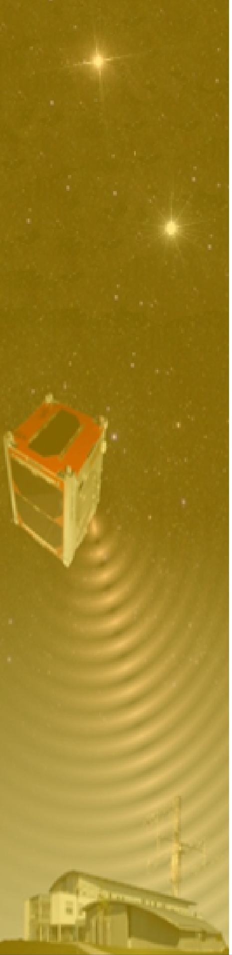


Institut für Informatik
Lehrstuhl für Robotik und Telematik
Prof. Dr. K. Schilling

Würzburger Forschungsberichte
in Robotik und Telematik

Uni Wuerzburg Research Notes
in Robotics and Telematics

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Band 1

Frauke Driewer

Teleoperation Interfaces in Human-Robot Teams

Die Schriftenreihe

wird vom Lehrstuhl für Informatik VII: Robotik und Telematik der Universität Würzburg herausgegeben und präsentiert innovative Forschung aus den Bereichen der Robotik und der Telematik.

Die Kombination fortgeschrittener Informationsverarbeitungsmethoden mit Verfahren der Regelungstechnik eröffnet hier interessante Forschungs- und Anwendungsperspektiven. Es werden dabei folgende interdisziplinäre Aufgabenschwerpunkte bearbeitet:

- Robotik und Mechatronik: Kombination von Informatik, Elektronik, Mechanik, Sensorik, Regelungs- und Steuerungstechnik, um Roboter adaptiv und flexibel ihrer Arbeitsumgebung anzupassen.
- Telematik: Integration von Telekommunikation, Informatik und Steuerungstechnik, um Dienstleistungen an entfernten Standorten zu erbringen.

Anwendungsschwerpunkte sind u.a. mobile Roboter, Tele-Robotik, Raumfahrtsysteme und Medizin-Robotik.

Lehrstuhl Informatik VII
Robotik und Telematik
Am Hubland
D-97074 Wuerzburg

Tel.: +49 (0) 931 - 31 - 86678
Fax: +49 (0) 931 - 888 - 6679

schi@informatik.uni-wuerzburg.de
http://www7.informatik.uni-wuerzburg.de

Dieses Dokument wird bereitgestellt durch den Online-Publikationsserver der Universität Würzburg.

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg

Tel.: +49 (0) 931 - 31 - 85917
Fax: +49 (0) 931 - 888 - 59 70

opus@bibliothek.uni-wuerzburg.de
http://www.opus-bayern.de/uni-wuerzburg/

ISSN (Internet): 1868-7474
ISSN (Print): 1868-7466
ISBN: 978-3-923959-54-9 (Paper)
eISBN: 978-3-923959-57-0

Zitation dieser Publikation

DRIEWER, F. (2009). Teleoperation Interfaces in Human-Robot Teams. Schriftenreihe Würzburger Forschungsberichte in Robotik und Telematik, Band 1. Würzburg: Universität Würzburg.

Teleoperation Interfaces in Human-Robot Teams

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Frauke Driewer

aus

Ettlingen

Würzburg 2008

Eingereicht am: 01.10.2008

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Klaus Schilling

2. Gutachter: Prof. Dr. Aarne Halme

Tag der mündlichen Prüfung: 08.05.2009

Danksagung

An erster Stelle gilt mein allerherzlichster Dank meinem Doktorvater Prof. Dr. Klaus Schilling für die Möglichkeit an diesem spannenden Thema zu arbeiten und für seine Unterstützung bei der Bearbeitung. In meiner Zeit an seinem Lehrstuhl habe ich viel von ihm lernen können und ich bin besonders dankbar für das Vertrauen, das er in den zurückliegenden Jahren immer in mich gesetzt hat. Weiterhin möchte ich Prof. Dr. Aarne Halme herzlich für die Erstellung des Zweitgutachtens danken, sowie den Mitgliedern der Prüfungskommission Prof. Dr. Rainer Kolla und Prof. Dr. Jürgen Albert.

Am Lehrstuhl für Robotik und Telematik des Instituts für Informatik an der Universität Würzburg hatte ich die Chance in einem aussergewöhnlichen Team mitzuarbeiten, in dem die Arbeit jeden Tag Freude gemacht hat. Danke dafür besonders an Markus Sauer, Florian Zeiger, Dr. Lei Ma, Rajesh Shankar Priya, Marco Schmidt, Daniel Eck, Christian Herrmann, Martin Hess, Dr. Peter Hokayem, Dieter Ziegler, Annette Zillenbiller, Edith Reuter und Heidi Schaber. Weiterhin möchte ich mich bei den Studenten bedanken mit denen ich zusammenarbeiten durfte, im Rahmen dieser Arbeit besonders bei Florian Leutert, Koami Edem Missoh und Matthias Wiedemann.

Ein Teil dieser Arbeit entstand im Rahmen meiner Mitarbeit im EU-Projekt "PeLoTe". Die Zusammenarbeit im europäischen Konsortium war immer spannend, bereichernd und erfolgreich. Dafür danke ich den beteiligten Kollegen an der Czech Technical University Prague, der Helsinki University of Technology, der CertiCon a.s., dem Steinbeis Transferzentrum ARS und insbesondere meinen Würzburger Projekt-Kollegen Dr. Herbert Baier sowie Grzegorz Zysko. Dr. Andreas Mühlberger danke für die Unterstützung bei der Auswertung der Benutzertests. Ebenfalls danke ich allen Teilnehmern, besonders der freiwilligen Feuerwehr Binsfeld und der Feuerwehrschule Würzburg.

Mein grosser Dank gilt Markus Schniertshauer für den vielfältigen Beistand und die permanente Motivierung während der ganzen Promotionszeit. Meinen Eltern und Geschwistern, sowie meinen Grosseltern danke ich für die jahrelange Unterstützung. Ihnen ist diese Arbeit gewidmet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The last decades have brought impressive advances in the area of robotics and consequentially many challenging applications are already a reality. Industrial robots assure fast and highly accurate assembly in production lines. The mars rovers “Spirit” and “Opportunity” demonstrate long-term planetary explorations since 2004[1]. First service robots enter private households in form of vacuum cleaning robots[2].

In recent years developments from using robots as highly-developed tools for a specific industrial application towards working alongside with multi-tasking robots in human populated environments have started. This results in a shift of requirements on different levels. The workspace is highly dynamic, unstructured, and partially unknown. Infrastructure for localization, navigation, and communication can often not rely on fixed pre-installations. The task range for a robot is larger. Tasks might change during the application and the robot might be used in a different way as intended. Finally, the way how human interacts with the robot changes. Human take nowadays mainly the role of operating from a separated environment. In future applications, human will interact with robots on different levels, as an operator, supervisor, co-worker, or first-time user.

One of the driving forces for this development is the desire to send robots in hostile, unpleasant or for human inaccessible environments and perform dangerous or tedious tasks with the robot. For example, in rescue operations the risk for human life could be reduced, if robots can be send to unknown and potentially dangerous environments as first explorers instead of human rescue workers. In construction teams robots can be used to bring heavy structures and equipment into the construction area or to work on repetitive and tedious tasks. Robots can support astronauts at missions on planets and thereby reduce the time in inhospitable environments.

Full substitution of people by autonomous robots is not possible nowadays nor is it desirable for most applications. Human still exceed robots in terms of cognition, the ability to adapt actions according to new situations, or the possi-

[1] http://marsrovers.jpl.nasa.gov/overview/

[2] Examples are the iRobot Roomba (http://www.irobot.com/) or the Kärcher RoboCleaner (www.robocleaner.de)

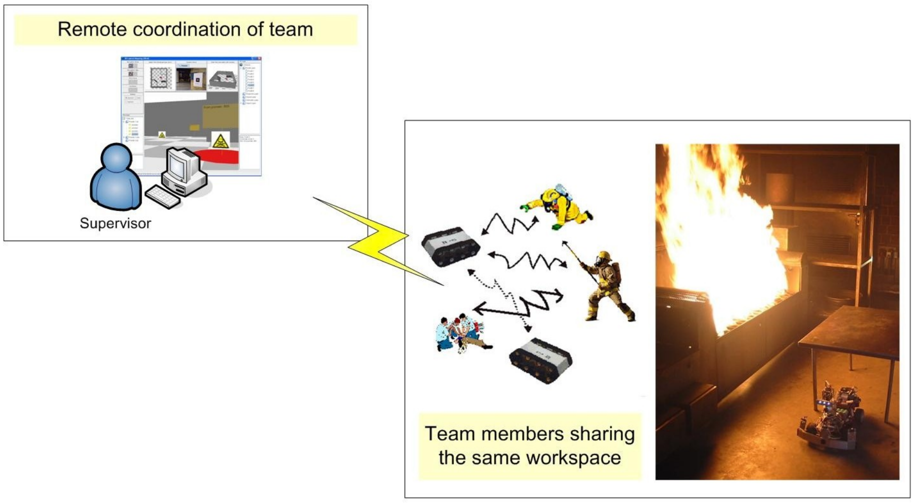


Figure 1.1: Exemplary application for human-robot teams: Robot-supported firefighting.

bility to integrate earlier experience. Effective human-robot teams combine the strength of both and achieve a higher performance than human alone or robot autonomy could. Introducing robotic team members into existing team structures is a challenging task, regarding communication, design, autonomy, task sharing, and interaction. At this point, there is only little understanding how human team members will percept the robots, how they will make sense out of the robots behavior, what degree and flexibility of autonomy is necessary and possible and after all if human teams understand robots as team members or rather as highly developed tools.

This work deals with teams in teleoperation scenarios, where one human team partner (supervisor) guides and controls multiple remote entities (either robotic or human) and coordinates their tasks (cf. Figure 1.1). Such a team needs an appropriate infrastructure for sharing information and commands. The robots need to have a level of autonomy, which matches the assigned task. The humans in the team have to be provided with autonomous support, e.g. for information integration. Design and capabilities of the human-robot interfaces will strongly influence the performance of the team as well as the subjective feeling of the human team partners. Here, it is important to elaborate the information demand as well as how information is presented.

Such human-robot systems need to allow the supervisor to gain an understanding of what is going on in the remote environment (situation awareness) by providing the necessary information. This includes achieving fast assessment of the robot's or remote human's state. Processing, integration and organization of data as well as suitable autonomous functions support decision making and task allocation and help to decrease the workload in this multi-entity teleoperation task. Interaction between humans and robots is improved by a common world model and a responsive system and robots. The remote human profits from a simplified user interface providing exactly the information needed for the actual task at hand.

The topic of this thesis is the investigation of such teleoperation interfaces in human-robot teams, especially for high-risk, time-critical, and dangerous tasks. The aim is to provide a suitable human-robot team structure as well as analyze the demands on the user interfaces. On one side, it will be looked on the theoretical background (model, interactions, and information demand). On the other side, real implementations for system, robots, and user interfaces are presented and evaluated as testbeds for the claimed requirements.

1.1 Exemplary Application: Supporting Fire-Fighters

Rescue operations, more precisely fire-fighting, was chosen as an exemplary application scenario for this work. The challenges in such scenarios are high (highly dynamic environments, high risk, time criticality etc.) and it can be expected that results can be transferred to other applications, which have less strict requirements.

Fire fighters and other rescue forces are highly imperiled, in particular if aside from fire, heat, and smoke other potential sources of danger are present, e.g. explosive or dangerous substances in industrial production facilities, or aggressive viruses in research laboratories.

Mobile robots can help to reduce risk for human life by taking over the critical first investigation. Based on the thereby gained insight, the rescue forces can enter the area better informed and equipped and are therefore better prepared. Furthermore, structured planning of the rescue mission is assisted by a higher information level. According to their equipment, mobile robots can also give other support, e.g. transport heavy life-saving equipment, hazardous waste cleanup, or search in blocked and difficult to access areas.

Until now their are only few incidents where robots have been used in real scenarios [115, 28]. Requirements and principles for rescue robot systems are not yet well-established. Participation with robots in response training exercises [25] or evaluation under rather idealistic conditions, e.g. in standardized test environments as in rescue competitions [187] helps to establish related guidelines. Often those refer to control of a single rescue robot and to the application domain of urban search and rescue (USAR), which is related to rescuing victims from confined spaces, e.g. caused by collapsed structures.

Today, rescue parties mainly have a strictly hierarchical organization. Normally, an operational command coordinates small teams that perform rescue tasks in the operational area. The distribution of specific tasks to certain team members is again organized in a hierarchical command structure. The demand on the correctness and suitable organization of the information flow during the mission is high. For the operational command it is important to have all information present that concerns the whole mission performance and situation, as well as the status, task performance and capacity of the teams. The team leaders need their assigned task, as well as details about the workspace and the team members.

Such existing team structures can guide the development of a human-robot team. The supervisor has, for example, comparable tasks as the operational command. Jones and Hinds [90] observed and analyzed a Special Weapons and Tactics (SWAT) team and designed a system for coordination of multiple remote robots. Adams [5] analyzes the work of special forces in a police and fire department for designing a human-robot system.

In larger disaster operations different organizations, as fire brigades, police departments, and hospitals, have to coordinate their actions, which results in an additional demand for systems to support organization and decision making, e.g. [130]. The human-robot system, developed in the work at hand, can be integrated in such crisis management tools. It can help to report and distribute up-to-date information and hence represents the local situation better. The human-robot rescue team is then a subpart in the rescue hierarchy, which is called if the actual emergency requires robotic support.

1.2 Contributions

The present work contributes to the introduction of human-robot teams in task-oriented scenarios, such as working in high risk domains, e.g. fire-fighting. It covers the theoretical background of the required system, the analysis of related human factors concepts, as well as discussions on implementation. An emphasis is placed on user interfaces, their design, requirements and user testing, as well as on the used techniques (three-dimensional sensor data representation, mixed reality, and user interface design guidelines).

Framework

First of all, the proposed human-robot system requires an infrastructure that allows the team member to communicate as required and to share all kind of information. Whereas in the area of networked robotics the primary focus is often on communication or architectures, here the emphasis is to define a suitable concept for a teleoperated human-robot team on system level. The well-known supervisory control [158] concept reflects the here selected team setup and is therefore chosen as a basis. This concept and its generic task design is adapted for controlling multiple entities. Interacting in a supervisory control scenario requires a certain amount of autonomy. The autonomy design of the system and the robots is elaborated. Situation awareness for the human-robot team is discussed. The developed framework provides the theoretical background of the human-robot system. The realization of the architecture was partly performed in the EU-project PeLoTe[3] and was extended for this work.

[3] PeLoTe - Building *Pre*sence through *Lo*calization for Hybrid *Te*lematic Systems. The project was performed in a consortium of 3 universities and 2 companies: Czech Technical University Prague, Julius-Maximilians-Universität Würzburg, Helsinki University of Technology, CertiCon a.s., and Steinbeis Transferzentrum ARS. An overview of the PeLoTe project can be found e.g. in [38, 99, 152].

Requirements

Based on the elaborated model for a teleoperated human-robot team it is analyzed in the next step what elements are required for the interfaces that connect humans and robots to a team. According to own user tests and available literature the requirements for such user interfaces are elaborated. A difference is made between user interfaces for the supervisor and the human team members that work in the remote area together with the robot. For the exemplary scenario, fire fighting, an end-user requirement analysis was conducted during the PeLoTe project. The results are especially analyzed with respect to the human-robot interaction and the user interfaces in the system. This evaluation is also a contribution to human-machine interaction in safety critical systems in general.

Design

The derived requirements are realized in the design of the user interface for the supervisor. Seven design principles are formulated based on existing guidelines from human-robot or human-computer interaction and the performed user studies. Several implementations for the user interfaces are introduced. A mixed reality approach for integration of a-priori information, sensor data as well as user input, e.g. based on observations, into a three-dimensional model of the environment is presented in more detail. This approach opens up the potential to integrate data from novel 3D-sensors as the PMD camera[4] for an improved environment perception. Furthermore, the user interface can be extended for visualization on large-scale stereo projection screens for an enhanced telepresence of the supervisor.

User studies

Finally, three user studies are described. The results are presented and conclusions for the presented user interfaces and the system are drawn.

1.3 Terms

Terms, as teams or robot, can be seen from different perspectives. This section gives short explanations for the most important terms of this work and shows how they are used in the remainder of the work.

Teams

The term *team* indicates a group of entities that perform together a common mission. These entities are typically heterogeneous (e.g. humans with different capabilities, different types of robots that are equipped with varying sensor systems,...).

[4]PMD - Photonic Mixer Device, the PMD camera provides additional to the gray scale images distance information [113, 140].

Humans and robots are not (yet) seen as equal, but each brings in complementary capabilities necessary for achieving the joint goal. Both have certain strengths as well as weaknesses, which are taken into account in the task division in the team.

Autonomy and Robots

Schilling, de Lafontaine, and Roth describe in [148] autonomy as the capability of a spacecraft to

- "to meet mission performance requirements for a specified period of time without external support,
- to optimise the mission product, e.g. the scientific measurements, within the given constraints."

A similar description of autonomy can be applied to the here considered mobile robots. For this work a *robot* is normally a semi-autonomous mobile vehicle equipped with different kind of sensors. Neither is the robot completely autonomous nor purely teleoperated by a human operator.

A *semi-autonomous robot* runs autonomous based on sensor information and programming for a restricted time duration without human intervention, e.g. following way-points while avoiding obstacles. The robot may have certain autonomous behaviors, e.g. it may recognize certain objects in the environment and treat the detection in an automated way. The robots are used to optimize the performance of the whole team, e.g. explore the emergency area and help the human rescuers to find safe and efficient routes.

Autonomous functions are not restricted to the robots, also other system components may support the user, e.g. automated selection of information which is presented to the supervisor.

Teleoperation

The term *teleoperation* is used in two ways. In the general case, it means operating/controlling at a distance, whereas otherwise it refers to direct and continuous human control, e.g. via joystick. The latter case is in the remainder denoted as direct teleoperation to prevent confusion.

1.4 Outline

The remainder of this work is organized as follows. Chapter 2 compiles the background and related research for this work. First, the field of human-robot interaction (HRI) is introduced and the present work is classified in relation to the different areas of HRI. Possibilities for evaluation of human-robot systems are shown. After a discussion of robot user interfaces and related studies, human-robot teams are introduced. Finally, mixed reality and its application for robotics is presented.

Chapter 3 elaborates the framework for teleoperated human-robot teams. Supervisory control is extended towards using this model for the proposed team structure. Problems with autonomous systems and situation awareness are discussed. Finally, the system design and the realization in this project are explained. Chapter 4 shows the challenges and information requirements for user interfaces in human-robot teams. Together with the related guidelines these are implemented in the interfaces, which are shown in Chapter 5.

Chapter 6 presents the performed user tests. The experiment, methods for evaluation, and results as well as observations are described. The work ends with a summary and conclusion in Chapter 7.

Chapter 2

Background

In this chapter the background for elaboration of the presented work is given. The field of human-robot interaction (HRI) is introduced and the topic of this work is classified in the area. Evaluation of human-robot system is discussed. Further, robot user interfaces are presented by showing a number of example implementation for single and multi-robot teleoperation as well as by presenting guidelines for designing such interfaces. The next section discusses related work on human-robot teams. The chapter ends with an introduction to mixed reality and its use for human-robot interfaces.

2.1 Human-Robot Interaction

Human-Robot interaction (HRI) concerns on the one side the robot (robot and autonomy design, behavior, sensors and sensor-processing, perception, algorithms ...) and the human on the other side (interaction roles, human capabilities and limitation, individual factors, as e.g. level of training, design of user interfaces, ...). Furthermore, it concerns the connection of both, which means the system and technology side as well as the model or theory side (interaction principles, understanding, system architecture, communication, modeling of interactions, ...).

HRI is a multidisciplinary area, where researchers from very different fields, as e.g. robotics, computer science, human-computer interaction, human-machine systems, artificial intelligence, cognitive science, or psychology, contribute with their expertise. In the past years HRI received high interest [3] and it was realized that related research is of significant importance for the introduction of robot technology into human environments [26].

2.1.1 HRI versus HCI

Human-robot interaction (HRI) as a relative new research area, can benefit from related fields, such as *human-computer interaction*[1] (HCI) and *human-machine*

[1] The following definition for HCI is given by the Association of Computing Machinery (Special Interest Group on Computer-Human Interaction): "Human-computer interaction is a discipline

interaction[2] (HMI). Well-established theories, models, and evaluation criteria can be utilized or used as a starting point for finding HRI-specific approaches. HRI can even be seen as a subfield of HCI [185].

Nevertheless, there are certain aspects where HRI differs from the other disciplines. In [71] it is emphasized that HRI differs

> "... from both HCI and HMI because it concerns systems (i.e., robots) which have complex, dynamic control systems, which exhibit autonomy and cognition, and which operate in changing, real-world environments."

Where HCI has its focus on users using the computer, HRI has its center on the combination of human and robot and all related problems. Methods from HCI or other related fields can only be used, if they enable to take the different nature of HRI into consideration [181]. Scholtz identifies six dimensions which specify the distinction [155]:

- *Interaction roles.* People with different background, expectations, and tasks may need to interact with the same robot. The interaction mechanism of the robot has to support interactions with different people [154].
- *Physical nature of the robot.* The robot actually moves and interacts with its surrounding. It needs to model the environment and this model has to be transfered to the user, such that the human can understand the robot's behavior.
- *Dynamical nature of the platform.* In many application domains it will be common that certain sensors or even the platform itself degrades, e.g. in smoky environments many sensors will have only limited functionality.
- *Environmental condition.* Depending on the application the robot's environment is dynamic, harsh or dangerous.
- *Number of systems the user interacts with.* The user might need to control several independent (and heterogeneous) robots.
- *Autonomous behavior of the robot.* A semi-autonomous robot can execute tasks autonomously for a certain period of time, i.e. it is able to sense and will adapt its task execution according to its surrounding. More intelligent systems are even able to learn and evolve their abilities on-line.

However, as long as there are only few theories, models, and special evaluation methods available, HRI researchers have to make use of known methods and evaluate if they can be adapted. Especially, in the area of user interface design for HRI many methods from HCI and human factors can be successfully applied (cf. Chapter 5).

concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." [83]

[2]Human-machine interaction considers similar problems as HCI in the interaction between human and machines in general.

2.1.2 Areas of HRI Research

The multidisciplinary nature of the topic entails that the research areas under the roof of HRI are very diverse. One can encounter a wide range of topics in the related conference proceedings, e.g. [1], or special issues, e.g. [101]. An extensive survey on HRI is given by Goodrich and Schultz [79].

A part of the HRI research is rather technology or engineering related, whereas other work is rather related to psychology and social sciences. A lot of research performed lies in between both forms. This work is rather technology related, but employs related approaches from psychology and social sciences were needed.

Research topics can also be classified according to e.g.

- application: e.g. rescue robotics, industrial robots, service robots, assistive robotics, toy and educational robots,
- robot shape: e.g. manipulator arms, wheeled robots, walking machines, humanoids,
- research approach: e.g. hardware or software implementation, algorithm, experimentation, design, study, observation, theoretical approach or
- type of experiment: e.g. field study, experimental laboratory experiment, Wizard of Oz experiment[3], simulation.

In this classification the present work concentrates on wheeled mobile robots for achieving special tasks, as e.g. necessary for rescue robots. It includes software implementations for the robots, the user interfaces and the system, which are based on design and modeling as well as experimentation. The studies are normally laboratory experiments with real or simulated robots.

Furthermore, a major distinction is made between *physical Human-Robot Interaction (pHRI)* and *cognitive Human-Robot Interaction (cHRI)*. pHRI concerns the physical interactions between humans and robot, e.g. haptic, forces, ekoskeletons, and safety issues [144], whereas cHRI concentrates on the combination of the user and the robot, e.g. situation awareness, human-robot teaming, adjustable autonomy, design principles, multi-modal interaction, and social interaction [26]. Even though pHRI is an important topic and needs to be considered for the presented human-robot team approach, this work focuses on cHRI[4].

2.1.3 Taxonomies

As HRI takes many different forms it is necessary to find a more formal categorization, which allows classifying the various systems. Yanco and Drury propose in [185] a taxonomy for HRI and give an update in [183]. They use taxonomies

[3]In Wizard of Oz experiments the test participants assume that they interact with an autonomous system, but actually a person generates the autonomous behavior.

[4]In the following HRI will therefore relate to cHRI

from human-system interaction [7], multi-robot systems (e.g. [51]), and computer-supported cooperative work (e.g. [53]) as examples.

In the following this taxonomy [185, 183] is used to describe the human-robot teams of this work[5]. Where needed the description is adapted. The different taxonomy categories are here further grouped according to the type of the category.

Application

The *TASK* classification describes the high-level task of the human-robot system. The task domain in this work is not completely fixed. Nevertheless, the system and interfaces were mainly designed with the search and rescue domain in mind (*TASK = search and rescue* or *support for fire fighters*). If the robots in the team have different tasks, a more detailed classification could be given for individual robots, e.g. $TASK_{robot\ 1\text{-}k}$=*exploration*, $TASK_{robot\ l}$=*carry equipment*,... The classification *CRITICALITY* concerns the type of task to perform. It is here set to *high*, since human life is affected in rescue operations.

Robot and Robot Team

The appearance of the robots in the teams (*ROBOT-MORPHOLOGY*) is *functional*. The value for the category *ROBOT-TEAM-COMPOSITION* is in general *heterogeneous* in this work, but *homogeneous* for the experiment described in Section 6.3.

Human-Robot Team Configuration

For this work, the ratio between human and robots in the team can be adapted to

$$\textit{HUMAN-ROBOT-RATIO} = \frac{\text{number of humans}}{\text{number of robots}} = \frac{s+n}{m}.$$

s is the number of supervisors in the team, which is for the here described system normally one. n is the number of human team members working in the same environment as the robot (teammates) and was in the experiments normally zero or one. m is the number of robots, which did not exceed three in the user studies.

INTERACTION denotes the approach how the robots are controlled. Hereby, *team* (human or robot) denotes that there is some kind of organization in issuing or performing the command involved. In contrast, *multiple* humans or robots implies that there is no agreement between the members of the group, everyone acts independently. Depending of n and m *INTERACTION* in the human-robot team is different. For $n > 0$ and $m > 1$ *INTERACTION = human team, multiple robots.* Otherwise, the description changes to *one human*, respectively *one robot.*

People can have a different values for *INTERACTION-ROLE* depending on how they interact with the robots [154]. For this work the human role is *supervisor*,

[5] The classification is here given in capital italics and the value(s) used for this work in italics. For a further description and other possible values [185, 183] can be used as reference.

operator, or *teammate*. A more detailed discussion of these roles follows in Section 4.1.1. The roles are not fixed over the complete mission, the human team members might change their roles accordingly.

Automated functions

Automated function can be located on the operator as well as the robot side. On the operator side decision support for operator is classified in subcategories:

- *AVAILABLE-SENSORS* is a list of all sensor types that are on-board the robot, e.g. {sonar, laser, odometry, video, ...}.
- *PROVIDED-SENSORS* is a subset of *AVAILABLE-SENSORS*, which holds all sensors that are directly visualized in the operator interface, e.g. {video, ...}.
- *SENSOR-FUSION* is a collection of functions, which fuse different sensor information and provide the result to the operator. In the list are the sensor types mentioned and the result for the operator support, e.g. {{{sonar,laser}→obstacles}, {{laser,odometry}→pose}, ...}
- *PRE-PROCESSING* is a list that includes the sonar types from which data is pre-processed as well as the result for the decision support for the operator, e.g. {{video→objects}, ...}

Depending on the scenario human team members also carry sensors. Obviously, this information and the related processing can be describes as above. On the other hand the human team members in the work area might also have a display and the decision support for them can be described in a similar manner.

AUTONOMY describes the autonomy level on the robot side measured as the percentage of time the robot is able to achieve sufficient performance in carrying out its assigned task on its own, i.e. without human intervention. *INTERVENTION* on the other hand measures the percentage of time when a human has to support the robot for achieving the wanted performance level (*AUTONOMY* + *INTERVENTION* = 100%). Robot control can vary between direct teleoperation (*AUTONOMY*=0%, *INTERVENTION*=100%) and full autonomy (*AUTONOMY*=100%, *INTERVENTION*=0%). In between these two extremes the values cannot be determined exactly. They rather represent an estimate for describing the autonomy design of a robot (cf. Section 3.3.1). Very often these values are also changing depending on the situation, e.g. in case of a communication drop-out a robot with normally a high value for *INTERVENTION* might increase the autonomy level and becames a fully autonomous robot until communication is re-established. If several robots are used the *AUTONOMY*/*INTERVENTION* relation can be given for each robot seperately (in a homogeneous robot team it will be equal for all robots). In some cases it might be also useful to give an estimation for several robots together, e.g. if the robots move in a formation.

Time and Location

In human-robot teams as considered in this work *TIME* = *synchronous*, i.e. all team members work on their tasks at the same time. For the supervisor *SPACE* = *non-collocated* and *PHYSICAL-PROXIMITY* = *none* as he/she works outside the workspace on coordinating the remote team. For the humans in the workspace *SPACE* = *collocated* and *PHYSICAL-PROXIMITY* is normally *passing* and in experiment of Section 6.1 *following*.

2.1.4 Social and Ethical Aspects of HRI

As the presented work is a rather technology oriented proof-of-concept, social and ethical aspects of HRI are not further taken into consideration. Nevertheless, they need to be kept in mind when designing intelligent systems and user interfaces that are to be used for supporting humans maybe even in life-threatening situations [163]. One important question is, for example, who is to be blamed if an error occurred? The user of the robot, the robot designer or the robot itself? The answer to this question not only concerns legal and therefore financial issues, but also influences the way how humans interact with robots if they solve a task together.

Kim and Hinds present in [96] their results from a study with a delivery robot in a hospital. They found that, the more autonomous the robot is, the more it will be blamed by users and the less the users will blame themselves or other co-workers. This and other similar findings may have significant influence on the autonomy design of a robot, but also on the design of the user interfaces.

2.2 Evaluation of Human-Robot Interaction

In the growing area of HRI the elaboration of broadly applicable evaluation methods and classification schemes is very important. Today, validation of interfaces for HRI is often highly dependent on the application area, which makes it difficult to compare different approaches. For task-oriented human-robot teams considered in this work, performance is an important evaluation factor, e.g. time to completely explore the whole emergency area. Nevertheless, pure performance measurement often does not explain why a certain performance was achieved and why other approaches may have reduced performance. Moreover, it does not tell anything about other important factors, e.g. user friendliness or workload. Here, one can draw on methods from the human factors area [4].

2.2.1 Methods

Scholtz proposes in [153] issues that need to be answered for a complete evaluation of the human-intelligent system interaction: availability and presentation of necessary information, efficiency of interaction language, efficiency of interactions from the human system perspective, scalability to multiple platforms and interactions, and support for the evolution of the platform.

The question whether the user has all needed information accessible can also be formulated in a question, which asks for situation awareness. Situation awareness can be evaluated e.g. by the Situation Awareness Global Assessment Technique (SAGAT) [55, 155] or the LASSO technique proposed by Drury, Keyes, and Yanco [48]. Situation awareness is further discussed in Section 3.4.

For the evaluation of interaction performance typical usability tests can be conducted [153]. The user interfaces can also be evaluated by heuristic evaluation [112, 125, 122], which allows informal usability analysis in an early stage and can be adapted for HRI [105, 30]. Another method from HCI is the Goals, Operations, Methods, and Selection rules (GOMS) technique [27, 49], which models the user interaction formally and allows prediction of performance time and learning.

For some applications it might be useful to analyze the workload of the user, e.g. by using secondary task performance measures [166] or the NASA-TLX (Task Load Index) [81]. Another subjective criteria, which can be adapted from psychology is the PANAS scale [179]. PANAS (Positive and Negative Affect Schedule) contains 10 positive (e.g. interested, excited, ...) and 10 negative (e.g. distressed, upset, ...) affects, for which the test participant gives a rating from 1 (very slightly or not at all) to 5 (extremely).

Kidd and Breazeal mention in [94] three measurement types for HRI: self-report, physiological, and behavioral measures, which all have advantages and disadvantages. They suggest a combination of self-report questionnaires and behavioral measures for HRI experiments. Furthermore, they recommend an experiment protocol with the following phases: introduction of robot and experiment, familiarization phase, start video recording, complete interaction experiment, apply questionnaire, conduct and record an interview, and explain the aim of the experiment to the subject. The authors also point out that there are additional difficulties in HRI compared to HCI studies, e.g. the need for a robust robot for different users.

The authors of [22] emphasize the use of methods from the area of sociology for describing and evaluating human-robot cooperation, here carrying a wooden bar together with a robot manipulator. They propose a classification scheme, which integrates technical details as well as sociological parameters.

2.2.2 Experiment Design

Walters et al. give in [176] a discussion on how to design and conduct human-robot interaction studies based on their experiences. They give practical advice, e.g. video camera placement or questionnaire design, explain experimental implementations, such as the Wizard of Oz methods, and discuss safety and legal issues. Finally, they show design and methodological considerations for planning a study. They also emphasize that sufficient time has to be taken to test and improve the experiment and the procedure.

The authors of [181] mention two problems in many of the HRI evaluations. First of all, test participants are often not drawn from the typical end-user community. Very often the systems are even only tested by the developers themselves, which can be seen as an upper limit, i.e. if they have trouble to operate the interface, normal users will have even more trouble. Secondly, the user tests are often performed rather informal, e.g. environment conditions are not kept equally for all tests.

In [103] it is presented that novice users are used as test participants, because the robot system to be tested is foreseen for several different applications. The integration of robots for certain tasks, e.g. characterization of radiation environment, will change the role of the human in the tasks, which means the system has to be designed for novice users. Even though the authors agree that end-users finally need to test the system, they argument that evaluation with novice users increases the impact of a study for different applications and reveals the lower limit of performance. Finally, for many applications testing with enough end-users is not feasible, whereas normally a high number of novice users can be easier achieved. They also show guidelines for testing the usability of an interface or an architecture, which suggest for instance that tests are to be held under real-world conditions with similar complexity and uncertainty in the environment and task.

2.2.3 Metrics for HRI

Crandall and Cummings explain in [34] that a set of metrics classes for human-robot teams should include key performance parameter of the system, determine the limit of robots and human operator, and allow prediction of the team performance. They propose a set of metrics and evaluate it in a user study.

Steinfeld et al. introduce in [169] common metrics for task-oriented HRI regarding human, robot, and system. They propose classes of task metrics, which allow evaluation for a wide range of applications and better comparison between different systems. The work at hand is of the area of task-oriented human-robot systems and therefore several of the proposed metrics apply. Hence, the task and common metrics from [169] are shortly presented in the following. The task metrics are given in five categories:

- Navigation (global and local navigation, obstacle encounter)
  Measurement of *effectiveness* (e.g. completeness of navigation task completed, area coverage, deviation from given path, obstacles avoided or overcome), *efficiency* (e.g. time to complete, time operator needs to spend), and *workload* (e.g. operator interventions, time operator spends for input compared to the time the robot navigates).

- Perception
  Passive perception (interpreting sensor data) can include *identification* measures (e.g. percentage detected, accuracy of classification), *judgment of extent* measures (e.g. absolute and relative estimation of distance, size, length), and *judgment of motion* measures (e.g. judgment of absolute robot velocity or relative velocity to other moving objects).

Active perception involves movements of robot or sensor/camera based on a detected possible search target or sensor fusion of different sensors. Measures are *active identification* (e.g. time to approve the identification, amount of movement of the camera), *stationary search* (e.g. accuracy for detection in sensor range, time to search), and *active search* (e.g. time and effort spent for target identification, errors).

- Management
  Fan out measures how many homogeneous robots can be controlled by a human [129, 33].
  Intervention response time indicates the delay between the moment a problem is encountered by the robot until the operator reacts.
  Level of autonomy discrepancies, e.g. the measurement of operator ability to determine the appropriate level of autonomy for the situation.

- Manipulation (degree of mental computation and contact errors)

- Social (interaction characteristics, persuasiveness, trust, engagement, compliance)

Especially the categories of navigation, perception, and management are relevant for this work. Further, common metrics are divided in:

- System performance:
  Quantitative performance including effectiveness, i.e. accomplishment of the task in percent, and efficiency, i.e. time to complete, *subjective ratings* and the *appropriate utilization of mixed-initiative.*

- Operator performance:
  Situation awareness, *workload*, and *accuracy of mental models of device operation.*

- Robot performance:
  Self-awareness of the robot of its capabilities, *human awareness*, and *autonomy*, e.g. neglect tolerance.

2.3 Robot User Interfaces

As nearly all of todays robot applications need human intervention at least to some extent, the robot user interface is certainly one of the most critical components in a robot system. On one hand the user interface has to enable the robot to convey information to a user, e.g. present its internal state, position, world-view, or just do human-like conversation. One the other hand the user wants to give input to the robot with the interface, e.g. navigate the robot, assigned a task, ask for specific information, or just answer to the conversation attempt.

Depending on the application domain very different paths are taken in the endeavor to design the optimal user interface, e.g. in the service robot domain gestures or natural language interaction are of high interest, e.g. [80, 73, 160]. In the area of remotely controlled vehicles graphical user interfaces are often used. A graphical user interface (GUI) allows to visualize a high amount of information and the input of complex command sequences. The challenges are to decide which information is relevant and how an efficient and intuitive interface has to look like. This section concentrates on graphical user interfaces for remotely controlled vehicles.

2.3.1 Vehicle Teleoperation

Fong et al. give in [70] four categories for vehicle teleoperation interfaces:

- With *direct interfaces* the user controls the robot manually, e.g. based on camera feedback with a joystick. The robot needs only little autonomy, but low-delay and high-bandwidth communication is required.
- *Multimodal/-sensor interfaces* allow the user different control modes or integrate information from various sources in one view.
- In *supervisory control interfaces* the user is able to monitor the remote scene and give related high-level commands [158].
- *Novel interfaces* have special input methods, e.g. gestures, or are used in certain novel application areas. The term novel is obviously not permanent. A today's novel interface might be state-of-the-art in a few years.

Even though some teleoperation interfaces for robots fall clearly in one of the four categories, many interface can have components from different categories, e.g. a multimodal user interface that applies a supervisory control approach. Especially, if tasks get more and more complex and environments and other external conditions (e.g. communication) are restrictive, a reasonable combination of all four is needed. These four categories can rather be considered as different design approaches, which can can complement each other in the same interface.

