DR 6.1: Multiple asynchronous sorties to assess a large-scale static disaster area

Rainer Worst[*], Hartmut Surmann[*], Erik Zimmermann[*], Timea Bagosi[†], Tomáš Svoboda[‡], Michael Achtelik[§] and the TRADR consortium

[*]*Fraunhofer IAIS, Sankt Augustin, Germany*

[†]*Technische Universiteit Delft (TUD), The Netherlands*

[‡]*CTU in Prague*

[§]*Ascending Technologies GmbH, Krailing, Germany*

⟨rainer.worst@iais.fraunhofer.de⟩

Project, project Id:	EU FP7 TRADR / ICT-60963
Project start date:	Nov 1 2013 (50 months)
Due date of deliverable:	December 2014
Actual submission date:	February 2015
Lead partner:	Fraunhofer
Revision:	final
Dissemination level:	PU

This document describes the setup of the technical system framework and the integration of the initial prototype of the TRADR system.

Figure 1: UGV (left) and UAV (right) in action at hospital site in Pisa

Executive Summary

This report describes the final result of WP6 in Yr1 – an integrated cognitive robot system with specific capabilities. To achieve this goal, two tasks were addressed:

Task T6.1 Technical system framework

Task T6.2 Multiple asynchronous sorties to assess a large-scale static disaster area

To create the **technical system framework**, we had to consider a large amount of code and experiences from the finished NIFTi project (`http://www.nifti.eu`). Due to the overall success of NIFTi, we re-used significant parts of the framework in TRADR, but we also used the opportunity to make the necessary changes according to our previous experiences. While we continued to use ROS as a middleware system for hardware abstraction and the implementation of the low- to medium-level parts, we decided to replace CAST, which had been used for the decomposition of the high-level parts in NIFTi, by an agent-based approach. The software development environment was updated by replacing SVN with GIT as the software repository and TRAC with Redmine for issue-tracking. Regarding the hardware components, the ground robots have been maintained and upgraded; the stock of aerial robots has been enriched by commercial platforms provided by the partner ASC. All developing partners of TRADR expressed their commitment to the chosen development environment and middleware-framework.

The second task was to integrate a system for the project-wide scenario of Yr1 **Multiple asynchronous sorties to assess a large-scale static disaster area**. Therefore, the functionality developed by WPs 1-5 was merged to an operable system for human-robot team operations. This integrated system was based on the user needs analysis and was used for the scenario-based evaluation, both performed by WP7 (see Figure 1).

Role of system framework and integration in TRADR

In Yr1, we have updated the system framework and migrated selected components from NIFTi to TRADR. During the next years, we will create at least three integrated robotic systems according to the scenario-based roadmap of TRADR. They will emphasize the issues of dynamic environments, multi-robot task adaptation, and persistence in long-term human-robot teaming. The system's evolution process is guided by applying user-centric engineering practices, which play a central role in TRADR.

Contribution to the TRADR scenarios and prototypes

The results presented in this report contribute to the TRADR project by providing an initial prototype of an integrated cognitive robot system, based on the technical framework that was initially developed in the NIFTi project and improved during the first months of TRADR. It enables us to collect practical experiences and to improve the system architecture. Future work of all other WPs benefits from the experiences gained by the assessment of its technical capabilities during the evaluation with end-users.

1 Tasks, objectives, results

1.1 Planned work

The tasks addressed by WP6 in Yr1 were:

Task T6.1 Technical system framework

Task T6.2 Multiple asynchronous sorties to assess a large-scale static disaster area

These tasks contributed accordingly to the overall objectives of WP6:

A Specify and set up the TRADR technical system framework

C Integrate WP components continuously into a single architecture

The first result to be achieved was a working technical system framework, including hardware components like ground and aerial robots as well as a software development environment with all necessary tools. A milestone was set at month 8 for the commitment of all devoping partners to the chosen development environment and middleware-framework for TRADR.

The second result to be achieved was an integrated system for human-robot team operation, in particular for the project-wide scenario of Yr1 "Multiple asynchronous sorties to assess a large-scale static disaster area". Therefore, the functionality developed by WPs 1-5 had to be merged to an operable system for human-robot team operations. This integrated system was based on the user needs analysis and it was used for the scenario-based evaluation, both performed by WP7.

1.2 Actual work performed

This description starts with an introduction to the TRADR system architecture in section 1.2.1, which is the result of a continuous discussion between all partners in Yr1. In section 1.2.2, we present the tools used for the integrated software development in TRADR. Section 1.2.3 is about the migration process from NIFTi to TRADR. The last three sections 1.2.4, 1.2.5, and 1.2.6 provide details about aspects that are particularly interesting from the integration point of view, namely user interfaces, data storage and retrieval, and network infrastructure.

A comprehensive overview of the finished NIFTi project is given in [32] (Annex Overview 2.1), the TRADR project is presented in [33] (Annex Overview 2.2), and the user-centric development paradigm used in both projects is summarized in [29] (Annex Overview 2.3).

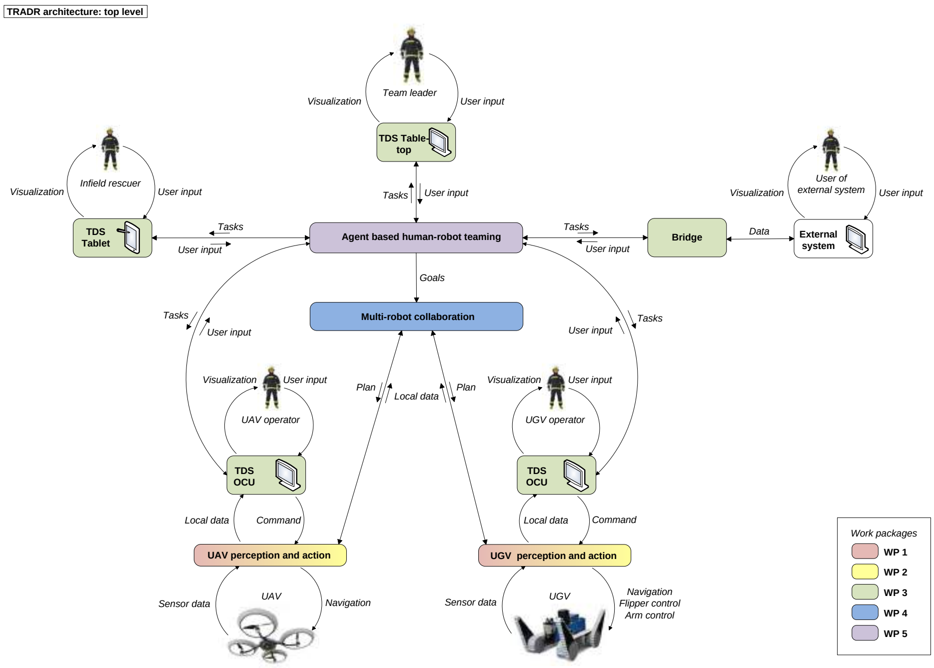


Figure 2: Top-level system architecture

1.2.1 Principles of the TRADR system architecture

The TRADR system enables humans and robots to work as a team, exchange information and operate together to accomplish complex rescue scenarios.

It is built in an incremental way by improving particular components according to the needs of the end-users as specified, e.g., during the annual end-user evaluation. This bottom-up approach is kept together by a system framework that is based on the common understanding of the project partners regarding the system architecture (see Figure 2).

Different aspects of the system architecture are described in the remainder of this section.

Team-members and agents A TRADR team consists of human and robotic team-members, which collaborate during a search and rescue mission.

Agents are the software representation of all users and robots inside the TRADR system with an 1:1 relation to each. All agents can communicate with each other and exchange data. They orchestrate information flow, manage the TRADR Display System (TDS) and coordinate high-level robot control. For a more extensive description of the role of agents in the TRADR architecture, see the annex "Agent architecture" from TRADR deliverable DR5.1.

Human team-members The human team-members can play different roles like team-leader, operator or in-field rescuer.

Team-leader The team-leader manages the entire mission. He uses a table-top as his main device for interaction, which visualizes the situation on the one hand and accepts user input on the other hand.

Operator The operator controls one or more robots. He uses an Operator Control Unit (OCU) to get the necessary information about the situation and the controlled robot(s). The actual control is achieved either by a GUI, by a dedicated control device, or by speech recognition.

In-field Rescuer The in-field rescuer is acting within the emergency scenario. He uses a tablet computer for the visualization of the situation and for his user input, which includes in particular to take pictures of points of interest.

Robotic team-members There are two different kinds of robots involved in a TRADR mission: Unmanned Ground Vehicles (UGV) and Unmanned Aerial Vehicles (UAV). These robots can be equipped with different types of sensors like RGB-camera, laser scanner, stereo-camera, thermo-camera etc.

UGV Each UGV is either tele-operated through the OCU or working autonomously. The operational mode is adapted during the sortie according to the needs. A classical Perception-Planning-Action loop is running on-board of the UGV for the autonomous operation, including low-level path planning, obstacle avoidance, flipper and arm control based on point clouds and image data.

UAV Each UAV is either tele-operated through the OCU or working automatically. The operational mode is adapted during the sortie according to the needs; according to the current state of the art, automatic flight is only possible under favourable conditions. A classical Perception-Planning-Action loop is running on-board of the UAV for the automatic operation, including low-level 3D path planning and obstacle avoidance based on point clouds and image data.

We prefer the term automatic instead of autonomous if we talk about UAVs, because autonomous flights are not allowed based on ICAO rules. This is justified, because there is always a human pilot in the loop being able to intervene during the operation. Details on the control of UGVs and UAVs are presented in DR2.1 of WP2.

