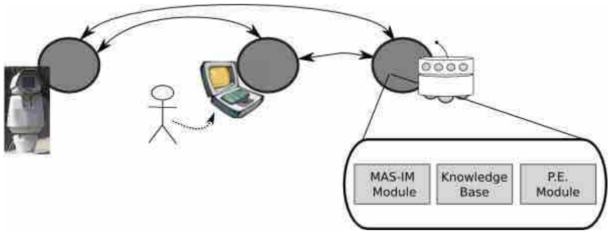


Figure 2. System Overview

3 Interaction Semantics

When different entities with different perceptions of the reality interact, it is necessary to provide not only a common language, but also, strict interaction rules and semantics. Like in any conversation, it is important to not only know the language, as well as, knowing the rules in which the actors are engaging. For instance, it is not considered polite to interrupt conversations or crosstalking. Surely it is important to know how to interact with others, however it is also important to know how to interact with ourselves (e.g. what stimuli to give our body as to accomplish a desired action).

Considering these concepts, interactions were tackled in two separate ways. One relates to the way the agent interacts with other agents (Inter-Agent Interactions); the other refers to the way the agent interacts with the physical entity's control structure (Intra-Agent Interactions). Inter-Agent interactions have a common structure to all who share the network, therefore are reusable and stereotyped. Intra-Agent interactions, on the other hand, are tightly coupled with the entity in question, e.g. the operator interacts with its' agent through a joystick or GUI. This implies that the interface responsible for the latter kind of interactions (Physical Entity Module) is not reusable, and it is mostly native to the entity.

3.1 Inter-Agent Interactions

As mentioned before, this type of interactions are common to all entities. In order to create a common language to every entity in the system it was necessary to create a supporting ontology. The next logical step was to define the conditions and rules for these interactions, where two approaches were taken. First, a *publish/subscribe* method for the registration and subscription of the services and information each entity provides was used. This means that, if an operator wishes to teleoperate a robot, it has first to search for a robot that has published a teleoperation service and subscribe it. This method is done taking advantage of JADEs' yellow page service. Secondly, similar in spirit to the distributed blackboard architecture, the oper-

---

[1] http://www.fipa.org

ator has to do is to update its desires of movement in the remote robot's knowledge base and the latter will comply with that desire.

These interactions are done through the MAS-IM module using the Knowledge Base (KB). The Knowledge Base was designed as to create a enable the build up of a shared mental state. This means that when two entities interact, each creates a knowledge base that refers to the other entity's beliefs, desires and intentions (BDI). It is important to note that this remote knowledge base is not necessarily a full representation of the entity's BDIs, just the BDIs necessary to the interactions. This information is defined in the ontology.

For instance a robot is available to be teleoperated, it publishes the teleoperation service. When the operator finds that robot, the operator subscribes that service. Then each of the physical entities create an empty knowledge base of the remote entity. From this time on each entity may choose whether to update its' BDI or simply to reply them when a query is made, as it is shown in Figure 3.

From an engineering point of view, Agent Communication Language (ACL) messages may induce a considerable network load when transferring large pieces of data. To this effect two design paths were followed. On one hand there is sparse information (e.g. sensor information), although high in quantity it has very little detail associated. Sensors are subject to false positives and negatives, thus meaning that sometimes the right events are not being communicated to the rest of the network. On the other hand, there is detailed information. This information is not generated as fast as sparse one, however it has more detail. For instance a robot may be constructing a map where it checks the events (e.g. obstacles) from different angles. This provides a way of verifying if the obstacle is still there or not, or if even existed. Other entities have no other way of knowing of the false positives, since the robot only updates new obstacles, it does not update removed ones. Other entities can only synchronise by accessing detailed information and discarding previous sparse information. Meaning that detailed information must be requested with a certain regularity, in order to, provide an improved view of the remote environment. However it can not be done in short time intervals, as it would create great latency in the network. A test on this trade off can be seen in Section 5.

When considering the information transfer method, two paths were followed. On the sparse information case, queries, replies or updates are done through ACL messages. Messages are filled with knowledge based information. This occurs in the content feeder module, part of the MAS-IM module, as shown in Figure 3. On the other end the MAS-IM module extracts the knowledge base information through the content extractor module updating the information in the knowledge base, also seen in Figure 3. When considering detailed information, protocols designed for larger data transfer are used, such as the FTP protocol. This information exchange is also allowed by the publish and subscribe wrapper is implemented, where the requesting entity subscribes to the information by reading the protocol definitions and target information.