2.3.2 Example Interfaces

There exists a wide range of different graphical user interfaces (GUIs) for teleoperation of robots. Most of them are experimental, i.e. not commercially available. Nevertheless, some GUI elements are now state-of-the-art for teleoperation interfaces (map of the environment, visualization of robot's position and state,...) and several studies have proved that they are needed for successful robot remote control. However, there exist many different ways to visualize these elements.

Robot user interfaces are often developed for specific robots or for dedicated tasks. Quite many were for example designed with the search and rescue domain in mind, e.g. for one of the rescue competitions. Some are implemented for single robot operation, some can be extended for multi-robot operation and others are foreseen especially for multi-robot operation.

Figure 2.1 and the following section shows some examples for single robots from the literature. The list cannot be complete due to the high number of different interfaces, but it presents some representative approaches or interfaces that apply certain features.

Single-robot teleoperation

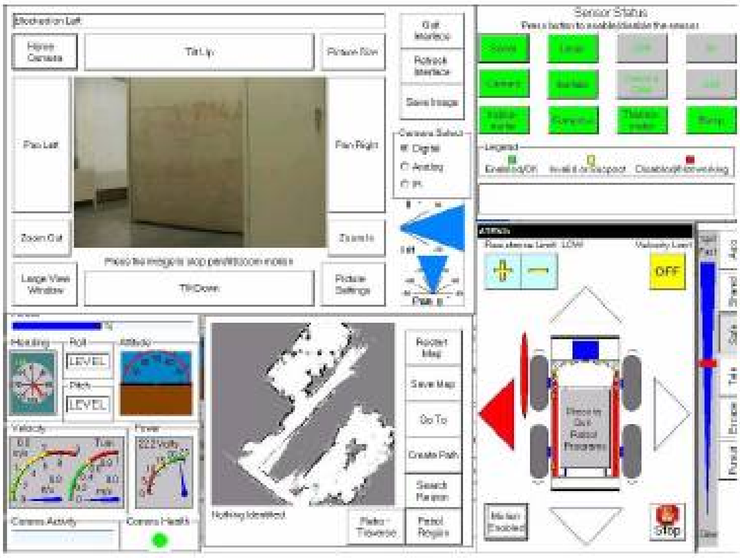
The interface [17, 19] in Figure 2.1(a) is an iteratively designed and well-tested graphical user interface. It is a rich information display with five windows: video from the onboard camera and camera control buttons, status and control for the robot sensors, distance information and control for the robot's movement as well as autonomy mode selection, a map of the environment based on SLAM[6], and state information, e.g. power, communication state. The interface provides an expert user with all necessary information for analyzing the robot state, sensor conditions and the direct surrounding of the robot. Novice user focus on the most relevant information, the video and the robot control [10].

Baker et al. [10] modified the system by designing a new interface, which is shown in Figure 2.1(b). Information is combined, reduced, or relocated and additional features are implemented. This GUI has a focus on video, which is augmented by additional information, e.g. pan and tilt indication. Distance information is shown as colored blocks around the video display and the environment map is arranged on the side. A rear camera image is shown above the front video. Rarely needed information, as e.g. battery level or sensor malfunction, is reduced or even removed, but the system gives an alert when a critical value is reached, such that the user attention is drawn to this data. Furthermore, the system gives suggestions for autonomy modes [11].

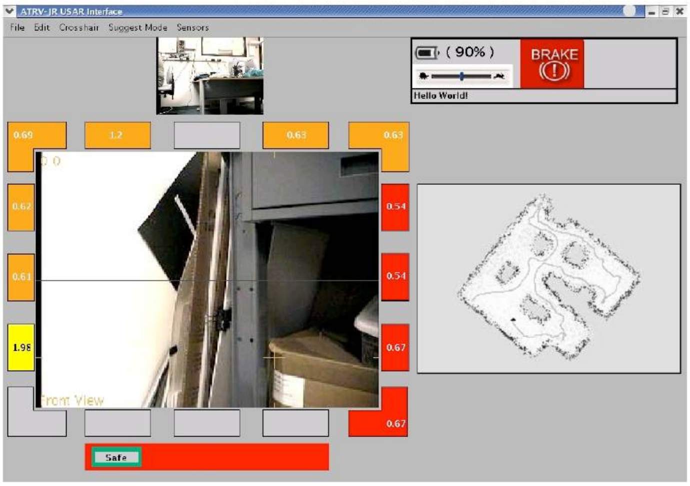
Bruemmer et al. [19] assert that it can be a problem for real application if the user interface relies heavily on video due to bandwidth requirements, image quality, and field-of-view. They compare the interface of Figure 2.1(a) (video and 2D map) with a Virtual 3D display [119] (cf. Figure 2.1(c)) in an exploration task, where the map is built online. In the interface of Figure 2.1(a) this map is presented with a two-dimensional map from a top-down (exocentric) view. In the Virtual 3D display the map data is presented three-dimensional and the user can change the perspective as needed from an egocentric (robot) to an exocentric view. The robot is modeled as an avatar and is moving around in the virtual environment according to the real robot's movements. The virtual world can be further enhanced by semantic information[7], e.g. still pictures taken at special locations with the robot's camera [119]. Life video images of the robot's onboard camera can be shown in front of the robot avatar, but these function is switch off for the experiment presented in [19]. The results showed no significant difference in task performance for both GUIs, but a decreased workload and slightly increased feeling of control for the participants that used the Virtual 3D display compared to the video-based display. Furthermore, the Virtual 3D display requires only a

[6]SLAM - simultaneous localization and mapping (robot maps the environment and at the same time tracks its pose [21])

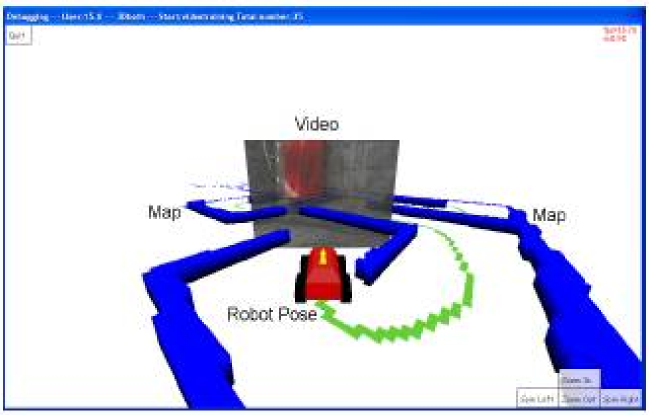
[7]*Semantic maps* are here environment maps that are augmented by additional information, which gives a meaning to the location where the information is inserted.



(a) Adopted from [19].



(b) Adopted from [10].



(c) Adopted from [120].

Figure 2.1: Examples for graphical user interface for single robot operation.

fraction of bandwidth compared to video transmission and allows teleoperation also in environments of low visibility.

The concept of ecological interfaces, in form of the previous mentioned Virtual 3D interface, was further developed and evaluated related to several aspects in simulation and real-world studies [138, 120, 121, 118]. A good overview of the development and experiments is given in [117]. User tests demonstrated improvements compared to a 2D interface in controlling the robot, building a map, arranging multiple, competing sets of information, robustness against delays, managing a pan tilt zoom camera during navigation, as well as searching the environment. Moreover, the test participants preferred the 3D interface, rated their own performance better, and expressed lower frustration. Finally, the operators kept the distance between obstacle and robot higher, which is a sign for a better awareness of the robot surrounding. The authors [117] explain the good results in evaluation of the 3D interface mainly with three principles: (a) use a common frame of reference, (b) correlate action and response in the display and (c) give the user a possibility to change the perspective.

Another study compares the interface concepts of [117] (map-centric, 3D map and video, Figure 2.1(a)) against a video-centric approach (video and 2D map, modified version of Figure 2.1(b) [10]) in a search task performed by eight test participants from the USAR domain. In [182] performance (area coverage, number of found victims, and number of bumps) and user preference were analyzed. The area covered in the test was significant larger with the map-centric approach. There was no significant difference for victims found and bumps on the front of the robot. Bumps to the rear appeared less with the video-centric approach, which had a small rear-facing camera image in the interface. The video-centric approach was preferred in ease of use and helpfulness of controls. Nevertheless, users preferred the 3D to the 2D map view. In [48] situation awareness regarding five awareness categories: location, activities, surroundings, status (health and mode), and overall mission, was analyzed. The map-centric interface scored better in location and status awareness, whereas the video-centric interface was more effective in surroundings and activities awareness. For the overall mission awareness none of the interfaces showed a benefit.

The GUI of Eck, Stahl, and Schilling in [52] was designed especially for teleoperation of a robot outdoors in rough terrain. Except from camera images and sensor data it also shows a large-scale map of the environment including building, streets, and other landmarks.

Apart from the presented concepts other ideas for supporting teleoperation tasks have been developed. In [88] the concepts of a sensory egoSphere as a short-term memory for supporting robot navigation is presented. Local information is projected on a sphere with the robot in the middle. In [12] scripts are used for improving human-robot coordination in robot-assisted search of void spaces. A fixed task sequence for the search (here localize the robot visually, observe the situation with the camera, scan for victims with a thermal camera and report finding to a partner) is to be performed by human and robot together. The GUI visualizes every step and sub-step of the sequence as well as if it is a task for the operator or an automated task in a script panel. Previous, current and future task states are shown and the operator can start, pause/resume or end the search task. Other authors, e.g. [104, 91], use computer games, i.e. first-person shooters, as examples and transfer features from games to single-robot teleoperation for USAR competitions.

The previous mentioned examples normally require at least a laptop computer due to the size of the GUI. Keskinpala, Adams, and Kawamura show in [93] a PDA (Personal Digital Assistant) based interface for controlling a robot. Control input is given with four large, transparent buttons by touching on the screen. Three information screens are implemented: vision-only (camera), sensory-only (ultrasonic and laser), and vision with sensory overlay. In [2] workload is evaluated for the different screens of the PDA interface.

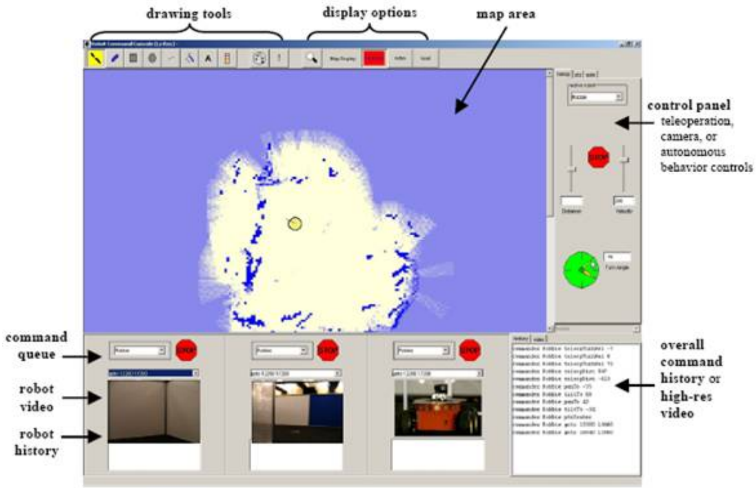
Multi-robot teleoperation

Some of the above mentioned concepts have the possibility to be extended to multi-robot control interface, but in the current state they are used for single-robot teleoperation. The following examples have been designed for controlling multiple robots.

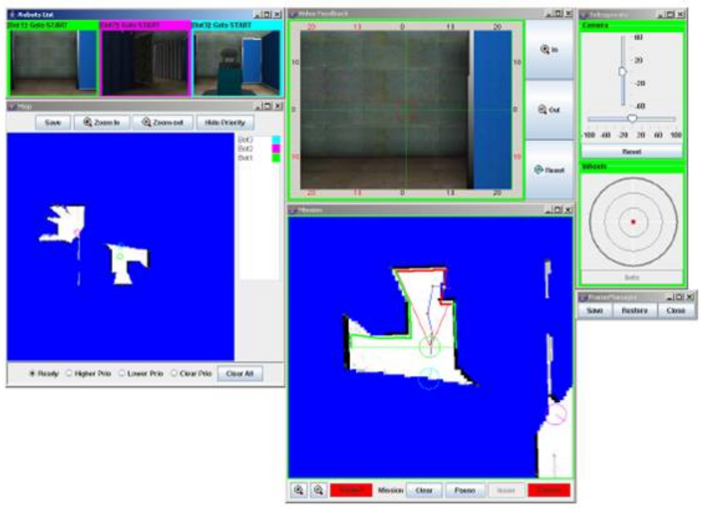
Drury et al. show in [46] an interface for a small number of robots, which was tested with one real robot. The main element of the GUI (cf. Figure 2.2(a)) is a map that shows the position of all robots as well as fused sensor data and has three layers (probability of obstacles, probability of victims, and explore/avoid layer). The interface provides drawing tools for operator input to the map and a control panel for teleoperation, camera control, and autonomous behaviors. For each robot a pane with video, status messages, command history and a stop button is visualized. Wang and Lewis show in [177] a graphical user interface (cf. Figure 2.2(b)) that includes a thumbnail of each robot's onboard camera and a global map. For a selected robot a larger camera window with related camera control buttons as well as the robot's local map including the robot actual plan is shown. The selected robot can be controlled by setting waypoints in this map or by direct teleoperation. Different control strategies (autonomous, manual control, mixed initiative) were tested.

A general problem of interfaces for controlling multiple robots is scalability. Especially if a pane with video is shown for each robot the interface can only scale to a certain number of robots. Moreover, it is difficult for an operator to focus on more than one video from different scenes at the same time.

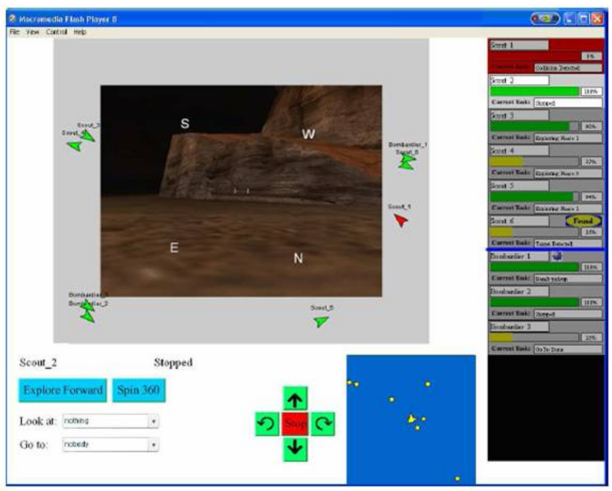
Humphrey et al. suggest the use of a so-called halo display to achieve a scalable multi-robot user interface [86]. Their GUI (cf. Figure 2.2(c)) shows the camera image from the selected robot. Relative position and orientation information of other robots is visualized with arrows at the appropriate position in the area surrounding the camera image (the halo area). The color of the arrows represents the status. Further, the interface has task status bars for each robot, a global view of the robot positions, and a control panel. In the evaluation the effect on workload, situation awareness, and robot usage are tested when six or nine simulated robot are used for a bomb defusing task. The results showed that the



(a) Adopted from [46].



(b) Adopted from [177].



(c) Adopted from [86].

Figure 2.2: Examples for graphical user interface in multi-robot operations.

interface supports scalability, e.g. increasing the number of robots from six to nine robots did not increase the workload equally strong.

For multiple robots it might be necessary to group some robots into smaller teams or formations [82]. In [85] three visualization possibilities for such robot teams are shown: individual robots, connection of the robots in one team by a semitransparent shape, and the complete team by a solid shape. The shape is a line connection of the different robots in the team. These visualizations of the team are shown in an environment map, different teams are differentiated by colors. A tab display can show information about an individual robot (position, status, distance to goal), about a specific team (status of each team member, identifier of the last goal accomplished), and about all teams (team color, status, distance to next goal). A simulation study showed that the two visualizations of the team were more often used than visualization of individual robots. In [62], another approach for interacting with teams of robots instead of individual robots is presented. Here, task list were evaluated in a simulation to supervise 18 robots in three teams. The two last examples can be useful in supervision of large groups of rather autonomous robots, which move as sub-teams in formations. Fitzgerald and Thomas propose in [167] stigmery-based control for exploration of large areas with a high number of robots. This allows giving high level goals to the team, since a behavior is send to the group of robots instead of instructions for each robot and enables to implement a very simple user interface, where the user enters the command vector for movement of the group. Interaction with individual entities is not foreseen.

In [155] an example of an interface for on-road driving with a robotic vehicle is shown. The GUI shows vehicle information (speed, fuel level, sensor information, current position including the road,...), environment information (traffic around, speed limit, road condition,...) and route information (distance to goal,...). The icon of a vehicle in the road map combines indication of vehicle and environment status. The outer shape (square, circle, diamond) and color (green, yellow and red) symbolize the environment status (normal, caution, trouble), whereas the inner shape visualizes the vehicle status.

2.3.3 Studies and Guidelines

As explained, there exist many different design approaches for teleoperation interfaces. In general, they are difficult to compare as they are developed in different laboratories all over the world and no common benchmark tests are established. Yet, some studies were e.g. conducted at occasions as the AAAI[8] Robot Rescue Competition, which are held in standardized environments [87]. In this and similar competitions different robot systems and therefore also different interface concepts are applied to explore a remote environment and search for victims. Based on these studies, the authors identified typical problems of user interfaces and suggested guidelines to overcome these problems.

Scholtz et al. conclude in [157] from observation at the Robocup 2003, that information displays for urban search and rescue robots should include a frame

[8]AAAI - American Association of Artificial Intelligence

of reference, robot status indication and integration of sensor data from various sensors. Furthermore, the robot should be able inspect itself and support for automatic presentation of contextually-appropriate information is needed.

Yanco, Drury, and Scholtz [181] derived initial guidelines for designing interfaces for HRI from the analysis of four different robot systems that took part in the 2002 AAAI Robot Rescue Competition. Two of the systems were also tested with rescue domain experts. In summary, the guidelines suggest to provide a map of areas in which the robot was moving, fused sensor information rather than different sensor streams separately, and more spatial information concerning the robot in the environment. Moreover, user interfaces for multiple robots should use a single display for all robots and in general the use of multiple windows should be minimized. Finally, suggestions for the appropriate autonomy degree can be helpful.

In [187] the previous study is extended by observing teams in the 2002, 2003, and 2004 AAAI Robot Rescue Competition. The resulting design guidelines primarily apply to USAR situations, but might apply for remote control of robots in general. The authors suggest to use a single monitor as switching the attention between multiple monitors seems to decrease the performance. The size of the video image window seems to influence the performance, i.e. a large video window supports the navigation and the search task. Different windows should not occlude each other, as switching to the needed window slows down the user. When multiple robots are applied, one can serve as an observer of the others, e.g. if one robot is stuck and needs to be recovered. Finally, the user interface has to be designed for the intended user.

Goodrich and Olsen [78] compile seven principles for efficient HRI on basis of lessons learned from experiments with their own interface approaches:

- Changes in the control or autonomy mode of the robot and in the information display should be possible implicitly, i.e. should not burden the user additionally. Though, implicit user commands have to be visualized in the interface to avoid confusion.
- The use of natural cues (icons, gestures, sketches) supports efficient interaction.
- The possibility of directly manipulating the world (e.g. the camera image) should be preferred before commanding the robot, i.e. draw attention to the task instead to the robot or to the interface.
- Otherwise the manipulation of the relationship between the robot and the world, should be preferred before manipulation of the robot.
- The user should be enabled to manipulate the presented information directly in order to navigate the robot or perform the task.
- Sensor information should be presented in an integrated fashion and kept as a history, which will decrease the demand on the short-term memory of the user.

- The user interface and the robot autonomy design should support the user with attention management, e.g. through colors, flashing, highlighting, or sorting.

Steinfeld [168] took a different approach for compiling interface lessons. He interviewed six experts (i.e. having extensive experience with interfaces for autonomous or semi-autonomous mobile robots) from the Robotics Institute of the Carnegie Mellon University. As a result seven categories, that need to be addressed when designing user interface for HRI, were identified: safety, remote awareness, control, command inputs, status and state, recovery, and interface design. For each category more detailed recommendations are given.

Ferketic et al. describe in [64] and [63] the way towards standards in human-robot interface especially for the domain of space exploration and express the demand for establishing such standards.

All the previously mentioned suggestions are more or less based on laboratory studies. Therefore, the few known investigations of robots in real scenarios are of high significance. Murphy and Burke describe in [115] lessons learned from three responses (World Trade Center 2001, Hurricane Charley 2004, and La Conchita mudslide 2005) and nine realistic field exercises. Their findings are related to human interaction with remote robots in general.

- The experiences had shown that building and maintaining situation awareness and not autonomous navigation is the major shortage, i.e. even if the robot is able to navigate fully autonomous, the human cannot be eliminated (e.g. for searching victims, assessing the surrounding). Robot operators spent a lot of time to understand the state of the robot and the environment, but communication with other team members helped to overcome this absence of situation awareness. This calls also for progress in sensor systems, data processing and interpretation for improving situation awareness as well as for strategies that support team communication.

- Human-robot interaction should look at the integration of the robot in a team of rescue experts as an active source of information. Information is filtered according to the hierarchy of the team member and distributed team members use the same data differently.

- Shared mental models as well as the team communication is supported by shared visual information from the robot's view. This requires reliable communication of high bandwidth.

- Rescue robots are normally not anthropomorphic. Nevertheless, humans working side by side with the robot interact socially with the robot, e.g. keeping *eye contact* and *personal space* etiquette, use of gestures.

If applicable, guidelines and suggestions from the literature mentioned in this section have been taken into consideration for the design as it will be discussed in the later sections.

2.4 Human-Robot Teams

Apart from the user interfaces an infrastructure that enables interaction between the team members has to be established. It has to provide maintenance of information and data sharing abilities as well as enable communication between team members. Another important aspect of a human-robot team is the robot design, i.e. data processing and autonomous functions. In this work, not the efficient realization of algorithms for autonomous behaviors is of interest, but the way how humans understand and how they interact with autonomous entities in the team. This knowledge has to be integrated in the design of the interfaces such that they support the understanding of autonomous behavior and allows interaction with autonomous functions. The human-robot system needs also functionalities to facilitate team work between human and robots.

2.4.1 Multi-Entity Systems

As the taxonomy classification *INTERACTION* of [185] shows a human-robot systems can have various configurations and therefore different requirements apply on the infrastructure. Also other classifications have a major influence on the system design, e.g. the *SPACE* taxonomy.

A high number of different architectures for multi-robot cooperation and distributed robot control (e.g. [133, 89, 143]), as well as robotic development environments (cf. [97]) have been developed in the past. Challenges are the coordination of heterogeneous robots, interoperability between existing teams, reaction on changing goals, and the integration of appropriate communication and teamwork models [126].

Most relevant for this work is the agent-based human-robot interaction operating system of Fong et al. [68, 67, 69]. This software framework enables peer-to-peer human-robot interaction by providing the facilities for a task-oriented dialog between humans and robots. A variety of user interfaces and therefore interaction modalities are supported and different robots can be integrated. The system was developed for human-robot collaboration in future space exploration.

2.4.2 Interacting with Autonomy

Even highly automated vehicles require interaction with humans [186]. Therefore, special care has to be taken in designing system and user interfaces for proper utilization of autonomy functions.

Parasuraman et al. describe in [131] that autonomous functions can be applied to (a) information acquisition, (b) information analysis, (c) decision and action selection, and (d) action implementation. They emphasize that careful selection of the autonomy design for each of the four functions is of high importance for any automated system and propose a framework that enables designers to choose the right autonomy level for every type. In Section 3.3 the proposed model is applied for the autonomy design of the human-robot system.

Stubbs, Hinds, and Wettergreen [170] apply the four functions of [131] to robotics as *autonomous sensing (information acquisition)*, *autonomous planning (information interpretation and decision selection)*, and *autonomous acting (action implementation)*. They compared three trials with a science robot in a field study, where in every trial the level of autonomy in sensing, planning, and acting was increased. The robot and an engineering team were located in Chile. A science team remotely operated the robot from Pittsburgh and searched for signs of life in a desert. The analysis of occurring problems showed that with increasing autonomy level the kind of problems changed from *missing contextual information related to data-collection* towards *lack of transparency with respect to robot's decision making.*

The autonomy level of a robot lies between the extreme cases of direct teleoperation and full autonomy. Depending on the application it is not set to a fixed level over the course of the mission. Related concepts are adjustable autonomy [76, 137], mixed initiative [102, 116], and shared control [100, 132].

In the past years a number of experiments with different autonomy levels respectively different control concepts have been performed. For example, in [17] two autonomy levels are compared in a search task with a remote robot: *safe mode*, where the robot is manually controlled, but protects itself from collision autonomously, and *shared mode*, where the user only gives directions and the robot navigates based on the local environment, but the user can take over control if needed. The results showed that participants that used higher autonomy (shared mode) performed better (found a higher number of objects). In [65] the *standard shared mode* is then compared to a *collaborative tasking mode*, where the user defines the task goal, whereas the robot is responsible to supervise the direction itself. The results for the task performance for both modes were similar. However, the participants that used the collaborative tasking mode showed a slightly better performance in a secondary task, they had to solve simultaneously. They also rated their feeling of control higher and for them less navigational errors were detected.

Wang and Lewis compare in [177] three different levels of autonomy in a victim search task with three simulated robots. In manual control users could give waypoints, or directly teleoperate the robot and operate the onboard camera. Under mixed initiative the robots used local environment information to explore the area and cooperated with other robots to prevent moving in already explored areas. The user could take over manual control of a robot if needed. In the fully autonomous mode the robots navigated and identified victims on their own without any human operator. The authors evaluated that with mixed initiative more victims were found and the area was searched more complete than with manual control or with full autonomy.

The experiments in the literature show that a certain amount of autonomous capabilities is highly beneficial compared to direct teleoperation. Nevertheless, fully autonomous systems that do not require human intervention at any time are not yet feasible and are not desirable in many application areas. Moreover, even the control of semi-autonomous vehicles requires a careful design of the autonomous functions and the related user interfaces. Yanco and Drury emphasize in [184] that it is difficult for a human operator to maintain awareness if the robot is in autonomous mode. Another problem is that users typically do not choose the most appropriate level of autonomy, which asks for methods as autonomy mode suggestions [11].

2.4.3 Acting as a Team

The wish to use robots in novel application areas, e.g. search and rescue, and the resulting need for higher degrees of autonomy and efficient instruments for interaction with human cause a change in the view on robots: away from teleoperated machines (*robot as tools*) towards team partners (*robot as peers*).

An essential property of a team is the joint work on a shared goal. Often human team members change their role according to new situations. Similarly, robot team mates have to be able to vary their level of autonomy depending on the environment and situation. However, this role switching needs to be present in the interfaces, as otherwise team performance might degrade. It is important that each team member knows the current capabilities, needs, and weaknesses of others as well as is able to understand and predetermine the action of others [18].

Fong, Thorpe, and Baur [71] propose *collaborative control* as a concept for teleoperation, where human and robots work as partners on achieving a common goal. In this control concept the robots need to be *self-aware* (e.g. know its limitations) and have *self-reliance* (e.g. protect itself). A system for collaborative control needs to enable human and robot to conduct a two-way *dialogue* and has to be *adaptive* with respect to the skills of the user. Based on this concept the human-robot interaction operating system (cf. Section 2.4.1, [67]) was developed.

When humans and robots work together side-by-side perspective-taking supports the naturalness of the human-robot communication [174]. Teamwork between human and robot in a shared location can also be facilitated by robots that anticipates the action of the human and adapts its own actions accordingly [84].

2.5 Mixed Reality

As seen in Section 2.3 a major problem of robot user interfaces is the representation of the high amount of data that is needed to efficiently control a robot or even a team of humans and robots in a remote environment. Mixed reality (MR) provides for this purpose a promising approach. In the following the concepts of MR and its application in robotics are discussed.

2.5.1 From Real to Virtual

The term *virtual reality* (VR) originally referred to immersive virtual reality systems, which use special hardware such as head-mounted displays or CAVE^{TM} systems[9] to enable immersion into an artifical, three-dimensional, computer-generated

[9] CAVE^{TM} - Cave Automatic Virtual Environment

world. Nowadays, the term goes beyond this classic definition and also systems with three-dimensional worlds on a normal desktop computer may be called virtual reality.

Milgram and Kishino show in [108] the concept of a *virtual continuum*, which runs from the complete virtual to the complete real world. Between both extremes lies the so-called *mixed reality* (MR), which can be divided into *augmented virtuality* (AV) and *augmented reality* (AR) depending on the ratio of virtual and real information that is combined in the system. Figure 2.3 shows several robot user interfaces arranged in the virtual continuum.

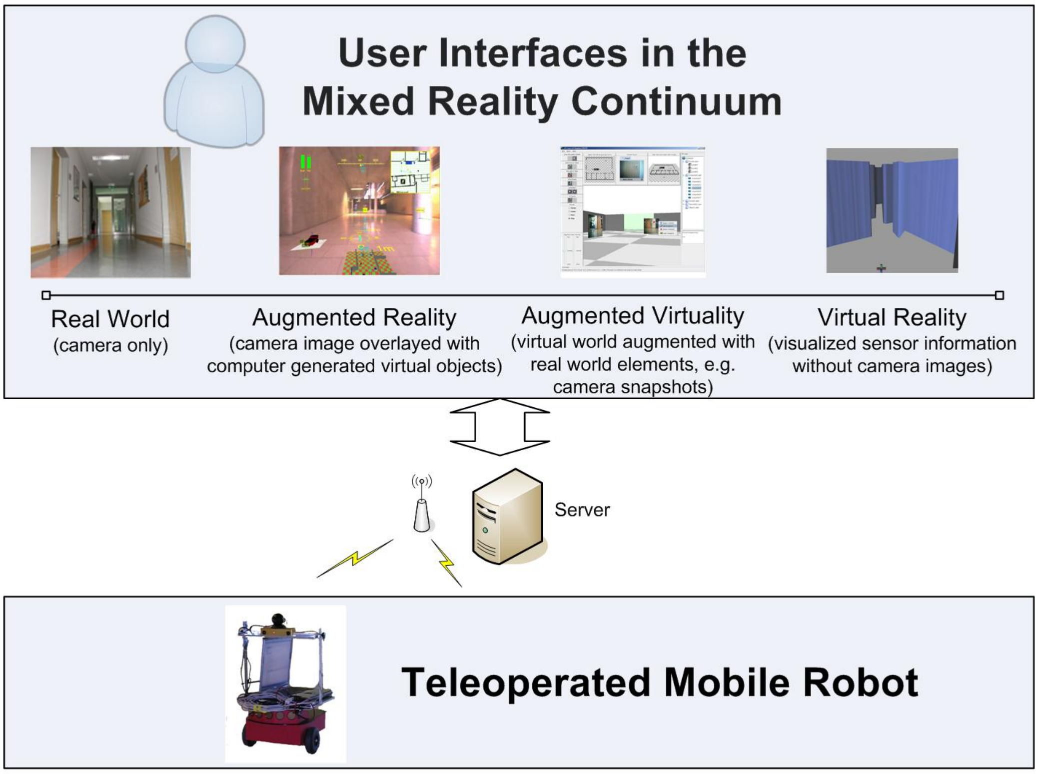


Figure 2.3: Robot user interfaces in the virtual continuum [145]

Azuma gives with [9] a survey on AR and explains that AR systems have three characteristics: (a) they combine real and virtual world (b) they are interactive in real time and (c) they are registered in three-dimensional space. This definition excludes films as well as two-dimensional overlays and entails certain technological challenges for the implementation of AR systems. One of the most important problems to solve is the registration problem, i.e. the proper alignment of virtual and real objects. Often optical tracking of markers is used, e.g. the ARToolkit [92, 175] was developed particular for AR systems. The advantage of this method for alignment is, that the real world (camera image) is used to determine the position of the virtual object, which is calculated in relation to the tracked markers that are attached to objects in the real world. This means, the virtual objects are not only very precisely positioned, but the overlay is also time synchronized with the camera image, i.e. delays do not disturb the user, which is very important especially for moving objects. The drawback of this method is that it needs markers attached in the environment, which makes it unusable for remote control of robots in unknown environments. Another challenge in AR is the occlusion problem, i.e. virtual objects may hide parts of the real world wrongly. AV system face the same challenges, but often the requirements e.g. on accuracy are less difficult to meet.

2.5.2 Mixed Reality in Robotics

There exists a variety of different VR/MR approaches for robotics applications, e.g. robot programming, plant layout simulations, supervisory and collaborative control or teleoperation with poor visual or delayed feedback [20]. Some examples are shown in this section.

In most systems the real world is represented by a camera image. Here, external cameras refer to cameras that are not attached to the robot. They observe the robot itself in its surrounding and are normally fixed on one location, but may be able to pan, tilt, and zoom. In contrast, onboard cameras are attached to the robot and are therefore moving around as the robot does. An remote operator can see the robot body not or only to a small extent through the image of the onboard camera. Some onboard cameras are attached in such a way, that a part of the robot can be seen, which simplifies the navigation task [171]. Onboard cameras may range from simple webcams to high resolution pan-tilt-zoom (PTZ) cameras.

External cameras are often used for AR support in robot programming. Collett and MacDonald [32] show an AR toolkit for intuitive visualization of sensor data (laser scans, sonar readings, odometry history,...). Here, the real world input comes from a wall-mounted camera or directly through a stereo video-see-through head mounted display. An innovative approach for communication with service robots is shown in [188]. The user has a tablet PC with an attached webcamera, which takes images from the robot and the environment. The display on the tablet PC shows the camera images augmented with virtual objects. The system expresses the robot state by cartoon-like annotations, e.g. a happy face shows it completed its mission successfully.

In [164] an AR-based robot programming prototype for manipulator arms is presented. The operator has a head-mounted display with an integrated camera and a handheld input device which is used to record waypoints or edit an existing program. A usability test emphasized the potential of AR supported programming mechanisms for robot manipulators. Milgram et al. [110] show another approach of AR for efficient communication between human and a manipulator arm. They augment video images from an external camera with virtual information such as pointers, tape measures and landmarks to enhance the spatial awareness of the operator. Experiments with the virtual tape measurement for teleoperation task are shown in [109].

For the previous systems with external cameras the registration problem can often be solved by using optical tracking. For mobile robot teleoperation with MR, especially in unknown environment registration cannot be solved in that way.

Sauer, Eck, and Schmidt [147] use AR for supporting teleoperation of a car-like robot. Sonar measurement and other information is faded in the view of the robot's onboard camera to enhance the spatial awareness of the operator. In [146] an AR interface is used to teleoperate a differential drive robot with a stereo camera image augmented with virtual objects, e.g. obstacles, compass, two-dimensional map data and waypoints. The user test showed the potential of AR systems for teleoperation of mobile robots. In both approaches the local robot reference frame is used for alignment of local data, e.g. distance measurement, and the robot onboard localization (here odometry) for data that has to be aligned globally, e.g. obstacles, waypoints. In [128] a feature-based localization method for robot localization is shown, which is used for aligning the real and virtual scene. The AR system overlays the camera image with planning data, a world model and sensory data and is used for supporting the operator in path planning.

Atherton et al. [8] present a concept based on AV and multiple perspectives for the coordination of a multi-agent team on a Mars mission. The three-dimensional world model is enhanced with camera frames of the robots, information for decision support, and agent icons. The user can choose different perspectives depending on the actual need, e.g. the overhead view supports the operator to get an overview about the whole environment. The Virtual 3D interface (Figure 2.1(c), e.g. [117]) introduced in Section 2.3.2 is also an example for AV in robot control.

2.5.3 Robot User Interface in the Virtuality Continuum

In [145] robot user interfaces from different locations in the virtuality continuum are compared for different tasks. The used interfaces are shown in Figure 2.3 and Table 2.1 shows the resulting advantages and disadvantages as well as requirements and the possibilities for operator support.

As seen in the table the AV graphical user interfaces allows best the support for human-robot team teleoperation as need for this work (e.g. support for multi-entity teleoperation, history, grounding and shared situational awareness).

Table 2.1: Comparison of user interfaces with different locations in the virtuality continuum [145].

		camera only UI	AR GUI	AV GUI	sensor-based VR UI
type	location in continuum [108]	real	real with virtual objects	virtual + real world (camera, sensor data)	virtual + sensor data
type	category [70]	direct	direct, multi-sensor	supervisory and/or direct, multi-sensor	direct, multi-sensor
support for	navigation	good, depending on quality/FoV	even better as with pure video	depends on the quality of sensor data	depends on the quality of sensor data
support for	awareness of status/ diagnosis	little	good	good	good
support for	local situation awareness	good, depending on quality/FoV	even better as with pure video	less good, requires good positioning/ mapping	less good, requires good positioning/ mapping
support for	history functions	no	locally (in the robot's view)	yes	locally (in the robot's view)
support for	multi-entity teleoperation	no	small (information from other robots in the view)	yes	small (information from other robots in the view)
support for	grounding/ shared situation awareness in multi-entity teams	no	no	yes	no
required	bandwidth	high	high	less and adjustable	less and adjustable
required	acceptable delay	small	small	less sensitive	small
required	video frame rate	high	high	less sensitive	-
required	sensor sampling rate	-	high	less	high
required	registration accuracy	-	high	less (integration into model)	high
required	a priory map information	not necessary	helpful, but not necessary	very helpful	helpful
required	mapping capabilities	not necessary	helpful, but not necessary	very helpful	very helpful

Chapter 3

Framework for Human-Robot Teams

The last chapter summarized related work in human-robot interaction, evaluation of human-robot systems, robot user interfaces, human-robot teams, and mixed reality. The literature showed that HRI has made advances in the recent years in the area of user interfaces for single robot teleoperation, adjustable autonomy, or multi-robot respectively human-robot teams. First guidelines for teleoperation interfaces have been developed. Some authors showed how to use techniques from HCI for developing human-robot interfaces. Others show systems that support human-robot teamwork.

Nevertheless, more work is necessary especially for interfacing humans in different roles and robots toward a joint team working on a common goal. This chapter gives a theoretical framework for such human-robot teams. The supervisory control concept is introduced and human-robot teamwork is modeled by applying this concept. The autonomy design of the presented system is shown. Next, situation awareness for humans, robots, and the team is discussed as an important factor for successful teleoperation of a human-robot team. After the theoretical background has been elaborated the implementation of the concept is shown. Finally, the chapter ends with an outline on how team support can further be integrated into the proposed system.

3.1 Sheridan's Supervisory Control

The term supervisory control is used in this work in accordance with the explanation of Thomas Sheridan [158]. In human organizations, supervisors interact with subordinate human staff, whereas in human-machine systems supervisors interact with (semi-)autonomous subsystems (here robots). In both structures the supervisor gives high-level instructions, which are then translated into low-level actions by the human subordinates respectively the intelligent machines. On the other side, they gather, pre-process, and fuse information and transfer it to the supervisor, who can then conclude on the state of the task and initiate further steps.

The following sections summarizes Sheridan's explanations and discusses the implementation of the supervisory control paradigm for human-robot teams.