Perception, planning, action To explore unknown or known environment, a closed loop between comprehensive perception, individual and collaborative planning, and reasonable acting is inevitable.

Perception Perception takes place in all different levels of the TRADR system, leading from the sensor data on the robotic level to local data (egocentric map) of a single robot, then to global data (allocentric map) created by merging the local data within one sortie, and finally to a world model (environmental 3D details, traversability, points-of-interest, no-go-areas, positions of robots, etc.) on the mission level by collecting the global data of all sorties. Details on this part are presented in DR1.1 of WP1.

Planning Planning is needed on-board of each robot for the autonomous resp. automatic operation. The goals for this planning can either be given by an operator or by a superior planning component that deals with multi-robot collaboration. The multi-robot collaboration on the other hand receives its goals by an agent-based human-robot teaming component at a higher level. Details on this part are presented in DR3.1 of WP3, DR4.1 of WP4, and DR5.1 of WP5.

Action Finally, all (sequences of) actions that are planned by the different planning components must lead to atomic actions, i.e., actions that actually affect the actuators of the robot. Details on this part are presented in DR2.1 of WP2.

Human-Robot Interface (TDS) In general, human team-members use interface devices (table-top, OCU, tablet computer) for the interaction with other team members incl. the robots.

In particular, the OCU offers the capability to control a specific robot (UGV or UAV) directly. This can either be done by assigning goals to the local planner of the robot or by classical tele-operation. In case of the UAV, there is always an additional pilot in the loop, who keeps the robot in line of sight and is able to intervene.

Agents are responsible to accept user commands and to process data to be visualized on display devices (see Figure 3).

Persistence (Database) TRADR has the objective to store the local and global data collected during the different sorties within one mission.

These snapshots of the world model at different points in time can be used during the mission by the team-members, mainly to improve their planning activities.

After the mission, the collected data provide a detailed documentation of the entire event. The structure of these data is a crucial feature of the whole

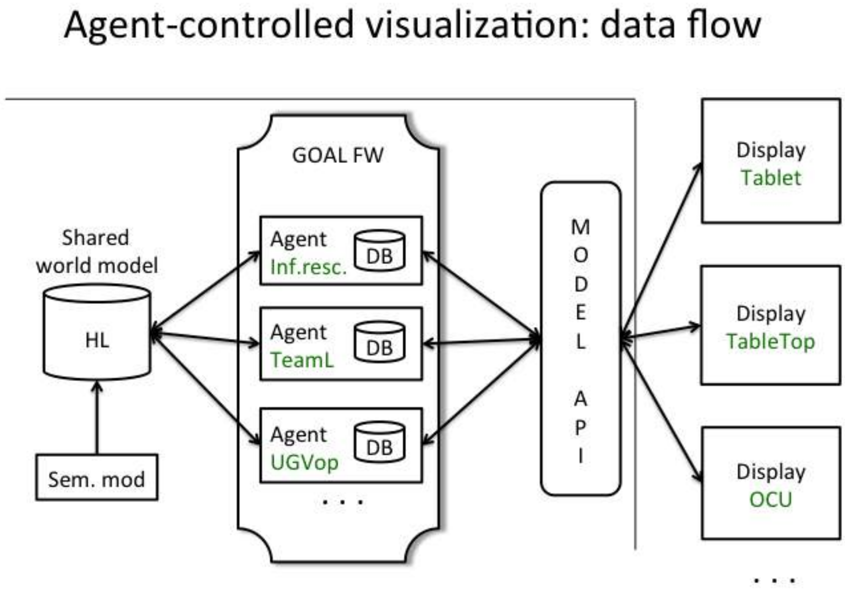


Figure 3: Basic structure of TDS

system and we will elaborate it in detail in the following years of TRADR. Currently, we have designed five data layers, in which the corresponding data types are stored (see Figure 4). Details of the prototypical implementation are described in section 1.2.5. First experiments have been started during TJEx, where a map – considered as a basic world model – created by a robot, was stored and reloaded between different missions as described in DR1.1 of WP1.

Communication Communication between the team members is achieved indirectly by shared memories (e.g., within the database), but also directly by message exchange (see Figure 5). This is based on LAN, using wired connections where possible. However, the mobile robots and the infield-rescuers must use wireless connections with all their inherent drawbacks.

Regarding the robots, they use the ROS-based publisher-subscribe pattern to send and receive messages.

The human team-members are either talking directly to each other (with or without any communication device, e.g., walkie-talkie) or they use their dedicated interface devices.

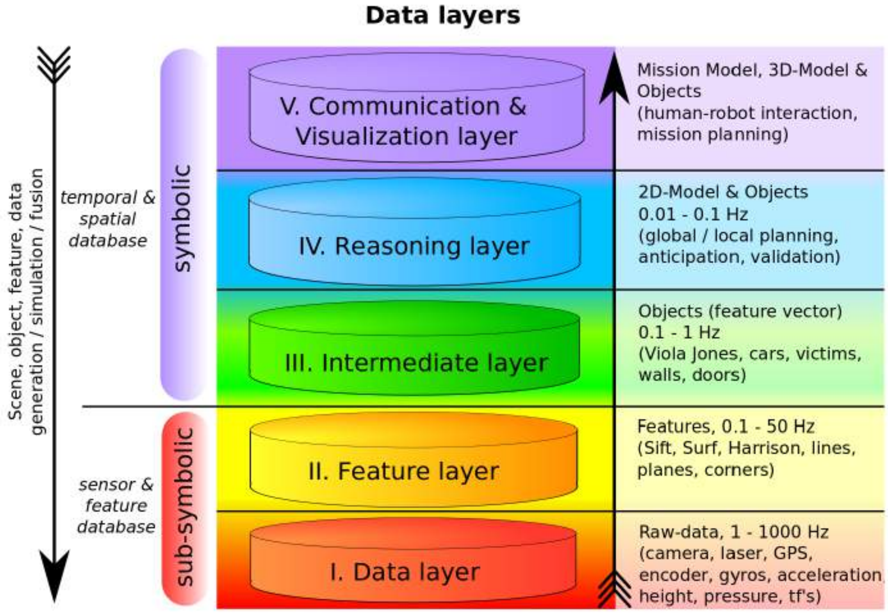


Figure 4: Basic DB-layers

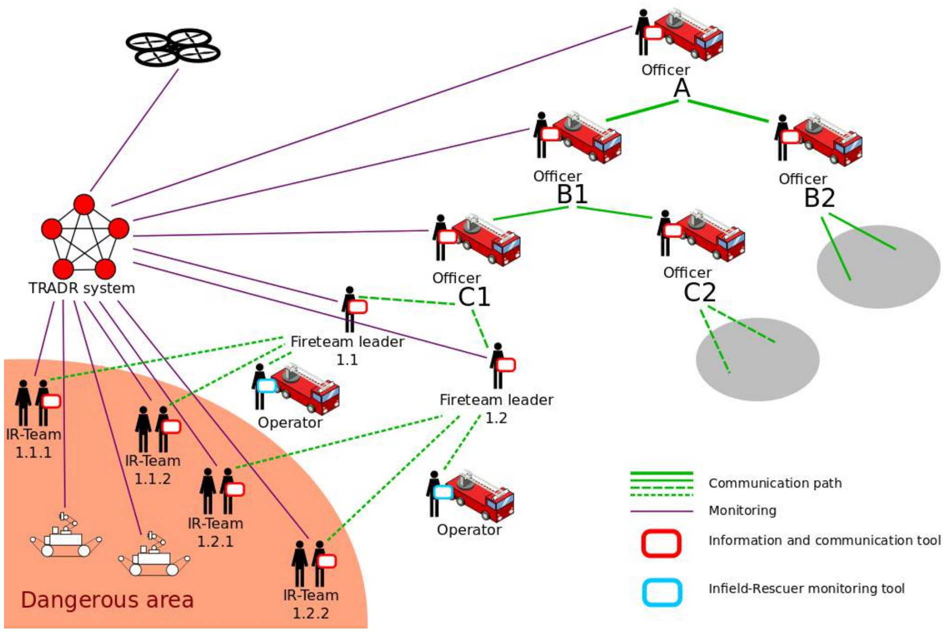


Figure 5: Communication structure

1.2.2 Means for system integration

As TRADR is a project with diverse partners from different countries it is vital to ensure that all parties have a common understanding of the system as a whole as well as the parts in it.

To achieve this goal, TRADR had a kickoff meeting at the beginning of 2014, where all partners presented their planned contributions in the various work packages and how they are related to or build upon other work packages. Furthermore, regular Skype meetings and conventions, which are listed in detail in the Periodic Report Year 1 (DR9.1), ensure a continuous exchange of current work package statuses, changes and decisions of all partners.

The software being developed in TRADR is composed of the parts provided by the partners. To integrate all these components into one system a clear definition of the boundaries between the parts is vital as well as which functionalities can be found where. Thus a clear definition and knowledge of the interfaces and data formats of the subsystems is crucial for a proper system integration.

To simply access all tools used in TRADR like GIT, GITLAB, Jenkins, Redmine, and SVN, the same SHA/MD5 login keys are used system wide.

ROS As in NIFTi we also use ROS as a middleware-framework in TRADR to facilitate reuseable, decoupled modules for an aggregated and integrated system. [1]

Comparison of available ROS-multimaster systems Obviously, it is no option for TRADR to launch a single ROS master only. At least each robot system will need its own ROS master, thus it will be necessary to establish a communication across these systems.