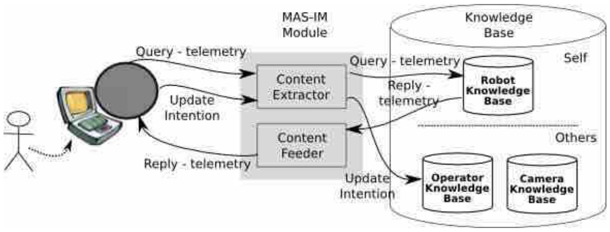


Figure 3. Inter-Agent

3.2 Intra-Agent Interactions

Intra-Agent interactions refer to the interactions between the entity's agent and the entities internal mechanisms, e.g. the robots' agent consulting the log files containing the telemetry data. These interactions are restricted to the type of entity they were designed for, meaning that they are native to a specific entity. Considering the significant differences between a robots' control system and an operator control system, these kind of interactions usually have to be native to the type of entity considered. Not only because the interfaces are different, in the operators' case a Human Interface Device (HID) (e.g. gamepad) is the obvious interface, while in the robots' case a socket or file interface is more suited.

Intra-Agent interactions are performed by the Knowledge Base and the Physical Entity (P.E.) module, in a similar way that the human brain assimilates the external information and then sends the information to the several muscles, or how the brain receives information from the muscles of fatigue or fitness. In order to simulate the human body, there is a daemon that regularly updates the Knowledge Base self beliefs. This daemon activates the Information Encoder on the P.E. module and generates Knowledge Base suitable information. The P.E. module checks the log file system in order to retrieve the most recent information (e.g. telemetry data) and transforming it into KB information, as to update the its' beliefs.

Considering that this work does not refer to an autonomous robot, the robot has not the desire of moving or accomplish anything. However with this representation, by having the other Knowledge Base (e.g. operator), the operator is able to provide the robots' agent with a 'will' of movement. This is done through a daemon that works as an hormone that navigates through the nerves in order to send the desired information, when the operators remote Knowledge base has its BDIs updated the daemon transforms the KB information into low-level information, using the Information Decoder on the P.E. module, and sends it to the robots' control system. All of this is depicted in Figure 4.

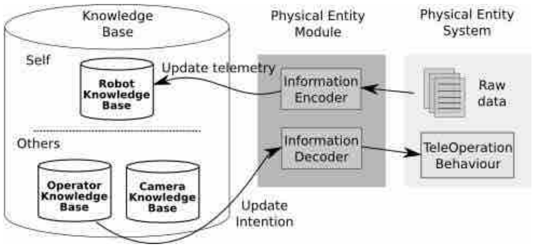


Figure 4. Intra-Agent

Figure 5. Ares Robot

4 Model Instantiation

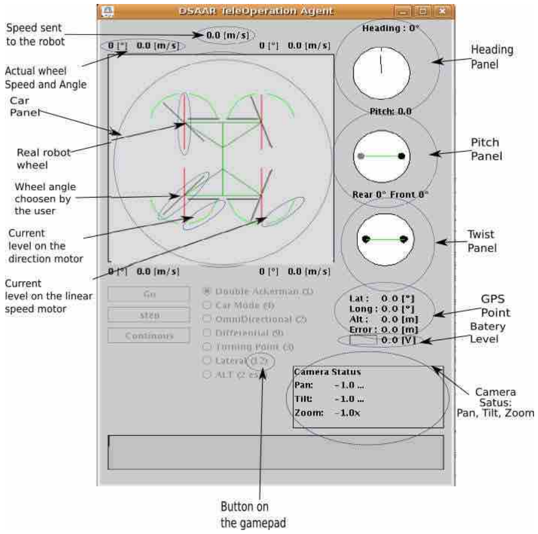
In order to validate the proposed approach, a series of experiments were carried out. Initial tests were made using the player/stage/gazebo tool [4] and with the RWI B12 indoor robot. However there was a need to bring it outdoors, in order to test not only communications as well as the navigation difficulties in a more demanding environment. The experimental setup is composed by a teleoperation camera, a robot and an operator. The robot and the camera may be seen in Figure 5. Although placed together, their characteristics were self sufficient to be considered different entities. When searching for specific targets or considering the scenario of autonomous navigation, the teleoperation camera may be used without interfering with the robots' reality.