3.1.1 Principle

Sheridan describes supervisory control as located between manual and fully autonomous control. In the traditional *manual control*, where no computerized automatic control is used, sensor information is directly shown on the display of the operator, who gives input via the controller directly on the actuators of the system. In computer-aided manual control the sensing and/or the control information exchange is supported by computer programs. This can already be seen as some kind of supervisory control applying a less strict definition. Nevertheless, in these systems all decisions are fully done by the human operator, which also means without him/her the system stops the task performance completely. A robot navigated by a human operator via direct joystick control is such a manual controlled system. In principle, there are no autonomous functions needed in this system, i.e. the joystick command represents input for each motor that drives the vehicle and raw sensor data is displayed, e.g. camera images. Nevertheless, most systems utilize some amount of autonomy. Supervisory control, in the less strict sense, is used to decrease the operator load and increase the navigation performance. For example, sensor data is pre-processed, such that the operator sees only relevant data and joystick commands represent motion commands of the vehicle rather than raw input to the actuators. However, all control loops in the system are closed over the human operator.

On the other side of the spectrum *fully automatic control* is located. The human operator can only start or stop the system and observe the task performance of the system. Actuators are controlled fully autonomous based on gathered sensor information. A fully autonomous robot would represent such a system. In current mobile robotics such control is rare, since for complex tasks human operators are still required. Often full autonomy can be guaranteed only for a short period of time. Nevertheless, there are already some examples such as mobile vacuum cleaners, which have a very restricted set of tasks and can perform their designated tasks without any human intervention.

Supervisory control in the strict sense is located in between manual and fully automatic control. A certain amount of control is executed through loops that are not closed via the human operator, but via autonomous functionalities. The extent of automatic control can vary. In the telerobot example, the robot could use autonomous obstacle avoidance based on range data to overwrite human commands in order to protect itself. In systems with a higher amount of autonomous functions the human operator gives waypoint coordinates and the robot plans and executes movements to follow these points. The autonomous functions are not restricted to the sensor/actuator side, also the display/control side can be supported by autonomy, e.g. preprocessing, integration and interpretation of displayed sensor data or decision support systems for planning of tasks.

Obviously, the borders between the control modes are indistinct and it might not be obvious to the human operator, which control mode is actually used. Related terms are *traded control*, where the human takes over direct control from time to time, and *shared control*, where the human serves as supervisor for some control variables and uses manual control for others.

3.1.2 Supervisor's Tasks

Sheridan describes five generic functions for the human supervisor, which can be modeled as three nested loops. Figure 3.1 shows this and the functions applied to supervisory control of a telerobot.

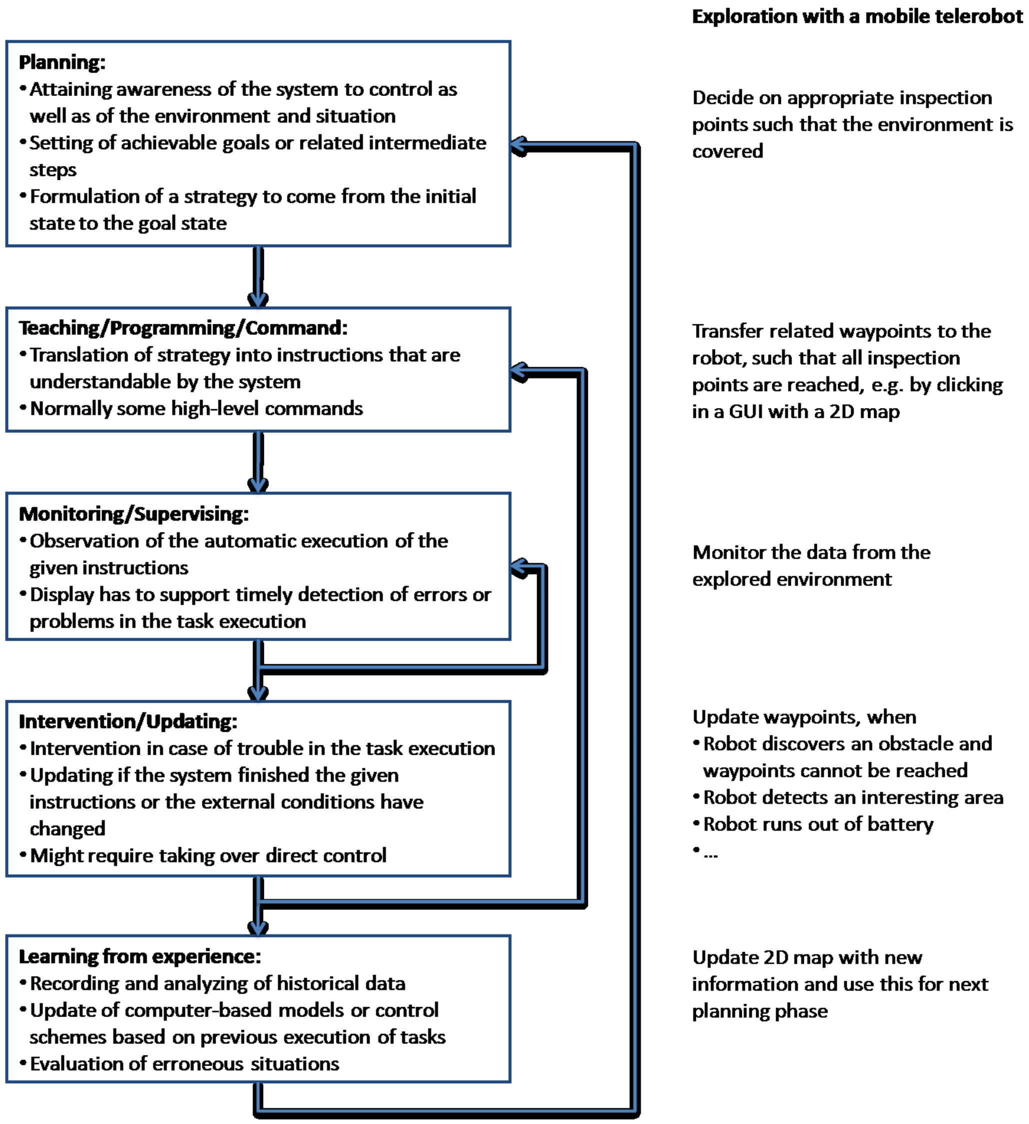


Figure 3.1: Five generic supervisory functions adopted from Sheridan [158] and exploration with a telerobot as example.

Figure 3.1 shows Sheridan's model for supervisory control applied to remote control of a singe robot. Actually, most remote controlled robot use some kind of supervisory control scheme. The following section shows the application of supervisory control for modeling human-robot teams. Here, the human-robot team is rather seen from a system perspective. Mechanisms that play a role in human team work are not considered in this first step. The goal is to establish a robust infrastructure, which allows cooperation in the team.

3.2 System Perspective on Human-Robot Teams

Compared to the above explained single robot teleoperation, supervisory control of multiple entities is much more complicated to model. The entities work indeed in the same workspace, but not necessarily at the same location and on the same task. This will have a great impact on the attention sharing of the operator and therefore the awareness.

Additionally, if heterogeneous entities are teleoperated they vary in equipment and abilities. In multi-robot teams differences appear in sensor or actuator equipment or in locomotion abilities. Differences occur also on the level of designated tasks or the intelligent behaviors. Especially, if a team of humans and robots is teleoperated the supervisor is interacting with two completely different entities. The robots are restricted in their own decisions and need more commands, whereas the human team members probably need protection in some situations, e.g. in search and rescue teams. Obviously, human team members cannot be commanded in the same way as robots.

Future human-robot teams will have to face strongly dynamic missions and unknown tasks, where little information about the upcoming challenges is known beforehand. Planning and execution have to be quickly adapted to new situations. Additionally, in safety critical missions, e.g. fire fighting, late detection of errors or problems will have serious impact on human life.

3.2.1 Supervisory Concept for Human-Robot Teams

In the PeLoTe-project, e.g. [38], the teleoperated human-robot team was modeled based on [158] according to Figure 3.2.

In Figure 3.3 this model is taken up. The loops with and without human intervention are emphasized here. Dashed lines show the interaction between team members in the environment. For example, a human team member can take over the control over a robot team member. This can be done by direct control or some kind of supervisory control. The degree of autonomy on the robot side will be normally lower than under control of the supervisor. Figure 3.3 shows also the different levels of autonomous functions that are included for each team member including the supervisor.

On the robot side autonomous behaviors are e.g. obstacle avoidance or navigating along waypoints. The human team member makes of course own decisions, but can be supported by an assistance system. Both will shape the behavior of the team member. He/she percepts information directly from the task environment or other team members as well as from the display of the assistance system, which

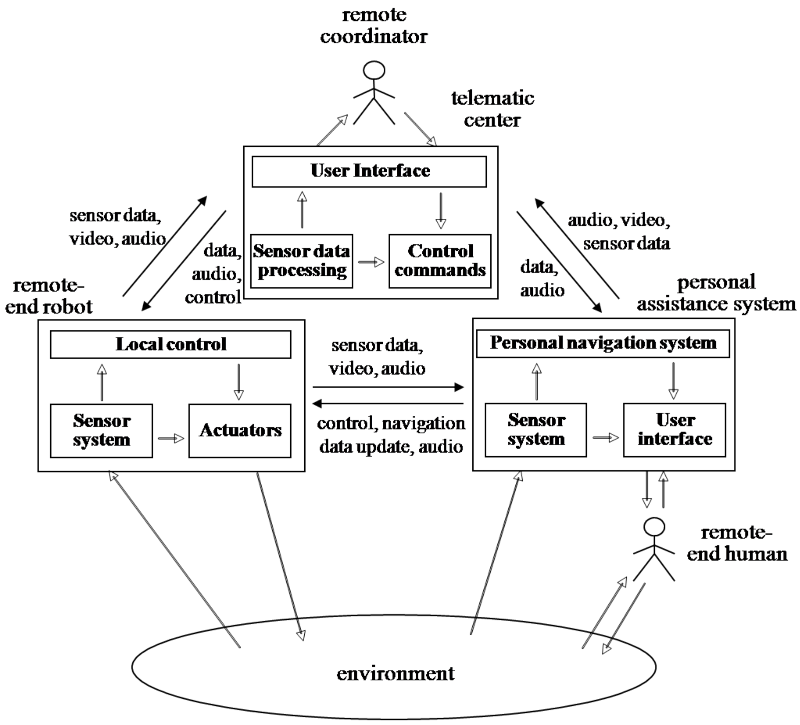


Figure 3.2: Supervisory control architecture proposed in the PeLoTe-project, e.g. [38].

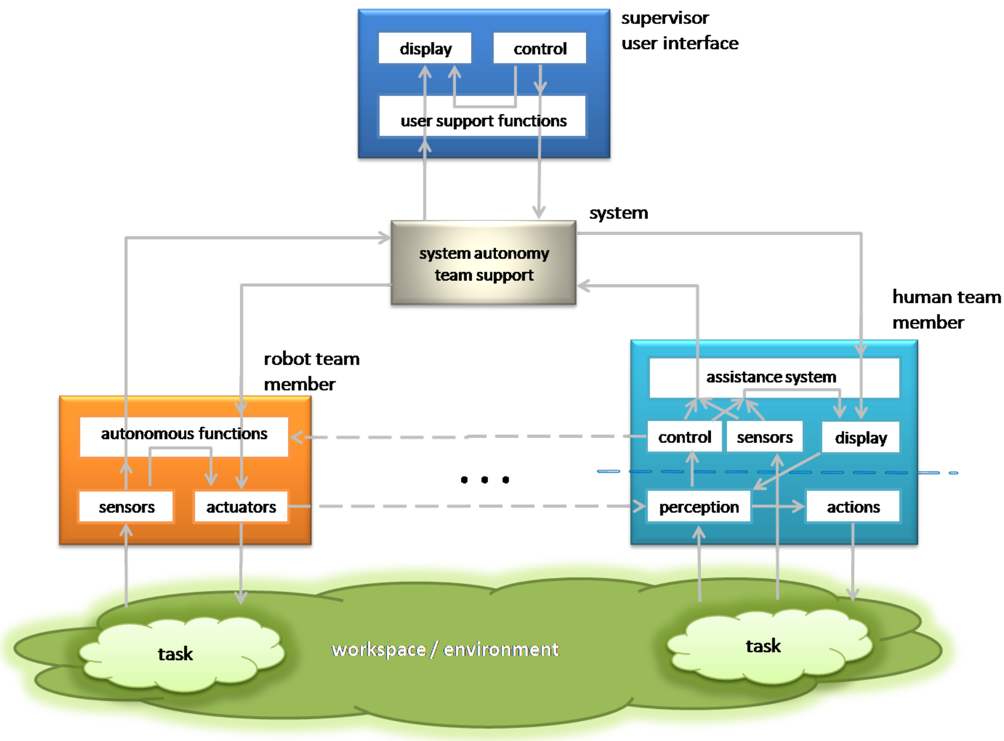


Figure 3.3: Expanded supervisory control architecture for human-robot teams.

visualizes information from own sensors or team knowledge. The supervisor block includes not only display and control unit, but also the execution of autonomous user support functions.

The system autonomy/team support block symbolizes all functions that integrate, filter, and distribute information and commands between the team members.

3.2.2 Supervisory Functions Extended to Teams

In Figure 3.1 the principle functions for a supervisor have been shown. These five functions can be used also to describe the supervisory control of multi-entity teams. Nevertheless, even though the functions describe the tasks for the supervisor well, the process cannot be modeled time-sequential anymore. The supervisor has to share his/her attention between different functions.

An approach for modeling multi-entity teams with supervisor would be to model each entity separately as in Figure 3.1. The other entities come into play as secondary tasks. The drawback of this model is, that it does not incorporate the team idea. It rather applies to a supervisor who controls separate processes which have nothing or little to do with each other.

Figure 3.4 shows an alternative model, which takes into account all team members and the interconnection between functions. The planning step is normally done taking into account all team members. After the strategy for the whole team was decided, tasks are allocated to different team members according to their abilities, purpose, position, status and situation. This planning step can be supported by automated assistance functions. For example, if the team should explore an area completely, e.g. in a search and rescue mission, a coverage planning could calculate a path for each entity. The human supervisor reviews the path distribution and makes necessary adoptions, e.g. if the coverage planning did not take into account certain external conditions (e.g. give a shorter path to the robot with low battery status).

Once the overall plan is clear, the supervisor distributes the tasks to the related entity. If she/he is supported by autonomous methods, this may be send in parallel to all entities. When the instructions are given manually, this function will be done time-sequentially.

As soon as an entity starts with the automatic execution the supervisor has to share the attention between the related entities. The supervisor monitors the execution of each entity, such that problems or errors can be found quickly. In these cases new instructions are given to the related entities. Errors or any other event that have occurred with team member i might require updating team member j (dashed lines in Figure 3.4).

Moreover, the supervisor has to monitor the overall performance of the team as well as external conditions, such as the environment. Should the situation arise that the team performance is not sufficient, the supervisor might need to intervene or update the tasks for certain team members or even re-start the planning with the gained experience. This will happen regularly in highly dynamic situations or

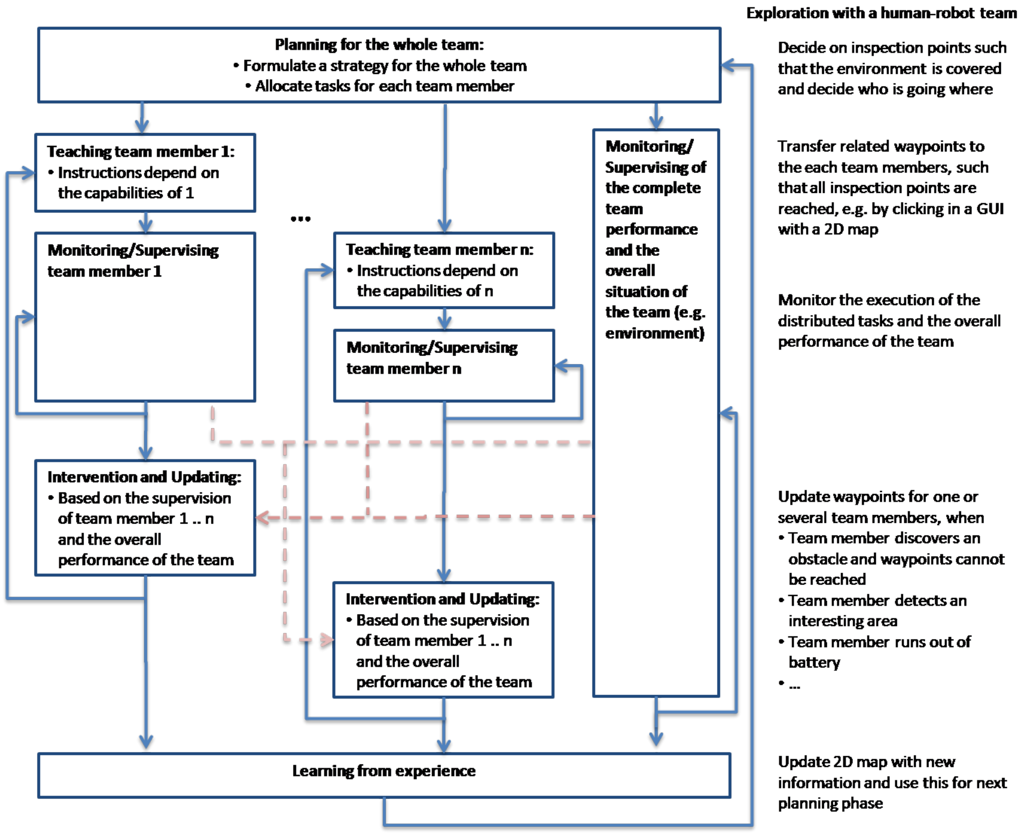


Figure 3.4: The five generic supervisory functions extended to the multi-entity teams. Dashed lines show the interconnection between team members. On the right side the example of exploration is shown.

if a priory information is doubtful. The more information is missing at the mission start, the more often re-planning has to be done.

The proposed model of Figure 3.4 does not make a difference between robot and human team members. This can be done since it does not include how the supervisor interacts with each team member. Instruction, supervision and intervention are top-level functions, which in detail might look different for each team member.

Interaction between team members is not formally modeled here, as only the control from the supervisor side is considered. Nevertheless, from the viewpoint of the supervisor team members that interact, i.e. cooperate locally, can be joined to sub-teams, which again receive instruction and are monitored until the supervisor intervenes (hierarchical organization).

3.3 Autonomy Design

Fully autonomous robots are not yet feasible in the proposed application domains (e.g. due to dynamic nature of the workspace and tasks), and very often they are

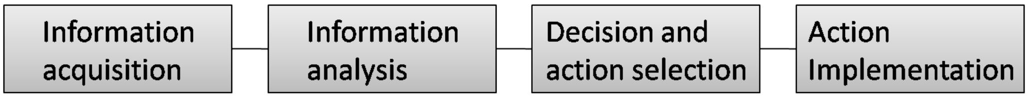


Figure 3.5: Four basics classes of functions where autonomous functions can be applied in a system [131].

not desired either. Leaving monitoring as the only task for the human may result in various problems [57], which are partly described in Section 3.3.2. On the other side, manual control cannot be used for operating a multi-robot system reasonably. Moreover, an appropriate autonomy design helps to overcome challenges of real-world environments (rough terrain, light conditions etc.) and to deal with possible bottlenecks of teleoperation (limited quality of presented sensor data, restricted frame rate, update rate, and field-of-view, lag, delay, or reduces telepresence due to missing information, e.g. sound). Therefore, the supervisory control concept is the most promising approach for the proposed system. Within this concept, the levels of automation/autonomy (LOA) have to be carefully designed. Several taxonomies for LOA can be found in the literature [57, 131, 136].

One specialty of the proposed teleoperated human-robot team is that it includes interaction on different stages: (a) human-robot, (b) human-system, and (c) human-human interaction, which may be mediated by the system. Whether the human-robot interaction resembles rather the human-system interaction or the human-human interaction is an open question and influences the role of the robot in the team (tool versus team partner). Possibly, it is not even fully comparable to any. Many factors can influence this question, e.g. robot appearance, intelligence, and behavior, as well as trust in the robot's autonomy and the way of communication. In the proposed framework, which shall provide a transition step from robot as tool towards human-robot teaming, the human-robot interaction depends mainly on the level of autonomy. For example, if the robot is autonomously following waypoints and detects objects on its own, the interaction is more similar to human-human interaction, as if the robot is directly teleoperated (human-system interaction).

3.3.1 Autonomy Levels

The framework of Parasuraman, Sheridan, and Wickens [131] identifies four generic functions, which are performed consecutively when proceeding with a task (cf. Figure 3.5). Each of the four functions can be implemented with a different level of autonomy. Taking into account the complexity of a human-robot system, the four are further divided into subtasks [111], which again can have different levels of autonomy.

The classification of Figure 3.6 can only be qualitative and does not give an absolute value for autonomy in the proposed framework. High LOA indicates that the subtask is performed autonomously, but the human is informed about the result (which might be filtered, selected, and sorted). Based on this information the human can overwrite the result or assist in accomplish the subtask. Subtasks with intermediate LOA can neither be done autonomously or by the human alone.

Successful performance is dependent on a suitable combination of human actions and autonomous functions. In low LOA subtasks the human has the main responsibility and utilizes the autonomous functions. In manual operation human does the task without any autonomous support.

It it important to note that there are several subtasks for which it is not yet clear how much autonomy is usable or whether the proposed level is the most appropriate one. Finally, not for all subtasks a fixed level is suitable. Often the level needs to be changed based on a human command (e.g. switching to direct teleoperation for further search of an area) or even due to some sensor measurement (e.g. go into a recovery mode, if wheels are blocked) or some result of previous subtasks (e.g. stop and inform supervisor that a certain object was found).

In Figure 3.6 path planning and navigation relates obviously in the first place to the robots, but also the human in the area can use similar functions for navigation support with the assistance system. The supervisor gives waypoints, the path can be calculated and visualized in the assistance system. The same applies for for self-diagnosis and self-protection, which can be supported by warning functions of the assistance system. Related actions are then of course performed manually.

The information acquisition has a relative high LOA, except for information that cannot be gathered easily by sensors. This includes information that is extracted from spoken language or even discussions, where from it is generally difficult to extract information autonomously. Here, the human has better skills as any state-of-the-art system. Information analysis happens with a relative high LOA for the local information and self-diagnosis, such that the entities know what happens around them and their own state. They can protect themselves from harm and are able to performed assigned tasks with little human support. Global information and evaluation of the overall mission state is performed with a low LOA. The analysis of the overall picture is rather difficult to automate, as they include cognitive skills and understanding as well as the ability to predict into the future (relates to Level 3 SA, cf. Section 3.4). Nevertheless, experiments (cf. Chapter 6) indicate that this is an area where autonomy should be increased, e.g. by providing more support functions in the display of information. Self-protection has also in the decision and action selection function a high LOA. Path planning is currently of intermediate LOA, such that it allows the human to have full control over the areas, where the team moves. The overall task planning is mainly done manually. Higher LOA for path or task planning would include autonomous exploration or cooperative planning approaches (as e.g. implemented in the PeLoTe project [98] and used for one of the user studies, cf. Section 6.1). Navigation is normally performed with a high LOA, where human can only start or stop the movement. For certain cases, human can take over manual control and directly teleoperate the robot e.g. by joystick or keyboard. The update of the environment model can happen manual, with intermediate LOA, or high LOA depending on the kind of update. The experiment in Section 6.3 analysis different LOA for environment model update.

The four functions are implemented in each of the four blocks of Figure 3.3 (supervisor user interface, robot, system, and assistance system of the human team

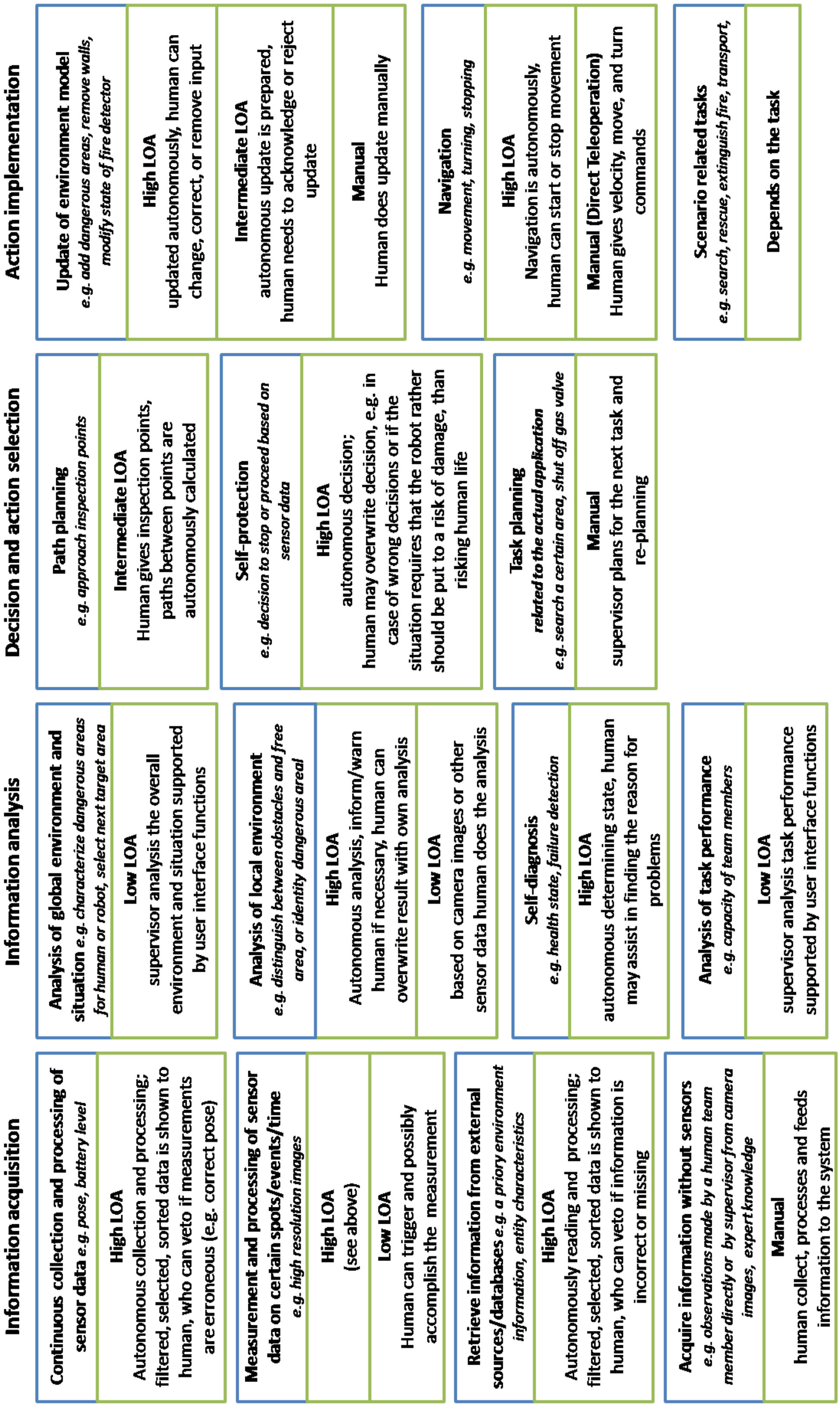


Figure 3.6: Autonomy design for subtask incorporated in the framework of [131]

member), but have different autonomy levels in each. With the autonomy design one has to decide which level is needed in each block for efficient application of the whole system. The design also influences the perception of the robot as a team member, i.e. higher autonomy on the robot side increases the perception of the robot as a self-determined entity in a team.

Other reasons for implementing functions, such as information acquisition and analysis, rather on the entity side, are of technical nature. Transmitting processed, filtered and selected data instead of raw data reduces demands on the communication bandwidths. The implementation of the full process from acquisition to action implementation on-board the robot helps to overcome problems as communication failure and delays. On the other side, global information, which often concerns data from different entities is evaluated by the system autonomy. The functions for supporting the display of information and the control inputs are mainly implemented directly in the supervisor user interface or in the assistance system. If the support functions are generally used (for both supervisor and human team member) they might also be implemented in the system. The implementation details are given later in this chapter (cf. Section 3.5) and for the user interface support functions in Chapter 5.

3.3.2 Problems with Autonomous Systems

Despite the obvious advantages that automated systems have, a number of problems may occur which might decrease the overall performance. These have to be taken into account in the design (i.e. choosing the right LOA for each subtask) and they have to be further considered in the user interface design. Common problems in human interaction with automated systems, which are also relevant for the human-robot system presented, include:

- Poor feedback from the system states [159].
  Clear indication of the robot's mode is mandatory (e.g. teleoperation vs. following waypoints). Otherwise, it is difficult for a human user to understand certain actions. Moreover, certain actions have to be explained, e.g. a robot stopping in front of an obstacle has to properly communicate this to the human to prevent further wrong input commands.

- Misunderstanding or lack of understanding of automation/autonomy [159].
  The humans in the system have to understand the limits and abilities of the autonomous functions, especially the autonomy of the robots. A descent understanding of the robot's autonomous functions helps to comprehend why a robot is doing what. This might be achieved with a high level of training, but can also be supported by explaining certain behaviors as well as the abilities in the user interface.

- Overreliance on the autonomous functions [159, 131].
  Reliability is very important for an autonomous function, such that the human can trust the system and will utilize it to its maximum. Nevertheless, overreliance ("complacence") may have serious consequences. For highly reliable systems, operator may pay less attention and potentially miss one of

the seldom failures of the automated system. The possibility that this effect appears is higher, if the operator has different tasks to perform [131] as it applies for a supervisor, who has to divide its attention to different team members and their situation.
Assuming the localization system of a robot had worked well in the past and robot has performed without the supervisors attention for a while, this may lead to overtrust at the supervisor side. If the localization system shows strong inaccuracy, e.g. the determined position is in a different room as the robot actually is, the human may not recognize that. In worst case, the supervisor will register other information according to wrong positioning and adjusts further planning for other team members based on this information (e.g. may result in sending a human team member into a dangerous situation). This calls on the one side for a highly reliable system, but as often correctness of sensor measurement cannot be fully guaranteed, the supervisor has to be warned about potential failures in information acquisition and analysis.

- Balanced mental workload [131].
  In the best case autonomous functionality should decrease the mental workload, but in no case it should increase mental workload. An increase can happen if the functionality is hard to initiate or requires entering a high number of data manually. Again, the user interface needs to be designed carefully to simplify the interaction and decrease mental workload.

- Reduced situation awareness [54, 131].
  This problem is particular important for the proposed system and user interface and is therefore discussed in the next section in more detail.

An outline of problems especially in robotics with varying levels of autonomy is given in Section 2.4.2.

3.4 Situation Awareness

Building and maintaining situation awareness (SA) can be seen as one of the most important issues in teleoperation of a single robot as well as of a human-robot team. This chapter gives a more detailed discussion on situation awareness.

Endsley gives in [54] a definition for situation awareness, which is applicable in human-robot teams, where SA is described as

> "... the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future."

According to this definition SA comprises different levels:

- *Perception (Level 1 SA)* - Receiving the important information, i.e. detecting significant cues from the environment.

- *Comprehension (Level 2 SA)* - Integrating different elements of the perceived information and determining the relevance.
- *Projection (Level 3 SA)* - Predicting future states of the environment, i.e. future situation events and changes.

Situation awareness is influenced by individual factors, e.g. expectations, objectives, automaticity, skills, and training as well as by task and system factors, e.g. system capabilities, interface design, stress, workload, complexity, and autonomy. Situation awareness, as the operator's internal model of the environment state, has a major influence on decision making [58]. Several phenomena derogate the effort to achieve a good SA, e.g. attentional tunneling, workload, out-of-the-loop syndrome [60, 15]. These are further discussed and considered in Section 5.2.

In [166] six categories for SA assessment techniques are identified: SA requirements analysis, freeze probe, real-time probe, self-rating, observer-rating, and distributed SA techniques. Their a number of different methods can be found for each category.

3.4.1 Situation Awareness in Robotics

The importance of SA in robotics is e.g. shown by Murphy and Burke [115], who report about the use of rescue robots in responses and field exercises. Other studies support this, e.g Burke and Murphy report in [23] from a field study where two operators teleoperated one robot in an USAR operation. Riley and Endsley present in [139] observations from an USAR operation with a robot and the influence of the interface design on SA. Yanco and Drury show in [184] that operators spent an average of 30% for building SA in an experiment with a remote controlled robot. They also observed that operators had difficulties to maintain SA when the robot moved autonomously.

Yanco and Drury [184] adapt Endsley's definition of SA to HRI as

> "... the perception of the robots' location, surroundings, and status, the comprehension of their meaning and the projection of how the robot will behave in the near future."

A more detailed definition that applies to human-robot teams splits awareness in five parts [47, 157]: human-robot, human-human, robot-human, robot-robot, and humans' overall mission awareness.

Several techniques have been developed to analyze situation awareness respectively compare situation awareness for different human-robot systems. For example, Drury, Keyes, and Yanco [48] developed a technique for analyzing human-robot awareness, which is based on an evaluation of operator comments regarding the robot's ***l**ocation*, ***a**ctivities*, ***s**urroundings*, ***s**tatus*, or ***o**verall mission* (LASSO). Others [155] use an evaluation based on the Situational Awareness Global Assessment Technique (SAGAT), a freeze online-probe technique proposed by Endsley.

Ricks, Nielsen, and Goodrich [138] describe that lack of visual cues and delays, e.g. resulting from limited bandwidth or sensor update time, add to the problem of losing SA while controlling a robot remotely. The absence of visual cues from the environment, e.g. in the camera image, makes it difficult to keep self-orientation and to estimate distances to surrounding objects. They propose ecological displays based on augmented virtuality for overcoming SA issues based on lack of visual cues and delays.

3.4.2 Situation Awareness in Human-Robot Systems

For the present work it is also important to consider SA with regard to teams. Depending on the responsibilities in the team each team member requires certain SA elements, which together form the SA for the team [59]. The overlap in SA elements, i.e. the related subset of information, determines the coordination in the team. Endsley describes in [59] the overall *team SA* as

> "... the degree to which every team member possesses the SA required for his or her responsibility ...".

Whereas *shared SA* is defined in [61] as

> "... the degree to which the team members have the same SA on shared SA requirements ...".

For a successful team it is therefore important

- that each team member has the information available that allows him/her to build the SA required to perform the assigned tasks and achieve the own subgoals as well as

- that the right and correct information is shared between different team members, such that shared SA can be built, which allows to cooperate and work efficiently together.

Burke and Murphy present in [23] a model for forming SA in robot-assisted technical search, where a robot operator and a tether manager teleoperate a robot together in a typical USAR task. The robot operator works in front of an operator control unit, whereas the tether manager is in charge for the tethered connection between control unit and robot. Both work together for navigation as well as for identification of victims from the transmitted video images. In this model each human has a role-specific mental model according to his/her responsibilities and a own mental model of the situation, which are fused by team communication into a shared mental model and SA, which then supports decision making for the next action performed by the robot operator.

Sharing visual spatial information can support the team SA in human-robot teams. Knowledge about the spatial context of other team members, certain objects, or events plays an important role for communication, comprehension, and memorizing of information [137]. The user interface needs to support filtering the data, such that the user only gets relevant information and should enable team members to view the data from the viewpoint of another teammate [8].

Halme explains in [80] that cooperation between operator and robot requires them to have a similar understanding of the work environment (spatial awareness). The concept of *common situation awareness*, also called *common presence*[1] [80], uses a virtual model of the environment for interfacing humans and robots in a cooperative task. The concept strongly emphasis on localization (providing a common frame of reference) as one of the key component for maintaining a common model of the environment, i.e. each entity (human [173] and robot) that works in the environment has a location as well as all objects in the virtual model have [172]. These objects, e.g. walls, task-related objects, images, verbal description, are a priory known or based on updates and modification made by humans or robots. Even so the common model is equal for all entities, the interpretation of the data that entities can retrieve from the model is different.

Situation Awareness of the Robot

If robots shall no longer be considered as pure tools used for performing a limited number of very specific tasks, they have to be seen as entities that also have own SA requirements for performing their assigned tasks and that they contribute as an active source to the shared SA. For example, in the common situation awareness concept [172, 80, 173] robots and humans equally have the possibility to modify the common model and have the same access to the information in the model, even if the output is used differently.

Since the robots are part of the SA problem, SA theory has to be extended and adapted to the certain requirement of robots. Adams proposes in [6] a formalization of SA for unmanned vehicles and human-robot systems. For the SA of an unmanned vehicle SA_V she proposes:

$$SA_V = (Level\ of\ autonomy \times C) \cup Envir \cup X. \tag{3.1}$$

The term $(Level\ of\ autonomy \times C)$ relates to the elements related to a certain level of autonomy (C can include factors as workload, stress, attention, perception, ...), $Envir$ to the environmental characteristics (e.g. weather, terrain, location), and X are other factors such as for example capabilities, task complexity, or communication errors.

Further, for a team of i humans and j robots system level team SA can be represented according to[6] as

$$SA_{ST} = \underbrace{\bigcup_{1}^{i} SA_{H_i}}_{\substack{SA\ of\ the\\ human\ team}} \cup \underbrace{\bigcup_{1}^{j} SA_{V_j}}_{\substack{SA\ of\ the\\ unmanned\\ vehicle\ team}} . \tag{3.2}$$

[1]The concept common presence was first investigated in the EU-project PeLoTe.

On the other side, shared SA in this team is described in [6] by

$$SA_{ST_{shared}} = \underbrace{\bigcap_{1}^{i} SA_{H_i}}_{\substack{\text{shared SA of} \\ \text{the human team}}} \cap \underbrace{\bigcap_{1}^{j} SA_{V_j}}_{\substack{\text{shared SA of} \\ \text{the unmanned} \\ \text{vehicle team}}} . \tag{3.3}$$

Equation 3.2 and Equation 3.3 are rewritten in the following two equations for application in the here considered human-robot system:

$$SA_{ST} = SA_{supervisor} \cup \left(\bigcup_{1}^{i} SA_{teammate_i} \right) \cup \left(\bigcup_{1}^{j} SA_{robot_j} \right) \tag{3.4}$$

$$SA_{ST_{shared}} = SA_{supervisor} \cap \left(\bigcap_{1}^{i} SA_{teammate_i} \right) \cap \left(\bigcap_{1}^{j} SA_{robot_j} \right) . \tag{3.5}$$

The SA demand of each entity might vary depending on the specific assignment, i.e. a fire fighter team has different needs than a human-robot astronaut team. Nevertheless, for the task range discussed in this work there are some basic SA requirements can be formulated (cf. Table 4.2).