During the developers' meeting in May 2014, an approach was proposed that is used in the SENEKA project [34]. After a follow-up discussion, it became clear that the so-called "SENEKA hub" does not really fit for TRADR. We should rather rely on standard packages from the ROS community like:

- Multimaster (deprecated)
- Foreign Relay (deprecated)
- Robotics in Concert: rocon
- multimaster_fkie

After functional tests of rocon and multimaster_fkie, it is recommended to use rocon for the needed ROS Multimaster feature in TRADR. Details of the comparison can be found in [55] (Annex Overview 2.6).

Issue-tracker system The development and information exchange in TRADR should be supported by a suitable tracker software. In NIFTi TRAC was used for this purpose offering ticket functionality and a Wiki.

The start of TRADR was a good point in time to reconsider which up-to-date software could be used instead, offering even more useful functionalities. The partners were asked to propose desirable features they would like to have included like task/issue/bug tracking, test integration, document/project management, meeting planning, workflow/user management etc. With this collection of favored features several up-to-date issue trackers were scrutinized and compared with TRAC. The collection can be found in [58] (Annex Overview 2.7). Also experiences of the partners with various trackers were factored in as well as impressions of the examined software.

At the end the choice fell upon Redmine [2] for several reasons:

Besides its mature impression it includes features like REST API, plugin mechanism, support for mobile devices, activity monitor, Gantt charts, a forum and a document management. Also the experience of one partner with Redmine was quite positive.

GIT repository As a successor of SVN, GIT [3] is used as a distributed version control system. To be able to develop on individual code parts and to minimize the effort of merging the contributions of different partners into the main codebase we use the *feature branch workflow* of GIT. Hence, new features are developed on separate branches without disturbing others or the main code base.

Partners can exchange branches whenever necessary and mutually request and accept merges to the main code base. This supports the continuous integration environment ensuring that the main code base is always buildable.

Jenkins server Jenkins, an automated build tool, provides a continuous integration environment [4]. It is triggered either manually or by the GIT repository *GITLAB* [5] each time a merge request between two branches is triggered by a partner.

Jenkins attempts to merge the given feature branch into the main code base and tries to build and test it (see Figure 6). On failure the respective developer gets an email with the occurred errors. All partners can see the current status of all projects.

Experiences Building upon the experiences made in NIFTi, the TRADR architecture emphasizes the following aspects to ease system integration:

- usage of same technical basis for similar components (like OCU and TDS)

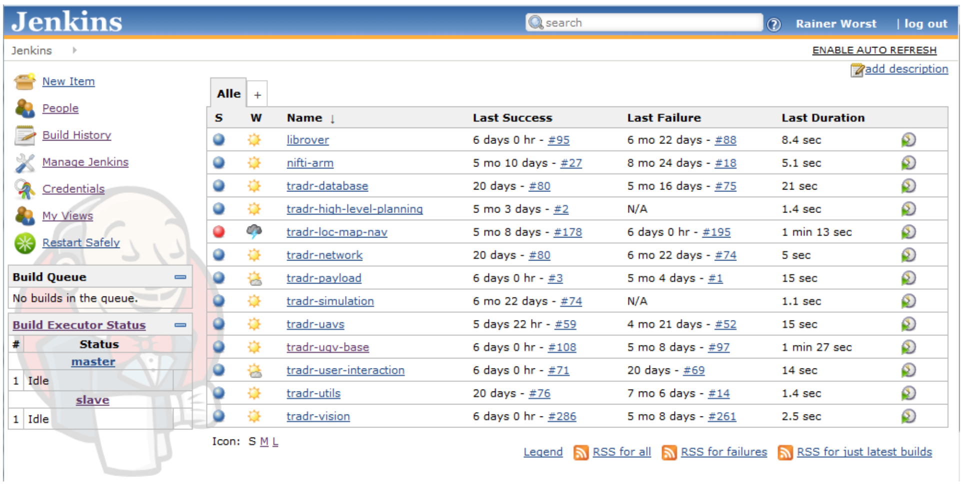


Figure 6: Jenkins build server

- software modules must be robust against connection losses
- usage of well-defined interfaces and APIs between software modules and subsystems
- usage of loosely coupled software modules
- usage of a central database for information storage and exchange

The architecture of TRADR simplifies the dynamic usage, expansion and exchange of all software modules and supports the future integration of new components and functions.

1.2.3 Migration from NIFTi to TRADR

As TRADR is a successor of NIFTi, it was obvious that the successful work created in NIFTi should become also available in TRADR. Besides available software, also the acquired experiences should be re-used. On the other hand, operating systems, third party components as well as tools also progressed in their development offering new versions or new strategies. Therefore, a managed process was needed for the migration, which comprised an upgrade of the UGVs, purchase of robotic arms, new UAV platforms, and the necessary software modifications. These topics are discussed in detail below.

Migration steps To migrate from NIFTi to TRADR, three main milestones were scheduled at the beginning of TRADR:

- Code selection, correction, and restructuring until March 2014

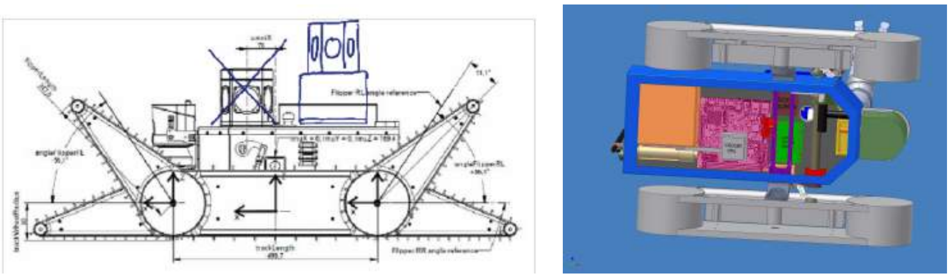


Figure 7: UGV upgrade process. Left: early drawing of the new design. Right: 3D model rendered by the CAD system

After this step only those parts of the NIFTi software remained that should be reused in TRADR. These parts were re-arranged to a new structure in SVN but still ran in the old ROS Fuerte Turtle and Ubuntu 12.04 environment. As part of this step, useful information in TRAC was transferred to Redmine.

- Migration from SVN to GIT and from ROS Fuerte Turtle to Hydro Medusa (catkin) until April 2014

  With this step, the versioning system was changed as well as the version of the underlying middleware-framework.

- Update to Ubuntu 14.04 until December 2014

  It emerged that with the update to Ubuntu 14.04 also ROS Hydro Medusa had to be updated to the newer version Indigo Igloo. After the update of all developers' PCs and of the continuous integration server Jenkins, also the robots were updated to Ubuntu 14.04 LTS.

In sum, the start of TRADR allowed to reconsider the used development strategies from NIFTi, use new versions of ROS and Ubuntu, remove obsolete code and outdated tools like SVN and trac and use up-do-date software tools like GIT and Redmine.

UGV upgrade At the beginning of the TRADR project, we went through a phase of collecting and discussing upgrade requirements. Several versions of technical drawings and CAD models (Figure 7) were created, discussed, and incrementally improved. The final re-design of the UGV was driven mainly by the requirement of integrating a manipulation robotic arm. Also, end-users often called for mounting additional sensors on-board. The locomotion, the main sensory field should remain intact as much as possible. The decisive constructive change is the re-design of the bearing part.

The old robot had an inside cage that held the remaining construction pieces, see Figure 8 left. In the upgraded design, the main bearing component is a profiled metal plate on top of the robot body. All other pieces are

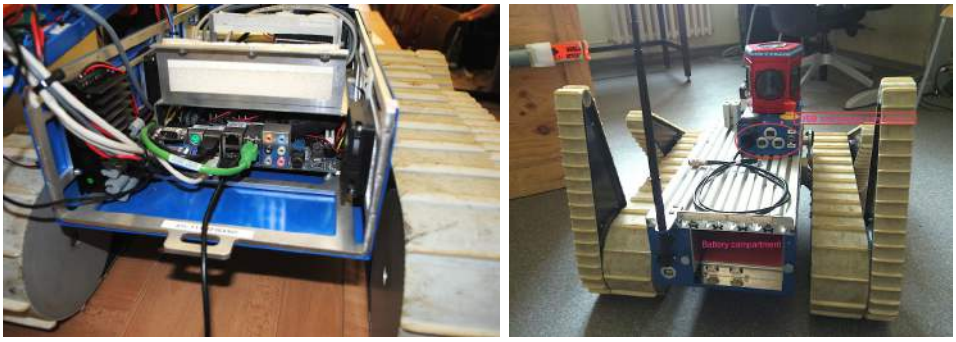


Figure 8: Construction differences. Left: bearing cage inside. Right: the top profiled metal plate is the main bearing component

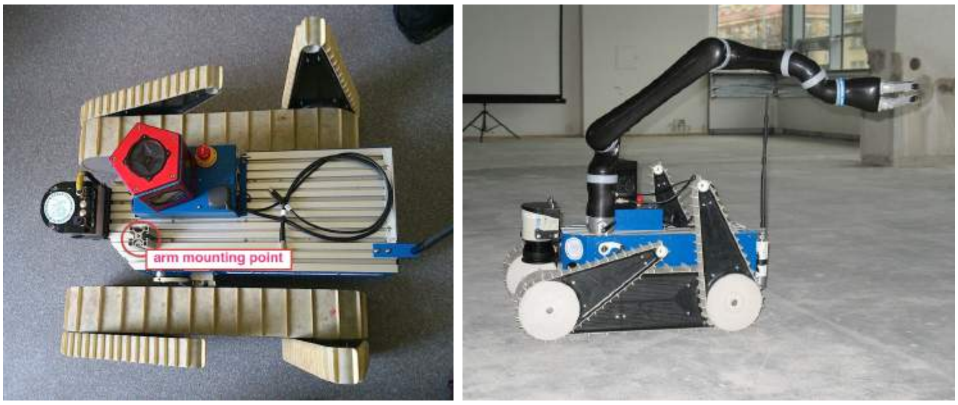


Figure 9: Left: a bird view of the robot, the flexible mounting point is visible next to the red Ladybug camera and the blue box. Right: a snapshot of the UGV with the arm mounted.

attached to it. A flexible top metal plate enables mounting the arm on any position, see Figure 9. A small metal box mounted on the plate contains several USB2/3 ports as well as cables for the arm, small light reflectors, and a 24V power supply for additional sensors.