The operator interacts with the system through a control centre depicted in Figure 6. This system is composed of a video screen and a laptop with a gamepad. The video screen displays a direct video feed from the teleoperation camera, this feed is considered detailed information so the service is subscribed and then with the necessary description the video server is contacted and the link is formed. The laptop provides two tools from which the operator is able to get a perception of what is happening with the robot and its surroundings. These tools are described in the following section.

4.1 Tools

In this section two tools designed to improve the operators' situation awareness, are presented. One which shows the operator the robots' and cameras' posture, as well as, the desired commands (Teleoperation tool) and the other which is a 2D representation of the world with several additional functionalities (2D Map tool).

Figure 6. Control Centre



Lat : 0.0 [°]
Long : 0.0 [°]
Alt : 0.0 [m]
Error : 0.0 [m]
0.0 [V]

Camera Status
Pan: -1.0 ...
Tilt: -1.0 ...
Zoom: -1.0x

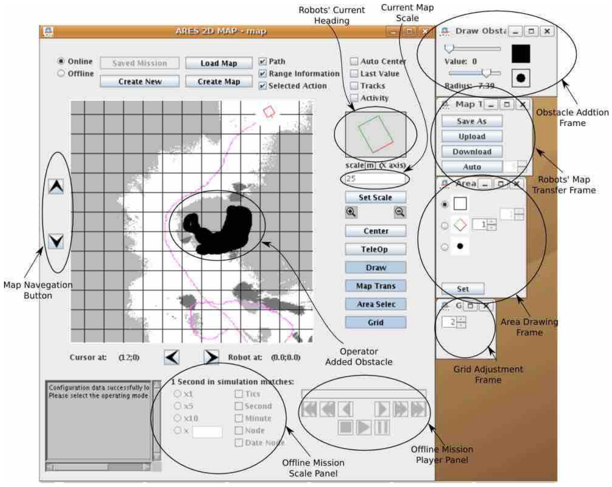
Figure 7. TeleOperation Tool

The teleoperation tool, allows the operator to control the robot and teleoperate it through a logitech gamepad[2]. As depicted in Figure 7, the operator may see the robots' posture, current wheel speed and internal status (e.g. battery level). This tool also allows the user to check the desired speed sent to the robot. Regarding the camera control and telemetry, the operator receives the pan, tilt and zoom information, as well as, the operator may choose between several control modes such as: follow heading; auto-pilot, where the camera follows the robots' movements while driving; speed mode, where the operator sends a rotation speed of pan and/or tilt; and position mode, which allows the operator to point click in any place shown in the robots' representation in the teleoperation tool. For further description on these modes please refer to [2]. The operator may also control the robot in a variety of modes such as double Ackerman, omnidirectional, turning point, and lateral. All of these modes are described in [13].

The 2D map tool, concedes a global perspective of the environment where the robot is conducting its operations

---

[2] www.logitech.com



Configuration data successfully lo
Please select the operating mode

Figure 8. 2D Map Tool

of the operator. This tool may also be used by a supervisor in order to check the operation progress, since it does not need to have a direct influence in the mission procedure. This tool provides all the common features of a regular map, such as: path tracking; obstacles identification; map navigation; scale adjusting; and an adjustable grid, which allows the operator to have a better notion of the travelled spaces. In addition to having these features, this tool also concedes the operator to add obstacles to the viewing map either to complete his view during teleoperation or to send to the robot in the case of autonomous navigation. This topic is out of the scope of this paper, for further refer to [11]. Finally this tool allows for users to watch a previous mission and study the operators' behaviour during the course of that mission. This is done by downloading the mission logs from the robot using the detailed information interaction exchange. Then the map launches a process that emulates the robots' agent behaviour and so providing the user to toggle back and forth through that mission in a variety of time scales. The 2D Map tool front end is depicted in Figure 8.

5 Field Trials

In order to test the ability of the proposed approach to handle scenarios with demanding network load, an experiment was designed that needed both sparse and detailed information. A supervisor was also added in order to take into account the notion of shared reality. The supervisor was added to the network in different time intervals to verify the perception of the reality. Taking into account the setup presented in the previous section, an outdoor environment was considered.