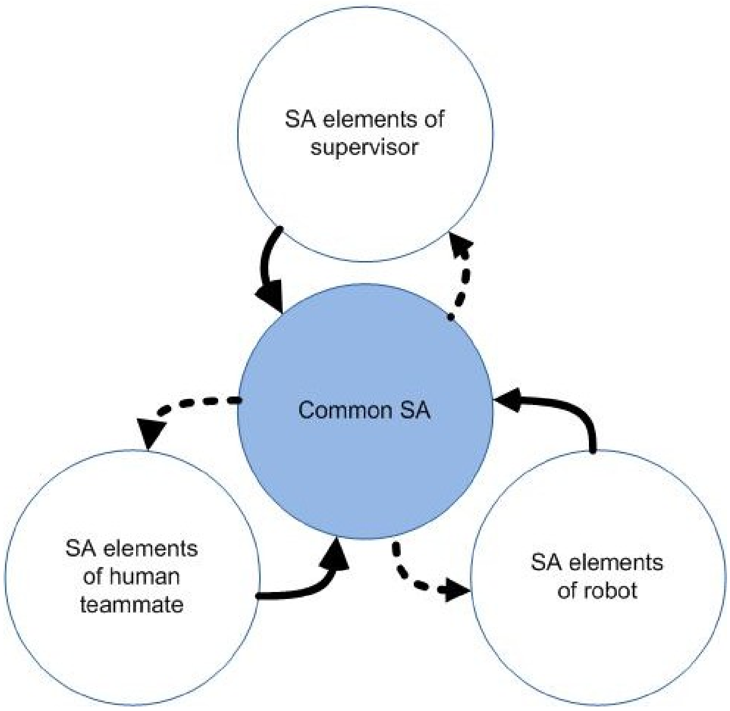


Figure 3.7: Common situation awareness in a system of one supervisor, one human teammate, and one robot.

Maintaining a Common Awareness

Basically, the concept of common SA [80] is closely related to both, team SA and shared SA. Common SA provides a more suitable representation for the SA requirements of the whole human-robot system in terms of mapping the concept to realization of a human-robot team framework. Figure 3.7 displays common SA for a typical human-robot system of this work.

Each team member maintains its own SA, contributes to the common SA, and has filtered access for gaining information from the common SA. The realization of the common SA takes care that all contributed information is in the same frame of references. Access is granted via related interfaces, which ensure that only relevant information arrives at each team member and that the information is understandable for the entity, e.g. related user interfaces for the humans [173].

3.4.3 Designing the Human-Robot Interfaces for Situation Awareness

As a result of the previous discussion on concepts for SA in teams obviously three main points are important for the design of the human-robot interface:

- Finding the SA requirements of each team member as well as analyzing what information has to be shared. This can be done for example with the goal-directed task analysis [56, 5] or end-user and task studies.
  Chapter 4 elaborates on this first point, which requires at least to some extent knowledge of the task. In Table 4.2 a more general list of required information was compiled for the here considered task area (exploration of an unknown, dynamic indoor environment as typical for fire fighting scenarios). A more concrete analysis of information needs was done for the fire fighting scenario based on an end-user requirement analysis (cf. Section 4.4).

- Providing user interfaces that allow the human to perceive and understand the necessary information. For this purpose guidelines have been developed [60, 16].
  Chapter 5 looks on this second point, and describes how different aspects are realized in the presented user interfaces.

- Analyzing the SA of the different team members during task performance.
  Chapter 6 describes the performed user studies. The awareness of different team members was assessed by a memory test, questionnaires, where test participants answered different questions related to their awareness, as well as through observations.
  The quality of the information gained during the task performance, i.e. collected in the map or analyzed in the memory test, gives an indicator for the common situation awareness. A further important indicator particular for the spatial awareness of the team is the quality of the localization of each team member. If the positioning of one team member is too erroneous or not available, they sensor data collected by this member cannot be integrated in the common model or is even integrated wrongly. As a result the common awareness does not benefit from the information collected by this team member or is even decreased.

3.5 Implementation of Concept

According to the previous discussion on supervisory control, autonomy and situation awareness a system has been set up. The basic architecture for supervisory control of the human-robot team was already shown in Figure 3.3. The framework has four major components: system for team support and system autonomy, robot, human assistance system with user interface, and supervisor user interface with related support functions. The two last components are mainly presented in Chapter 5, whereas the first two are explained in the next sections.

3.5.1 System for Team Support and System Autonomy

The functionality of the system goes beyond pure data and command exchange. It is the main component

- for enabling common situation awareness by keeping a common environment model and related (filter) functions to distribute information from the model to the different team members and
- for providing system autonomy functions that relate to global information and decisions.

Having these two points in mind a centralized system is required, whereas a pure data and command exchange could also be realized with a peer-to-peer architecture. A client/server architecture was chosen as a pragmatic approach, where the team support and system autonomy are implemented in the server. Robots, assistance systems of human team members, and user interfaces of supervisors are clients. The current implementation does not consider any security mechanisms for failure of certain components, e.g. breakdown of server, or communication losses. Here, other architecture types, e.g. a web-service based middleware that contains the server as one service, and a communication infrastructure based on ad hoc networks might be a feasible approach [189]. Nevertheless, this is not in the scope of this work. A comparison between client/server and event-based architecture can be found in [36].

Technically, the main functions of the server[2] are:

- Maintaining the common environment model. A layered-based map integrates a priory known information (building map and task related information, e.g. emergency exits) and up-to-date information (map updates and task related data, e.g. found dangerous areas). The human and robot team members of the remote team provide their data to keep the model up-to-date and get the updates they need to fulfill their own task.
- Managing the team. This includes keeping the actual state of each team member, including pose (2D or 3D), measurement from sensors (e.g. range

[2] The client/server concept and related first implementations were developed in the PeLoTe project. The server was later adapted and new clients for other robot types and user interfaces were developed, respectively new functionalities were added.

measurement), and status (health, battery), as well as taking care for authentication and authorization of team members (clients) that request a connection to the server. Moreover, information about the team members is deposited (characteristics, abilities, or technical details).

- Providing system autonomy functions. The functions support autonomy on different stages (information acquisition and analysis, as well as decision and action selection) and on different levels (cf. Section 3.3.1).

- Handling of system and team messages. *System messages* are generated automatically, e.g. all team members receive a message when the environment model was updated, such that they can update their own representation. Another example is, that the robot receives a message when a new path was issued. The content of system messages is normally not visible for the team members. The information is automatically processed, which results either in an action (e.g. start the new path) or in a suitable visualization (e.g. the updated environment model). *Team messages* provide a tool for more explicit and natural communication between the team members, similar to a chat facility. For example, the robot can send a message about an interesting object it has detected. In contrast to system messages, the content of team messages is normally directly visualized.

- Storing important information. Error messages, warning, or different events can be stored as required. Furthermore, the environment model is saved at the end of the mission.

All communication between team members goes over the server, except for video and audio data, as well as direct teleoperation control commands. There, the connection is established directly between the clients, whereas the details to establish the communication are again retrieved from the server.

Communication is currently realized over IP (Internet Protocol) connections (Ethernet or Wireless Local Area Network). Server and the interface to the server on the client side are implemented in Java. Communication with the clients is realized via Java-RMI (Java-Remote Method Invocation). Further implementation details about the server can be found in [36].

3.5.2 Robot Team Member

Software clients, which provide an interface to the server, exist for several different robots. Figure 3.8 shows the robots that can be connected. Connection to robots via the Player/Stage framework [75, 31] is also possible as well as connection to simulated robots of the USARSim environment [178]. Furthermore, an own simulation client was implemented providing robot behavior and typical mission related events (e.g. detection of objects).

The differential drive Pioneer robots are often used as they implicate fewer difficulties for navigation and path planning. The research taken out with the human-robot teams was performed in unstructured indoor environments to keep the general navigation task rather simple, but still realistic, and that experiments

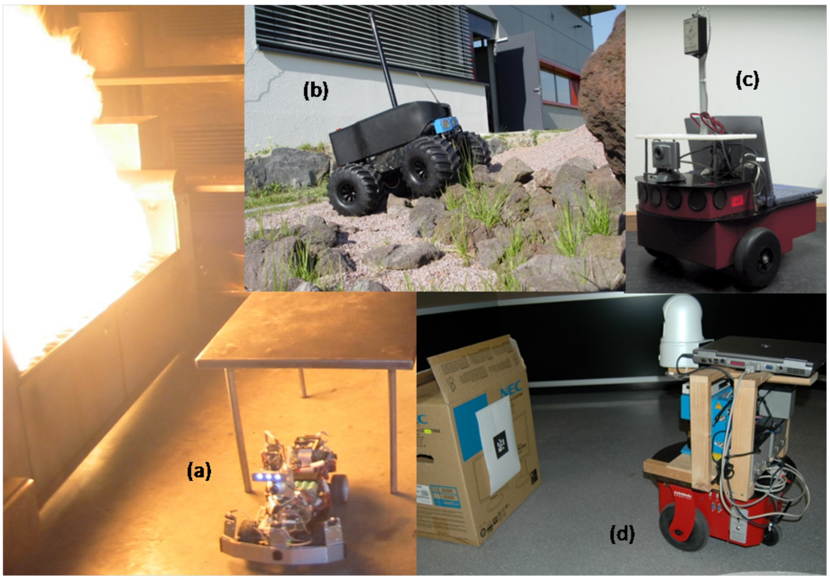


Figure 3.8: Robots for which interface to the server are implemented. (a) Ackerman steering vehicle Merlin [141, 151]. (b) Outdoor version of Merlin [72, 52]. (c) Pioneer I. (d) Pioneer II.

are concentrated on HRI issues. The robots are equipped with a localization system, ultrasonic sensors or laser scanner for obstacle avoidance and normally a camera for environment perception. The client software regularly updates the robots' position in the server and creates messages if the robot encounters problems, e.g. the battery is down, the robot got stuck, or detects an obstacle in front.

The robots have some autonomous behaviors, e.g. they can move along given waypoints. They can also detect markers in the environment and moves then towards the marker position. When used together with the waypoints mode, the robot will stop at each point, move around its pan and tilt camera and searches the images for markers. The marker detection system is based on the ARToolkit [92, 175], which has initially been developed for tracking in augmented reality. If a marker has been detected, the robot sends a message with the position and the marker identification. The markers are used to represent different objects in the environment.

3.5.3 User Interfaces

Technically, the user interface for the supervisor as well as the assistance system including the user interface for the human team member, are realized as clients that can connect to the server similar as the robot clients. Both user interfaces provide related support functions, that process information for more efficient visualization. The user interface design, as well as related support functions for visualization, e.g. sorting, highlighting etc. are topic of Chapter 5.

The assistance system may also provide the interface to other equipment that the human carries, e.g. a personal navigation system [142]. This system provides human localization similar to robot localization methods. Moreover, a local map based on range measurements supports the navigation task in low visibility conditions. The location data is needed for registration of new information in the common environment model. If no localization system for the human is available, the registration has to be done manually based on the human's own perception and estimation of his/her position.

3.6 Integration of Team Support

As indicated before, the proposed framework is a first step towards teleoperated human-robot teams. Some elements for team work, as the common environment model, have been identified and included. Nevertheless, further steps into the direction of real team work have to be taken. In this section further improvements for the described system are elaborated.

3.6.1 Autonomy Support for Information Analysis

In the experiments it could be observed that it is very difficult for the supervisor to keep track on all team members as well as on all changes in the environment. As the tasks, for which such systems are foreseen, are rather complex, it is also challenging to fuse the high amount of information available into one situation model and predict how different actions would influence the situation (Level 3 SA).

Information analysis has a great potential for further autonomy. Hereby, it is not necessary to go to a full autonomy, but rather increase the level step by step, such that the human is still aware about the changes in the situation, but is relieved from low level fusion and registration of different data sources. Comprehension of the situation (Level 2 SA) can be further support by autonomous interpretation of the data. Again, full autonomy is most probably not desired. As an example, different team members measure an increased temperature at certain spots. The system could draw the conclusion that a fire is somewhere close by and inform the supervisor about this observation.

Prediction to future situations can also be supported on different levels. E.g. if a robot detects a dangerous area and the system determines that a human team member is soon entering this area, if he/she follows the planned direction it should warn the human and the supervisor about this potential danger. More, complex predictive functions could simulate the movement of different team members as well as the effect of possible events and estimate the impact of a certain decision.

3.6.2 Improvement of the Robot's SA

The robot's awareness (cf. Equation 4.1) is currently designed, such that the robot is mainly aware about its own state, local environment, and its own task. Comprehension (Level 2 SA) is only little supported and projection (Level 3 SA) not at all. Nevertheless, for being a self-determined team member the robot's should have a deeper understanding of the information and predict future state. This will also support decision and action selection mechanisms especially in exceptional situations, e.g. in case of malfunctions.

For example, if the robot detects that the connection to the rest of the team has broken down it should determine autonomously what to do next. A standard procedure would be to move the path backwards until communication is established again. Anyway, this might not be the best decision, e.g. if the way back would take rather long and the battery level is already down, a more appropriated choice would be to move to a nearer exit. This requires prediction of the effect of each decision possibility. Here, the robot has to decide such that the human safety has higher priority than the own safety. For example, if the robot does a task with low priority and it detects a low battery level, it might move back to charge the battery. In case the robot does a task, which has an impact on human life, e.g. assure a communication link between human team member and supervisor, it has to stay and safe battery power by shutting down all systems that are not necessary for the task.

Currently, the navigation type is chosen by the supervisor. An improvement would include the condition of the supervisor or other factors, e.g. the current reliability of the communication link, into the decision to switch the type autonomously or propose the more appropriate choice to the supervisor. For example, if the robot is on direct teleoperation, but waits for the next command for a long time, it might go into autonomous navigation. Again, this has to be carefully designed, such that not accidentally the decision of the supervisor is overwritten. Moreover, such autonomous decisions and the reasons for taking the choice have to be obvious to the humans in the team. Otherwise, confusion, annoyance, and finally mistrust in the robot's behaviors follows.

3.6.3 Team Organization

Currently, the common situation model includes mainly environmental information as well as pose and state information of the team members. A more extended information base of the team members, that includes data about e.g. capacity, current task performance, or free time, can support the overall organization of the team. That means based on the information the system can propose team members that could take over upcoming tasks.

Further on, together with the position of the team members the organization of the distributed team without the supervisor intervention can be further improved. For example, if a team member needs help with a task a query can be given to the system, which answers with a list of free team members that have the required capability and are close enough. According to this list, the most appropriate team member can be chosen and a help request can be send directly. Such functions make the supervisor of larger teams also feasible.

Chapter 4

Interfacing Humans and Robots

In the last chapter the general framework for the here considered human-robot team structure was elaborated. It was shown that sharing the right situation awareness elements is one of the key factors for successful team work. This chapter starts with a discussion on the team configuration and the challenges that appear for an interface between human and robot. Then, generic information elements that each team partner needs are evaluated. Next, the requirements for the exemplary scenario, fire fighting, are determined and the shared information elements are specified for the example application. The chapter closes with with a discussion of further concepts for the information organization in the environment model.

4.1 Role Allocation in Human-Robot Teams

4.1.1 Interaction Roles

As humans that interact with a robot are no longer restricted to a pure operator role, other types of interaction have to be identified and requirements for the needed interfaces have to elaborated. Scholtz defines in [154] five different types of interactions (interaction roles) with robots:

- In the *supervisor role* the human monitors the complete situation and controls the mission as a whole, i.e. he/she specifies the overall goals and/or modifies the plans accordingly. The supervisor might monitor one robot or, in the case of multiple robots, the robot team as well as each single platform and is responsible for adapt the long term plan or larger goal.
- In the *teammate role* (also called *peer role*) people have face to face interactions with the robot, i.e. they work co-located and are able to control the robots as long as the commands are consistent with the overall strategy given by the supervisor.

- The *operator role* is taken by the human, when software or models e.g. control parameters or the behavior of the robot needs to be changed or e.g. when direct teleoperation is necessary.
- The *mechanic role* includes changes and adaption in the robot hardware, e.g. adjusting the onboard camera.
- People in the *bystander role* have only restricted possibilities for interaction. Nevertheless, this interactions are most difficult to design, predict and evaluate [156]. Bystanders are normally not familiar with the robot and have only few possibilities to influence the robots actions, e.g. stopping the robot's movement by stepping in front of the robot. However, bystanders need to understand the robot's behavior to some extent, e.g. he/she needs to know if the robot really stops or if the persons is rather supposed to move out of the robots path. An example for the bystander role is a victim in the rescue application [14].

Obviously, the boarders between these roles are not strict and the allocation is not fixed, i.e. the same person can have several roles at the same time or can switch between roles. Nevertheless, they describe ways of interaction that entail different demands on the user interfaces, the robot design, and the system support.

In general, a robot in real-world human environments needs to support interactions with all possible roles. Nevertheless, this work has an emphasis on the supervisor and the teammate role. The people in both roles can also switch to the operator role. Therefore, two types of user interfaces are needed. One is for the supervisor, who sits outside the workspace and thus has less restrictive hardware requirements (e.g. a standard computer with one or two monitors). The other user interface type is for the teammates, who work co-located with the robot in the workspace and move normally around. Thus, they have to rely on portable devices (e.g. laptop or even smaller devices). In the following the two user interface types are called supervisor user interface and teammate user interface, even if both types support also the operator role. Moreover, the teammate here works in the same area as the robot, but not necessarily in directly line of sight, i.e. interactions might be face to face, but they do not have to.

4.1.2 Team Configuration

Figure 4.1 visualizes the team configuration used in this work. The number of robots and teammates sharing the workspace vary depending on the experiment. However, the system is tailored for small teams, i.e. teleoperating a team of less than ten remote (human and robotic) team members. Even though, the system and user interface are scalable to much larger teams, it is diffcult for a human operator to manage such large teams without additional support, e.g. establishment of a hierarchical structure, navigation in formations and self-contained sub-teams.

In several areas humans still perform much better than robots, e.g. in cognition, integration of earlier experience, planning in dynamic environment and adaptation to new situation. The robots bring in their ability to accomplish dangerous or repetitive tasks with constant performance. Moreover, depending on

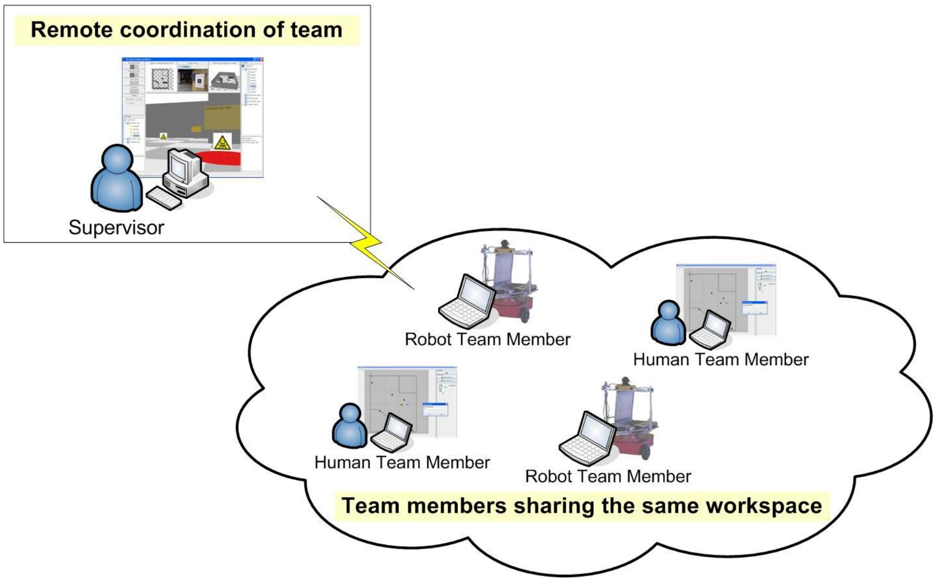


Figure 4.1: Team configuration: supervisor and remote team. The remote team consists of several human team members (teammates) and robot team members.

their equipment they can outperform human in their sensory perception and data processing. Every entity in the team has its own capabilities (strengths, equipment, ...) for contributing toward the common goal. Therefore, each entity has different responsibilities and tasks in the team.

Supervisor

The supervisor is located outside the workspace and uses a standard computer. In addition, he/she may have access to a phone line, internet, external databases, expert knowledge and so on. the main tasks are the coordination of the team according to the current situation and overall status of the mission. This includes distribution of plans and tasks to all team members, commanding the robots, guiding the human teammates and providing support on basis of an external, overall view on the situation. The supervisor will also update manually the common knowledge base according to sensor data that was not added automatically and according to human observations.

The supervisor takes over mainly the supervisory interaction role and only in special case, such as failures, the operator interaction role.

Teammate

Mainly the tasks of the teammates are dependent on the application scenario. They may be supported by a portable system for localization, navigation and assistance [142]. In fire-fighting for instance their main priority would be rescuing victims and extinguishing fire. Moreover, they collect data from the work area either by observation or human-attached sensors. They can take over control of one of the robots, e.g. for detailed search of void space or some other special tasks. Finally, if necessary they might also assist the robot directly, e.g. if the robot got stuck or if the camera needs re-adjustment.

The teammate has obviously the teammate (peer) interaction role, but may take over the supervisor and the operator role. In case of hardware problems with the robot in few cases the teammate may also be in the mechanic role.

Robotic Team Members

The tasks of the robot team members depend also on the application scenario and the robot type. In search and rescue heavy work machines can be used to clear the path from obstacles, transport heavy equipment or to extinguish fires. Small agile robots can be used for exploration, map-building, as sensor carrier or node, and as communication relay. The robots are normally equipped with sensors for localization and environment perception, such as distance sensors and cameras. Depending on the scenario they additional have other equipment, e.g. temperature sensors, thermal cameras, or microphones and loudspeakers. In this work the main tasks of the robots is exploration and search of unknown and unstructured environments.

The main role of the robots in the human-robot team is to extend the human capabilities, e.g. by moving in void or dangerous space or by transporting heavy parts. Nevertheless, in future applications the robots need also to be seen as representatives of the team, which means interaction mechanisms with bystanders become more and more important.

In the present system the main responsibility for overall mission planning lies with the supervisor role. The robots can get commands from both, supervisor and teammate. Nevertheless, the robot needs the option to overwrite certain human decisions, e.g. to protect itself or others in the near surrounding. This extent of autonomy is needed to adapt rapidly to new situations without human intervention, but also in case of communication dropouts or delays. Additionally, a remote operator might lack situation awareness, hence give the wrong command and navigate the robot in a harmful situation.

The team approach, where the humans regard the robots as peers rather than tools, even rises the question if a robot is finally able to give orders to other robots or even human team members, i.e. taking over the supervisor role [102]. Due to the current limitation, e.g. with respect to reliability, accuracy, or adaptability to dynamic situations, such teams are not yet possible and might not be desirable for many situations. Nevertheless, it is already now feasible to use the robots in the team as decision support, where the final authority of the decisions stays with the human.

These decision support capabilities as well as the autonomy functions, that overwrite human commands, imply high demands on the autonomy, system and user interface design. The following section discusses the challenges that appear for user interfaces in such team configurations.

4.2 Interface Challenges

In the proposed scenarios, graphical user interfaces provide the main source for the humans to receive information from the environment and to interact with other team members. Therefore, the interface can be a bottleneck in the system, i.e. it can either hinder or support the task performance.

Related literature and own experiments revealed several challenges for interface design in human-robot teams [42]. The next sections summarize different interface challenges regarding

- information display,
- communication,
- control and navigation,
- awareness of autonomous behaviors, and
- support for coordination and task allocation.

These five points cannot be seen as separated issues, but they are interdependent. Separation into different areas supports a more structured requirement analysis and design. *Information display*, *communication* and the *awareness of autonomous behaviors* are necessary, such that the operator can give input for *control and navigation* based on the *support for coordination and task allocation*, which assists him/her in performing towards the overall goal.

4.2.1 Information Display

It is essential to analyze, which information is relevant for which team member at what time, i.e. determine the elements of the common situation awareness and the related access functions. It has to be decided how data from different sources is pre-processed, fused, and presented (information acquisition). Actual sensor data and information that is known beforehand have to be combined with observations made by the human team members into the common environment and situation model (information analysis). If the supervisor has to share the attention between several entities it is required that he/she can quickly recover the necessary knowledge (position, status, task, local surrounding, capabilities ...) when switching to another entity.

Display of information is maybe the most elaborated challenge for human-robot interfaces and possibly one of the most important issues since without information from the remote scene also other requirements cannot be met. Many evaluations of user interfaces for teleoperation of mobile robots have already been performed and some have resulted in guidelines (cf. Section 2.3.3).

Information display is concerned with presentation of data from different team members. Often this kind of data is updated frequently and the user sees only the most actual data (e.g. video, pose) or data fused with older data (e.g. distance measurement integrated in an occupancy grid). For most data types the robot's onboard software performs related pre-processing, e.g. fusion of several position sensors. The concerned data is normally provided by one team member for the rest of the team, i.e. whoever needs the data should visualize it. The union of all data from the team together with the information known before the mission builds the team knowledge of a certain mission.

However, in a team often a member might want to contact directly a certain other team member or a certain group inside the team. For example, the robot tells the supervisor that it needs help in navigation because its onboard software had determined that it might be stuck. This is summed up in the next section, communication.

4.2.2 Communication

Communication can happen one-way, then it rather has an information character or two-way, i.e. the asker expects a certain reaction from the receiver. Appropriate visualization of this communication data is a crucial performance factor. For some data is might be questionable whether it should be handled by visualization means under the information display or the communication aspect. For example, battery status is continuously measured on the robot and could be update regularly, but in general it is only of interest when it drops under a certain limit. Therefore, the communication seems to be more suitable. Some data might be even updated regularly and critical changes might be communicated additionally, e.g. temperature. This redundancy is particular helpful in supervision of multiple entities as the supervisor will not be able to follow all changes from the remote scene and therefore might miss important data. Table 4.1 gives some examples for visualization as continuous updates (information display) and occasionally updates (communication).

Communication between the human team members is most naturally and fast done by spoken language (audio transmission). Communication between humans and robots seems to be more difficult, as current artificial systems do not provide the ability to discuss a situation or a decision. Nevertheless, the robots (and humans) might send messages e.g. that they found an interesting object or that they reached their goal position. If the supervisor is contacted by several entities at the same time the presentation of these messages has to be very efficient. Incoming messages have to be prioritized and sorted.

Fong, Thorpe, and Baur [66] describe the concept of collaborative control, which is based on an event-driven human-robot dialogue. The robot asks questions to the human when it needs assistance for e.g. cognition or perception, i.e. the human acts as a resource for the robot. Since the robot does not need continuous attention from the operator, collaborative control is also useful for supervision of human-robot teams. Other forms of communication between human and robot are e.g. gestures for direct communication or an approach introduced by Skubic et al. [165], which uses sketches to control a team of robots.

If the communication in the team is further advanced, mechanisms from human-human communication, e.g. common ground theory [29], have to be investigated

Table 4.1: Data Visualization as Communication or Information Display

Location of entities	Information Display	Data that is needed for all further decision with the current value and possibly also the history.
Video	Information Display	
Environment map	Information Display, actual fused with previous data	
Item detected	Communication	Data that requires (immediate) human intervention/reaction.
Known or unknown problem, e.g. robot got stuck	Communication	
Battery	Information Display, when level drops under a certain limit Communication	Data that might be necessary for further decisions, but also requires (immediate) human intervention/reaction if a certain value is reached.
Mode	Information Display, and mode change Communication	
Temperature	Information Display, when certain temperature is reached Communication	
Task performance	Information Display, but Communication when task (path) end is reached	

and transfered to human-robot communication [170, 95]. Burke and Murphy showed in [24] that shared visual presence from the robot's view works as a source for common ground for distributed human teams and supports team performance.

Based on the visualized data the supervisor gives then control commands or navigation aids to the robot and the human team members.

4.2.3 Control and Navigation

Typical input devices for control and navigation are joysticks, gamepads or the keyboard. More advanced methods are based on speech or gesture recognition.

Navigation of the robots can vary from full teleoperation to autonomous movements. When multiple entities are controlled by the same supervisor some autonomy should be provided for the navigation. Nevertheless, in most applications it is necessary that the robots can also be teleoperated, e.g. for moving close to some object or even move the object itself. When a team member teleoperates a robot directly he/she takes over the operator role.

For robots with a rather high level of autonomy, the previous described supervisory control approach is often used (cf. Section 3.1). It allows the user to enter high-level commands for monitoring and diagnosis of the robot. Providing this type of control makes the system capable to work even under low-bandwidth conditions or time delay in the communication link. Autonomy of the robots or the system requires a careful consideration of these features in the user interface design and implies the next interface challenge.

4.2.4 Awareness of Autonomous Behaviors

If the robots are not completely manually controlled, i.e. they can take over control about themselves by certain autonomous behaviors, the human operator has to be properly informed about the action of the robot. Otherwise, frustration and mistrust might result. The user has to fully understand why a robot behaves like it does. Particularly, changes in the level of autonomy are critical. At best, the user interface supports combining the skills and capabilities of humans and robots. Various studies analyze how humans interact with robots of different autonomy levels (cf. Section 2.4.2).

4.2.5 Support for Coordination and Task Allocation

In the presented team configuration, the supervisor is responsible for task allocation and coordination of the team during task performance. Therefore, the interfaces need methods to support the supervisor in understanding the status of the overall mission, the task performance of the group, and the individuals. Furthermore, support for communicating the allocated tasks to the related team member has to be implemented.

4.2.6 Interaction between Human Team Members

The discussed interface challenges are of high significance for the interaction of the supervisor, but also of the teammate, with the robots. However, similar issues appear in the remote interaction between supervisor and human teammate. If they are able to communicate via audio much of the requirements for the user interface are already solved by the direct audio communication. The whole communication part (cf. Section 4.2.2) can be seen as an imitation of human communication. Nevertheless, a graphical user interface can support the information exchange between human team members. Up-to-date information can be shared, whereas otherwise a paper map on both sides may be used.

Simple control and navigation input can be given via audio (e.g. *go 50 meters ahead and then turn to the right and go in room 345*), but more complex instructions might be easier given with a computer-based map. The awareness of autonomous behaviors corresponds to the knowledge about the other team members (their role, behavior, performance, capability, and competence), which is given if the team members do know each other and have already worked together. Supervisors with several remote entities need also some support in coordination and task allocation. Obviously, the division of challenges can also work between the human team members, but they are differentially expressed.

4.3 Information Demand for Team Members

4.3.1 Supervisor and Teammate

Table 4.2 compiles the different SA requirements for the supervisor and for the teammate, which have to be reflected in the user interface. The table results from own user tests and a literature review (cf. Chapter 2). The listed data is general and relates mainly to tasks as exploration and search in an unknown environment by humans and robots. The table can be used as a start point for determining the information demand in all similar applications. Obviously, the next step is to evolve the general requirements towards application-related requirements. This can be done e.g. by a end-user requirement analysis (Section 4.4). In Section 4.5 a concretized form of Table 4.2 is given.

4.3.2 Robot

The situation awareness elements for the robot team members are similar and can be composed as follows:

$$\begin{aligned} SA_{robot} &= \{environment\} \cup \{own\ task\} \cup \{self\} \\ &= \{building\ plan,\ surrounding\} \\ &\cup\ \{teleoperation,\ waypoint\ following,\ model\ update\} \\ &\cup\ \{pose,\ communication\ state,\ battery,\ movement\ blocked\} \end{aligned} \tag{4.1}$$

The environment characteristics are described by the *building plan*, which is retrieved from the common environment model and the *surrounding*, i.e. the local map from range measurements and object data, retrieved e.g. by camera and related image processing.

The robot needs also information about the own task. The two possibilities for navigation, *teleoperation* and *waypoint following*, are initiated by the supervisor. For teleoperation the robot has to know the movements that relate to the direct commands. For waypoint following the robot has to be aware about the points and how to reach the next point. For the environment *model update* the robot has to know which type of update requires which level of autonomy (cf. Section 6.3).

Finally, the robot needs to be aware about the own state. This includes the own *pose* in the same frame of reference as the building plan. Moreover, for self-diagnosis the robot has to know its state (*communication state*, *battery*, and *movement blocked*).

As Equation 4.1 shows, the robot currently has only implemented awareness about itself. For further facilitation of team work also awareness elements about the other team members shall be integrated. For example, the robot might need to known about the position of human teammates to support them in emergency without supervisor intervention. Heterogeneous robots might also support each other when they know about their different abilities and task performance.

Table 4.2: Requirements for user interfaces for supervisor and teammate.

		Supervisor user interface	Teammate user interface
ENVIRONMENT	Overview about the complete environment	A global map/model of the environment is very important, such that the supervisor can execute the main task of monitoring and guiding the team. Information in the map includes structural data, semantic information (emergency exits, gas valves, or any other related to the mission), etc.	Information about the global environment should be present only if it is relevant to the actual task (e.g. structural, path data and gas valve if a certain gas valve should be found) or influences the teammate's situation (fading in dangerous areas close by, which might endanger the human).
ENVIRONMENT	Knowledge about local environment	The representation of the local environment is required if the supervisor interacts with a certain entity (e.g. teleoperating a robot, analyzing a certain behavior or communicating with a human team member).	The teammate needs knowledge about the own local environment and if interaction with a certain other team member is required knowledge about their surrounding.
MISSION / TASKS	Goal and task allocation	The supervisor has to keep in mind the overall goal and is in charge to adapt the overall goal/plan. Therefore, a representation of the allocated specific tasks and support for associating and communicating new tasks to the related entities is needed.	The teammate should know the overall goal, but needs to know basically the own current task and potentially future tasks. If necessary, the teammate has to get access to the task allocation of the robots.
MISSION / TASKS	Work load and progress of each entity	As the supervisor has to manage the resources of the team, she/he has to keep track about the work load and the progress of task execution of each entity. This should be visualized appropriate in the interface.	The teammate should be able to request work load and progress of other team members in case he/she needs support with the own task.
ENTITIES	Pose, state and capabilities of entity	Pose, state and capabilities show the supervisor if a robot is able to perform a certain task. Moreover, the state visualization informs the supervisor if an entity needs help.	Important are the own pose and state (e.g. of the communication link). Data from others should be available on request, e.g. if support from another team member is needed the teammate can check, which entity has the needed capability.
ENTITIES	Comprehension of entity relations	The supervisor has to understand from the interface if two or more entities interact directly, e.g. if a teammate teleoperates a robot.	The teammate needs information about entity relation only if he/she wants to directly cooperate with a robot or another human.
ENTITIES	Comprehension of entity behavior	Understanding the current level of autonomy (e.g. when the robot starts an autonomous behavior to avoid an obstacle or if the supervisor changes the attention to a new entity) is difficult for an operator. The interface has to provide adequate support for understanding the entities' actions and behaviors.	The teammate has to be informed about the behavior of robots near to the own position.

4.4 Requirement Analysis for Exemplary Scenario

Table 4.2 shows the information requirements for the general case of teleoperated human-robot team. However, especially the aspects environment and mission/task require a careful examination of the application. Rescue operations are well-established procedures, which have been analyzed in personal interviews and in a questionnaire survey during the PeLoTe-project. The aims were to understand today's rescue operations, to learn about the needs of rescue workers, especially the information requirements, and to analyze how robots can support in rescue scenarios.

Over 75 questionnaires were send to fire brigades and other rescue organizations. Professional fire brigades, including plant and airport fire departments and fire fighter training academies, represent the largest feedback group with 16 returned questionnaires (10 from Germany, 4 from Czech Republic, and 2 from Finland). As the exemplary scenario is fire fighting, only the results from the fire brigades are analyzed with an emphasize on the requirements for human-robot interaction in the following. The complete evaluation can be found in [37].

4.4.1 Information Demand

Fire fighters use, if available, two-dimensional building maps based on corresponding DIN standard. These maps often integrate the layout of the building with positions of fire detectors, evacuation and action plans, gas storages, and other important knowledge. Sometimes the maps are also available electronically. Plant fire brigades often have access to more detailed information, e.g. storage rooms for dangerous material.

To sum up, the following information can be of interest during a rescue operation depending on the actual mission:

- building layout, including rooms, windows, doors, stairs, etc.,
- lines and valves for electricity, water, or gas,
- purpose of the building or area,
- storage rooms and work areas for dangerous substances, and
- information that is important for rescue and fire fighting, e.g. attack routes, fire alarm system control unit, fire detectors, extinguishing water pipes and hydrants, as well as exits and evacuations routes.

Obviously, certain operations require specific information, as e.g. temperature, atmospheric composition, or other sensor data, e.g. radioactivity. The locations of people that need to be rescued, as well as the position of operational units were also mentioned. This data needs to be updated continuously during the operation. Moreover, spoken information might be useful to be integrated, e.g. from the caller that reports about the emergency case.

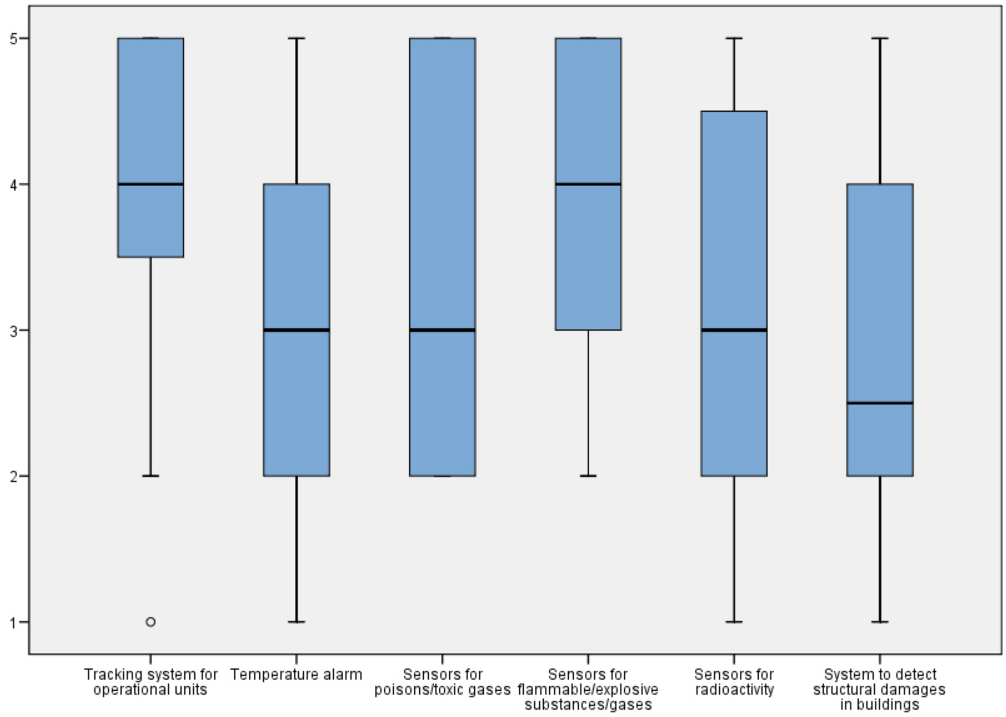


Figure 4.2: Importance of different information during a rescue operation rated by fire brigades (1 - minor ... 5 - critical).