Besides the mechanical construction upgrade, we also upgraded the on-board CPU. The miniITX mother board format remained the same, the computational power, RAM, and capacity of SSD increased. Also the side covers were re-designed in order to allow faster service of malfunctioning tracks. Speaker, mic, powerful light LED sources were also added in order to increase the overall utility. The residual electronics, motor controllers etc. remained the same.

During the design process, we decided not to engage BlueBotics, the original manufacturer of the UGVs, to perform the HW upgrade, but to assign this task to Neovision Ltd [6].

Company	Product	Weight	Range	Price	Payload	Power supply	IP-rating
Kinova	JACO Research edition	5.7 kg	90 cm	35.000 USD	1.0 kg	24 V	IPx2
Kinova	MICO Research Edition	5 kg	**70 cm**	21.700 USD	0.75 kg	24 V	IPx2
KUKA	Youbot arm	6.3 kg	**54 cm**		0.5 kg	24 V	
Barret	WAM	**25-27 kg**	100 cm	160.000 USD	3-4 kg	24-80 V	
Festo	Elephant trunk robot arm	**?**	?	?	?	?	
Inven-science LC	Advanced Robotic Manipulator (ARM 2.0)	**11.3 kg**	1.37 m	5.538 EURO	6.8 kg	12 V	
Jaguar	Jaguar Robotic Arm	**<10 kg**	71 cm	8.750 USD	4.0 kg	?	
Pioneer	Seekur Jr Outdoor Manipulator	10 kg	1 m	US 74,995 USD	3 kg	24V, **280W**	IP65
KUKA	Lightweight Robot (LBR)	**22.3 kg**			7 kg	220 V	
DLR	MIRO KineMedic	**9.8 kg**			3 kg		
Mitsubishi	PA 10-7C	**40 kg**			10 kg	100-240 V	

Table 1: Comparison of robotic arms for the TRADR UGV

Choice of robotic arm To add manipulation capabilities to the UGVs, we needed to integrate a robotic arm into the system. We considered all the models listed in table 1, and ended up choosing the Kinova JACO arm. The key disadvantages of the other options were as follows. The Mico and the Youbot were too small. The Barret, Inverscience, Jaguar, Kuka L. and Mitsubishi were too heavy. The Jaguar had no ROS support and the DLR and Festo were not available on the market. Finally, the Pioneer needed too much power.

UAV replacement In TRADR, the NIFTi-UAV should be replaced by Ascending Technologies systems. Doing so in a first step, the following UAVs (see Figure 10) were evaluated and tested by the consortium before and during the TRADR Joint Exercise in Italy, September 2014 (TJEx):

AscTec Pelican: [7] This UAV is one of the research platforms providing open access to a second micro controller enabling the user to easily test software onboard the UAV. Therefore, the NIFTi-UAV results could be transferred to this system and could continuously be used in TRADR. In particular, we prepared the Pelican for communication with ROS. This includes the development of a driver which exchanges basic commands and sensor information by using ROS messages, e.g., for the usage of RGB and infrared cameras. We implemented further algorithms, which estimate the motion of the UAV based on its inertial measurement unit. So, it is possible to control the UAV via ROS components and query status information. We implemented hector_slam [8] in combination with the Hokuyo laser scanner [9] as a proof of performance. We developed several plug-ins, which allow the control

Figure 10: Different UAV platforms provided by Ascending Technologies. Top left: Pelican. Top right: Firefly. Bottom: Falcon 8

of the Pelican by clicking GPS positions on a map of the current environment. The Pelican communicates via the so-called ACI interface, so that the final Ascending UAV, which will be used in the TRADR project, can also communicate with ROS via this driver.

AscTec Firefly: [10] The software architecture of this system is very similar to the Pelican. Main differences are the payload options resulting in a first preference to use the Pelican.

AscTec Falcon 8: [11] Falcon 8 is designed for professional use by end-users. The challenge in TRADR is to combine the researchers' needs with the end-users' demands. The Pelican is a platform to be used, if experiments on navigation, planning and obstacle avoidance are made. The team can add sensors and the onboard computer can maintain CPUs up to Intel Core i7. In contrast, the Falcon is preferred by the end users – it can provide stabilized images and provides many comfortable features matured over the years based on real experience.

In TRADR, features of both systems are important: to work with the end users and understand their needs while the research team is replacing features performed by skilled pilots step by step with intelligent algorithms.

Figure 11: Photogrammetic 3D model of the TJEx hospital site

In Yr1, the following software features were used, evaluated, and integrated into the command structure:

Waypoint/trajectory control: The waypoint control is based on the new autopilot system AscTec Trinity and was enhanced during the TJEx preparation. For more details, please see DR2.1.

3D mapping: Another very promising feature tested during TJEx was to generate a photogrammetric 3D model of the hospital site. For a preliminary model, a flight of 5 minutes and thereafter 10 minutes offline computation time was needed (see Figure 11). It still needs to be evaluated how to match photogrammetric point clouds with Lidar data. From the TJEx experience it became apparent that such a 3D model could be provided quickly at system setup time and made available to the system administration. Doing so, topics like geo-referencing the model or ortho-map are addressed as well as the needed workflow.

Camera integration: On the Falcon, a variety of cameras were evaluated. Most promising for the scenario was on one hand a high resolution camera with a fix lens to generate data for the 3D model or a simple first overview picture. On the other hand a combination of stabilized zoom camera together with an infrared camera was positively evaluated during TJEx. The ability to zoom in immediately in order to inspect details was also evaluated favourably. Also the ability to switch to the infrared camera was very useful to detect persons.

Split control: For now, and as long as automatic systems can't avoid collisions, the pilot has to have direct sight to the vehicle and is responsible for its proper operation. During TJEx nevertheless, it was very promising to split the command and to give a second operator the ability to control the camera. This was done using the existing split control. For the next step this feature will be integrated into the command structure and controls (OCU), so that an operator can control the camera by himself watching the live video stream. Additionally, the operator can command the pilot to change position in order to provide the desired field of view.

UAV roadmap So far the TRADR specific development is done on two different platforms, the Pelican and Falcon because of the different tasks performed by the research team and by end-users. During Yr2 it is now planned to utilize results from the FP7 project EuRoC to provide a more powerful platform with the newest auto pilot technology. In a next step, the goal is to bring both platforms together so that newly developed algorithms can be evaluated concurrently with providing the comfort functions of the end-user platforms. This will lead to a first version of a TRADR specific UAV in Yr3 and a final version in Yr4.

1.2.4 User interfaces for shared SA

An existing user interface inherited from the NIFTi project was used during TJEx. Based on this experience and on the results of WP5 as presented in DR5.1, we decided to re-design the NIFTi components OCU (Operator Control Unit) and TREX (Tagging based Realtime EXhibitor repository).

We identified the main causes of instability and avoided them in the design of the new TRADR Display System (TDS). For a detailed description of this process, see "Overview of the development towards the TRADR Display System" in DR3.1 of WP3.

Plugin-based user interfaces To avoid the found drawbacks, we decided to use a mutual plugin-based user interface architecture. We chose Qt as framework for the OCU and the TDS since it is available on many platforms, supports plugin-based development, is multitouch aware, comes with many ready to use widgets and libraries (e.g., a map view) and has a good ROS integration (called rqt) to easily visualize arbitrary robot sensor data.

This strategy allows us to integrate new features and plugins without being compelled to write two separate solutions or to modify the main program code each time. We decided to keep plugins clear and simple avoiding local caching or separate interpretation of data from the database. In addition, OCU and TDS should be loosely coupled and exchange information only through ROS messages or the database.

Prototype of new OCU We developed a collection of loosely coupled plugins, which restored the functionalities of the former OCU, e.g., map view, 3D laser data view, camera live view, screenshot functionality, iconized previews, status indications. We can add arbitrary rviz views to embed

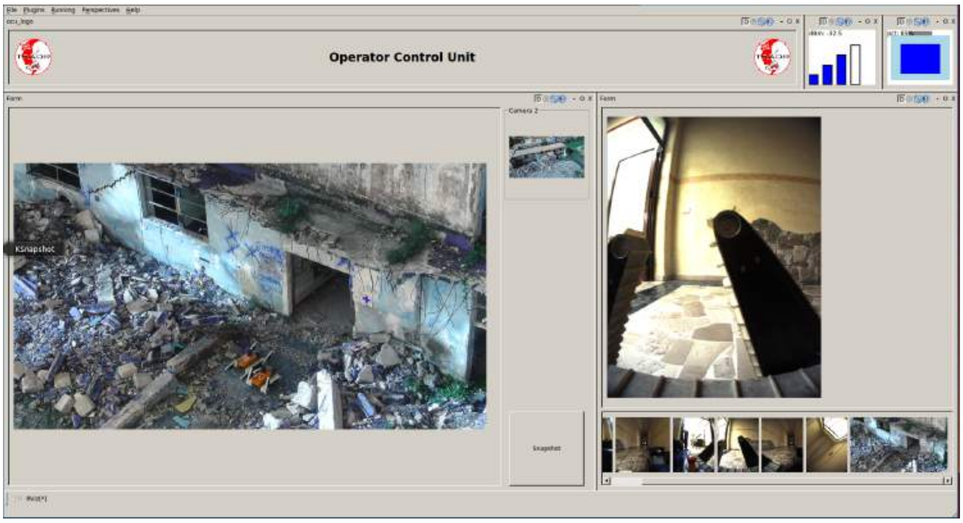


Figure 12: Operator-Control-Unit (OCU)

already developed and tested visualizations of robot data. This was verified during the TJEx, where we adapted the initial version of the new OCU, originally written only for the UAV, also to the UGV (see Figure 12).