In this trial the operator updates ones desires and intentions through the teleoperation tool, and requests the robots' beliefs through the teleoperation and map tool. The operator request both sparse and detailed information.

5.1 Experimental Setup

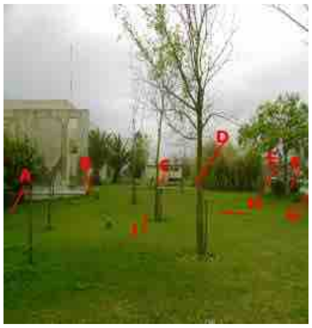
In this experiment the operator has to move the robot through a circuit formed by 9 points, from point A to point I. In this course the operator faces several obstacles which have to be avoided, as well as, navigation through narrow areas. The goal of this test is to find the best relation between the amount of detailed information that may be asked from the robot, without losing real time control, as well as, the minimum detailed information necessary for competent teleoperation. The test course can be seen in Figure 9, where the real test site and a robots' representation made previously to the test of the site are shown. In this figure it can be seen which points the user has to pass during the test. In Figure 10 it can be seen some of the most difficult challenges of this course, for instance in Figure 10(a) the operator had to take the robot through an area whose width was few centimetres wider than the robots' width. In point C where the operator is faced with an S turn within an area with great movement limitation, (see Figure 10(b)). In point D the operators' greatest challenge was to avoid hitting any of the concrete pins depicted in Figure 10(c). The last of the operators' challenges was in the section composed of the points E, F and G, where cars were parked and the operator could not hit them. This course section ended with a right turn which could easily be overlooked, (see figure 10(d)). The operator was placed near point H turned with his back to the testing course. During the several runs a supervisor was seldomly added to the network in order to monitor the status of the run.

In this test the operator performed several runs where the detailed information, e.g. the map, was requested from the robot in several time intervals which varied form 1 to 10 seconds, while still receiving sparse information, e.g. telemetry data, and sending orders. Also important to refer is that sparse information had a 300 ms transfer rate and teleoperation orders had a 100 ms transfer rate. In the end the more relevant runs were repeated in order for a better understanding of the results.

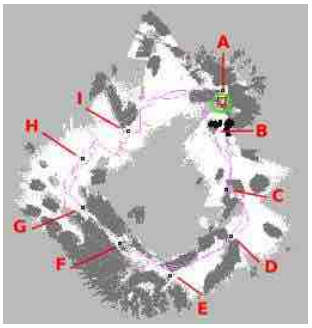
5.2 Results and Analysis

During the course of this test it was found that detailed information, i.e. robots' map, made control unreliable under the 5 seconds transfer rate. This fact got worse with distance of the robot and with the size of mapped area. At the start of each run the operator had access to the robots' map at that time, which is depicted in Figure 11, after that, the rate at which the robots' map is refreshed is according to the run. Here it will be analysed the 5 seconds and 10 seconds run. All runs with a 5 seconds or larger refresh rates were successfully completed. The ones under 4 seconds were interrupted, as a result of the operator not being able to have full robot control and there was danger when approaching the cars near the points E, F and G.

An example of a complete run may be seen in Figure 12. This run was made with a 5 seconds refresh rate. During this run the operator was able to keep control dur-



(a) Test field picture



(b) Robots' map of the test field overview. (37x39m)

Figure 9. Test field overview

Figure 10. Test course main challenges'

Figure 11. Initial run (15x13m)

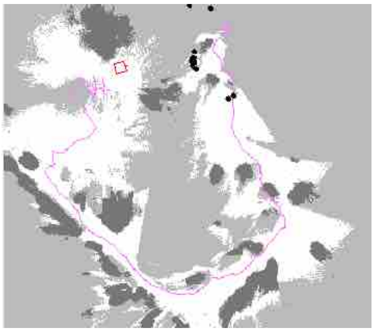


Figure 12. Result of complete 5 seconds run (32x28m)

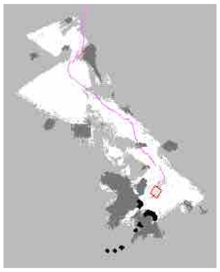


Figure 13. Situation awareness in a 5 seconds run near point D (23x28m)

ing the whole trial, with few misleadings from false positives from telemetry data. One example of misleading telemetry information may be seen in Figure 13, in which telemetry data gave the operator the notion that the robot would not be able to pass through point D, and forced the operator to wait for the map to be refreshed in order to proceed. The problem of misleading telemetry information increased as the refresh rate was smaller, which caused a performance drop. In several situations the operator was forced to wait several times, in order to proceed, this may be seen in Figure 14. In this situation the wait for detailed information, and sensor false positives gave the operator the impression of a dead end.