Figure 4.2 shows a rating of the importance of different information. The needs for information vary a lot between the different answers. For example, many answers indicated that the position of the operational units (tracking system) could be of high importance. Nevertheless, other did not find this useful at all. This depends on the background (e.g. city fire department may have different demands than fire fighter in an airport), but also on the personal experiences the responder has made or on the potential specific application the responder was having in mind. Reservations with respect to the functionality of the system or sensor may have further biased the result. That said, as a general result it can be concluded that their is indeed a need for such dynamically updated information. Additionally to the suggested information (cf. Figure 4.2), visualization and supervision of breathing protection was mentioned several times. Moreover, traveled path, emergencies within the rescue team, alarm for escaping, are other examples for visualization that were suggested. The highest rating was reached for the tracking of operational units and sensors for flammable/toxic gases/substances (e.g. oil, gasoline). Further personal discussion as well as written comments on the questionnaires reveal that there is especially a need to get warned from unexpected dangers.

Depending on the operation there is a multitude of data, which shall be provided to the officer in charge, as well as in some degree also to the operational units in the area. This requires user interfaces, that visualizes a high amount of data, but supports clarity of the different information. It was stressed, that existing norms and standards have to be followed. Therefore, the main element of the user interface of this work is a map (or three-dimensional model), which is adapted from the already used maps in Germany, e.g. by using standardized symbols.

It was further required that the visualization of the information shall be dynamically updated and allows a good overview of the situation with the possibility to get more detailed data on demand. It is important that the most critical information is always visible and less important information should be rather omitted than risk information overflow. The classification of information in multiple levels supports the clearness of the visualization, as data that is for the moment not relevant can be faded out and is quickly available if required. One answer explained also that it would be important to have all most critical information in one layer.

Except one, every responder that required pictures from the positions of the operational units (14 out of 16 answers), would save the picture for later use. Most responders liked to have pictures from the positions of the operational units over the complete missions. Others required only pictures from certain significant places or situations, e.g. from victims, from the seat of a fire, condition of access routes and facilities at risk, or always when dangerous activities are performed.

Commanding officer and operational unit in the area require different user interface, which have to adapted according to their information requirements, but are based on similar graphical elements. A head-mounted display was selected most often as the display of choice for the operational unit before an arm-mounted display, a display that is kept in the pocket until it is needed or any other solution. In general, the weight for such displays that are carried into the emergency area is restricted to less than a few hundred gramme. If head-mounted displays are used, they need to be integrated into existing helm concepts.

4.4.2 Rescue Robots

Search, exploration, and detection tasks are typical topics in which a mobile robot can support the rescue operation. Especially, the search for injured persons received a rating of high importance. Exploration of dangerous areas, as well as the detection of dangerous areas and hazardous material was rated as highly important.

Robots shall be used to identify as much critical information as possible, so that based on the additional knowledge together with the prior known information fast decisions can be taken, which expose the fire fighter to as little risk as possible. Therefore, the user interfaces have to include adequate control mechanisms for the robot as well as provide the collected sensor data in an processed and integrated fashion.

Figure 4.3 shows that technical characteristics, as ability to climb stairs, rapid locomotion capability, or efficient working even at high temperatures, get high importance ratings. The ability of transferring data can be seen as the most important feature. While the communication with the control center (supervisor) is therefore also of high interest, interaction with the operational units in the emergency area seems to be less important for the responders. There might be

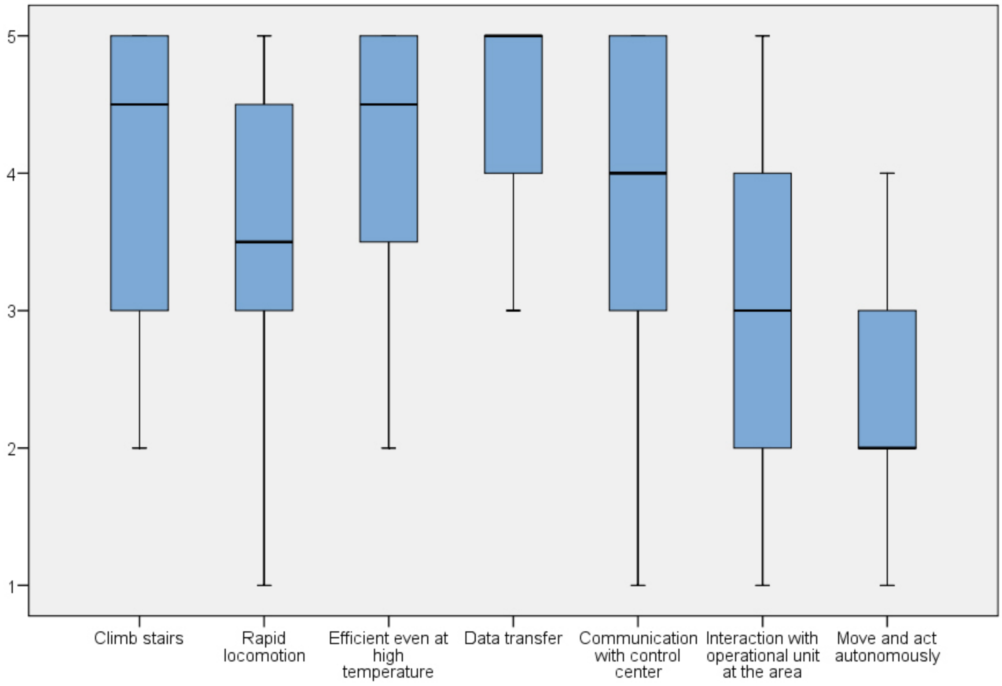


Figure 4.3: Characteristics of a rescue robot (1 - minor ... 5 - critical).

concerns about the additional workload or the constraint possibilities because of the emergency situation and equipment. Moreover, the possible benefit of a robot interacting directly with the rescuers in the area might not be conceivable yet. Moving and acting autonomously received the lowest importance rating. This might also be because the benefits were not visible to the responders. Furthermore, trust in the reliability in such autonomous actions might be low. In the questionnaire, the autonomous function of following a person was given as an example. This example might have been to advanced, i.e. the autonomy level was too high, for such a new system as rescue robots are. It can be assumed that fire fighter rather would like to have more control about the robots when starting to use such systems.

4.4.3 Rescue Map

In the PeLoTe-project a proposal towards a standard for electronic maps was elaborated, the Search and Rescue Map (SRM) [106]. In the described system the SRM was integrated as the information base for map data. It was designed based on the user requirement analysis and an analysis of existing maps and standards. The map is stored in XML (Extensible Markup Language) format and is implemented as a module in the server (cf. Section 3.5.1). It contains previous known information as well as updates that have been added during the operation. The SRM is the main component for the environment model, i.e. the common situation awareness.

Information in the SRM is sorted in layers, e.g. there are layers for building map, electricity infrastructure, fire fighting related objects, rescue related information, or dangerous objects. The layered structure is reflected in the user interface. Layers that contain information, which is currently not of interest, can be faded out and the amount of present data is reduced (cf. Figure 4.4)

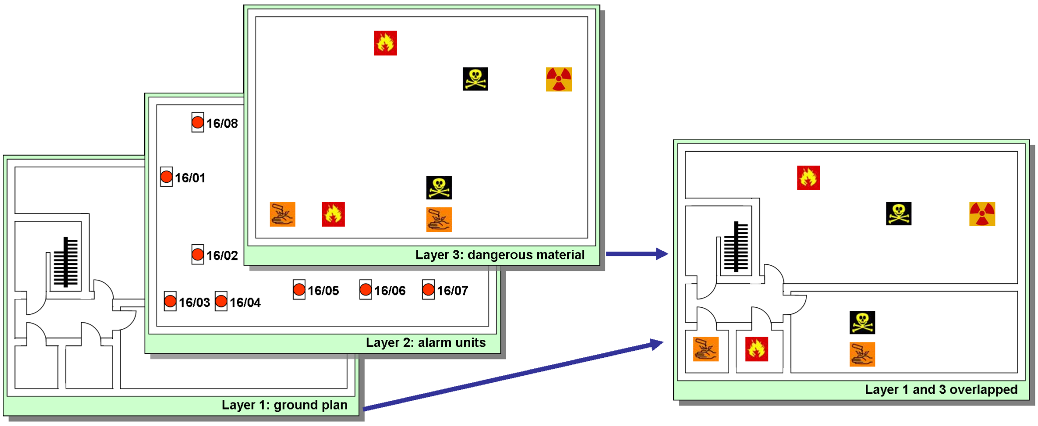


Figure 4.4: Scheme of the SRM: single layers on the left side are combined (here layer 1 and 3) in the user interface, such that only the currently relevant information is shown at once.

Each object in the layers has certain attributes, e.g. position, name, textual description, identification. Depending on the type of object the representation also includes the shape or the name of the used icon. Based on these data, the objects are then re-constructed in the user interfaces and appropriately visualized. The SRM may be updated in the user interface. Objects can be added or deleted. Some objects can also be modified in their state.

Even though the SRM was developed for rescue operations, the layout can be used for other applications as well as. It is particular useful if a high amount of data is available, which should be stored and visualized in a structured way.

4.5 Information Demand as Basis for the User Interface Implementation

Whereas Table 4.2 provides a general elaboration of information requirements in a user interface, Table 4.3 shows the implemented features for this work on basis of the exemplary scenario. As several user interface versions were implemented, this outline shows the latest results in summary.

Table 4.3: Elements of the user interface implementations for supervisor and teammate in the exemplary scenario.

		Supervisor user interface	Teammate user interface
SITUATION	Overview about the complete environment	A environment model (2D or 3D), currently based on the SRM, including building structure and information as proposed in Section 4.4.1. Images from the scene can be stored and visualized in the map for later reference. Short written notes can be placed on the related location in model.	A simplified environment map (2D) is shown.
	Knowledge about local environment	The local environment is mainly perceived by video images from the related team member. Additionally, data from distance sensors might support the understanding of the local surrounding.	The teammate is present in the local surrounding, but if a personal navigation system is available may be supported by range measurement especially in low visibility conditions. For controlling a robot directly, the information demand is similar to the supervisor.
MISSION/TASKS	Goal and task allocation	In order to prevent exploration of the same section several times, the already covered area is marked. The planned paths for all team members are visualized.	The own path is visualized in the user interface.
	Work load and progress of each entity	Additionally to the planned path, the progress in traveling the assigned path is visualized for each robot.	Workload and progress of other team members can be requested from the supervisor if needed.
ENTITIES	Pose, state and capabilities of entity	The pose is shown as an arrow (2D) or a model (3D) in the map. The state (communication state, battery state, movement state) and capabilities are shown for each entity.	Own pose as well as important state characteristics, e.g. communication state, are shown. The position of other entities is also visualized in the map. State and capabilities of other have to be requested from the supervisor.
	Comprehension of entity relations	No special support is implemented yet. In the 2D user interface it is visible if the robot follows the human teammate or if he/she directly teleoperate this robot.	Necessary information has to be requested from supervisor.
	Comprehension of entity behavior	The mode (teleoperation or waypoint following) is shown. If a robot did an autonomous action, e.g. stopped because it found an object, this is represented and already hints for how to proceed are given.	No special interaction methods for direct interaction are implemented on the robot. The necessary information has to be requested from the supervisor.

4.6 Extending the Map Organization

4.6.1 Improved Human-Robot Communication

The sorting of data into different layers, as the SRM proposes, has major advantages in terms of structuring information. Nevertheless, with respect to the human-robot interaction other formats might be more suitable. In the area of service robotics concepts (e.g. [74, 114]) were elaborated that support human-like communication, by giving the robot a knowledge of rooms and corridors as well as of the concept of rooms, e.g. a coffee maker belongs normally to the kitchen. The map of the robot is extended by semantic information, which should makes sense to both human and robot. More natural human-robot communication can be supported, e.g. the command *"drive to pose (x, y, θ)"* can be formulated to the easy understandable, human-like command *"drive to room A121"*, or even more detailed *"drive to the table in room A121"*. The classification of rooms and corridors as well as other semantic information can be hand-coded or a user can teach the robot, but approaches for retrieving this data from grid maps are also investigated in the literature.

The approach to describe an area by topological information, here rooms, corridors, and doors, can also be advantageous for the proposed system and exemplary scenario. The intended application environments include large-scale companies, offices, universities with laboratories, or hospitals. The building structures follow in principle the room concept. The SRM can include such kind of information by a separate layer including all room names, corridor and door knowledge. The problem is that, currently this data has to be included by hand at the correct position. For large-scaled areas this can get tedious and changes during the mission are not easily included. Though, the major problem is that the building structure (in the SRM geometrical primitives that describe walls etc. and objects, as doors and windows) are only related to the labeling by the coordinates. A real connection is not available, i.e. the system can answer questions like *"Is robot A in room A121?"* only after a calculation step. Moreover, queries like *"Show all objects of room A121!"* requires the system to go through all objects and identify the ones in the related room.

The sorting of information with relation to the room or corridor seems advantageous, not only for the human-robot communication, but also for the visualization and the human-human interaction. Nevertheless, the layered structure can be preserved by given each object an attribute with the layer. This is important as the layered structure provides an important tool for sorting data easily. Nevertheless, data can then be faded in for one rooms, whereas for other rooms it might be rather faded out.

Such a room-based concepts allows also easier inclusion of information that does not relate to a specific location, i.e. no exact coordinate can be given. The information can sorted into the rooms and/or corridors to which it relates to. In that way the positioning of a human teammate without localization system is also easier, e.g. he/she can easier inform the supervisor via the audio communication, e.g. *"Leaving A121 and following corridor to the left"*.

4.6.2 Decisions on Visualization

The experiment of Section 6.1 showed that, even though the layered structure was rated as useful also by potential end-users and is vital if a large amount of data is available, the possibility to fade information in or out was rarely used. On the one side, this could be because the data amount in the experiments was rather low, e.g. normally data visualization in the map did not overlap, and information that was not relevant, i.e. candidates for fading out, was rarely included in the experiments. On the other side, the workload in such tasks is already high and a careful selection of information requires some effort. Moreover, removing information from the screen involves the risk to loose awareness of the whole part of data.

Autonomous functions may help to choose the right visualization for certain situations. Obviously, it is of high importance to carefully design this functions. Such autonomous display functions should be used rather sparse, since the user might get annoyed by a continuously changing display. That means, not ever possible action should be automated. Less important or less time-critical decisions on what to visualize should probably be rather left to the user. On the other side, in critical situations, if there is a risk that needed information is currently not visible, the user has to be supported. For some situation it might better to only recommend fading in certain information then to fade in the information autonomously.

Some examples for autonomous functions are:

- There has to be special notification, if updates of data types that are currently invisible, are received.
- If a human teammate enters a room, it has to be ensured that no important information, e.g. dangerous objects, are set to invisible.
- Initiation of a new task can include to fade in all information that is normally needed for the task. E.g. if a injured person was found and should now be rescued, information about the accessibility of exits and save routes shall be shown.

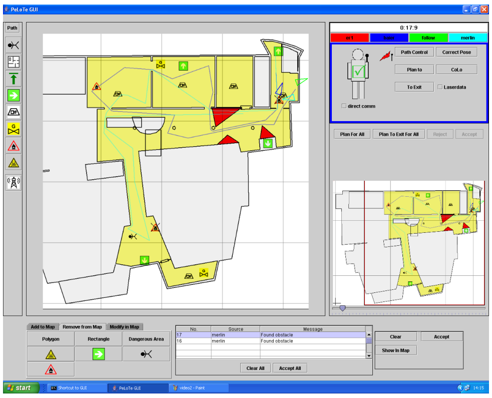
Chapter 5

Designing the User Interfaces

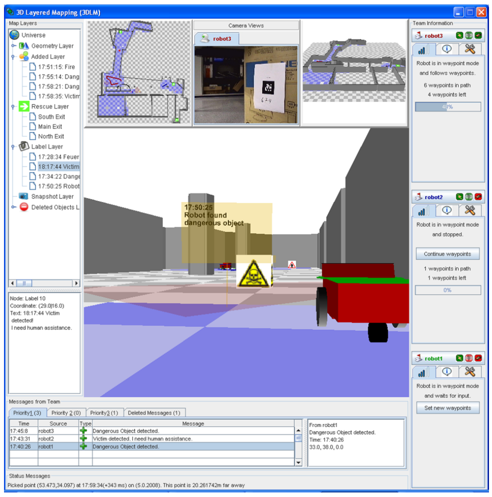
Whereas the last two chapters analyzed the structure and information requirements of human-robot systems, this chapter discusses design issues. However, design aspects can not be seen isolated from the underlying system. The design of user interfaces for human-robot interaction implies other challenges as the design of typical human-computer system. One has to expect much larger response times if commands are sent to a robot, are processed and transformed into actions. These aspects have to be included in the design phase and result in different criteria, which have to be considered in the implementation of the user interfaces.

Moreover, the user interface of each team member has to interface with several other heterogeneous systems, which requires a careful and consistent specification of the data flow. As the other systems might evolve, e.g. robots get new features or the human team members use different equipment, the interfaces between user interface and these systems also have to be easy adaptable. Calibration and registration of the information, which is retrieved from various areas and by different team members, into a common frame is another critical issue. This registration of information as well as the specification of data exchange interfaces is supported by a defined framework as presented in Chapter 3. Nevertheless, the design of the user interface has to consider these aspects. For example, localization systems on-board the robots together with the common model of the situation provide a basis for registration of the information. Anyway, the supervisor user interface has to enable the user to discover errors and uncertainties in the localization and to correct information respectively manually register information.

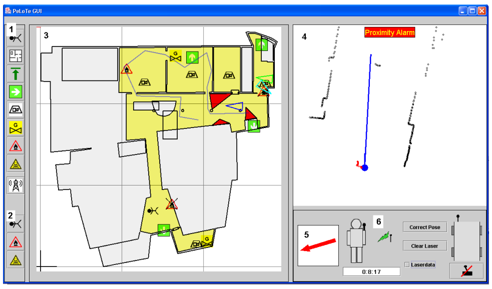
This chapter discusses principles for user interface for human-robot teams and the implemented mixed reality approach is shown. An outlook on the potential of integrating sensor data from innovative 3D sensors is given. The extension of the interface towards distributed 3D scenes for large-scale stereo projection screens is shown.



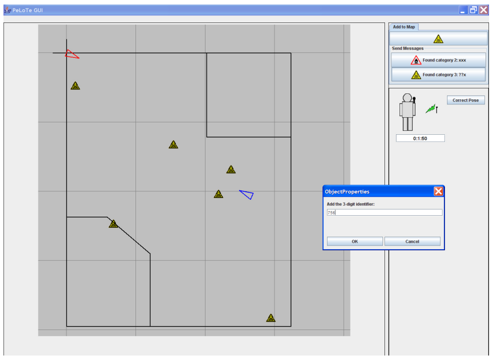
(a) 2D supervisor GUI



(b) 3D supervisor GUI



(c) Teammate GUI with personal navigation system support



(d) Teammate GUI without personal navigation system support

Figure 5.1: The four developed graphical user interfaces.

5.1 Graphical User Interfaces

According to the discussion on required information in the human-robot system the following main elements are needed in the graphical user interfaces:

- global map, including building structure as well as all other information sorted in layers and abilities to manipulate the map (fade in/out layers, add objects, ...),
- space for local information (camera image, laser scans, ...),
- visualization of entities (pose, capabilities, task performance, ...),
- message visualization and related control, and
- input possibilities for sending commands to entities.

Several different interfaces have been developed (cf. Figure 5.1), a 2D and a 3D GUI for the supervisor and a 2D GUI used with and without a personal navigation system for the teammate. Most of the subsequent discussion is based on the newest implemented supervisor interface, the 3D GUI. The design principles apply primarily to the supervisor GUI. For the teammate GUI more specific testing and analysis of the task has to be performed.

The teammate has more direct interaction with the robot, often called peer to peer interaction. There, other forms of interaction have to be considered in the future, e.g. signs, gestures, pointing devices, speech [80]. In this work only user interfaces in the form of graphical representation are considered.

5.2 Designing a Usable Human-Robot Interface

In Section 2.3.3 existing studies and guidelines for human-robot interaction are presented. Additional to this guidelines one can get guidance for the design of the user interfaces from the areas of human-computer interaction or human factors [4, 50, 150, 135].

In the following, design criteria for the human-robot interfaces are derived. For this purpose Shneiderman's eight golden rules for user interface design [161], and ten general usability heuristics developed by Nielsen [123, 124], both documented in Table 5.1, where used as a basis. HRI guidelines, as documented in Section 2.3.3 and other related references are integrated. The principles concern mainly user interface design problems, but often affect the overall system design.

Bad design in information presentation can result in low situation awareness (SA). Bolstad, Costello, and Endsley explain in [15] SA demons that hinder the user in building and maintaining SA. In the following sections some of the phenomenas (highlighted in italics) are discussed with respect to human-robot systems and are taken into account for designing for SA.

Shneiderman's eight golden rules	Nielsen's usability heuristics.
S1 Strive for consistency	N1 Visibility of system status
S2 Enable frequent users to use shortcuts	N2 Match between system and the real world
S3 Offer informative feedback	N3 User control and freedom
S4 Design dialogs to yield closure	N4 Consistency and standards
S5 Offer simple error handling	N5 Error prevention
S6 Permit easy reversal of actions	N6 Recognition rather than recall
S7 Support internal locus of control	N7 Flexibility and efficiency of use
S8 Reduce short-term memory load	N8 Aesthetic and minimalist design
	N9 Help users recognize, diagnose, and recover from errors
	N10 Help and documentation

Table 5.1: Eight golden rules of Shneiderman [161] and the usability heuristics of Nielsen [123, 124]. Numbering is included for easier referencing in the following text.

5.2.1 Provide Feedback from System and Team Members in a Suitable Manner

Respond to User Input

The GUI should give feedback after each user action (S3), whether it has influence only on the own display (e.g. fading out a layer) or other human or robot team members (e.g. sending a new path). A user always has to be aware about the system state, i.e. what the system currently does and how certain user input has been interpreted (N1).

Appropriate feedback shows the user that the command was received, interpreted, and processed. In the design of the 3D supervisor user interface fading out (or in) of layers was for example made visible not only by setting the layer objects to invisible, but also by moving the objects and a colored layer base upwards out of the screen. Hereby, the change receives more attention from the user and is also obvious even if only few objects are in the layer or if no objects are in the current view. Another example is drawing in the map for updating of certain objects or setting a path. After the user has initiated the action, the drawing mode is started and a line is drawn between the last pose and the cursor. This function is especially important if the user gets interrupted, e.g. by an audio comment from the remote scene. The implementation of known concepts, as graying out a button when it is deselected or changing an icon, is self-evident.

Account for Longer Response Times

Other commands in the user interface give indirect feedback. If an object is added to the map in the supervisor GUI, the object data is send to the system, which adds it to the common environment model. The update in the environment model is then again displayed in the GUI. Under normal circumstances response times are here small . Nevertheless, problems with the delay in response appear if commands are send to other team members. A typical example is sending of a new path to the robot, where delays of several seconds might occur between sending the command and receiving feedback that the robot started the movement, e.g. visible only due to changes in the sensor data visualization or camera image.

According to [123] the user feels as if the system reacts instantaneously, if the response time is less than 0.1 seconds. If the response time is between 0.1 and 1 second the user will realize the delay, but no special care has to be taken in the way feedback is given. The user will keep the focus on the certain action for up to 10 seconds. If delays of more than 10 seconds appear users normally want to do other tasks in between.

Therefore, with a response time of several seconds until a movement start becomes visible in the GUI the delay is obvious to the user. This could also be observed in experiments, when the user would wonder if the robot ignores the command and sends it again. Especially if the users have other expectations, e.g. based on computer games, they would get annoyed or frustrated about a longer delay. Moreover, even if the delay is less than 10 seconds it is highly probably that another task or team member will require the user's attention during the waiting time. To ensure the user that the system processed the input, feedback is given in the form of switching the task display (e.g. from waiting for a new path to moving on path). An improved feedback would inform the user also about the intermediate step, that the robot has received the path and now initiates path planning and navigation.

Explain Unusual and Unexpected Response

If no path was found, the user also has to be informed about this. This relates also to rule S7, which requires that the humans in the team shall feel that they are in control of the system. This rule relates strongly to the problem of interacting with autonomy (cf. Section 2.4.2 and Section 3.3). If the robot reacts differently than the user would assume under normal circumstances, the feedback message should contain the reason for the different behavior, such that the user does not get confused, angry, or unsatisfied about the robot or system performance.

5.2.2 Present Sufficient Information about the Situation with a Suitable Design

Present the Right Information According to the Task

In the human-robot system changes are often not a response to certain user actions, but rather related to the development of the situation in the remote environment. Such changes (e.g. a robot stopping without human intervention) typically contribute to the feeling of loosing control over the system or single team members (S7). Related visualization of these changes supports the user in understanding the system or team member behavior. The GUI has to provide enough information that the user can work on his/her tasks in the scope of the overall goal. On the other side useless information has to be filter, such that data overload is prevented [30], (N8), (SA demon *data overload*). Sorting the information in layers (cf. Section 4.4.3) helps to organize and reduce visualized information. As the main task of the supervisor is to organize the team and coordinate the overall mission more information is needed in the supervisor GUI. The teammate, who works on a more specific task and is also more affected by the situation, needs less and rather local than global information.

The information demand for the applications of interest has been elaborated in Table 4.2 and more specifically in Table 4.3. Hereby, it is not only important that the information is presented, but also how the information is processed and visualized. The display of a large amount of unstructured, raw data will rather contribute to confusion and reduce task performance.

Provide Methods for Attention Management

The proposed human-robot system requires the supervisor to share the attention between several tasks and team members. The interface needs to provide support for attention management ([77], last item of the seven principles of [78], cf. Section 2.3.3). Important information, that needs immediate reaction, has to be emphasized in contrast to the rest of the information.

Nevertheless, strong accentuation of certain information elements or also specific team members might lead to *attention tunneling* (SA demon). The supervisor focuses strongly on a certain information element or team member and becomes blind for other emerging problems, i.e. loose the awareness about the overall situation. *Misplaced salience* (SA demon), e.g. using red color for marking a lot of different information, might lead to confusion. Therefore, attention management has to be used only for selected cases and different nuances of highlighting have to be used for data of different importance.

For example, a detected victim in an emergency area receives highest priority, whereas reduced battery power of a robot is less important. Nevertheless, if no other, more pressing problems exist, the low battery power should become obvious for the supervisor. That means, the information should be highlighted more than e.g. the information that a robot has reached the end of the path. Attention management has been implemented by sorting of incoming messages according to different priorities and color coding of critical state changes. For future improvements on how to inform a user about new incoming and important information in complex, information-rich displays and multi-tasking environments, research in notification systems[1] [107] may inform about problems, e.g. distraction, as well as give ideas for visualization techniques, e.g. using icons with simple motions for notifying about new information [13].

Support Switching between Team Members

In the experiments with two robots and one human in the remote scene (cf. Section 6.1), it could be observed that the attention of the supervisor is often focused strongly to the human (attention tunneling in terms of entities). On the one side this is important, as the human rescuers should get priority before the robots, but obviously it might lead the supervisor to oversee important information that the robot has detected, e.g. a victim. Here, careful designed attention management can also support the supervisor.

Concentration on the human team member was probably facilitate by the use of audio communication as well as by the fact that the environment was challenging for the human, i.e. low visibility reduced navigation capabilities and the supervisor was needed for support. The robot in this experiment run rather autonomously and was not affected by the low visibility. On the contrary, in a follow-up experiment (cf. Section 6.2) the supervisor showed the opposite behavior. There, the environment was rather unstructured, but no lighting constraints were added, i.e. the human could easily navigate and perform the task. Therefore, the supervisor concentrated on the robot, which had a lower autonomy level that time. In conclusion, there is evidence that the supervisor concentrates on the entity that needs most intervention and neglects the ones that act more autonomously.

[1] Examples for notification systems are instant messaging systems, email alerts, or news tickers.

As preference of a team member cannot prevented in a realistic scenario, switching to entities that have worked autonomously for a long time has to be supported by the interface, i.e. awareness of the actual situation has to be quickly regained. Externalization of memory supports context acquisition after switching from one team member to the other [129].

The same problem appears if all entities performed autonomously, i.e. the supervisor role is pure observation of the team, for some time. If a team member needs the supervisor, it might then be difficult to understand the actual situation (SA demon *out-of-the-loop syndrome*).

If the supervisor turns to a team member most important information about this member must directly be accessible. First of all this includes the position and the local surrounding (map, 3D model, camera etc.). The 3D GUI provides additionally information about the state and the task performance (following inspection points, waiting for input, etc.). The supervisor can get information about the capabilities, which is important as each team member may have different equipment. It is also possible to see a history for each team member, where all important events are logged with a time stamp. Therefore, the supervisor can reconstruct what has happened to the entity in the time it was performing unobserved, e.g. that the connection was lost for some time or that the robot had problems with the movement.

Provide Support and Allow for Reducing Short-term Memory Use

Providing such kind of information about each team member in the GUI helps to reduce the load for the short-term memory, which is a limited resource (S8). Overloading of the short-term memory might lead to decreased SA (SA demon *requisite memory trap*). Additionally to the information about the team members and the environment information in the model, the 3D GUI provides the possibility to store images from the scene [119] and short notes. Both are added to the 3D environment model and hence carry besides the actual content also the positional information as well as the context from the surrounding.

Integrate Data in a Common World Model

Attention tunneling does not only relate to highlighted information. In the supervisor GUI a lot of different information competes with each other. In order to reduce the information elements much of the information is fused in the environment model. It is the main element in both, the 2D and 3D GUI. Hence, it is placed in the center of the screen and takes most of the space. Other elements are grouped around.

Autonomous functions for subtasks in information acquisition and analysis (cf. Figure 3.3.1) support the integration of data into the world model as well as other data preparation mechanisms (e.g. visualization of state information by color-coded icons). Additionally, the world model can be updated manually, which provides the possibility to integrate data that was not gained by sensors, e.g. observations from the scene, external knowledge, or conclusion drawn from audio conversations.

The environment model contains level 1 and level 2 SA information elements (cf. Section 3.4). Together with the task performance visualization of each team member this allows the supervisor to predict, which team member needs attention next (level 3 SA) [15]. As the same model is used by all team members, even though the form is adapted to the actual need, the SA elements are shared and support common ground in the team communication (cf. Section 4.2.2). It could be observed in the experiment of Section 6.1 that teams, where supervisor and teammate had a similar 2D up-to-date map in the GUI, discussed a lot over the audio channel, but used the 2D map in the GUIs as reference. The maps provided a basis for their discussion on the actual situation and the future strategy and supported the team work. In the control group, which had only a paper-map and an audio channel, communication was more difficult, a lot of time was used to find out where the teammate currently is positioned. Future plans were difficult to agree, as the supervisor did know little about the actual situation and they had no common basis for discussion.

With an up-to-date map differences in the situation understanding can be easier cleared. For example, if the supervisor updates the model according to his/her understanding, which may not be correct, other team members can immediately see this in their map and intervene accordingly. Based on the model, team members can discuss their different views and find an agreed, common awareness. Currently, only human team members will hold such discussions. Robots can be information sources used to confirm a particular view. In the future, robots may actively step in, if their software realizes a difference between common world model and the internal world model of the robot.

Most of the studies in Section 2.3.3 also recommend to fuse data from different sensors. For example, in [157] it is recommended to display integrated information from multiple sensors. Moreover, a frame of reference is suggested to see the position of the robot in relation to the environment information. The 2D map as well as the 3D model put all team members with a localization system in the same frame of reference by visualizing an arrow or a 3D model. Goodrich and Olsen [78] emphasize the need for maintaining past sensor data to relieve short-term memory, as it is given with a common environment model.

Even though a common model that integrates environment information into a common view is most useful, sometimes the raw or filtered data of single sensors might be needed. Then, the user should have the possibility to switch them on for visualization in the GUI. Raw or filtered, information should nevertheless be shown in a suitable context to the rest of the environment information. For example, distance measurements from ultrasonic sensors are normally only used onboard the robot. In direct teleoperation this raw data could also be used for navigation the robot. The distance measurement of the moment can then be overlaid over the 3D model data.

5.2.3 Design Input and Command Methods Carefully

Structure the Sequences of Actions and Allow Cancellation by the User

A sequence of action in the GUI shall have a start point, middle part, and an end point, which closes the sequences with giving related feedback (S4). For example, adding an object into the environment model starts with choosing the pose and type of object. The middle part includes setting the object properties, e.g. name, shape etc. Finally, the object adding ends with closing the dialog and the object turns up in the list of objects and in the map. Giving a new path to the robot starts with choosing the robot for setting way points. In the middle, the points are set one after the other. The sequence is closed by sending the points to the robot, which will then start its movement to the first point. The set of points is also visualized in the map.

Each sequence of actions should provide the user the possibility to abort the chain (N3). For example, the dialog for adding objects has a button for cancel the sequence. When entering way points the user can remove the last point or chancel the whole path.

Use Natural Control Mechanisms

Goodrich and Olsen [78] suggest to let the user rather manipulate presented information directly. In the 3D supervisor GUI most interaction happens directly in the 3D model (main view). A right mouse click on the robot opens a menu with the actions that can be done with this robot (e.g. assign a path, teleoperate directly, or move viewpoint to the robot). A right mouse click on objects will open another menu that is adapted to the object clicked (e.g. delete, or edit object). If the user clicks on free space a menu is opened, where the user can chose to add a certain object or change the viewpoint to the clicked position.

Provide Support for Task Assignment and Map Updates

Previous it was mentioned that switching to an entity needs support in visualizing the important information about each team member. After the supervisor has gained an understand about the entities situation, it has to be decided what the entity should do next. In order to support this decision process, support functions suggest the next action and provide a button that starts this next action or sequence of actions. If the robot stops because it arrived at the end of the path, a notification will be generated that the robot is waiting for a new path and a button that starts entering the way point setting is shown. If the robot stops, because it detected an object, this will be noticed and the button will show "*Continue waypoints*". After pressing the button the robot will go on to follow the original path. Obviously, here the user does not have to follow the suggestion and can assign a complete new path to the robot.

Furthermore, map updating based on incoming message from the team members is also supported. A right mouse click on the message will show a menu with actions that are determined by the content of the message. The user can choose one option and continue or decide for another solution. Such assistance functions support the supervisor in accomplishing his/her complex task. Further autonomous message handling or task assignment could relive the supervisor even more. Nevertheless, highly automated functions for this tasks bear a number of problems (e.g. the previous mentioned out-of-the-loop syndrome). In one of the experiments (cf. Section 6.1) the possibility was given to process all incoming message autonomously at once. When this function was used by the test participants, it could be often observed that the performed map update were not comprehensible anymore (lack of level 2 SA).

Support features, as recommending the next action, provide also some kind of online help in using the system and can make it easier for novice users to start working with the system (N10). A real online help or documentation is not implemented. For time critical tasks, as fire fighting, this may not useful at all. Nevertheless, for other applications, which are less critical it should be considered to implement an online help system as it will support performing the complex task of the supervisor.

5.2.4 Consistency within the System and the Task

Consistency in the system and within the team

The GUI should be consistent in the required sequence of actions, in terminology, and commands (S1, N4). For example, the dialogs for updating the environment map should use the same terminology independent of the type of update that is done. A consistent sequence of actions should be required for different objects.

Task/path assignment from the supervisor GUI should require similar commands for human and robot teammate, if possible. There shall not be a difference in the terminology of presented information, whether the information comes from the robot or from the human teammate. Messages that the human teammate sends by buttons in his/her GUI have the same format as the messages send by the robot. The terminology of the teammate and the supervisor GUI shall be consistent, such that they can easily communicate via the audio channel.

Take into account the Background of the User and the Application

Consistency should be given not only within the system, but also for the application and therefore the background of the user. That means, information on the display should be easy recognizable for a user. The 2D map reminds on a normal paper-map and the 3D model is similar to the real world. Goodrich and Olsen [78] recommend to use natural cues as sketches or labels.

For the exemplary scenario, fire fighting, the maps are based on maps currently in use by fire brigades to ensures that the system speaks the user's language (N2). This includes not only the terminology, but also the use of symbols that are known to potential end-users. Obviously, this GUI elements have to adapted according to the application area. A human-robot system for support of a construction team needs different terminology and map design (icons, colors, ...) than a human-robot system for a fire brigade.

5.2.5 Allow Flexibility

Provide shortcuts and various ways to interact

There is a number of actions that are frequently used, which can be potentially implemented as shortcuts (S2, N7). Selection of a team member, adding or deleting an object, or deleting a message are typical examples that can be supported by using special keys as shortcuts. More advanced shortcuts may provide several commands in one keystroke or mouse click, e.g. sending a robot the path backwards, sending all team members to the next exit, or processing all received messages autonomously. Here, the question has to be answered, how far the input shall be supported by automated functions. For example, the processing of all received messages resulted in confusion as discussed before. This example shows, that shortcuts shall be used only if appropriate and that the support function has to be tested for potential misunderstandings.

The user shall have some freedom in interacting. For example, a robot can be chosen by clicking the 3D model, but also by clicking the entity information panel. Objects can be deleted directly in 3D model or from the layer panel.

Allow Different Viewpoints

Different tasks require the supervisor to take over different viewpoints. For example, setting waypoints is easiest performed by a top-down view. For direct teleoperation the robot perspective is often more suitable. Observation of the complete team is best done by seeing all team members e.g. from above. Often a more detailed view of a particular scene is useful, which requires zooming.

The user interface has to provide access to different view points and allow quickly adapting the viewpoint. The 3D GUI has additional to the main view, which is adaptable, a small top-down view and small view from the side. The main view can be changed by mouse (zooming and moving) or by clicking in small top-down view. Moreover, the user can choose to move over or at the position of a team member. He/she can move through the 3D model as the team members in the real world. The main view can also be attached to a certain team member, such that the view point adapts according to the real position change.

5.2.6 Errors and Recovery, Safety

Prevent errors and take care for safety of team members

The system should prevent serious errors (S5, N5). The proposed system architecture assures that failure of one of the clients (GUIs or robot software) will not lead to a complete system failure. Unexpected changes in the environment or certain actions in the GUIs should not lead any component or the system into an undefined state. The robots protect themselves, e.g. by stopping in front of unexpected obstacles instead of continuing the path. If they are sent to a unreachable point, they do not start their movement.

Help to Recover from Errors

Autonomous functions in the GUI may further support the user by detecting erroneous commands and preparation of error handling mechanisms. Understandable error messages help the user to recognize the error (N9). Wrong commands, e.g. sending the robot to an unreachable point, should be made visible and explained well, so that the supervisor can consider this for future path assignments. At the same time, the error message should already suggest the next step, e.g. repeat the waypoint setting.