The system is completely configurable, which means that we can obtain any available TDS functionality without altering or adding any code. The OCU is also able to save and restore profiles, which enables the user to personalize his instance of the OCU like shown plugins or location of subframes. More details about the implementation of the TDS are described in DR3.1.

1.2.5 Data storage and retrieval

A main goal of TRADR is to store and retrieve data that is updated across different sorties and extendable over a long period of time; hence, the design of proper persistent models is essential. As a result of the requirement analysis, we found that we have to split the database in two parts in order to fit our needs. The low-level database must be able to store and handle rather big binary blobs like images, videos, or pointclouds, while the high-level database maintains information about the semantic model of the data.

General design of low level DB A database system, which is suitable for robots in disaster scenarios, has to be able to handle the following issues:

handling blobs The low-level database has to be able to handle binary large objects often referred as blobs. Binary in this context means that it doesn't have to be necessarily understood/indexed by the database.

network load Since there is only a limited bandwith available for the robots, not all data can be transfered to a central database.

autonomy and consistency The robot must be able to operate even without a network connection. As there might be connection losses even during data transfer, it is impossible to always ensure consistency on the database level. The TRADR consortium decided to prioritize autonomy over consistency for the low-level database. This implies that the application level must be aware of connection losses and inconsistencies.

To integrate the database into the whole TRADR system, a database API will be established. At the TJEx, a prototype of this API was successfully tested, integrating the control and display components (OCU, TDS) into the system to share image data with the robots and amongst each other. As the low-level database will mainly be used to share data in a simple way, there is no need for a complex query language like SQL. The API is REST-inspired, so there are only a few commands to access and modify the data. The database client itself specifies the kind of data which should be stored in the database. The communication to the database is implemented by pure ROS service calls, which makes it simple to access the database from all ROS components.

To fulfill the mentioned requirements, we decided to use MongoDB [12] with a thin wrapper to automatically convert ROS messages into JSON documents. MongoDB has native support for blobs, is able to synchronize different database instances and also provides geo-spatial indexing.

Choice of high level DB The high-level DB is an RDF triple store. Based on our initial research, many triple store implementations exist that could serve our purpose. The most performant triple stores are discussed in the referenced list [13]. Other works on triple store evaluation [47] [51] support that view. As a lot of high-level data also originates from and references to low-level data, our initial choice fell on AllegroGraph [14] [19] which provides some kind of integration with MongoDB [15].

During the development of the project's overall architecture it became plausible that between the low-level and high-level DB a module called the *Semantic Modeler* is needed for abstracting and filtering information and thus transfer and enrich data from the low-level to the high-level DB. The full set of functionalities and RDF abilities of this modeler is yet to be specified.

Since this is the most likely scenario, our choice fell on Stardog [16] triple store including the Pellet [50] reasoner. Stardog, developed by Clark & Parsia [17] is able to handle ontological rules as described in the annex "The TRADR Ontology" of DR5.1. It also can provide explanations to the

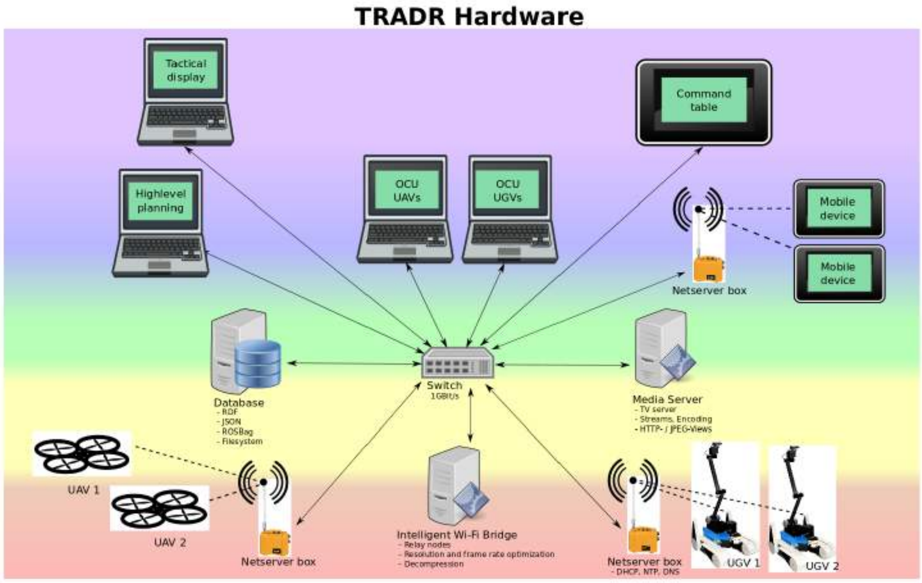


Figure 13: Network topology

inferred logical consequences. It is third on the list of the most performant triple stores right after Oracle Graph and the already mentioned Allegro-Graph, and it's a lightweight, easy to use graph database that scales up to 50 billion triples.

Further extensive use and tests will determine if the choice of triple store technology is suitable for the requirements that the project imposes.

1.2.6 Network infrastructure

To function properly even under adverse conditions, TRADR's network infrastructure needs to support reliable connections as well as robust data transfer rates (see Figure 13).

As TRADR includes the utilization of mobile robots, the concept of a stable WiFi network is crucial and the most sensitive point in the network infrastructure [52] (Annex Overview 2.5). Due to the experiences of NIFTi, we are confident to avoid the lacks that came up during in-field-usages of robots in former years.

From the software point of view this means to reckon with occasional WiFi connection losses and to properly handle corrupted data sets as well as transfer delays or interruptions without hanging or crashing. We improved the reliability of data exchange through the decoupling of modules that were closely coupled before.

As the bandwidth of WiFi connections is strictly limited, it is vital to deal with huge data sets accordingly. Transferring data without unnecessary repetitions over the WiFi connection can save a lot of trouble. To accomplish this, a relay node was implemented to receive the data, e.g., a video stream, over the WiFi connection once and distribute it within the fast wired LAN as often as needed.

Concerning network protocols for data transfer, we analyzed situations showing a blocked network transport. With this knowledge we successfully tested more fault tolerant protocol alternatives for broadcasting video and streaming data to prevent transfer standstill. An available solution to dynamically deal with changing data transfer bandwidth in ROS communication can be found at [18].

To further improve the data transfer over WiFi we use two common frequency bands at 2.4GHz and 5GHz instead of only one. By doing this, we could trouble-free fly the UAV without initiating an emergency landing routine due to connection loss.

The NetserverBox offers a rugged and waterproof WiFi access point for mobile robots into TRADRs wired network [53] (Annex Overview 2.4). The NetserverBox was also used in NIFTi and showed a quite robust performance. With one battery it can be powered for at least 10 hours. For TRADR a few improvements have been developed and included to ease the handling and enhance its functionalities:

- box internal status and monitoring webpage
- second LAN adapter with routing capabilities to external internet
- passive mode (client/server switch)
- power injection by wall outlet or truck outlet
- status LED for battery charge level and box status
- safety beeper and deep discharge protection
- battery hot swap
- update of existing capabilities like NTP, DNS, DHCP

Two of these boxes were delivered to other partners in TRADR. All in all it shows that with a fully charged battery the NetserverBoxes can endure a whole scenario day and thus allow to successfully work with mobile robots.

Network setup in TRADR As the TRADR setup consists of a PC cluster, the connection speed between the single machines should be fast. All computers are connected with each other by a fast 1 GBit/s LAN switch. WiFi NetserverBoxes are used to also connect the mobile robots to the LAN. The main features of our network setup can be summarized like this:

- Robots are connected via WiFi to the internal fast gigabit network, all other components are connected via cables.
- Within the internal network DHCP, NTP, and DNS services are provided.
- Because WiFi is relatively slow, all topics of the robots are relayed through the WiFi bridge. Thus they are transmitted over the weak connection only once. The bridge between WiFi and LAN is also responsible for dynamic data rate and resolution adjustments over the WiFi connection.
- A dedicated media server processes and provides suitable visible data (image views, films, scenario overviews with additional information, etc.) to arbitrary display devices like tablets or browsers.

Analysis of network issues during TJEx 2014 We experienced several WiFi breakdowns during TJEx. We took some network logs to be able to analyze the problem further. We found out that our communication stack (ROS messages on top of TCP) is not working well in a noisy environment, as packages might get altered during transmission and thus being rejected on the TCP level, even though they might still be useful, or could be made useful by, e.g., using forward error correction.

To improve the network stability we brought up these points for evaluation in future TRADR exercises:

UDP As UDP does not try to retransmit packages, nor rejects corrupted packages, it can improve the performance of, e.g., video streams.