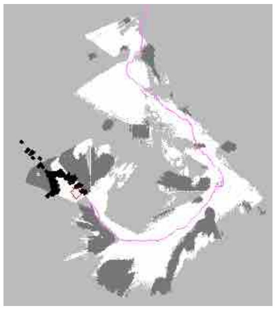


Figure 14. Situation awareness in a 10 seconds run near point G (29x32m)

6 Conclusions

A multi-agent based teleoperation system is proposed. The approach was validated both in simulation and the real world, under a teleoperation scenario. The Multi-Agent System, allows the addition and removal of several entities in a seamless way. When performing the experiment, this situation was tested with the addition and removal of the supervisor. This approach fosters the concept of shared reality, by having recently added entities with need of being updated with other's state information.

The field trials shown a clear trade off between sparse information and detailed information. When the operator, asked for too much detailed information, the ability for controlling the robot was clearly decreased. However results have shown that there is no need in having detailed information in short time frames, as sparse information can easily complement that gap. During the field trials, a trade off was found at a 300 ms cycle for sparse information and with a 5 seconds cycle for detailed information. This combination provided the operator with a comfortable 100 ms reaction control. Using this transfer rate the operator was able to have real time control, along with a proper situation awareness.

As future work, attention will be put into the some ways of improving the shared mental state of each entity in the network. Further studies will be done in the short term regarding the detailed information exchange, in order to, provide possible improvement to the transfer rate while to decreasing the network load.

Acknowledgements

We would like to thank Carlos Cândido and Paulo Santos for their support on preparing the experimental setup. This work was partially supported by FCT/MCTES grant No. SFRH/BD/27305/2006.

References

[1] F. Bellifemine, A. Poggi, and G. Rimassa. JADE – a FIPA-compliant agent framework. In *Proc. of PAAM'99*, pages 97–108, London, April 1999.

[2] C. Cândido, P. Santana, L. Correia, and J. Barata. Shared control of a pan tilt camera on an all terrain mobile robot. In *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (to appear)*, Hamburg, Germany, 2008.

[3] P. R. Cohen and H. J. Levesque. Teamwork. 25:487–512, 1991.

[4] B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, 2003.

[5] B. Grosz and S. Kraus. The evolution of sharedplans. In *Foundations and Theories of Rational Agency*, pages 227–262.

[6] B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[7] M. L. Joeseph Manojlovich and J. Gennari. Camera control and decoupled motion for teleoperation. In *Systems, Man and Cybernetics, 2003. IEEE International*, pages 1339–1344. IEEE Press, 2003.

[8] C. W. Nielsen and M. A. Goodrich. Comparing the usefulness of video and map information in navigation tasks. In *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 95–101, 2006.

[9] L. Nolle, K. Wong, and A. A. Hopgood. *Research and Development in Intelligent Systems XVII*, chapter DARBS: A Distributed Blackboard System. Springer, 2001.

[10] I. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion. Human-robot teaming for search and rescue. In *IEEE Pervasive Computing: Mobile and Ubiquitous Systems*, pages 72–78, January 2005.

[11] P. Santana, M. Salgueiro, V. Santos, L. Correia, and J. Barata. A knowledge-based component for human-robot teamwork. In *Proceedings of the 5th IEEE International Conference on Information in Control, Automation and Robotics (to appear)*, Funchal, Portugal, May 2008.

[12] P. Santana, V. Santos, and J. Barata. Dsaar: A distributed software architecture for autonomous robots. In *Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 20–22, Prague, Czech Republic, September 2006.

[13] V. Santos, C. Cândido, P. Santana, L. Correia, and J. Barata. Developments on a system for human-robot teams. In *Proc. of the Scientific Conf, of the 7th Edition of the National Robotics Festival*, Paderne, Portugal, April 2007.

[14] J. Wang, M. Lewis, and S. Hughes. Gravity-referenced attitude display for teleoperation of mobile robots. In *Proceedings of the 48th Annual Meeting of the Human Factors and Ergonomics Society*, pages 2662–2666, 2004.