Give the Possibility to Undo Actions

Taking an unsuitable decision cannot be prevented completely, but undo mechanisms minimize loosing important information (S6, N3). Deleted objects from the environment model are easily restored in the supervisor GUI. Deleted messages are also moved to the removed list instead of deleting them completely. This helps the user to maintain clarity, since messages and objects that are not used any longer, can be deleted. If they are immediately removed, the user might not do this out of the fear to make a mistake and loose the information.

Other reversal of actions is currently not implemented, but can be done by further automated steps. For example, the team members can log the path and move backwards if required. Nevertheless, these actions are not really undo functions, since they do not mean that the situation can be changed into an earlier state. Therefore, design has to be careful, such that no wrong expectations from the user side will be associated with this functions.

5.3 Mixed Reality Approach

After experience in the first experiment (cf. Section 6.1) was gained, a new user interface was designed for the supervisor. The principle elements are the same, but instead of the map visualization a 3D model was used. In Section 2.5.3 it was shown how different interfaces represented in the virtuality continuum are suitable for human-robot interfaces. For the supervisor of a human-robot team augmented virtuality, where the virtual model is built from real data, is most suitable.

Two main reasons for switching from a 2D map to a 3D model can be mentioned. First, the environment in which the human-robot team works is three-dimensional. If sensor data is integrated (e.g. distance measurement from ultrasonic sensors), the spatial relation between different sensors or a priori data is more intuitive. It is easier for the user to register camera images in the model.

For example, in Figure 5.2 the camera images shows a marker, which in this test simulated a dangerous object. In the 3D view the object is represented by dangerous object symbol. From having the camera and the 3D view next to each other it is easier to see the correlation than it would be between the camera image and a 2D map.

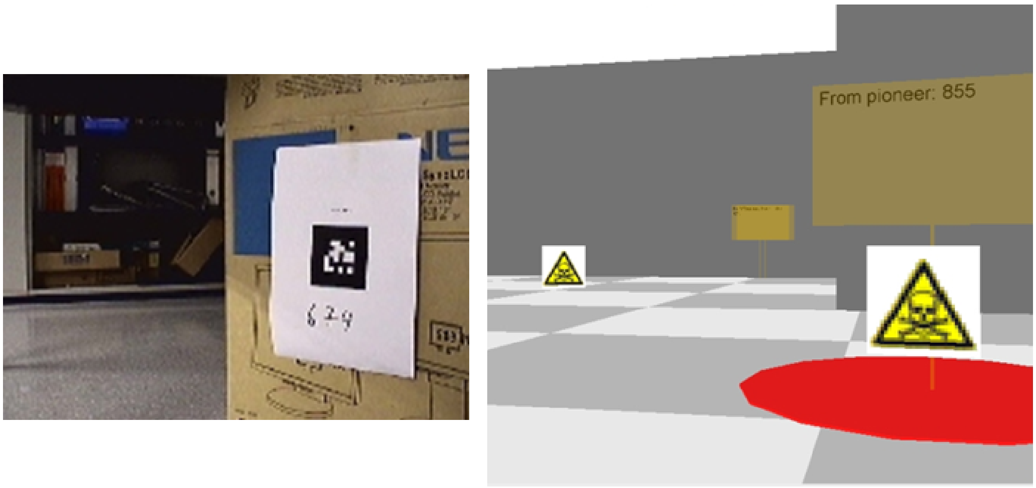


Figure 5.2: Correlation between camera image and 3D model.

The second reason for using 3D instead of 2D is the ongoing development in sensor technology and 3D mapping. Data from laser scanners can be used to construct 3D models of the environment, e.g. [127]. A recent development is the Photonic Mixer Device (PMD) which enables small, light-weight cameras that generate 3D images based on time-of-flight measurements [180]. As it becomes feasible to use such sensors onboard a robot and process data accordingly there is a requirement to have a user interface, which can include 3D sensor data or 3D maps constructed by sensor data.

Shneiderman shows in [162] the potential of enhanced 3D interfaces, but also mentioned problems. Enhanced 3D interfaces have "design features that make the interface even better than reality", for example it is possible to change the color of an object, group components, attach labels, or go back in the time line. Collaboration with team partners in other locations can be supported by enhanced 3D interfaces. Nevertheless, 3D interface are not always the better choice, e.g. if the third dimension only adds to confusion or if task are simply easier performed in 2D. Shneiderman suggests a set of guidelines as a starting point for designing 3D interfaces and enhanced 3D features. In the following significant issues for this work are discussed.

The 3D world is a simple model, i.e. it is not aimed at photorealism. That means effects as shadowing, lighting or extensive textures are not or only if necessary sparsely used.

Navigation through the model is simplified, i.e. it does not use all degrees of freedom that are possible in 3D. For the human-robot interface navigation in the 3D world can only be done as it could be done in the real world, i.e. the user cannot go through walls. Nevertheless, the user can choose a different viewpoint, e.g. the view of a robot, or the top-down view ("teleportation"). An overview, top-down and side, is always present, such that the user does not loose the overall picture of the situation.

History is kept to some extent, i.e. removing objects can be made undone. The user can directly act on the objects (e.g. editing or deleting) and command the robot.

Tools are provided for measure distances in the 3D world. The floor has a checkerboard pattern with each element is one on one meter. This allows the user to quickly estimate e.g. the distance between object and robot. Moreover, the distance between actual viewpoint and clicked object or position is displayed. This allows e.g. more detailed evaluation of the available space when planning the next movement. The last clicked object or robot gets a red, blinking marker such that it can be easily identified from the rest of the model.

5.4 Realization of the User Interfaces

In this section only the latest implementations are explained in more details. The process of user interface development can be comprehended by several publications [36, 150, 45, 149, 41, 145, 44].

5.4.1 Graphical User Interface for Supervisor

Figure 5.3 shows a screen shot of the mixed reality user interface with the environment map from the experiment with three robots. In the following the main GUI elements are presented according to the three categories of Table 4.2 (environment,entities, mission/tasks). Interaction and input methods as well as other support functionalities are introduced.

Situation/Environment

The main view for the environment is the 3D model in the middle. Two other views of this model show an overview of the complete view from above and a side view. The middle field shows camera images from the robots.

The main view can be switch to different viewpoints: above (overview and zoomed), entity-view (movable like a virtual robot) and attached to the real entity (following the entities movement). The user can store viewpoints and move back to these later.

The environment model does not only include information about the building structure, but also task-related information as e.g. exits or dangerous areas. Therefore, the model is not only a view on the environment, but also documents the situation of the team in the remote environment. The information is arranged in layers, which can be seen in a tree-structure on the left side of the GUI.

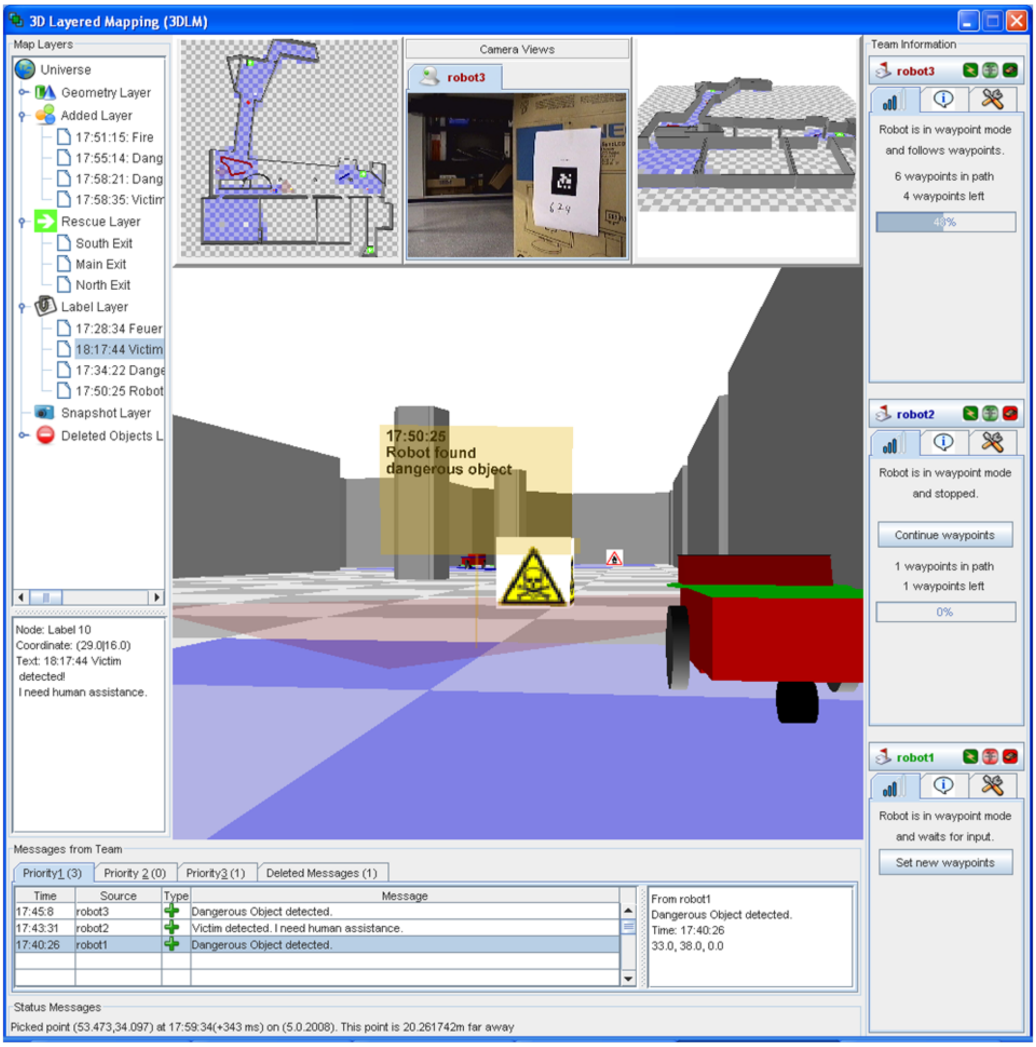


Figure 5.3: Mixed reality user interface for supervisor of human-robot teams in complex tasks.

If new information was gathered in the environment the team members can automatically update the map or send a team message. The team messages are sorted according to their priority into a table (Figure 5.3 below main view). The priorities are set according to the task. Here, messages that concern human life are of highest priority (e.g. detected fires, victims), messages that concern the robots safety get middle priority (e.g. battery nearly empty) and information-only messages have lowest priority (e.g. an invalid waypoint).

The team coordinator can add new objects according to the team messages. He/she can also add 3D-labels, which can keep more information and can be used as a memory functions for later use or for other team members.

Entities/Team Members

The pose of team members (that have a localization system) are shown directly with avatars in the 3D model of the environment as well as the assigned set of waypoints.

The most important information of each team member is shown summarized on the right side of the GUI: mode (waypoint or teleoperated), name, connection status, battery status, and mobility.

More details are given in the tabbed panes take can be opened on demand. Currently there are panes for progress of the assigned waypoint, general information (history of important events) and the team member's capability (sensors, equipment, special function etc.).

Mission/Tasks

In this work, the overall task is exploration and mapping of a partly known environment, i.e. the main task of each team member is following a set of waypoints autonomously and inform about detected fires, victims, dangers and obstacles. The 3D model changes the color of the underground, such that the team coordinator gets a good overview which area was investigated already. Obviously, this feature requires reliable self-localization of the robots; otherwise it might imply a wrong assumption to the coordinator. In cases, where the robots are not able to localize correctly, this feature should be switched off or the robot could transmit a probability for the correctness of the position and this could be color-coded in the display of the investigated area.

Each team member has a progress pane, which shows the advance in the way point following. The pane shows if a robot is waiting for input, if it was stopped or if it is following the way points. As it can be seen in Figure 5.4, a hint is given what to do next and a related button is shown. For example, if the robot had stopped because it found a new object, the text informs about this and a button allows the user to directly send a command to the robot to continue on its old path. If the robot follows waypoints, it is shown how many way points were given and how many are still left. A progress bar additionally visualizes this. This information should give the user always best feedback about the robot's current situation, workload and progress and should reduce free time, where no task is assigned. The general information tab keeps the most important information of each team member as a history, such that the team coordinator can refer to this record, even after some time.

Interaction/Input

The main input devices are at the moment mouse and keyboard. Most input is done by right mouse clicking on a certain pose in the model or GUI element. This opens a popup menu with the related actions that can be taken. For example, Figure 5.4 (left) shows the popup menu when the user right-clicks in the model.

On the right side of Figure 5.4 the menu is shown when the user clicks on the robot avatar in the model or the entity view on the right side of the GUI. According to the robot's mode the menu adapts (e.g. no "*teleoperate robot*" when the mode is teleoperation).

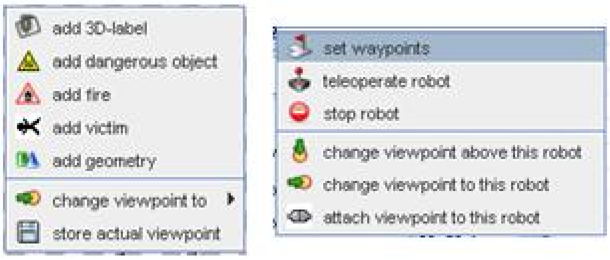


Figure 5.4: Popup menus for updating map (left) and commanding robot (right)

Other Support Functionalities

Team messages are shown below the main view in the GUI. By right clicking on the message the user already gets a popup menu of the option that are suggest as a reaction on this message ("*add a 3d label*", "*delete message*" etc.). As teleoperating a human-robot team is a rather complex task and it is likely that the user will make errors the GUI should enable redoing actions as far as possible. Here, map elements are not deleted completely, but are moved to a deleted object list. From this list they can be restored easily. Likewise, messages are not immediately deleted, but can also be restored.

5.4.2 Graphical User Interface for Teammate

Figure 5.5 shows a screenshot of the GUI, when the teammate is equipped with a personal navigation system. On the left side (1) buttons are located for fading in and out layers as in the supervisor GUI. Below three message buttons are located (2). When one of this buttons is pressed, a team message is sent to the supervisor GUI containing the kind of message (found victim, found fire, found dangerous area) and the actual position at the time when the button was pressed. According to this message the supervisor can update the map and re-organize the mission.

In the middle (3) the map is shown with the same icon as for the supervisor GUI. The two-dimensional map enables the user to find out, where the position is in the emergency area, where the other team members, victims, fires etc. are and where the exits are.

On the right side (4) an egocentric view of the laser data give the user a view what is just in front. The circle (blue) represents the position of the user. The smaller circles represent the distance the laser actually measures. Colors are used to identify obstacles that are very close to the user. A proximity alarm sign as well as an alarm sound warns the user if obstacles are very close. The blue line represents the next path segment that was given by the supervisor. This

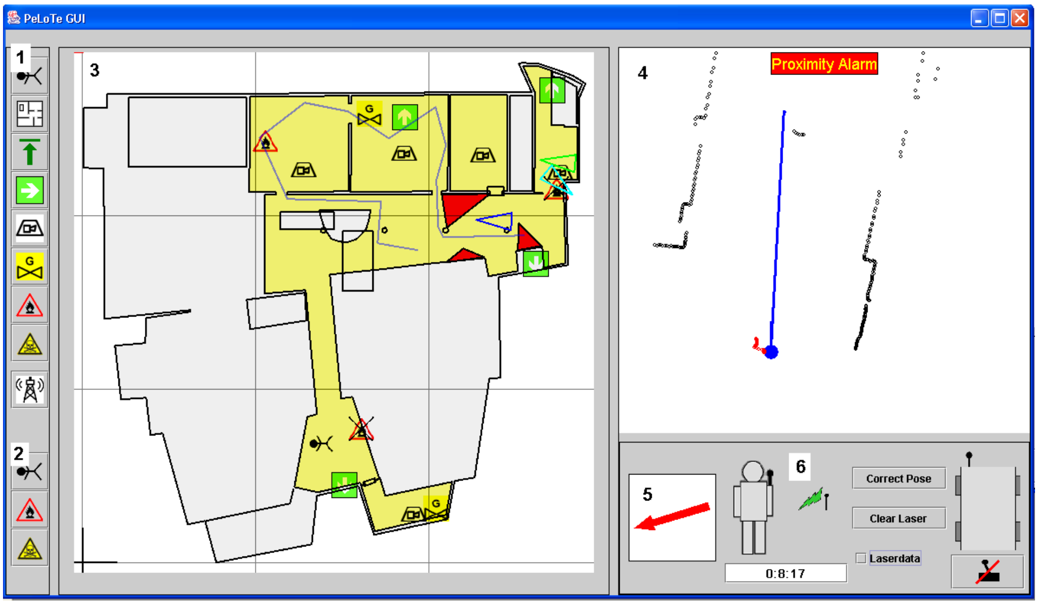


Figure 5.5: Teammate GUI. if personal navigation system is used.

egocentic laser view enables relatively fast movements and navigation through a dark environment.

On the right bottom a direction arrow (5) shows the current direction the human is looking with respect to the map. Other information, as connection status, is shown (6). Here, the user can also start the teleoperation mode of an accompanying robot in order to use it in dangerous area or in areas with difficult access.

Figure 5.6 shows a screen shot of a simplified version of the GUI, if the teammate only uses an audio communication device and a laptop connected to the system by a wireless connection and no localization system. The GUI only shows the two-dimensional map as well as buttons to add items to the map and send messages to the supervisor. If the teammate wants to communicate the own pose to other team members he/she has to update the pose in the map manually.

5.5 Expanding the User Interface Concepts

The principles for designing a human-robot user interface have been based on developments and testing with the user interface shown in the last section. Future work in terms of usability includes testing with different tasks and team configurations in different environments for verification of the drawn principles and establishment of further guidelines. Design of the teammate user interface has to be further investigated depending on the restriction of equipment he/she can take in the area.

Apart from further work in usability, technical advances can be introduced in the user interfaces. Therefore, the integration of 3D sensor data as well as first tests with stereo projection systems are shown in conclusion of this chapter.

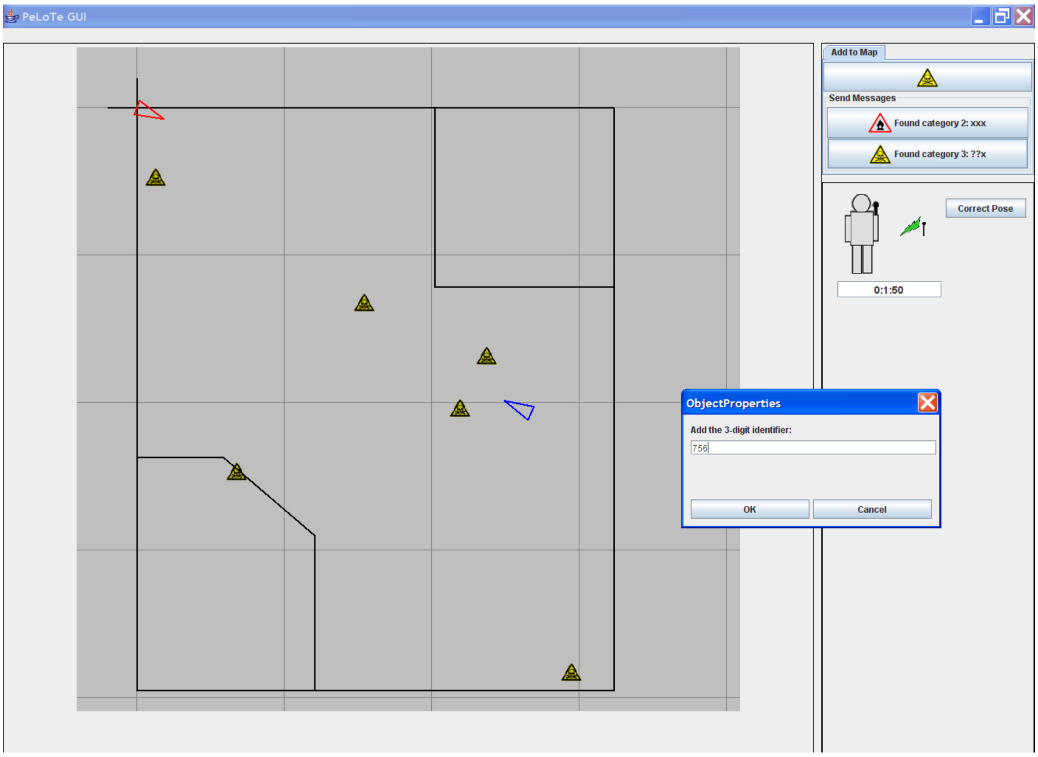


Figure 5.6: GUI, if teammate is only equipped with audio communication and laptop (no personal navigation system)

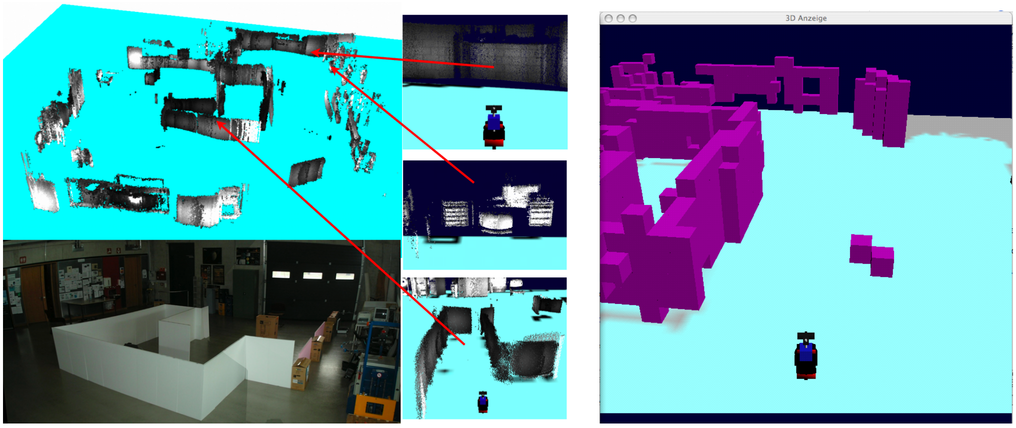
5.5.1 Integration of 3D Sensor Data

Recent developments in sensor techniques offer potential to use 3D sensors on mobile robots. An example is the PMD[vision]®19k, based on the Photonic Mixer Device (PMD) concept [113, 140]. It provides 160×120 pixel in one measurement. For each pixel three values are given: distance information, amplitude, and an 8 bit gray scale value.

An evaluation of this camera model for application on mobile robots and filtering, calibration and adjustment methods for camera parameters, can be found in [180]. From user tests it could be concluded that teleoperation based on live PMD distance images requires a high cognitive effort for the user and is not suitable. The field-of-view is limited and motion noise decreases the ability to percept the environment during robot movements.

In [180] two other possibilities are shown that allow indeed inclusion of the PMD camera for teleoperation tasks. First, a map can be built from PMD data and a 2D laser scanner. The map can be represented by a point cloud (cf. Figure 5.7(a)) or by an octree (cf. Figure 5.7(b)).

The second possibility uses only PMD data to create 3D panoramas. The robot stops, turns around and thereby takes PMD images, which are then merged to a panorama image. This results to an increase field-of-view and therefore better awareness about the surrounding.



(a) Point could representation (b) Octree representation

Figure 5.7: 3D map built from PMD data and 2D laser scanner in different representations [180].

The 3D panoramas can be directly integrated in the 3D GUI. A layer for 3D panoramas can be arranged and the panoramas can be stored with the location they were taken. Storing 3D panoramas from significant locations supports the supervisor in maintaining the overall awareness and keeping a history of past situations. The advantage of 3D panoramas before normal camera images is the increased field-of-view and the spatial knowledge that is included. This can be for example of importance if the supervisor wants to estimate if a robot fits through a hole. Furthermore, the distance information may support the supervisor in identifying objects in the remote environment. Here, he/she might combine information from stored camera image and 3D panoramas to gain more knowledge. The acquisition of 3D panoramas might be initiated by the supervisor or more advanced by an autonomous decision of the robot.

Integrating the map built from PMD data and laser scanner in the 3D GUI is more challenging. Currently, the mapping is done offline [180]. The 3D map increases the data amount significantly compared to the current used 2D information. For very unstructured, especially outdoor, environments employing 3D mapping data might have a significant benefit. Nevertheless, it has to be tested in which cases the visualization provides improvement compared to a 2D map. Moreover, it has to be evaluated whether the complex 3D data does not add to confusion and increase workload respectively decrease situation awareness. In particular, the point cloud representation might be confusing under some circumstances. The octree representation appears to be more suitable e.g. for navigation.

An advanced method to integrate the 3D mapping data would let the robot decide if the data should be shown to the supervisor. For example, if the robot preceives that it has entered a rather unstructured area, it could inform the supervisor that 3D map data includes a higher information content and should be visualized. On the other side, if the robot moves through a plane environment, e.g. through building corridors, the 3D data contains not more information than the 2D data and it is not necessary to transmit or visualize the data. Such methods requires a high level of autonomy in data analysis onboard the robot, but might be necessary especially if communication abilities are restricted.

5.5.2 Application on Stereo Projection Systems

Combined 2D/3D GUI

The previous presented 3D GUI provides some amount of telepresence to the supervisor attributable to the 3D environment representation and the sensor data from the remote team. Nevertheless, real immersion into the scene cannot be achieved with such desktop systems. For this, stereo projection systems are used, e.g. head-mounted displays, CAVE[2] systems, or multi-plane stereo projection screens.

For the supervisor CAVE-like system, as the for this work available three-sided stereo projection screen[3], are suitable. They allow multiple users working on the same screen, which might be necessary for complex supervision tasks. The large-scale of the screen allows to arrange the information amount clearly. Nevertheless, the user(s) work basically with one display, compared to using multiple desktop monitors, which may have drawbacks for robot teleoperation [187]. The stereo effect provides immersion into and can help to understand the remote situation. In particular, the spatial orientation can be supported. The integration of head-tracking systems further increases the feeling of immersion.

Figure 5.8 shows a prototype for using the human-robot system of this work on a three-sided screen. Figure 5.8(a) and Figure 5.8(b) show example applications. Figure 5.8(c) shows the system architecture, which is integrated in the client/server architecture explained in Section 3.5.1. For each wall of the screen an extra client is running. For the right side this is a version of the 2D supervisor GUI and for the other two sides a 3D GUI client was implemented. The 3D model on the left and middle side are visualized with different viewpoints such that a continuous picture is shown. More details on the implementation and the human-robot interface features can be found in [40].

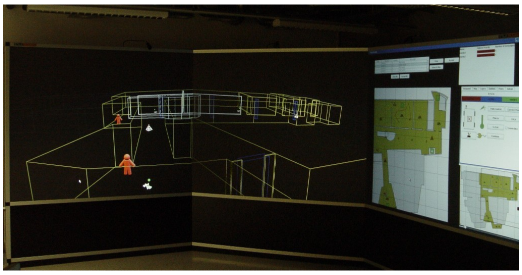
The described setup is customized for the human-robot system and showed the potential of such stereo projection systems. It initiated the development of the previously introduced desktop 3D supervisor GUI as well as of a framework for distributed 3D models, which is presented in the next section.

Distributed 3D models

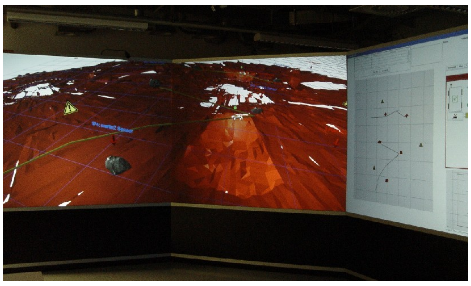
Telematic applications, as the supervision of human-robot teams, involve different requirements than other typical virtual reality (VR) applications on CAVEs or

[2]CAVE - Cave Automatic Virtual Environment [35]

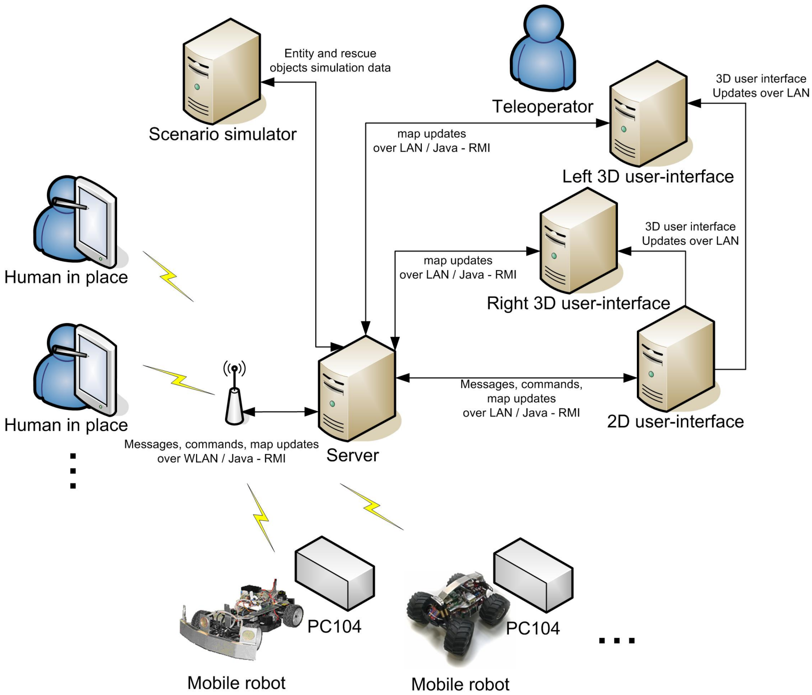
[3]The used system consists of three projection walls. Each wall has width of 2 meters and height of 1.6 meters. They are arranged with an angle of 135° between two walls. Two beamers project the pictures for the right and the left eye over a system of mirrors on each wall. The pictures are polarized orthogonal, such that user wearing glasses with polarization filters experience the stereo effect.



(a) Example for fire fighting application.



(b) Example for control of rovers in planetary exploration.

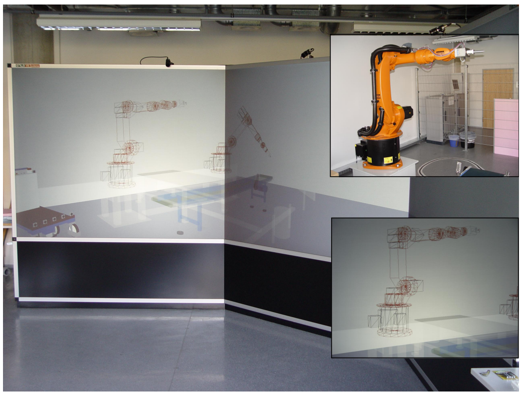


(c) System architecture for 3D stereo GUI in the human-robot system.

Figure 5.8: Mixed reality user interface. 2D map on the right plane and 3D model one the two left planes.

multi-plane stereo projection screens, e.g. car design process, architectural planning, or entertainment. These application areas often require very realistic and detailed models, which is not the case for the here considered applications. Details are only needed if they are significant for the task or if they give important visual cues for the user's situation awareness. The aim is not on realism, rather on an improved reality, e.g. by including symbols, color-coding, 3D labels etc. (cf. Section 5.3).

Whereas the mentioned other applications often use 3D models with none or pre-calculated, restricted movements, telematic applications use highly dynamic models, which are updated based on sensor data or user input. Moreover, interaction with the real world is facilitated by interacting with the 3D model. This requires a tight coupling between 3D model and real world. On one side, sensor



(a) Industrial assembly line with two robot arms, conveyor belt and transport vehicle.

(b) Sensor data visualization of PMD data.

Figure 5.9: Example for distributed 3D models.

data updates the model. On the other side, interaction with the model operates equipment, e.g. the robots. Based on these requirements a framework was developed. A last requirement was the ability to integrate with existing systems, e.g. the in this work proposed human-robot system.

The developed software provides a flexible framework for online, synchronous updates of distributed dynamic 3D models. The framework broadcasts updates of the viewpoint adapted to the actual screen as well as object manipulations (e.g. adding, deleting, changing color or position). Two examples are shown in Figure 5.9. Details on the implementation and the examples can be found in [39].

The main difference for the implementation of the supervisor 3D GUI compare to the approach introduced in the last section is that only one client is necessary for the GUI. The framework for distributed 3D models needs to be integrated into this client software.

Various configurations are feasible with the shown framework. For example, the supervisor can use the three-sided projection screen with a 2D/3D combination similar as in the previous shown approach. The teammate can use the same 3D model on his/her display. Thereby, the GUIs may connect to the server and build the models separately. It is also possible to connect both more directly and share the 3D models over the framework methods. This kind of interaction then provides an advanced approach for the common situation awareness model.

Chapter 6

User Studies

In the last chapter principles for user interfaces in human-robot teams were elaborated and explained. Techniques for representation of three-dimensional sensor data, e.g. the PMD camera, were worked out. Methods for visualization of human-robot interfaces or similar applications with a large amount of online updates and interaction possibilities on large scale stereo projection screens were developed.

The guideline development was guided by a review of the theoretical background and user-centered requirements as well as on several different implementation of user interfaces and system features. This implementations were evaluated in three major user studies:

1. Robot supported Fire Fighter Operation[1]
2. Cooperative Exploration
3. Multi-Robot Teleoperation in an Exploration Task

Partly, lessons learned from this tests were already explained in the previous discussions. Detailed results and conclusions drawn from these studies are summarized in the present chapter.

6.1 Robot Supported Fire Fighter Operation

The system developed in the PeLoTe project was tested with different test participants in a simulated rescue operation. 12 students and 12 fire fighters from a voluntary fire brigade participated in the experiment. Always two students or fire fighters composed a team of one supervisor and one teammate[2]. Half of the team performed the experiment with the developed system (with the GUIs of Figure 5.1(a) and Figure 5.1(c)) including two robots (similar to Figure 3.8(a)). The

[1]This study was performed during the PeLoTe project. Here, the result concerning the user interfaces are discussed.

[2]The terms supervisor, (human) teammate, robot or robotic teammate are used as introduced in Section 4.1.2.

other half used a map in paper form only. Both groups had a radio link for spoken communication available.

Parts of the experimental area were darkened to different levels or marked as dangerous areas. Signs symbolized fire scenes, fire detectors, exits and gas valves. Dolls represented victims, that had to be taken to the next exit. All items should be marked by the supervisor either in the GUI or the paper map. Before the experiment the teams could practice with the system in a separate environment. The evaluation was performed on basis of self-reported data from questionnaires, performance measurements (time, rescued victims, found objects etc.), an evaluation of the memorized situation after the test, and observation of the teams.

6.1.1 Results and Discussion

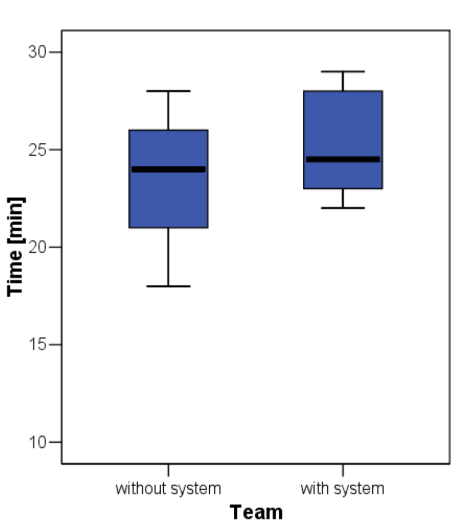
In the following particular the results and observations related to the GUIs and interaction between humans and robots are analyzed. The evaluation of the whole PeLoTe system can be found in [45, 152, 134].

Performance

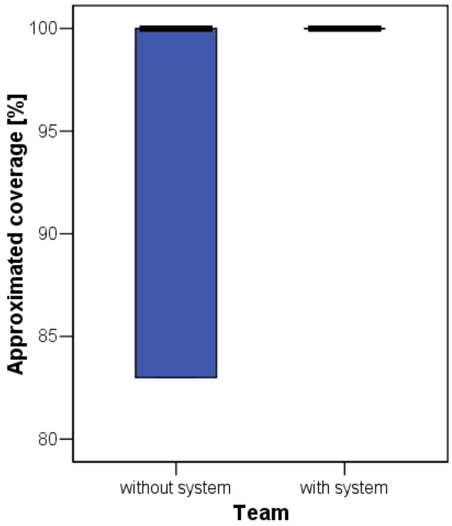
Figure 6.1 shows the result of the most important performance measures comparing the teams that used the system and the control group without the system. Teams with the system used the robots for performing part of the exploration and could therefore act more target-oriented, e.g. the human teammate rescued victims that were already identified by the robot. Most of the time the robots acted autonomously, but the teammate or supervisor could take over direct teleoperation. This was e.g. done for the exploration of dangerous areas, which were not accessible for the humans. Direct teleoperation (here with a joystick) costs much time and slowed down the teams. Nevertheless, the time duration between teams with and without system did not significantly differ, since on the other hand navigation through darken areas was speed up with the system (Figure 6.1(a)). The experiments showed that the teams that had performed the experiment with the support system found slightly more objects and could rescue more victims (Figure 6.1(c)) and searched the area slightly more complete (Figure 6.1(b)).

Subjective measures

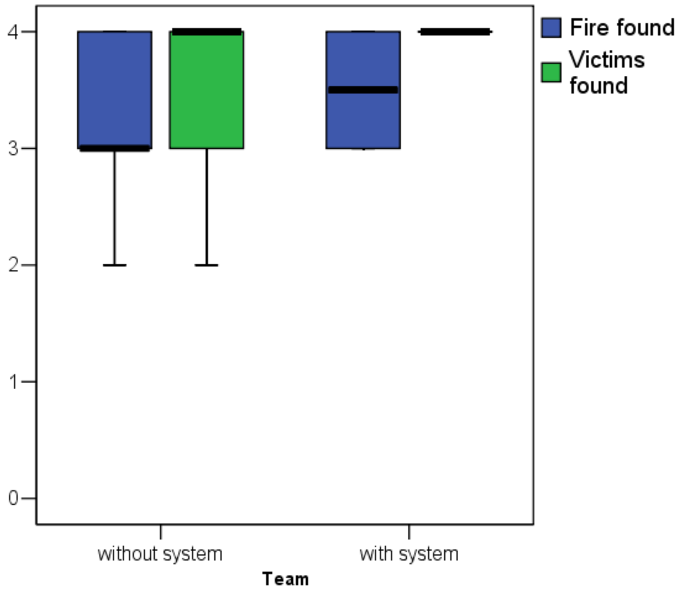
These pure performance based measures do not fully show the potential of tested system. Especially, navigation in darkened areas and the communication about the actual situation could be highly improved with the system. Observations and the questionnaires showed that the supervisors in teams with system understood the situation on-site better, could take a more active part and provided an improved support for the teammate. Figure 6.2(a) shows the PANAS scale (cf. Section 2.2), which revealed that there is no difference in the positive affect for teams with or without the system, but more negative affect for teams without the system. Large differences in the rating occurred in the negative categories *worried*, *nervous* and



(a) Time teams needed to finish the task (No fixed time limit was given. Teams were told to use about 25 minutes and finished then, when they thought they had completed the task).