FEC With Forward Error Correction, we would be able to send messages through a wireless connection, even if it alters the content of the messages.

Adaptive content aware (lossy) compression To make the best of the available bandwith, we must adapt the compression level of (lossy) compression algorithms according to the wifi strength.

Relay and cache node To avoid to retransmit data over WiFi we must use a relay and cache node as already described.

To effectively measure the impact of the different improvement ideas, we started to design a network benchmarking toolkit. Its main purpose is to measure the effective and potential throughput of a wireless setup.

1.3 Relation to the state-of-the-art

Research of the last decades showed open questions that are essential for rescue robotics, e.g., the need of having self-cleaning sensors, highly mobile robots, low power consumption, reliable high bandwidth communication channels, shape shifting capabilities or semi-autonomous robotic behaviors [41, 39, 43, 46, 38, 40, 35, 27, 36]. In addition to those robot-related aspects, there are also issues regarding usage and performance, like the need for portable robots, a minimum cognitive load and stress level for the operator, and the ability of interpreting natural communication of the operator while simultaneously covering best situational awareness [24, 23, 42, 21, 45]. Since 2012, the DARPA Robotics Challenge tries to promote the development of disaster reponse robots; few participants from Europe are currently involved [31].

The TRADR system addresses some of these open questions. Omnidirectional cameras are covering a 360 degrees view of the scenario to the operator; a virtual PTZ camera offers a natural "through the eyes" view. Such means can boost the situational awareness of the operator as they have the potential to limit the cognitive load and still keep the flexibility of the system high [39, 48]. In addition to raw camera views, a mapping system helps to keep track of the current position and state of the robot. By using laser range finders (2D and 3D) and a state-of-the-art mapping technology, the robot is autonomously recording a representation of the environment and presenting itself correctly aligned in this representation to the user [45, 56, 30, 44, 54, 57]. Concurrently to the support of raw images and maps, the robot is autonomously analyzing the raw data and searching for objects in the environment (e.g. cars, humans, hazardous material).

To keep the human operator in the loop, we use a graphical user interface to present the preprocessed information and to acquire new commands. This technology is well elaborated [20, 21] and has several advantages. In comparison to a raw video/sensor display and joystick-like control system like those presented in [23, 42, 35], such an advanced interface can give support to the operator and hence limit his cognitive load. It allows him to keep several facts in mind without over-stressing his attention.

Mobility is a key feature for a rescue robot. As rescue scenarios are usually less structured, the robot is forced to deal with rubble, holes, uneven terrain or even with objects that must be overcome. A large number of technologies has been elaborated and the research in this field continues. Wheel based systems are economical in terms of power consumption, but have often problems to handle scenarios with holes or which require climbing skills. Legs are extremely flexible, offer good climbing skills and have a high mobility, but are complex, uneconomic in terms of power consumption and can usually carry less payload. Track based systems are economic between wheels and legs. They have usually high friction and a wide footprint, which makes them a good compromise in terms of payload, overcoming objects and climbing skills [41, 39, 49]. For the TRADR system, we continue to use the highly adaptive NIFTi-UGV, which is a track-based platform and able to traverse complex terrain. In addition, we apply a Pelican and other platforms of Ascending Technologies as UAVs, which can provide a bird's eye view on the scenario.

Several other European projects address the deployment of (teams of) UGVs and UAVs in various disaster response scenarios. ICARUS [26] and DARIUS [25] target the development of robotic tools that can assist during disaster response operations, focusing on autonomy. SHERPA [37] is focused on the development of ground and aerial robots to support human-robot team response in an alpine scenario.

None of these projects addresses persistence issues. In TIRAMISU [22], a toolbox is developed for removal of anti-personnel mines, submunitions, and Unexploded Ordnance (UXO). It includes a component called TIRAMISU Repository Service, which provides a centralized data-sharing platform that contains the locations of detected landmines and UXOs. The TRADR concept of persistent situation awareness goes beyond this in various respects. On the other hand, the EU project STRANDS [28], aims at modeling the spatio-temporal dynamics in human indoor 3D environments in order for a single robot to adapt to and exploit long-term experience in months-long autonomous operation. In contrast, TRADR deals with multiple sorties into an unstructured outdoor environment carried out by a human-robot team.

References

[1] `http://wiki.ros.org/indigo`.

[2] `http://www.redmine.org/`.

[3] `http://git-scm.com/`.

[4] `http://jenkins-ci.org/`.

[5] `https://about.gitlab.com/`.

[6] `http://neovision.cz`.

[7] `http://www.asctec.de/downloads/public/AscTec-Pelican_Safetydatasheet.pdf`.

[8] `http://wiki.ros.org/hector_slam`.

[9] `https://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html`.

[10] `http://www.asctec.de/downloads/public/AscTec-Firefly_Safetydatasheet.pdf`.

[11] http://www.asctec.de/downloads/public/AscTec-Falcon-8_Safetydatasheet.pdf.

[12] http://www.mongodb.org/.

[13] http://www.w3.org/wiki/LargeTripleStores.

[14] http://franz.com/agraph/allegrograph/.

[15] http://franz.com/agraph/support/documentation/current/mongo-interface.html.

[16] http://stardog.com.

[17] http://clarkparsia.com/.

[18] https://github.com/WPI-ARC/datalink_toolkit/wiki.

[19] Jans Aasman. Allegro graph: RDF triple database. Technical report, Technical report. Franz Incorporated, 2006.

[20] M. Baker, R. Casey, B. Keyes, and H.A. Yanco. Improved interfaces for human-robot interaction in urban search and rescue. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 3, pages 2960–2965. IEEE, 2005.

[21] Bradley J. Betts, Robert W. Mah, Richard Papasin, Rommel Del Mundo, Dawn M. McIntosh, and Charles Jorgensen. Improving situational awareness for first responders via mobile computing. Technical memorandum, NASA Ames Research Center, Moffett Field, CA 94035-1000, March 2005.

[22] L. Cantelli, M. Mangiameli, C.D. Melita, and G. Muscato. UAV/UGV cooperation for surveying operations in humanitarian demining. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–6, Oct 2013.

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

[24] Jennifer Casper. Human-robot interactions during the robot-assisted Urban Search And Rescue response at the World Trade Center. Master's thesis, Department of Computer Science and Engineering, College of Engineering, University of South Florida, May 2002.

[25] P. Chrobocinski, N. Zotos, E. Makri, C. Stergiopoulos, and G. Bogdos. DARIUS project: Deployable SAR integrated chain with unmanned systems. In *Telecommunications and Multimedia (TEMU), 2012 International Conference on*, pages 220–226, July 2012.

[26] G. De Cubber, D. Doroftei, D. Serrano, K. Chintamani, R. Sabino, and S. Ourevitch. The EU-ICARUS project: Developing assistive robotic tools for search and rescue operations. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–4, Oct 2013.

[27] B. Doroodgar, M. Ficocelli, B. Mobedi, and G. Nejat. The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2858–2863, 2010.

[28] Tom Duckett, Marc Hanheide, Tomas Krajnik, Jaime Pulido Fentanes, and Christian Dondrup. Spatio-temporal representations for cognitive control in long-term scenarios. In *International IEEE Workshop on Autonomous Cognitive Robotics*, 2014.

[29] Mario Gianni, Ivana Kruijff-Korbayová, Rainer Worst, and NIFTi Project Team. Human-robot teaming in disaster response – a user-centric approach. Submitted to ICRA Workshop RATHA 2015 "Robotics & Automation Technologies for Humanitarian Applications: Where we are & Where we can be", May 2015.

[30] A. Kleiner. *Mapping and exploration for search and rescue with humans and mobile robots*. PhD thesis, Universitätsbibliothek Freiburg, 2008.

[31] Stefan Kohlbrecher, Alberto Romay, Alexander Stumpf, Anant Gupta, Oskar von Stryk, Felipe Bacim, Doug A. Bowman, Alex Goins, Ravi Balasubramanian, and David C. Conner. Human-robot teaming for rescue missions: Team ViGIR's approach to the 2013 DARPA robotics challenge trials. *Journal of Field Robotics*, pages n/a–n/a, 2014.

[32] G.J.M. Kruijff, I. Kruijff-Korbayová, S. Keshavdas, B. Larochelle, M. Janíček, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, M.A. Neerincx, R. Looije, N.J.J.M Smets, T. Mioch, J. van Diggelen, F. Pirri, M. Gianni, F. Ferri, M. Menna, R. Worst, T. Linder, V. Tretyakov, H. Surmann, T. Svoboda, M. Reinštein, K. Zimmermann, T. Petříček, and V. Hlaváč. Designing, developing, and deploying systems to support human-robot teams in disaster response. *Advanced Robotics*, 28(23):1547–1570, 2014.

[33] Ivana Kruijff-Korbayová, Francis Colas, Mario Gianni, Fiora Pirri, Joachim de Greeff, Koen Hindriks, Mark Neerincx, Petter Ögren, Tomáš Svoboda, and Rainer Worst. TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response. *KI - Künstliche Intelligenz*, pages 1–9, 2015.

[34] Helge-Bjoern Kuntze, Christian Frey, Thomas Emter, Janko Petereit, Igor Tchouchenkov, Thomas Mueller, Martin Tittel, Rainer Worst, Kai Pfeiffer, Moriz Walter, Sven Rademacher, and Fabian Mueller. Situation responsive networking of mobile robots for disaster management. In *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*, pages 1–8, June 2014.

[35] Thorsten Linder, Viatcheslav Tretyakov, Sebastian Blumenthal, Peter Molitor, Hartmut Surmann, Dirk Holz, Robin Murphy, and Satoshi Tadokoro. Rescue Robots at the Collapse of the Municipal Archive of Cologne City: a Field Report. In *8th IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR-2010)*, July 2010.

[36] Yugang Liu and Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, pages 1–19, 2013.

[37] L. Marconi, S. Leutenegger, S. Lynen, M. Burri, R. Naldi, and C. Melchiorri. Ground and aerial robots as an aid to alpine search and rescue: Initial SHERPA outcomes. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–2, Oct 2013.

[38] M.J. Micire. Evolution and field performance of a rescue robot. *Journal of Field Robotics*, 25(1-2):17–30, 2008.

[39] R. Murphy, J. Casper, J. Hyams, M. Micire, and B. Minten. Mobility and sensing demands in USAR. *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE*, 1:138–142 vol.1, 2000.

[40] R. Murphy, J. Kravitz, S. Stover, and R. Shoureshi. Mobile robots in mine rescue and recovery. *Robotics Automation Magazine, IEEE*, 16(2):91 –103, 2009.

[41] R.R. Murphy. Marsupial and shape-shifting robots for Urban Search And Rescue. *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems]*, 15(2):14–19, Mar/Apr 2000.

[42] R.R. Murphy. Human-robot interaction in rescue robotics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):138–153, May 2004.

[43] R.R. Murphy. Trial by fire [rescue robots]. *Robotics & Automation Magazine, IEEE*, 11(3):50–61, Sept. 2004.

[44] Johannes Pellenz, Dagmar Lang, Frank Neuhaus, and Dietrich Paulus. Real-time 3D mapping of rough terrain: A field report from disaster city. In *IEEE International Workshop on Safty, Security and Rescue Robotics*, 2010.

[45] Kai Pervölz, Hartmut Surmann, and Stefan May. 3D laser scanner for tele-exploration robotic systems. In *Proceedings of the International Workshop on Safty, Security and Rescue Robotics (SSRR 2006), Gaithersburg, Maryland, USA*, August 2006.

[46] K.S. Pratt, R.R. Murphy, J.L. Burke, J. Craighead, C. Griffin, and S. Stover. Use of Tethered Small Unmanned Aerial System at Berkman Plaza II Collapse. In *IEEE International Workshop on Safety, Security and Rescue Robotics, 2008. SSRR 2008*, pages 134–139, 2008. ISBN: 978-1-4244-2031-5.

[47] Kurt Rohloff, Mike Dean, Ian Emmons, Dorene Ryder, and John Sumner. An evaluation of triple-store technologies for large data stores. In *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, pages 1105–1114. Springer, 2007.

[48] N. Shiroma, N. Sato, Y. Chiu, and F. Matsuno. Study on effective camera images for mobile robot teleoperation. In *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, pages 107 – 112, 2004.

[49] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Book, 2004.

[50] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.

[51] Florian Stegmaier, Udo Gröbner, Mario Döller, Harald Kosch, and Gero Baese. Evaluation of current RDF database solutions. In *Proceedings of the 10th International Workshop on Semantic Multimedia Database Technologies (SeMuDaTe), 4th International Conference on Semantics And Digital Media Technologies (SAMT)*, pages 39–55. Citeseer, 2009.

[52] Hartmut Surmann and Rainer Worst. Design principles for the connection of mobile sensor platforms. Accepted for Workshop “Networked Robots” at the European Robotics Forum, March 2015.

[53] Hartmut Surmann, Rainer Worst, Stefan Wilkes, Tom-Marvin Liebelt, and Christopher Eulering. Simple Mobile Robots and Self Adaptive Wireless Networks. In *Proceedings of Joint 45th International Symposium on Robotics (ISR) and 8th German Conference on Robotics (ROBOTIK)*, Munich, June 2014.

[54] T. Wiemann, A. Nüchter, K. Lingemann, S. Stiene, and J. Hertzberg. Automatic Construction of Polygonal Maps From Point Cloud Data. In *IEEE International Workshop on Safty, Security and Rescue Robotics*, 2010.

[55] Stefan Wilkes. ROS Multimaster Concepts. Unpublished presentation, 2014.

[56] Zhe Zhang, Hong Guo, G. Nejat, and Peisen Huang. Finding disaster victims: A sensory system for robot-assisted 3D mapping of Urban Search And Rescue environments. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3889 –3894, 2007.

[57] Zhe Zhang, Goldie Nejat, Hong Guo, and Peisen Huang. A novel 3D sensory system for robot-assisted mapping of cluttered Urban Search And Rescue environments. *Intelligent Service Robotics*, 4(2):119–134, 2011.

[58] Erik Zimmermann. Comparison of issue tracker capabilities. Unpublished table, 2014.

2 Annexes

2.1 Kruijff et al. (2014), "Designing, developing, and deploying systems to support human-robot teams in disaster response"

Bibliography G.J.M. Kruijff, I. Kruijff-Korbayová, S. Keshavdas, B. Larochelle, M. Janíček, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, M.A. Neerincx, R. Looije, N.J.J.M Smets, T. Mioch, J. van Diggelen, F. Pirri, M. Gianni, F. Ferri, M. Menna, R. Worst, T. Linder, V. Tretyakov, H. Surmann, T. Svoboda, M. Reinštein, K. Zimmermann, T. Petříček, and V. Hlaváč. "Designing, developing, and deploying systems to support human-robot teams in disaster response." In *Advanced Robotics*, 28(23):1547–1570. 2014.

Abstract This paper describes our experience in designing, developing and deploying systems for supporting human-robot teams during disaster response. It is based on R&D performed in the EU-funded project NIFTi. NIFTi aimed at building intelligent, collaborative robots that could work together with humans in exploring a disaster site, to make a situational assessment. To achieve this aim, NIFTi addressed key scientific design aspects in building up situation awareness in a human-robot team, developing systems using a user-centric methodology involving end users throughout the entire R&D cycle, and regularly deploying implemented systems under real-life circumstances for experimentation and testing. This has yielded substantial scientific advances in the state-of-the-art in robot mapping, robot autonomy for operating in harsh terrain, collaborative planning, and human-robot interaction. NIFTi deployed its system in actual disaster response activities in Northern Italy, in July 2012, aiding in structure damage assessment.

Relation to WP This paper provides a comprehensive overview on the NIFTi project, which prepared the ground for TRADR; its content was reflected when performing T6.1.

Availablity Unrestricted. Pre-print included in the public version of this deliverable; full paper available for download at:
http://dx.doi.org/10.1080/01691864.2014.985335

2.2 Kruijff-Korbayová et al. (2015), "TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response."

Bibliography Ivana Kruijff-Korbayová, Francis Colas, Koen Hindriks, Mark Neerincx, Petter Ögren, Mario Gianni, Tomáš Svoboda, and Rainer

Worst. “TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response.” KI - Künstliche Intelligenz, 2015

Abstract This paper contains a description of the project TRADR: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response. As robotic disaster relief systems are still scarce, any incident serious enough to render robot involvement will most likely involve a sequence of sorties over several hours, days and even months. TRADR focuses on the persistence of environment models, multi-robot action models, and human-robot teaming, in order to allow incremental capability improvement over the duration of a mission. TRADR applies a user centric design approach to disaster response robotics, with use cases involving the response to a medium to large scale industrial accident by teams consisting of human rescuers and several robots (both ground and airborne). The paper overviews the project objectives and motivation, the structure and approach, as well as the partners and related work.

Relation to WP This paper is an extended summary of the Description of Work and describes the base for the technical system framework, which was developed in T6.1.

Availablity Unrestricted. Pre-print included in the public version of this deliverable; full paper available for download at http://dx.doi.org/10.1007/s13218-015-0352-5

2.3 Gianni et al. (2015), “Human-robot teaming in disaster response – a user-centric approach.”

Bibliography Mario Gianni, Ivana Kruijff-Korbayová, Rainer Worst and the NIFTi Project Team. “Human-robot teaming in disaster response – a user-centric approach.” Submitted to ICRA Workshop RATHA 2015 “Robotics & Automation Technologies for Humanitarian Applications: Where we are & Where we can be”, May 2015.

Abstract We have been pursuing the goal of supporting people and robots working together in teams in close and continuous human-robot interaction in the EU FP7 ICT project NIFTi. The core of our design methodology is a usercentric approach. We have applied this methodology to design and develop a complex system supporting human-robot collaboration in search and rescue (SAR) scenarios. We first describe the components of the user-centric development cycle, discussing the benefits of this approach in particular with respect to situation awareness and human-robot collaboration. Then, we summarize its application in NIFTi and the lessons we learned

along the way about human-robot teaming for SAR. To complete the picture we give a brief technical overview of the complex system we developed employing this methodology.

Relation to WP This paper describes the user-centric development approach that was successfully used in the NIFTi project and also in TRADR when achieving task T6.2.

Availablity Restricted. Not included in the public version of this deliverable.

2.4 Surmann et al. (2014), "Simple Mobile Robots and Self-Adaptive Wireless Networks"

Bibliography Hartmut Surmann, Rainer Worst, Stefan Wilkes, Tom-Marvin Liebelt, and Christopher Eulering. "Simple Mobile Robots and Self-Adaptive Wireless Networks." In *Proceedings of Joint 45th International Symposium on Robotics (ISR) and 8th German Conference on Robotics (ROBOTIK)*. Munich, June 2014.