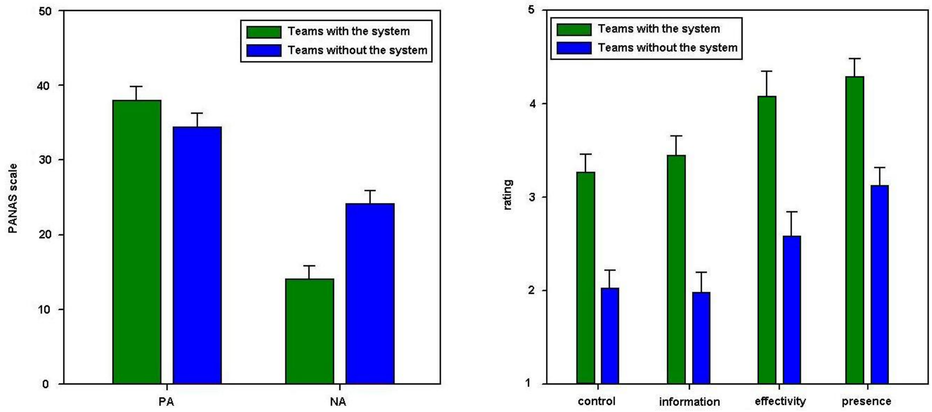


(b) Approximate coverage achieved by the teams (Only the area that could be covered by human team members was considered, i.e. areas that could only be reached by the robot (dangerous areas) were excluded to ensure a fair comparison between both groups).



(c) Number of victims that were rescued respectively fires that were identified in the area (The maximum that could be achieved was four for both).

Figure 6.1: Boxplots of performance measures.



(a) PANAS scale for teams with and without the system.

(b) Self-assessment of control, information, efficiency, and presence.

Figure 6.2: Subjective measures.

confused. Figure 6.2(b) shows the result of a survey that asked different questions analyzing how test participants rated their feeling of control, of being informed about the situation, efficiency in task performance, and immersion/presence.

Memory test

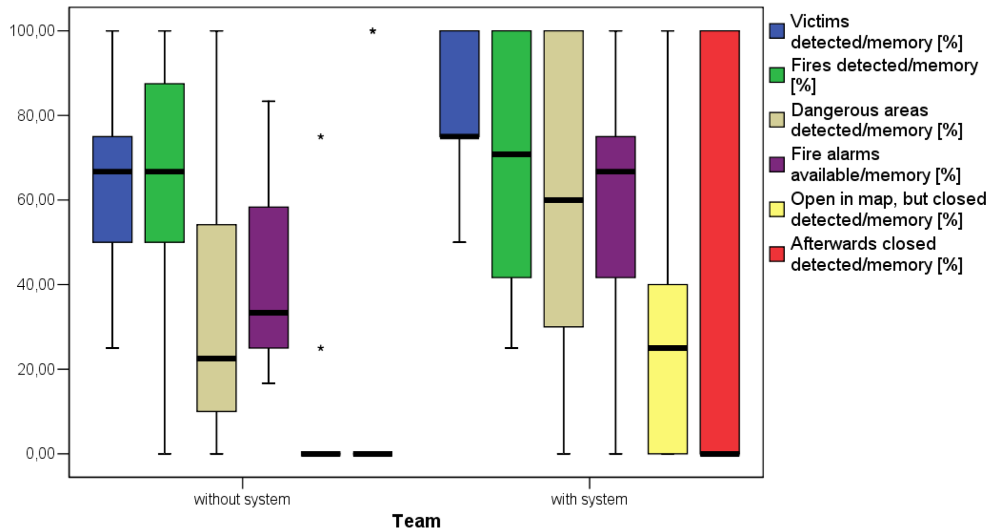
As a reference for the spatial awareness of each team member the participants where asked to fill a blank map after they finished the test from memory. In general, the teams with the system could re-call the items/events better than the teams without the system (Figure 6.3(a)), indicating also that their common situation awareness was improved by the system.

Figure 6.3(b) points out the difference between supervisors and human teammate for teams without and with system. Without the system the human teammate is able to remember slightly better for most of the items. Observations support the finding, that the supervisors without the system were poorly involved in the remote situation. Moreover, both team member seemed to concentrate strongly on the main task to rescue victims and find fires. Both items could be memorized better than other events, e.g. map changes or fire detectors.

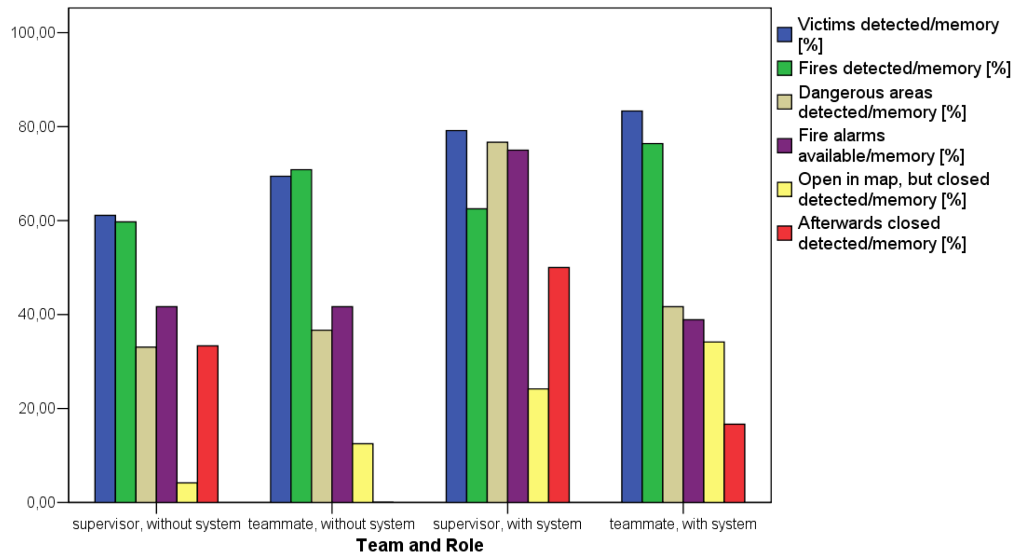
Similarly as in team without the system, in teams with the system teammates could remember victims and fires slightly better than the supervisors. Nevertheless, the supervisors in team that used the system had a better memory about the surrounding tasks (dangerous areas, fire detectors, map changes). This seems to be a result of the better overview gained by using the GUI.

Observations

Several principal issues could be observed in the interaction of the team with the user interfaces. To some extent, this issues have already been discussed in



(a) Boxplot of the items that test participants correctly marked in the map related to items they encountered during the experiment.



(b) Average of items memorized related to items found for different actors in the teams with and without system.

Figure 6.3: Spatial awareness based on the memory of test participants shortly after the experiment.

the previous chapters and are summarized here in the context of the performed experiment:

1. The supervisor was strongly concentrating on the teammate, especially if he/she is moving through areas with low visibility, i.e. in difficult environments. Moreover, they could talk a lot over the radio communication. Only if the teammate did not require any assistance the supervisor would concentrate on the robot or on messages sent by the robot.

2. Communication between supervisor and teammate is not restricted to pure information or command exchange. The conversation includes also discussion about the situation and future strategies.

3. The message system showed to be a critical component of the interface. Often several message arrived at the same time at the supervisor, who was then often overloaded by the number of messages. This got worse if the supervisor concentrated a long time on the conversation with the teammate (as described in 1.). As a result many messages accumulated and the supervisor initiated automatic processing of all messages, e.g. all in the message reported objects were added to the map in one step.

4. The information content of automatically processed message often got lost, i.e. the related object was indeed marked in the map, but the new information did not receive the supervisor's attention. As a consequence he/she was not aware about this objects and necessary action were not initiated. In particular, problems occurred when the information in the message was not correct, e.g. for the few cases the robot did position itself wrongly. The result was not only a wrong map, but also a complete lack of understanding and confusion about the situation.

5. Autonomous behavior of the robot (e.g. the robot moved back to the exit if it arrived at the end of the path and did not get a new command for a longer time) result in confusion and gave the user the feeling the robot was acting arbitrarily.

6.1.2 Conclusion

The results from this test entered the design of the 3D supervisor GUI, further developments on system level and inspired the next experiments.

The message system showed to be a problematic element and was therefore improved. In future versions of the system, messages were sorted according to priority, i.e. messages that affect human safety have a higher priority than messages about robot safety, which on the other hand have higher priority than task related messages, e.g. reached end of path. The visualization of the message system was also improved. Autonomous processing of all messages was removed as it led often to confusion when applied.

In following tests the robot(s) used a lower level of autonomy, e.g. no autonomous navigation to the exit without human command. Moreover, visualization of robot behavior was added in the supervisor GUI to prevent confusion about autonomous behaviors.

The results that the supervisor strongly concentrated on the human teammate and the way how supervisor and teammate communicate guided the design of the subsequent experiment, which is shown in the next section.

6.2 Cooperative Exploration

Observation in the user tests discussed in the last section gave evidence that spoken communication has a major impact on interaction in the human-robot team and on the way how the supervisor shared his/her attention. Therefore, an experiment was conducted in which every team had to perform two runs, one with and one without audio communication.

Before the experiment it was expect that the supervisor will treat remote human and robot more equal if no audio communication is allowed, i.e. he/she is able to consider them as rather coequal team members. The audio communication is not only used for naturally and fast exchange of information. In the previous experiment the humans discussed a lot about observations and resulting strategies. This is a feature, which current robots hardly provide. Nevertheless, it might be a key point for successful teams. By cutting the human-human communication to the same level than the human-robot communication it was assumed that the overall performance of the team as well as the supervisor situation awareness drops.

6.2.1 Experimental Setup

Team Configuration and Task

Each team performing the experiment consists of a supervisor working outside of the exploration area (using a previous version of the GUI shown in Figure 5.1(b)) as well as a human team member (using the GUI shown in Figure 5.1(d)) and a robot as shown in Figure 3.8 (b) that move through the area.

The team task was to explore a room, find and identify objects. The basic structure of the room was known by the team, i.e. the walls and blocked areas were marked in the map. The positions of tables and smaller obstacles, as e.g. carton boxes that were distributed to make the environment more cluttered, were not marked in the map. The robot was able to move below the tables.

Ten objects were distributed in the area. The objects are defined by a unique pattern[3] and a three-digit number (cf. Figure 5.2). Each object belongs to one of three categories, which require the team to cooperate at different levels for achieve identification of the object.

[3]The recognition of the patterns is based on the ARToolkit [92, 175], as described in Section 3.5.2.

Table 6.1: Object Categories

	Category 1	**Category 2**	**Category 3**
Required level of cooperation	robot alone or teammate alone	robot and supervisor together or teammate alone	robot, teammate and supervisor together
Correct three-digit number can be found by	by the robot autonomously or by the teammate	by the supervisor based on the camera image or by the teammate	one digit from each team member

From first sight all categories look the same, the unique pattern and below a three-digit number, which was not correct for all object categories. The category of an object can be identified by the robot autonomously. The teammate can find the category of an object with a printed list that contains 84 patterns and the related category. The high number of patterns in the list were used to make the task more difficult for the human teammate. In the next step the correct three-digit number has to be determined.

For the first category this could be done by the robot or the human teammate alone (by image processing respectively by the printed list). The number of objects from category two could be derived either by the human team member alone or by the robot and supervisor together. Category three needed cooperation of all three team members, each had to contribute one digit. Table 6.1 summarizes the object categories.

The classification into three categories was done to foster cooperation, i.e. each team member has to bring in their capabilities. The robot is able to quickly categorize objects. The human is faster in exploration (in the environment used in this experiment) and therefore finds the markers faster. The supervisor has the overview about the whole situation. All three are needed to identify the category three objects. The tasks for each team member and the strengths are summarized in Table 6.2.

Test Participants

Five test teams performed the experiment as it was described before. The ten test participants were recruited on a voluntary basis from students and postgraduate students of Computer Science. All ten participants were male and had prior experience with remote-controlled cars, computer-games or even mobile robots. Except for one group, the two team members did know each other beforehand.

Table 6.2: Task sharing between the team members

	Robot	**Teammate**	**Supervisor**
Tasks	explore	explore	maintain map
	find objects	find objects	operate robot
	identify category	identify category	guide team
	identify number on object category 1	identify number on object categories 1 and 2	identify number on object category 2
	contribute one digit to object category 3	contribute one digit to object category 3	contribute one digit to object category 3
Strength	fast identification of category	fast in exploration and therefore in finding objects	overview about situation from the detailed map

Procedure

The teams performed the task twice, one time with audio communication via headset and one time without. The environment for both runs was the same. Nevertheless, different objects were used as well as the position of the objects was changed between the two runs. The order of using audio and not using audio was altered between the teams. All experiments with audio were performed in the same set-up of objects as well as the non-audio experiments.

The test participants were asked to fill in a questionnaire about their background beforehand and questionnaires about the experiment after each run. One run lasted until the team agreed that all objects were found, completely identified and marked in the map. The teams knew that they had to search ten objects.

Before the first run the participants were allowed to train about 15-20 minutes in a different area. They were also allowed to discuss their strategy before starting each run.

6.2.2 Results and Discussion

Performance

The performance was evaluated by the time for task completion, the number of found and correctly marked objects.

Figure 6.4 shows the time duration each team needed to explore the area, i.e. the time until they thought they had succeeded to find and identify all ten objects.

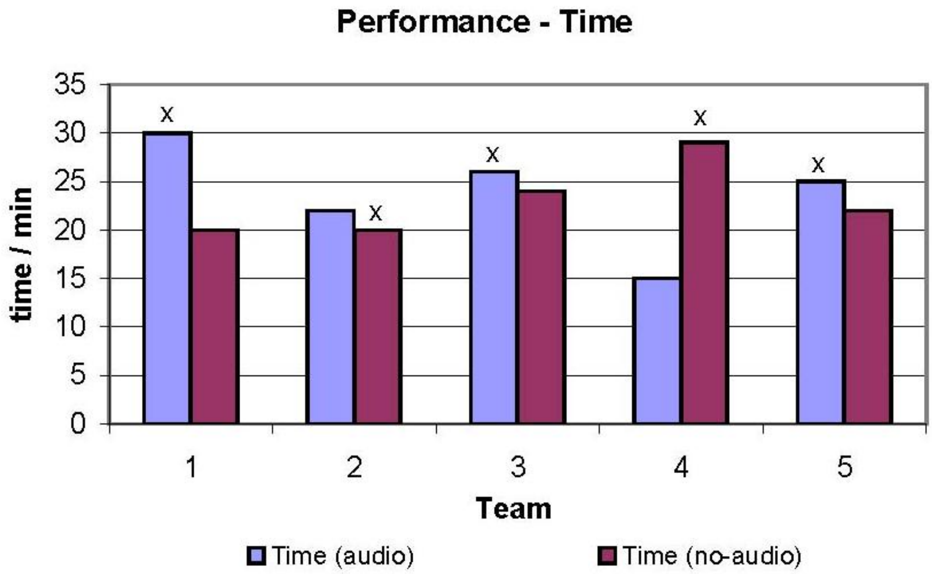


Figure 6.4: Performance measure time. The x indicates the first run.

The average time needed with audio is 23.6 min and the time without audio is 23 min. There is no significant difference between the two runs concerning time. Contradictory to what was expected most teams were slightly faster without audio communication. Nevertheless, one team performed extremely fast with audio (Team 4).

Figure 6.5 presents how many objects the teams really found and how many of them were correctly marked in the map. The number of found objects is an indicator for the awareness of the system status. The teams were asked to complete ten objects. Nevertheless, it could happen that they assumed that they had found ten objects, but in reality had found and marked one double and missed another. In average with audio 9.2 objects were found and without audio 10 objects, i.e. all objects. 91% of the found objects were marked in the case of audio communication was allowed and 88 % were marked in the case of no audio.

It was expected that the team would perform faster as well as find and mark more objects correctly for the runs with audio. This assumption was not met as both runs were equally successful in performance. Several reasons might explain this fact.

First of all the number of teams was very small and every team used a slightly different strategy. Some teams mentioned afterward that, if they had used another strategy they would have been much faster. The training time was too short for them to find an optimal way of performing the task. Moreover, in general the teams mentioned that the second run was much easier as they learn to use the system more efficiently and as they could do some optimization in the strategy.

Moreover, most teams did not communicate a lot via audio contradictory to the last experiment (cf. Section 6.1). The task were clearly distributed. The supervisor operated the robot and concentrated on the robot and the teammate

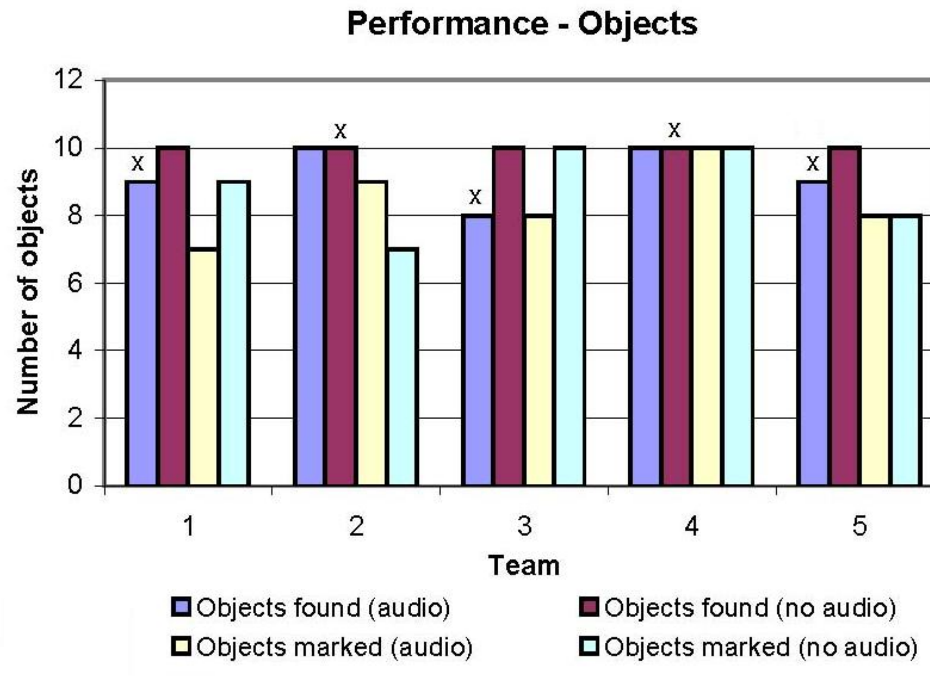


Figure 6.5: Performance measure found and marked objects. The x indicates the first run.

searched alone until an object of category 3 was found (cooperation between all three team members needed). The team configuration was rather a sub-team of supervisor and robot cooperated loosely with the human inside the area. One reason for this configuration was that most supervisors preferred to operate the robot manually instead of using the waypoint and search function. Therefore, the robot needed all their attention. The teammate did not really need the attention of the supervisor, i.e. no input was needed expect for objects of category 3.

The performance of team 4 supports this view. This team has found and marked all objects correctly during both runs. Additionally, their second run (the one with audio) was much faster then all other runs. A major difference compared to other team was that this team used the audio communication intensively during the mission. At the end, when only category 3 objects were left the teammate and supervisor worked together to bring the robot to this objects. Exceptionally from other teams, the teammate of this team also used the message system to send longer messages in the run with no audio communication instead of transmitting only the found digits. (Example: "Category 3. Come with the robot to my position"). This might be a reason for their good performance also during this run.

Survey Questions

The questionnaires between the experiments investigated perception of the robot, trust into the data coming from robot or human, what the participants thought

Table 6.3: Trust in the information from human team partner vs. robotic team partner. Answers range from 0 (did not trust the information at all) to 4 (trust the information completely).

information source	**teammate (audio)**	**teammate (no audio)**	**supervisor (audio)**	**supervisor (no audio)**
robot	3 (0)	3.25 (0.25)	2.2 (2.7)	3.4 (0.3)
teammate	-	-	3 (3)	3.8 (0.2)
supervisor	3.25 (0.92)	2.5 (0.3)	-	-

about robot control. After each run the same questions were asked. At the end the participants were asked to compare both runs.

The test participants were asked if they thought the robot was rather a tool, an intelligent tool or an equal team partner. The most given answer was "a tool". An "intelligent tool" and a "team partner" was chosen one time each. As most of the participants had been working with robots before this experiment might have influenced their opinion.

Table 6.3 shows how much the team members have trusted the information coming from their team partners. The values are the averages, where 0 means they did not trust the information at all, 4 means they trusted the information completely. The numbers in the brackets are the variance of each value.

The teammate trusted the information from the robot equally and the information from the supervisor slightly less when using no audio. The supervisor trusted the information from the robot more when using no audio communication and the information from the teammate slightly less with audio communication. In general, the supervisor trusted the information from the human slightly more.

The results from the survey questions did not show significant difference between audio and no audio communication. One reason might be the small number of tested teams. Moreover, the performances metrics did also not show any significant difference what might be due to the different strategies that could be used. For example, some did a complete independent search, whereas others did a joint search, where both teammate and robot were working side by side. The different strategies had advantages and disadvantages.

6.2.3 Conclusion

The experiment has shown that a critical point for human-robot teams is the way of communication. The implemented message system was an improvement to our previous tests. Nevertheless, it was still a major point that participants criticized and was further improved for the next test. Semi-automatic adding of 3D labels based on messages has been proved to be a very helpful feature.

The number of found objects is an indicator of awareness of mission status, i.e. if the participants agreed that all objects were found, but indeed missed one or several and counted and marked instead another double they were not well informed about the overall status of the mission. The learning effect seems to influence this result more than the usage of audio communication. This result indicates that with some practice the user could keep track of the mission status, but the visualization was not intuitive enough. This result was also supported by informal comments the users gave after the experiment.

The different views of the virtual environment helped the supervisor to understand the situation a lot. But the fixed sizes of the views hindered. E.g. the first person view was helpful to compare video image and virtual model. On the other hand the general overview was better maintained from above. In the next version of the supervisor GUI the viewpoint control was improved and the user was enabled to select which view should be shown in the main view.

Moreover, the state of the robot as well as task performance visualization was improved for the next user test.

6.3 Multi-Robot Teleoperation in an Exploration Task

The observations in the both previous described user tests raised the question if automatic map updates are at all useful. Therefore, the next experiment [43] compared different information presentation methods (messages vs. autonomous updates) besides evaluation of the latest version of the 3D supervisor GUI (cf. Figure 5.1(b) and Figure 5.3).

6.3.1 Experimental Setup

Task

For the test scenario the supervisor has three simulated robots available, which are used as first explorers of an emergency area. The task for the test participants was to investigate the whole presented area as fast as possible. The robots could be used in waypoint mode or can be teleoperated with the keyboard, what was not recommended to the participants.

The robots start at three different entrances and will report different events while they are moving through the environment. These events will cause a certain behavior of the robot, which might require actions from the human. For some actions the robot needs assistance and sends a message. Other information can be added autonomously by the robot to the map. Table 6.4 gives an overview of the possible events.

For the experiment two victim events, two fire events, four dangerous object events, four obstacle events, two battery events and one motor problem event were simulated. Obstacles were also used to simulate blocked doors. Figure 6.6 gives a map overview how these events were distributed in the area.

Table 6.4: Task Sharing between the Entities in the Team

Event	**Robot behavior**	**Expected action from supervisor**
Victim found	stops and sends a message to the operator	add the respective icon and a 3D-label containing information about the victim to the map
Dangerous object found	sends a message to the operator	add the respective icon with a dangerous area around to the map
Motor status critical	sends a message to the operator	move the robot out of the area
Fire found	stops and adds autonomously an icon to the map	add a 3D-label containing information about the fire to the map
Obstacles found	updates the map autonomously	none directly - but consider new obstacle in future assignment of paths
Battery status	status display for the operator changes first to yellow, than to red	move the robot back to an exit

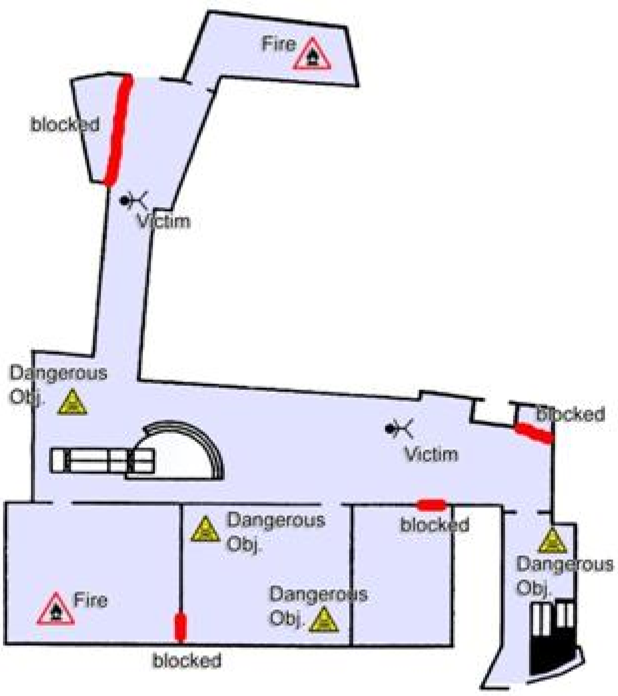


Figure 6.6: Test Scenario.

Test Participants

Seven test participants in the age between 26 and 35 years performed the experiment. They were volunteers, mainly PhD-students with different study background.

Procedure

First the experiment scenario, the task and the user interface was explained to each participant followed by a 15 to 20 minutes practicing period with a training simulation.

During the experiments the different events, reactions and commands of the user were written to log files to measure e.g. reaction times for the later evaluation. After the experiment the test participants were asked to fill a questionnaire. Additionally, they were observed during the experiment to get an impression how they react on different events and how they interact with the interface.

6.3.2 Results and Discussion

Task and Time

As it was very easy to see the covered area from the GUI, every test participant was successful in reaching the first mission goal: covering the complete area. The completion time in average was 896 s (std 119.42) with the fastest run of 736 s and the longest 1080 s.

Log data was available for dangerous objects, victims, and fires. All test participants marked all dangerous objects and all victims. Two test participants missed both fires and two missed one fire.

Reaction Times

Figure 6.7 shows the reaction time in seconds of the participants on dangerous objects, fires and victims.

The events that were visualized with messages (dangerous objects and victims) show in average lower reaction times than events that were automatically updated (fires). Nevertheless, no statistical significant difference could be proved. This may be attributed to the small number of test persons. Moreover, both dangerous objects and victims show a number of extreme outliers. It could be observed that for the few cases a message was overseen or forgotten, it took a longer time to realize that. In particular, with dangerous objects, where the robot did not stop, this was a problem. For future work a kind of reminder function for messages that seem to be overseen can support here.

For obstacles, motor and battery status no log data was available. Observations showed that obstacles (autonomous updates without stop of the robot) were often

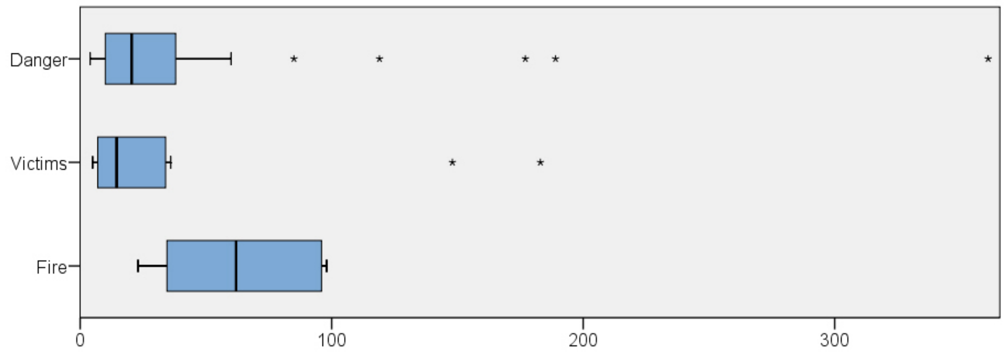


Figure 6.7: Reaction time in seconds for different events.

not immediately recognized, e.g. it could happen that the test person was sending the robot to a room, which was blocked. The robot updated the map correctly, but as the test person did not realize the update, the robot was sent to the same room several times again. Hardly any of the test participants realized the change of the battery status, which was shown by an icon. In contrast, the change of motor status, which was visualized with a message of middle priority, was recognized.

Visualization Method

The participants were asked about the method of visualization (message vs. autonomous update/icon). Figure 6.8 shows the opinion of the test participants about the visualization method. For visualization of victim and dangerous object events the message visualization was preferred by most test participants, as it is also supported by the measure of reaction time (cf. Figure 6.7). Most of participants would prefer message visualization also for battery status, as mostly this information was overseen.

GUI elements

The test participants were asked to rate different GUI elements according to how difficult it was to use them (cf. Figure 6.9). Compared to the previous experiment in Section 6.2 the viewpoint control and the handling of messages could be improved as also visible from the rating towards *Easy*.

Awareness

Figure 6.10 shows the rating of awareness about the three categories that have to be represented in a GUI for human-robot teams: entity, environment, and mission status (cf. Table 4.2). Mission and environment awareness were rated high. According to the rating as well as observations the visualization of the robot status needs some more improvements.

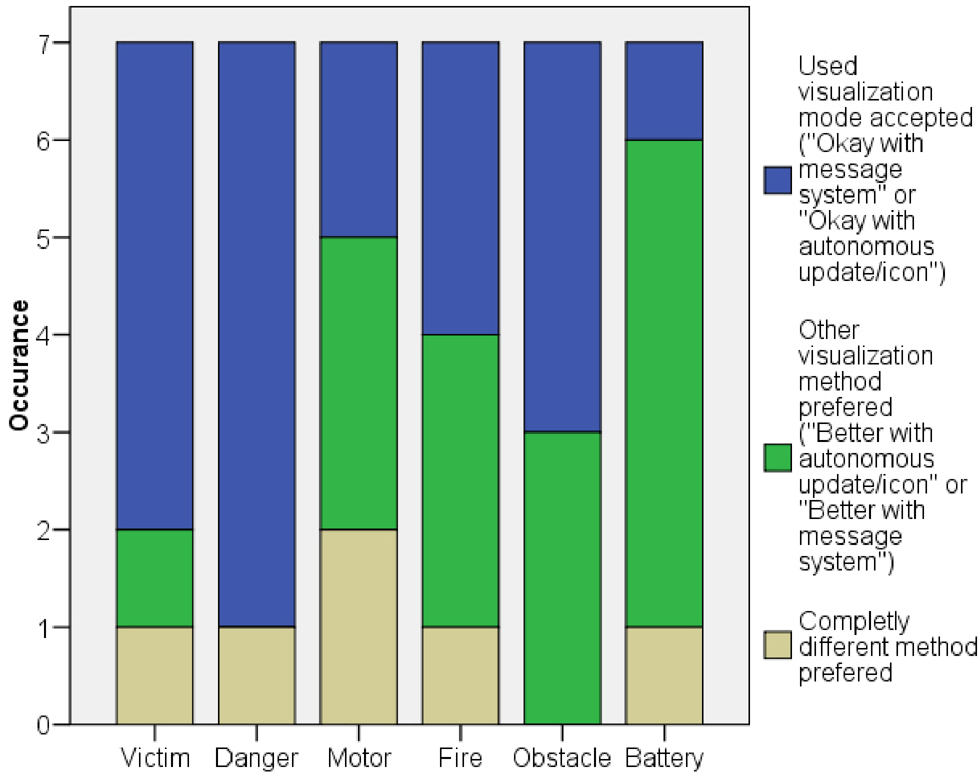


Figure 6.8: Subjective evaluation of visualization methods. Victims, danger, and motor events were transmitted with the message system. Fire, obstacle, and battery events were autonomously updated in the map or shown by an change in the related icon.

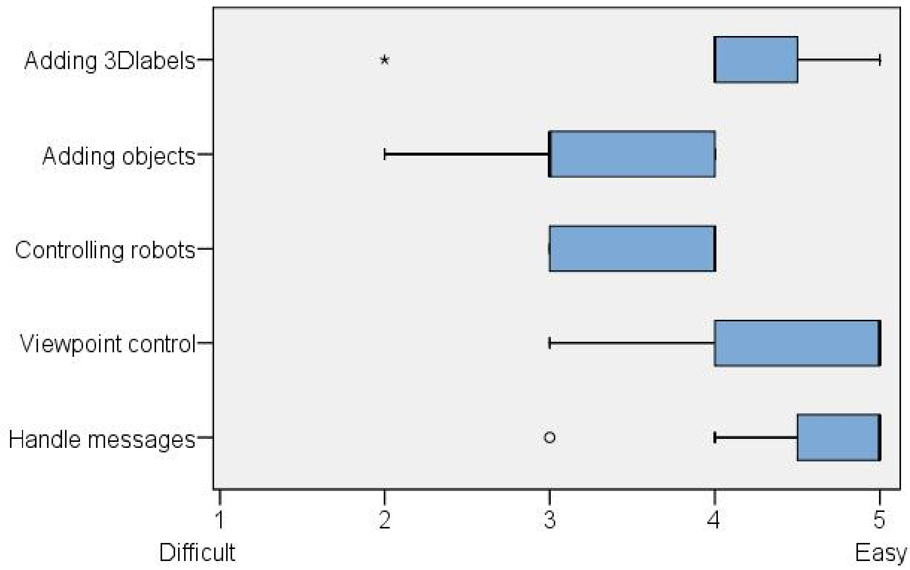


Figure 6.9: Subjective rating of GUI elements.

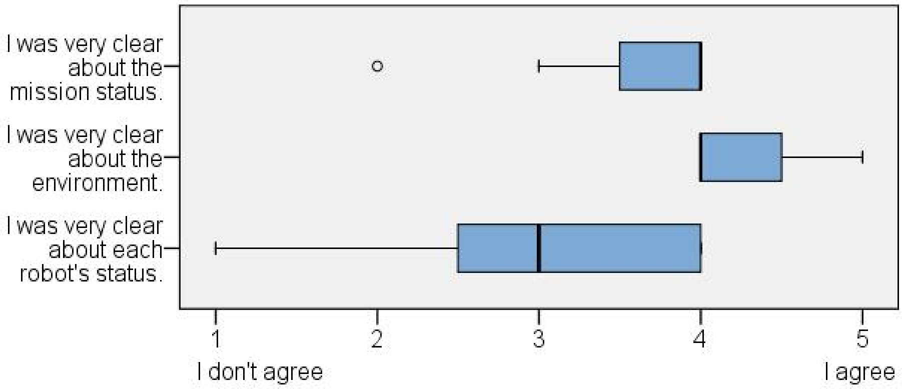


Figure 6.10: Awareness on entity, environment, and mission status based on rating of the test particpants.

Self-assessment of workload

The test participants were asked to evaluate the workload of the test (cf. Figure 6.11). The first two statement show a significant negative correlation, as expected.

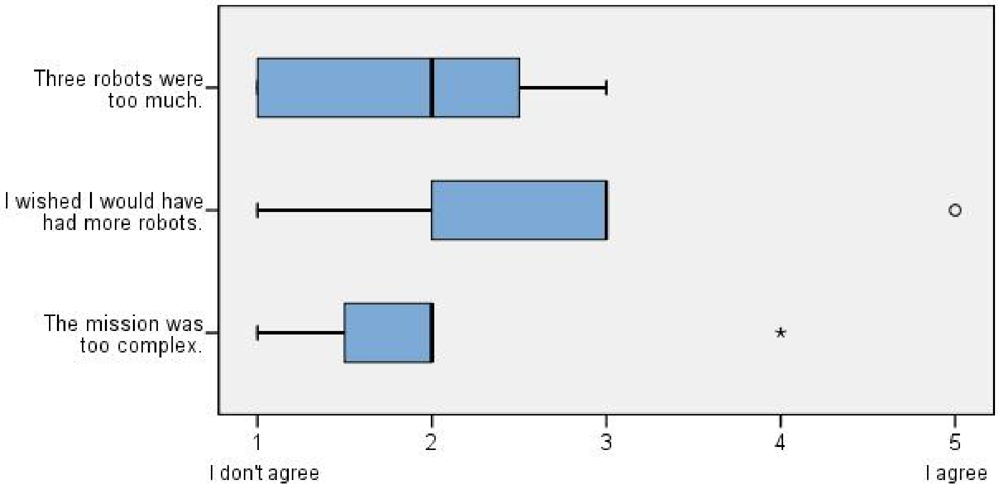


Figure 6.11: Subjective evaluation of workload.

6.3.3 Conclusion

In the previous experiments described in Section 6.1 and Section 6.2 the focus of the supervisor was strongly on one team partner, mainly the one that needed more support. In the experiment described in Section 6.1 this was the human teammate, since most of the experimental area was darkened and navigation was difficult for him/her. In the second experiment (cf. Section 6.2) the supervisor concentrated more on the robot, as most participants decided to teleoperated the robot directly. Now, the team was homogeneous (3 robots) and several supporting features were implemented to reduce the need for focusing on one team member.

For example, a robust path planning in the waypoint setting was implemented, such that the coordinator was released from teleoperation.

Earlier experiments showed the difficulty of the coordinator to integrate information that was autonomously updated by the team members. This last experiment supported these observations. Automatic update (fire and obstacles) was compared against presentation through messages (victims and danger objects). As Figure 6.7 shows the reaction time on message visualization was in average lower than for fires. Additional some participants overlooked autonomous update of fires completely and did not react at all. It could be observed that obstacle, which were also update by the robots directly in the map, were sometimes overseen, e.g. when test candidates did not understand why a robot did not move into a blocked room. Battery status change, which was visualized by an icon was overseen, whereas the motor status change, which was represented with a message was usually processed. The rating of the visualization methods from the questionnaire supported this view. This result indicates that communication over messages is indeed useful depending on the kind of information. The experiment showed that the implementation of the message system helps the supervisor to better keep track on objects discovered by the robots. Nevertheless, in realistic scenarios not all information can be reasonably visualized with messages. In order to allow the supervisor to process also a high amount of updates autonomous map updating is necessary. Then, suitable highlighting has to ensure that the user integrates the information in his/her situation model, e.g. by appropriate attention guidance.

In general, the usability of the interface could be improved versus the last tests. Table Figure 6.9 shows that the GUI elements were rated as easy to use. Additionally, it could be observed that novice users could learn most functions quickly and considered the usage as easy.

Chapter 7

Conclusions

The wish to integrate mobile robots in the workflow of human teams requires advances in the interaction capabilities. This work contributes thereby to the development of efficient teleoperation interfaces for task-oriented human-robot teams. Particular, human-robot teams in high-risk domains, as fire-fighting, were investigated.

First, a framework for human-robot systems was developed based on the supervisory control concept of Sheridan [158]. Based on the model introduced in [131] the autonomy design of the proposed system was designed. The different subtasks that are typically performed, e.g. in an exploration task in a rescue operation, were analyzed and a suitable level of autonomy was selected for each subtask. Related problems, especially the requirements on situation awareness, were discussed. The implementation of the proposed concept, which was used also for the user tests, was explained. Finally, the concept was further developed towards the realization of team work between human and robot.

Furthermore, requirements for interfaces in human-robot teams were elaborated. In summary, efficient interaction requires information about the team members (robot and humans), about the environment, as well as the mission and task performance. This information demand is shown generally and based on an evaluation of potential end-user requirements for the special case of rescue operation. The representation of required information in a map and the related interaction mechanism are further extended.

Based on the elaborated information requirements, on results from user tests, and on available guidelines, e.g. from human-computer interaction principles for graphical user interfaces for human-robot teams were developed. These do not only concern visualization, but influence the design of the system as as whole. In the following the concluded principles are summarized:

- Provide feedback from system and team members in a suitable manner.
  The system has to respond correspondingly to user input. In particular, long response times when sending commands to other team members and unexpected behaviors have to be explained.

- Present sufficient information about the situation with a suitable design.
  The user interface needs to provide only the suitable information according to the human's task in the team and has to provide support for reducing the short-term memory use. Information needs to be shown in an integrated manner and important data has to be highlighted. Information re-gain after switching the attention to a team member has to be supported. Support for estimating distances are useful.

- Design input and command methods carefully.
  Dialog use and input of command have to be structured and the user has to be able to exit command input anytime. Control methods shall be natural and the user shall be able to act directly with the objects in the map. Support for efficient task assignment and comfortable map updating are needed.

- Consistency within the system and the task.
  Terminology, actions, and commands have to be consistent in the user interfaces, system components and over the team. Terminology and symbolic has to be adapted to the users and the application.

- Allow flexibility.
  The user interface should provide shortcuts and different ways to interact. A user interface based on a 3D model of the environment has to allow freedom over the viewpoint control, but simplify navigation through the model.

- Errors and Recovery, Safety
  The system and team members shall be equipped with autonomous methods to protect themselves. Erroneous commands shall be prevented or help for recovery has to be given. The user has to be able to undo actions.

- Use 3D visualization techniques where appropriate.
  3D models can support the supervisor, especially when sensors for 3D measurements are used. Typical virtual reality techniques, as shadowing or lighting, shall be used sparsely to prevent confusion of the user. The aim is not a realistic model or full immersion of the user, but the presentation of information as required for the actual task at hand.

Finally, this thesis shows the potential for using 3D user interfaces for the supervisor on the example of a novel 3D sensor technique (PMD camera) and the introduction of stereo projection systems. Methods and techniques for implementation of user interfaces on basis of three-dimensional data or on large scale stereo systems are contributed.

The work shows the potential and the challenges for human-robot teams in high-risk, time-critical, and dangerous tasks. The results of this thesis contribute to the development of future teleoperation interfaces by an analysis of the needed system structure and information demand, as well as by an elaboration of the needed guidelines for interfaces design. Example implementations and the lessons learned from the user studies give a references for the claimed teleoperation interface design requirements. Technical contributions in the area of 3D user interfaces in general and especially for fire fighters further broaden the study.

The proposed infrastructure enables the integration of robots in human teams. The current state of technology allows real application of such human-robot teams only for very specific tasks. Nowadays, such systems imply also extreme costs, which for example cannot be carried by most fire brigades. Furthermore, the area of human-robot interaction has to be investigated in many different facets. Nevertheless, the work provides a prototype and a important basis for further development of teleoperation interfaces in future human-robot rescue teams.

Concrete examples for future work includes the integration of further 3D data, and extension to the stereo system, as well as the implementation of more autonomous functionalities for the robot. This is especially interesting if the system is applied in outdoor environments. In uneven terrain the use of 3D sensors and 3D visualization will become mandatory. This advance will also require accordant user testing, where it has to be assured that introduction of new data, visualization techniques or autonomous functions do not lower the performance of the supervisor. Related measures, as situation awareness or workload, should also be taken into account as they allow to draw conclusion on the interaction mechanisms behind pure performance measurement.

Bibliography

[1] *Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction*, New York, NY, USA, 2007. ACM.

[2] J. Adams and H. Kaymaz-Keskinpala. Analysis of perceived workload when using a pda for mobile robot teleoperation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 4128–4133, Apr 26-May 1, 2004.

[3] J. Adams and M. Skubic. Introduction to the special issue on humanrobot interaction. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(4):433–437, July 2005.

[4] J. A. Adams. Critical considerations for human-robot interface development. In *2002 AAAI Fall Symposium on Human-Robot Interaction*, November 2002.

[5] J. A. Adams. Human-robot interaction design: Understanding user needs and requirements. In *Proceedings of the 2005 Human Factors and Ergonomics Society 49th Annual Meeting*, September 2005.

[6] J. A. Adams. Unmanned vehicle situation awareness: A path forward. In *Proceedings of the 2007 Human Systems Integration Symposium*, 2007.

[7] A. Agah. Human interactions with intelligent systems: research taxonomy. *Computers & Electrical Engineering*, 27:71–107, 2001.

[8] J. A. Atherton, B. Hardin, and M. A. Goodrich. Coordinating a multi-agent team using a mulitple perspective interface paradigm. In *Proceedings of the AAAI 2006 Spring Syposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, Stanford University, California, March 2006. AAAI.

[9] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

[10] M. Baker, R. Casey, B. Keyes, and H. Yanco. Improved interfaces for human-robot interaction in urban search and rescue. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2960–2965, 10-13 Oct. 2004.

[11] M. Baker and H. Yanco. Autonomy mode suggestions for improving human-robot interaction. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2948–2953, 10-13 Oct. 2004.

[12] L. E. Barnes, R. R. Murphy, and J. D. Craighead. Human-robot coordination using scripts. In *Proceedings of the 2006 SPIE Defense and Security Symposium*, pages 126–135, April 2006.

[13] L. Bartram, C. Ware, and T. Calvert. Moticons: detection, distraction and task. *International Journal of Human-Computer Studies*, 58(5):515–545, May 2003.

[14] C. L. Bethel and R. R. Murphy. Non-facial/non-verbal methods of affective expression as applied to robot-assisted victim assessment. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 287–294, New York, NY, USA, 2007. ACM Press.

[15] C. Bolstad, A. Costello, and M. Endsley. Bad situation awareness designs: What went wrong and why. In *International Ergonomics Association 16th World Congress*, Maastricht, Netherlands, 2006.

[16] C. Bolstad and M. Endsley. Tools for supporting team collaboration. In *Proccedings of the Human Factors and Ergonomics 47th Annual Meeting*, 2003.

[17] D. Bruemmer, R. Boring, D. Few, J. Marble, and M. Walton. "I Call Shotgun!": An evaluation of mixed-initiative control for novice users of a search and rescue robot. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2847–2852, 10-13 Oct. 2004.

[18] D. Bruemmer, J. Marble, and D. Dudenhoeffer. Mutual initiative in human-machine teams. In *Proceedings of the 2002 IEEE 7th Conference on Human Factors and Power Plants*, pages 7–22 – 7–30, 15-19 Sept. 2002.

[19] D. J. Bruemmer, D. A. Few, M. C. Walton, R. L. Boring, J. L. Marble, C. W. Nielsen, and J. Garner. "Turn Off the Television!": Real-world robotic exploration experiments with a virtual 3-d display. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9*, 2005.

[20] G. Burdea. Invited review: the synergy between virtual reality and robotics. *IEEE Transactions on Robotics and Automation*, 15(3):400–410, June 1999.

[21] W. Burgard and D. Fox. *Probabilistic robotics.* MIT Press, 2005.

[22] C. R. Burghart and R. Haeussling. Evaluation criteria for human robot interaction. In *Proceedings of the Symposium on Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, pages 23–31, University of Hertfordshire, Hatfield, UK, April 2005. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

[23] J. Burke and R. Murphy. Situation awareness, team communication, and task performance in robot-assisted technical search: Bujold goes to bridgeport. Technical Report CRASAR-TR2004-23, Department of Computer Science and Engineering, University of South Florida, 2004.

[24] J. Burke and R. Murphy. Rsvp: An investigation of remote shared visual presence as common ground for human-robot teams. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 161–168, New York, NY, USA, 2007. ACM Press.

[25] J. Burke, R. Murphy, M. Coovert, and D. Riddle. Moonlight in miami: A field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Human-Computer Interaction*, 19(1-2):85116, 2004.

[26] J. Burke, R. Murphy, E. Rogers, V. Lumelsky, and J. Scholtz. Final report for the darpa/nsf interdisciplinary study on human-robot interaction. *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):103–112, May 2004.

[27] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of human-computer interaction*. Lawrence Erlbaum Associates, 1983.

[28] J. Casper and R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 33(3):367–385, June 2003.

[29] H. H. Clark and S. E. Brennan. Grounding in communication. In L. B. Resnick, J. M. Levine, and S. D. Teasley, editors, *Perspectives on socially shared cognition*, chapter 7, pages 127–149. American Psychological Association, 1991.

[30] E. Clarkson and R. C. Arkin. Applying heuristic evaluation to human-robot interaction systems. Technical Report GIT-GVU-06-08, Georgia Institute of Technology, 2006.

[31] T. H. Collett, B. A. MacDonald, and B. P. Gerkey. Player 2.0: Toward a practical robot programming framework. In *Proceedings of the Australasian Conference on Robotics and Automation*, Sydney, Australia, December 2005.

[32] T. H. J. Collett and B. A. MacDonald. Developer oriented visualisation of a robot program. In *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 49–56, New York, NY, USA, 2006. ACM.

[33] J. Crandall, M. Goodrich, J. Olsen, D.R., and C. Nielsen. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(4):438–449, July 2005.

[34] J. W. Crandall and M. L. Cummings. Identifying predictive metrics for supervisory control of multiple robots. *IEEE Transactions on Robotics*, 23(5):942–951, 2007.

[35] C. Cruz-Neira, D. Sandin, and T. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 135–142, New York, 1993. ACM Press.

[36] F. Driewer, H. Baier, and K. Schilling. Robot / human interfaces for rescue teams. In *Proceedings IFAC Symposium on Telematics Applications in Automation and Robotics, Helsinki*, 2004.

[37] F. Driewer, H. Baier, and K. Schilling. Robot/human rescue teams: A user requirement analysis. *Advanced Robotics*, 19(8):819–838, 2005.

[38] F. Driewer, H. Baier, K. Schilling, J. Pavlicek, L. Preucil, N. Ruangpayoongsak, H. Roth, J. Saarinen, J. Suomela, A. Halma, M. Kulich, and J. Kout. Hybrid telematic teams for search and rescue operations. In *Proceedings of the SSRR04, IEEE International Workshop on Safety, Security, and Rescue Robotics, Bonn, Germany*, May 2004. ISBN: 3-8167-6556-4.

[39] F. Driewer, M. Sauer, F. Leutert, and K. Schilling. A flexible framework for distributed three-dimensional models in telematic applications. In *Proceedings of the IASTED International Conference on Telematics*, Würzburg, August 2007. IASTED.

[40] F. Driewer, M. Sauer, and K. Schilling. Mixed reality for teleoperation of mobile robots in search and rescue scenarios. In *12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, volume II, pages 255 – 260, St. Etienne (France), May 2006.

[41] F. Driewer, M. Sauer, and K. Schilling. Design and evaluation of a teleoperation interface for heterogeneous human-robot teams. In *10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (HMS)*, Seoul (Korea), September 2007.

[42] F. Driewer, M. Sauer, and K. Schilling. Discussion of challenges for user interfaces in human-robot teams. In *3rd European Conference on Mobile Robots*, Freiburg (Germany), September 2007.

[43] F. Driewer, M. Sauer, and K. Schilling. Design and evaluation of an user interface for the coordination of a group of mobile robots. In *17th International Symposium on Robot and Human Interactive Communication*, pages 237–242, Munich (Germany), August 2008.

[44] F. Driewer, M. Sauer, and K. Schilling. Mensch-roboter interaktion in rettungseinstzen (human-robot interaction in rescue operations). *i-com*, 7(1):5–11, April 2008.

[45] F. Driewer, Schilling, and H. K., Baier. Human-computer interaction in the pelote rescue system. In *IEEE International Workshop on Safety, Security and Rescue Robotics, International Rescue System Institute*, Kobe (Japan), June 6-9 2005.

[46] J. Drury, L. D. Riek, A. D. Christiansen, Z. T. Eyler-Walker, A. J. Maggi, and D. B. Smith. Evaluating human-robot interaction in a search-and-rescue context. In E. Messina and A. Meystel, editors, *Performance Metrics for Intelligent Systems PerMIS '03*, 2003.

[47] J. Drury, J. Scholtz, and H. Yanco. Awareness in human-robot interactions. In *IEEE International Conference on Systems, Man and Cybernetics, 2003*, volume 1, pages 912–918, 5-8 Oct. 2003.

[48] J. L. Drury, B. Keyes, and H. A. Yanco. LASSOing HRI: analyzing situation awareness in map-centric and video-centric interfaces. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 279–286, New York, NY, USA, 2007. ACM Press.

[49] J. L. Drury, J. Scholtz, and D. Kieras. Adapting GOMS to model human-robot interaction. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 41–48, New York, NY, USA, 2007. ACM Press.

[50] J. L. Drury, J. Scholtz, and H. A. Yanco. Applying CSCW and HCI techniques to human-robot interaction. In *Proceedings of the CHI 2004 Workshop on Shaping Human-Robot Interaction*, pages 13–16, Vienna, April 2004.

[51] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, December 1996.

[52] D. Eck and K. S. M. Stahl and. The small outdoor rover merlin and its assistance system for tele-operations. In *Proceedings of the 6th International Conference on Field and Service Robotics*, Chamonix, France, 2007.

[53] C. A. Ellis, S. J. Gibbs, and G. Rein. Groupware: Some issues and experiences. *Commun. ACM*, 34(1):39–58, 1991.

[54] M. Endsley. Design and evaluation of situation awareness enhancement. In *Proceedings of the Human Factors Society 32nd Annual Meeting*, pages 97–101, Santa Monica, CA, 1988. Human Factors Society.

[55] M. Endsley. Direct measurement of situation awareness: Validity and use of SAGAT. In M. R. Endsley and D. J. Garland, editors, *Situation awareness analysis and measurement*, chapter 7, pages 147–174. Lawrence Erlbaum Associates, 2000.

[56] M. Endsley. Designing for situation awareness in complex system. In *Proceedings of the Second intenational workshop on symbiosis of humans, artifacts and environment*, Kyoto, Japan 2001.

[57] M. Endsley and D. Kaber. Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, 42(3):462–492, 1999.

[58] M. R. Endsley. Theoretical underpinnings of situation awareness: A critical review. In D. J. G. Mica R. Endsley, editor, *Situation awareness analysis and measurement*, chapter 1, pages 3–26. Lawrence Erlbaum Associates, 2000.

[59] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, March 1995.

[60] M. R. Endsley, B. Bolte, and D. G. Jones. *Designing for situation awareness: an approach to user-centered design.* London; New York: Taylor Francis, 2003.

[61] M. R. Endsley and W. B. Jones. A model of inter- and intrateam situational awareness: Implications for design, training, and measurement. In M. McNeese, E. Salas, and M. R. Endsley, editors, *New Trends in Cooperative Activities: Understanding System Dynamics in Complex Environments*, pages 46–67. Human Factors & Ergonomics Society, Santa Monica, CA, 2001.

[62] I. Envarli and J. Adams. Task lists for human-multiple robot interaction. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 119– 124. IEEE, August 2005.

[63] J. Ferketic, L. Goldblatt, E. Hodgson, S. Murray, R. Wichowski, A. Bradley, T. W. Fong, J. Evans, W. Chun, R. Stiles, M. Goodrich, and A. Steinfeld. Toward human-robot interface standards I: Use of standardization and intelligent subsystems for advancing human-robotic competency in space exploration. In *SAE 36th International Conference on Environmental Systems*, July 2006.

[64] J. Ferketic, L. Goldblatt, E. Hodgson, S. Murray, R. Wichowski, A. Bradley, T. W. Fong, J. Evans, W. Chun, R. Stiles, M. A. Goodrich, A. Steinfeld, D. King, and C. Erkorkmaz. Toward human-robot interface standards II: An examination of common elements in human-robot interaction across the space enterprise. In *AIAA Space 2006*, September 2006.

[65] D. Few, D. Bruemmer, and M. Walton. Improved human-robot teaming through facilitated initiative. In *The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 171–176, Sept. 2006.

[66] T. Fong, C. Thorpe, and C. Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 50(4):699–704, Aug. 2003.

[67] T. W. Fong, C. Kunz, L. Hiatt, and M. Bugajska. The human-robot interaction operating system. In *2006 Human-Robot Interaction Conference*. ACM, March 2006.

[68] T. W. Fong, I. Nourbakhsh, R. Ambrose, R. Simmons, A. Schultz, and J. Scholtz. The peer-to-peer human-robot interaction project. In *AIAA Space 2005*, September 2005. AIAA-2005-6750.

[69] T. W. Fong, J. Scholtz, J. Shah, L. Flueckiger, C. Kunz, D. Lees, J. Schreiner, M. Siegel, L. Hiatt, I. Nourbakhsh, R. Simmons, R. Ambrose, R. Burridge, B. Antonishek, M. Bugajska, A. Schultz, and J. G. Trafton. A preliminary study of peer-to-peer human-robot interaction. In *International Conference on Systems, Man, and Cybernetics*. IEEE, October 2006.

[70] T. W. Fong and C. Thorpe. Vehicle teleoperation interfaces. *Autonomous Robots*, 11(1):09–18, July 2001.

[71] T. W. Fong, C. Thorpe, and C. Baur. Collaboration, dialogue, and human-robot interaction. In *Proceedings of the 10th International Symposium of Robotics Research, Lorne, Victoria, Australia*, London, November 2001. Springer-Verlag.

[72] M. Frank, S. Busch, P. Dietz, and K. Schilling. Teleoperations of a mobile outdoor robot with adjustable autonomy. In *Proceedings 2nd ACIDCA-ICMI'2005, International Conference on Machine Intelligence*, Tozeur (Tunisia), 2005.

[73] B. Fransen, V. Morariu, E. Martinson, S. Blisard, M. Marge, S. Thomas, A. Schultz, and D. Perzanowski. Using vision, acoustics, and natural language for disambiguation. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 73–80, New York, NY, USA, 2007. ACM.

[74] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, and J. González. Multi-hierarchical semantic maps for mobile robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3492–3497, Edmonton, CA, 2005.

[75] 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 11th International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, June 2003.

[76] M. Goodrich, D. Olsen, J. Crandall, and T. Palmer. Experiments in adjustable autonomy. In *Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, 2001.

[77] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall. Managing autonomy in robot teams: observations from four experiments. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 25–32, New York, NY, USA, 2007. ACM Press.

[78] M. A. Goodrich and D. R. Olsen. Seven principles of efficient interaction. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pages 3943–3948, 2003.

[79] M. A. Goodrich and A. C. Schultz. Human-robot interaction: a survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[80] A. Halme. Common situation awareness as basis for human-robot interface. In M. Tokhi, G. Virk, and M. Hossain, editors, *Climbing and Walking Robots*, pages 3–18. Springer Berlin Heidelberg, 2006.

[81] S. Hart and L. Staveland. Development of NASA-TLX (task load index): Results of empirical and theorectical research. In P. Hancock and N. Meshkati, editors, *Human Mental Workload*. Elsevier, Amsterdam, The Netherlands, 1922.

[82] M. Hess, M. Saska, and K. Schilling. Formation driving using partical swarm optimization and reactive obstacle avoidance. In *Proceedings of the 1st IFAC Workshop on Multivehicle Systems (MVS06)*, Salvador (Brazil), October 2006.

[83] T. T. Hewet, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, and W. Verplank. Acm sigchi curricula for human-computer interaction. Technical report, ACM Press, New York, NY, USA, 1992.

[84] G. Hoffman and C. Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 1–8, New York, NY, USA, 2007. ACM Press.

[85] C. M. Humphrey, S. M. Gordon, and J. A. Adams. Visualization of multiple robots during team activities. In *Proceedings of the 2006 Human Factors and Ergonomics Society 50th Annual Meeting*, San Francisco, CA, 2006.

[86] C. M. Humphrey, C. Henk, G. Sewell, B. W. Williams, and J. A. Adams. Assessing the scalability of a multiple robot interface. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 239–246, New York, NY, USA, 2007. ACM Press.

[87] A. Jacoff, E. Messina, B. Weiss, S. Tadokoro, and Y. Nakagawa. Test arenas and performance metrics for urban search and rescue robots. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 3396–3403vol.3, 27-31 Oct. 2003.

[88] C. Johnson, J. Adams, and K. Kawamura. Evaluation of an enhanced human-robot interface. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 900–905vol.1, 5-8 Oct. 2003.

[89] E. Jones, B. Browning, M. Dias, B. Argall, M. Veloso, and A. Stentz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, pages 570–575, May 2006.

[90] H. Jones and P. Hinds. Extreme work teams: Using SWAT teams as a model for coordinating distributed robots. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 372–381, New York, NY, USA, 2002. ACM Press.

[91] M. W. Kadous, R. K.-M. Sheh, and C. Sammut. Effective user interface design for rescue robotics. In *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 250–257, New York, NY, USA, 2006. ACM Press.

[92] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top AR environment. In *Proceedings of the International Symposium on Augmented Reality*, pages 111–119, 2000.

[93] H. Keskinpala, J. Adams, and K. Kawamura. PDA-based human-robotic interface. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3931–3936, 5-8 Oct. 2003.

[94] C. D. Kidd and C. Breazeal. Human-robot interaction experiments: Lessons learned. In *Proceedings of the Symposium on Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, pages 141–142, University of Hertfordshire, Hatfield, UK, April 2005. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

[95] S. Kiesler. Fostering common ground in human-robot interaction. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 729–734, 13-15 Aug. 2005.

[96] T. Kim and P. Hinds. Who Should I Blame? Effects of autonomy and transparency on attributions in human-robot interaction. In *The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 80–85, 2006.

[97] J. Kramer and M. Scheutz. Robotic development environments for autonomous mobile robots: A survey. *Autonomous Robots*, 22(2):101–132, 2007.

[98] M. Kulich, J. Faigl, and L. Preucil. Cooperative planning for heterogeneous teams in rescue operations. In *IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 230–235, June 2005.

[99] M. Kulich, J. Kout, L. Preucil, R. Mazl, J. Chudoba, J. Saarinen, J. Suomela, A. Halme, F. Driewer, H. Baier, K. Schilling, N. Ruangpayoongsak, and H. Roth. PeLoTe a heterogeneous telematic system for cooperative search and rescue missions. In *Urban search and rescue: from Robocup to real world applications, in conjunction with the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai (Japan), September 28 2004.

[100] A. Lankenau. Avoiding mode confusion in service-robots. In M. Mokhtari, editor, *Integration of Assistive Technology in the Information Age, Proceedings of the 7th International Conferece on Rehabilitation Robotics*, page 162 167. IOS Press, 2001.

[101] F. Laschi, C. Breazeal, and C. Nakauchi. Guest editorial special issue on human-robot interaction. *IEEE Transactions on Robotics*, 23(5):837–839, October 2007.

[102] J. L. Marble, D. J. Bruemmer, D. A. Few, and D. D. Dudenhoeffer. Evaluation of supervisory vs. peer-peer interaction with human-robot teams. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 5*, page 50130b, 2004.

[103] J. L. Marble, D. A. Few, and D. J. Bruemmer. I want what you've got: Cross platform usability and human-robot interaction assessment. In E. Messina, editor, *Proceedings of PerMIS'04*, 2004.

[104] B. A. Maxwell, N. Ward, and F. Heckel. Game-based design of human-robot interfaces for urban search and rescue. In *Conference on Human Factors in Computing Systems - CHI 2004 Fringe*, 2004.

[105] O. Mayora, L. Sucar, S. Aviles, and C. Miranda-Palma. From HCI to HRI - usability inspection in multimodal human - robot interactions. In *Proceedings of the 12th IEEE International Workshop on Robot and Human Interactive Communication*, pages 37–41, 31 Oct.-2 Nov. 2003.

[106] R. Mazl, J. Pavlicek, and L. Preucil. Structures for data sharing in hybrid rescue teams. In *Proceedings IEEE International Workshop on Safety, Security, and Rescue Robotics*, pages 236 – 241, International Rescue System Institute, Kobe, Japan, June 2005.

[107] D. S. McCrickard, M. Czerwinski, and L. Bartram. Introduction: design and evaluation of notification user interfaces. *International Journal of Human-Computer Studies*, 58(5):509–514, May 2003.

[108] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D(12):1321–1329, December 1994.

[109] P. Milgram, S. Yin, and J. Grodski. An augmented reality based teleoperation interface for unstructured environments. In *ANS 7th Topical Meeting on Robotics and Remote Systems*, pages 966–973, Augusta, Georgia, USA, May 1997.

[110] P. Milgram, S. Zhai, D. Drascic, and J. Grodski. Applications of augmented reality for human-robot communication. In *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 3, pages 1467–1472, July 1993.

[111] C. A. Miller and R. Parasuraman. Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(1):57–75, February 2007.

[112] R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3):338–348, 1990.

[113] T. Möller, H. Kraft, J. Frey, M. Albrecht, and R. Lange. Robust 3D measurement with PMD sensors. In *Range Imaging Day*, Zrich, 2005.

[114] O. M. Mozos, P. Jensfelt, H. Zender, G.-J. M. Kruijff, and W. Burgard. From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *Proceedings of the IEEE/RSJ IROS 2007 Workshop: Semantic information in robotics*, San Diego, CA, USA, 2007.

[115] R. Murphy and J. Burke. Up from the rubble: Lessons learned about HRI from search and rescue. In *Proceedings of the 49th Annual Meetings of the Human Factors and Ergonomics Society*, 2005.

[116] C. Nielsen, D. Bruemmer, D. Few, and M. Walton. Mixed-initiative interactions for mobile robot search. In *American Association for Artificial Intelligence Mobile Robot Workshop*, Boston, Massachusetts, July 2006.

[117] C. Nielsen, M. Goodrich, and R. Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics*, 23(5):927–941, October 2007.

[118] C. Nielsen, M. Goodrich, and R. Rupper. Towards facilitating the use of a pan-tilt camera on a mobile robot. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 568–573, 13-15 Aug. 2005.

[119] C. Nielsen, B. Ricks, M. Goodrich, D. Bruemmer, D. Few, and M. Few. Snapshots for semantic maps. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2853–2858, 10-13 Oct. 2004.

[120] 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, New York, NY, USA, 2006. ACM Press.

[121] C. W. Nielsen and M. A. Goodrich. Testing the usefulness of a pan-tilt-zoom (ptz) camera in human-robot interactions. In *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*, volume 50, pages 757–761, 2006.

[122] J. Nielsen. Finding usability problems through heuristic evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 373–380, New York, NY, USA, 1992. ACM.

[123] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, San Francisco, 1993.

[124] J. Nielsen. Enhancing the explanatory power of usability heuristics. In *CHI '94: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 152–158, New York, NY, USA, 1994. ACM.

[125] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA, 1990. ACM.

[126] I. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion. Human-robot teaming for search and rescue. *IEEE Pervasive Computing: Mobile and Ubiquitous Systems*, 4(1):72–78, January 2005.

[127] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM-3D mapping outdoor environments: Research articles. *Journal of Field Robotics*, 24(8-9):699–722, 2007.

[128] R. Nunez, J. Bandera, J. Perez-Lorenzo, and F. Sandoval. A human-robot interaction system for navigation supervision based on augmented reality. In *IEEE Mediterranean Electrotechnical Conference*, pages 441–444, 16-19 May 2006.

[129] D. R. Olsen and M. A. Goodrich. Metrics for evaluating human-robot interactions. In *Proceedings of PERMIS 2003*, 2003.

[130] A. Oomes. Organization awareness in crisis management. In *Proceedings of the First International Conference on Information Systems for Crisis Response and Management (ISCRAM 2004)*, 2004.

[131] R. Parasuraman, T. Sheridan, and C. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 30(3):286–297, 2000.

[132] S. P. Parikh, V. Grassi Jr., V. Kumar, and J. Okamoto Jr. Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform. *IEEE Intelligent Systems*, 22(2):33–41, 2007.

[133] L. E. Parker. ALLIANCE: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, April 1998.

[134] J. Pavlicek, R. Mazl, L. Preucil, F. Driewer, and K. Schilling. Experiencing new capabilities of humans in hybrid telematic teams. In *IEEE Biomedical Robotics and Biomechatronics BioRob2006*, February 20-22 2006. ISBN 1-4244-0040-6.

[135] A. Powers. What robotics can learn from HCI. *interactions*, 15(2):67–69, 2008.

[136] R. W. Proud, J. J. Hart, and R. B. Mrozinski. Methods for determining the level of autonomy to design into a human spaceflight vehicle: A function specific approach. In E. Messina and A. Meystel, editors, *Performance Metrics for Intelligent Systems PerMIS '03*, September 2003.

[137] J. Reitsema, W. Chun, T. W. Fong, and R. Stiles. Team-centered virtual interactive presence for adjustable autonomy. In *AIAA Space 2005*, September 2005. AIAA-2005-6606.

[138] B. Ricks, C. Nielsen, and M. Goodrich. Ecological displays for robot interaction: A new perspective. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2855–2860, 28 Sept.-2 Oct. 2004.

[139] J. Riley and M. Endsley. The hunt for situation awareness: Human-robot interaction in search and rescue. In *Proceeding of the Human Factors & Ergonomics Society*, 2004.

[140] T. Ringbeck and B. Hagebeuker. A 3D time of flight camera for object detection. In *Optical 3-D Measurement Techniques*, Zrich, 2007.

[141] N. Ruangpayoongsak, H. Roth, and J. Chudoba. Mobile robots for search and rescue. In *IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 212 – 217, International Rescue System Institute, Kobe, Japan, June 2005.

[142] J. Saarinen, S. Heikkil, M. Elomaa, J. Suomela, and A. Halme. Rescue personnel localization system. In *Proceedings IEEE International Workshop on Safety, Security, and Rescue Robotics*, pages 218 – 223, International Rescue System Institute, Kobe, Japan, June 2005.

[143] J. Saarinen, A. Maula, R. Nissinen, H. Kukkonen, J. Suomela, and A. Halme. Gimnet - infrastructure for distributed control of generic intelligent machines. In *Proceedings of the IASTED International Conference on Telematics*, Würzburg, August 2007. IASTED.

[144] A. D. Santis, B. Siciliano, A. Luca, and A. Bicchi. An atlas of physical human-robot interaction. *Mechanism and Machine Theory*, 43(3):253–270, March 2008.

[145] M. Sauer, F. Driewer, K. E. Missoh, M. Gllitz, and K. Schilling. Approaches to mixed reality user interfaces for teleoperation of mobile robots. In K. Schilling, editor, *Proceedings of the 13th IASTED International Conference on Robotics and Applications (RA 2007)*, August 2007.

[146] M. Sauer, F. Driewer, and K. Schilling. Potential and challenges of stereo augmented reality for mobile robot teleoperation. In *10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*, Seoul (Korea), September 2007.

[147] M. Sauer, D. Eck, and M. Schmidt. Augmented reality user interface for controlling a mobile robot with onboard camera. In *Mechatronics & Robotics Conference, Aachen*, September 2004.

[148] K. Schilling, J. de Lafontaine, and H. Roth. Autonomy capabilities of european deep space probes. *Autonomous Robots*, 3(1):19–30, March 1996.

[149] K. Schilling and F. Driewer. Remote control of mobile robots for emergencies. In *16th IFAC World Congress, Prague, Czech Republic*, 2005.

[150] K. Schilling, F. Driewer, and H. Baier. User interfaces for robots in rescue operations. In *Proceedings IFAC/IFIP/IFORS/IEA Symposium Analysis, Design and Evaluation of Human-Machine Systems*, Atlanta (USA), 2004.

[151] K. Schilling and Q. Meng. The merlin vehicles for outdoor applications. In *SPIE conference proceedings Unmanned Ground Vehicle Technology IV*, Orlando, 2002.

[152] K. Schilling and L. Preucil. Tele-presence methods supporting cooperation of robots and humans. In *IEEE International Workshop on Safety, Security and Rescue Robotics, International Rescue System Institute*, Kobe (Japan), June 6-9 2005.

[153] J. Scholtz. Evaluation methods for human-system performance of intelligent systems. In E. Messina and A. Meystel, editors, *Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 PerMIS Workshop*. NIST Special Publication, August 2002.

[154] J. Scholtz. Theory and evaluation of human robot interactions. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 6-9 Jan 2003.

[155] J. Scholtz, B. Antonishek, and J. Young. Evaluation of a human-robot interface: Development of a situational awareness methodology. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, 5-8 Jan. 2004.

[156] J. Scholtz and S. Bahrami. Human-robot interaction: Development of an evaluation methodology for the bystander role of interaction. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3212–3217, 5-8 Oct. 2003.

[157] J. Scholtz, J. Young, J. Drury, and H. Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2327–2332, Apr 26-May 1, 2004.

[158] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control.* The MIT Press, 1992.

[159] T. B. Sheridan and R. Parasuraman. Human-automation interaction. In R. Nickerson, editor, *Reviews of Human Factors and Ergonomics*, volume 1, pages 89–129(41). Santa Monica: Human Factors and Ergonomics Society, 2006.

[160] M. Shiomi, T. Kanda, H. Ishiguro, and N. Hagita. Interactive humanoid robots for a science museum. *IEEE Intelligent Systems*, 22(2):25–32, 2007.

[161] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human Computer Interaction.* Addison-Wesley, Reading, Mass., 2nd edition, 1992.

[162] B. Shneiderman. Why not make interfaces better than 3D reality? *IEEE Computer Graphics and Applications*, 23(6):12–15, 2003.

[163] B. Shneiderman. Human responsibility for autonomous agents. *IEEE Intelligent Systems*, 22(2):60–61, 2007.

[164] C. Skourup, T. Pettersen, and J. Pretlove. Augmented reality system for robot programming. In *Proceedings IFAC/IFIP/IFORS/IEA Symposium Analysis, Design and Evaluation of Human-Machine Systems*, Atlanta (USA), 2004.

[165] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz. Using a qualitative sketch to control a team of robots. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pages 3595–3601, May 15-19, 2006.

[166] N. Stanton, P. M. Salmon, G. H. Walker, C. Baber, and D. P. Jenkins. *Human Factors Methods: A Practical Guide for Engineering and Design.* Ashgate Publishing, Ltd., 2005.

[167] F. Steele, Jr. and G. Thomas. Directed stigmergy-based control for multi-robot systems. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 223–230, New York, NY, USA, 2007. ACM Press.

[168] A. Steinfeld. Interface lessons for fully and semi-autonomous mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[169] A. Steinfeld, T. W. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich. Common metrics for human-robot interaction. In *2006 Human-Robot Interaction Conference*. ACM, March 2006.

[170] K. Stubbs, P. J. Hinds, and D. Wettergreen. Autonomy and common ground in human-robot interaction: A field study. *IEEE Intelligent Systems*, 22(2):42–50, 2007.

[171] M. Sugimoto, G. Kagotani, H. Nii, N. Shiroma, M. Inami, and F. Matsuno. Time follower's vision: A teleoperation interface with past images. *IEEE Computer Graphics and Applications*, 25(1):54–63, 2005.

[172] J. Suomela, J. Saarinen, and A. Halme. Creating common presence for a multientity rescue team. In *Proceedings of the 16th IFAC World Conference*, July 2005.

[173] J. Suomela, J. Saarinen, A. Halme, and P. Harmo. Online interactive building of presence. *Field and Service Robotics*, 24:395–404, 2006.

[174] J. Trafton, N. Cassimatis, M. Bugajska, D. Brock, F. Mintz, and A. Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(4):460–470, July 2005.

[175] D. Wagner and D. Schmalstieg. ARToolKitPlus for pose tracking on mobile devices. In M. Grabner and H. Grabner, editors, *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, February 2007.

[176] M. L. Walters, S. Woods, K. L. Koay, and K. Dautenhahn. Practical and methodological challenges in designing and conducting human-robot interaction studies. In *Proceedings of the Symposium on Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, pages 110–120, University of Hertfordshire, Hatfield, UK, April 2005. The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

[177] J. Wang and M. Lewis. Human control for cooperating robot teams. In *HRI '07: Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 9–16, New York, NY, USA, 2007. ACM Press.

[178] J. Wang, M. Lewis, S. Hughes, M. Koes, and S. Carpin. Validating USARsim for use in HRI research. In *Proceedings of the 49th Annual Meeting of the Human Factors and Ergonomics Society*, September 2005.

[179] D. Watson, L. A. Clark, and A. Tellegen. Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology*, 54(6):1063–1070, June 1988.

[180] M. Wiedemann, M. Sauer, F. Driewer, and K. Schilling. Analysis and characterization of the PMD camera for application in mobile robotics. In *Proceedings of the 17th IFAC World Congress*, 2008.

[181] H. Yanco, J. Drury, and J. Scholtz. Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Human-Computer Interaction*, 19(1-2):117–149, 2004.

[182] H. A. Yanco, M. Baker, R. Casey, B. Keyes, P. Thoren, J. L. Drury, D. A. Few, C. Nielsen, and D. J. Bruemmer. Analysis of human-robot interaction for urban search and rescue. In *In Proceedings of the IEEE International Workshop on Safety, Security, and Rescue Robotics*, Gaithersburg, MD, August 22-24 2006.

[183] H. A. Yanco and J. Drury. Classifying human-robot interaction: An updated taxonomy. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, October 2004.

[184] H. A. Yanco and J. Drury. "Where Am I?" acquiring situation awareness using a remote robot platform. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, October 2004.

[185] H. A. Yanco and J. L. Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, pages 111–119, Falmouth, Massachusetts, November 2002. AAAI.

[186] H. A. Yanco and J. L. Drury. Guest editors' introduction: Interacting with autonomy. *IEEE Intelligent Systems*, 22(2):16–17, 2007.

[187] H. A. Yanco and J. L. Drury. Rescuing interfaces: A multi-year study of human-robot interaction at the AAAI robot rescue competition. *Autonomous Robots*, 22(4):333–352, May 2007.

[188] J. E. Young, M. Xin, and E. Sharlin. Robot expressionism through cartooning. In *Proceeding of the ACM/IEEE international conference on Human-robot interaction*, pages 309–316, New York, NY, USA, 2007. ACM Press.

[189] F. Zeiger, N. Krämer, M. Sauer, and K. Schilling. Challenges in realizing ad-hoc networks based on wireless lan with mobile robots. In *Proceeding of Workshop on Wireless Multihop Communications in Networked Robotics*, Berlin, Germany, April 2008.