Abstract Disaster areas require mobile robots with extreme capabilities. This paper presents an approach for setting up a network infrastructure to operate such mobile robots. We present a waterproof netserver box as a main component and mobile router robots (RC cars) to extend the network capabilities. The simple mobile robot consists of cheap standard RC components. It provides a view into the disaster area with its sensors, e.g., cameras, and in addition and very important, it sets up a persistent communication between all mobile robots and rescuers. All components use ROS as a middleware and can be integrated in the overall system. In addition to infrastructure networks, mesh networks are also supported to replace the destroyed network infrastructure. Furthermore, the network and robots are prepared for cloud computing.

Relation to WP This contribution describes the usage of simple mobile robots and the necessary network setup in NIFTi, which has been partially re-used in the context of T6.2.

Availablity Unrestricted. Pre-print included in the public version of this deliverable.

2.5 Surmann, Worst (2015), "Design principles for the connection of mobile sensor platforms."

Bibliography Hartmut Surmann, Rainer Worst. "Design principles for the connection of mobile sensor platforms." Accepted for Workshop "Networked Robots" at the European Robotics Forum, March 2015.

Abstract Urban search and rescue (USAR) missions need support from modern computer technologies to achieve their task to help people. In the past decade robotic researchers and computer scientists enhanced robots in particular to help in USAR missions. These projects lead to complex and expensive robots, e.g., tracked robots, humanoids, or drones, but one big problem is still open: how to ensure communication. Targeting to team building between human rescuers, tracked robots (UGVs), and drones (UAVs), the EU-funded projects NIFTi and TRADR have also to solve the communication problem and therefore to provide robust network nodes to set up a network for the robots. In the presentation, we show the projects approach to connect humans and robots, especially with the new designed outdoor WiFi box and related design principles of the system architecture.

Relation to WP This contribution describes network communication issues in USAR missions and in particular the usage of the NIFTi Netserver-Box, which was developed to simplify the network setup for wireless connected ROS-based devices, e.g., in the context of T6.2.

Availablity Restricted. Not included in the public version of this deliverable.

2.6 Wilkes (2014), "ROS Multimaster Concepts."

Bibliography Stefan Wilkes. "ROS Multimaster Concepts." Unpublished presentation, July 2014.

Abstract Currently, two implementations of a Multimaster feature are maintained within the ROS community: rocon and multimaster_fkie. They have been compared to find a suitable basic component for TRADR.

Relation to WP This presentation shows the result of the comparison between available ROS Multimaster packages in the context of T6.1.

Availablity Restricted. Not included in the public version of this deliverable.

2.7 Zimmermann (2014), "Comparison of issue tracker capabilities"

Bibliography Erik Zimmermann. "Comparison of issue tracker capabilities." Unpublished table, March 2014.

Abstract This collection lists the capabilities of different issue trackers to be able to select a suitable tracker for TRADR.

Relation to WP This list was used to find a suitable issue tracker in the context of T6.1.

Availablity Restricted. Not included in the public version of this deliverable.

To appear in *Advanced Robotics*
Vol. 00, No. 00, Month 201X, 1–27

FULL PAPER

Designing, developing and deploying systems to support human-robot teams in disaster response

G.J.M. Kruijff[a], I. Kruijff-Korbayová[a*], S. Keshavdas[a], B. Larochelle[a], M. Janíček[a]
F. Colas[b], M. Liu[b], F. Pomerleau[b], R. Siegwart[b]
M.A. Neerincx[c], R. Looije[c], N.J.J.M Smets[c], T. Mioch[c], J. van Diggelen[c]
F. Pirri[d], M. Gianni[d], F. Ferri[d], M. Menna[d]
R. Worst[e], T. Linder[e], V. Tretyakov[e], H. Surmann[e]
T. Svoboda[f], M. Reinstein[f], K. Zimmermann[f], T. Petříček[f], V. Hlaváč[f]

[a]*DFKI, Saarbrücken, Germany*;
[b]*ASL, ETH, Zurich, Switzerland*;
[c]*TNO, Soesterberg, The Netherlands*;
[d]*ALCOR, Sapienza University, Rome, Italy*;
[e]*Fraunhofer IAIS, Sankt Augustin, Germany*;
[f]*Department of Cybernetics, CTU, Prague, Czech Republic*

(*submitted May 14, 2014, revised October 21, 2014*)

This paper describes our experience in designing, developing and deploying systems for supporting human-robot teams during disaster response. It is based on R&D performed in the EU-funded project NIFTi. NIFTi aimed at building intelligent, collaborative robots that could work together with humans in exploring a disaster site, to make a situational assessment. To achieve this aim, NIFTi addressed key scientific design aspects in building up situation awareness in a human-robot team, developing systems using a user-centric methodology involving end users throughout the entire R&D cycle, and regularly deploying implemented systems under real-life circumstances for experimentation and testing. This has yielded substantial scientific advances in the state-of-the-art in robot mapping, robot autonomy for operating in harsh terrain, collaborative planning, and human-robot interaction. NIFTi deployed its system in actual disaster response activities in Northern Italy, in July 2012, aiding in structure damage assessment.

Keywords: robot-assisted disaster response; human-robot team; user-centric design; disaster response

1. Introduction

NIFTi was a large-scale four year integrated project funded by the EU Cognitive Systems unit [1]. [1] The NIFTi consortium consisted of six academic partners (i.e., the institutions of the authors of this paper), sharing experience in human-robot interaction, human factors and cognitive user modeling, field robotics, spatial and visual modeling of outdoor environments, and flexible planning and execution; two end user organizations (the Italian National Firebrigade Corps and the Firebrigade of the City of Dortmund) and BLUEBOTICS[2], a company who developed

*Corresponding author. Email: ivana.kruijff@dfki.de
[1]NIFTi was funded within the EU FP7 ICT programme, Jan 2010 – Dec 2013, grant No. 247870.
[2]http://www.bluebotics.com/

the novel rover platform Absolem for NIFTi (Fig. 20). NIFTi's aim was to investigate cognitive architectures which could meaningfully sense, act and cooperate with humans in real-life environments.

Regarding the issue of cooperation in cognitive architectures, when NIFTi started, it was entering a research landscape that primarily focused on autonomy, and high-level communication. Little or no attention was given to making the cognitive architecture adapt to the human in understanding the environment, planning and acting, communicating. In the words of the leading experts on human-robot teamwork: "Whereas early research on teamwork focused mainly on interaction within groups of autonomous agents or robots, there is a growing interest in better accounting for the human dimension. Unlike autonomous systems designed primarily to take humans out of the loop, the future lies in supporting people, agents, and robots working together in teams in close and continuous human-robot interaction." (Sierhuis & Bradshaw, p.c. 2009). NIFTi adopted the goal to bring the human factor into cognitive architectures while developing robots capable of collaborating with human team members under the complex outdoor circumstances of a disaster response.

To address this aim, NIFTi put strong emphasis on system integration, embedded within a user-centric approach to system development. The two firebrigade organizations included as partners in the NIFTi consortium enabled close involvement of end users, the ultimate stakeholders in this game, throughout the entire R&D cycle. They provided input to system specifications, participated in yearly exercises and evaluations, and provided feedback for further iteration cycles of the development process. Emphasis on system integration required that all developed functionality be integrated in the NIFTi system, and exposed to evaluation by the end users. To facilitate this across-the-board integration, NIFTi adopted a scenario-driven roadmap. The roadmap defined progressively more complex real-life scenarios: A tunnel accident (years 1 and 2), a chemical freight train accident (year 3), and an earthquake disaster response (year 4). These scenarios were all instantiated at end user training areas, subjecting the NIFTi system to realistic circumstances.

The roadmap simultaneously drove R&D, and brought the resulting systems closer to the possibility of real-life deployment. Needless to say, the lessons we learnt along the way were hardly ever foreseeable in advance. Experiments and tests each year brought new insights, to which we continuously adjusted our plans and the direction our R&D would take from that point on. The result was a convergence of R&D with real-life needs, real-life possibilities, and a real-life ability to actually aid in a disaster response. This ability was put to a test successfully during the summer of 2012, when NIFTi deployed its then-actual system to support a human-robot team in the structural damage assessment in Northern Italy during the aftermath of the July 2012 earthquake in the Emilia-Romagna region.

This article describes the experience NIFTi gained in designing, developing, and deploying human-robot team systems for robot-assisted disaster response. §2 provides more details on the NIFTi roadmap and scenarios. In §3 we describe the integrated system and its functional modules. §3.1 outlines the system architecture. §3.2 describes novel methods for building up robot-centric, three-dimensional models of dynamic environments – integrating various forms of perception (3D laser, visible range panoramic camera, thermal imaging). §3.3 addresses the functionality needed to bridge between system- and human-centric situation awareness. §3.4 then describes how we built on that to provide the robot capabilities for autonomously traversing complex terrain, including crossing gaps and climbing stairs. §3.5 provides an insight into handling human-robot collaboration. §3.6 provides more detail on how information about the situation is made available through various views in the graphical user interfaces for the human team members – portable, as well as a "static" setup in a remote command post. §3.7 addresses one crucial aspect of teamwork, namely handling cognitive load. §4 shows how our end user experiments reveal that this has a genuine, positive impact on ability for human team members to build up a situation awareness "through," together with, our robots. Finally, §5 recapitulates the experience of the actual deployment of our system during the earthquake response in Northern

(a) Year 1 (b) Year 2

(c) Year 3 (d) Year 4

Figure 1. Scenario impressions over the years.

Italy in July 2012. §6 provides the conclusions.

2. Scenario-driven R&D

NIFTi organised its R&D around a sequence of scenarios that gradually increased in complexity, including operational context complexity (from flat 2D, to semi-unstructured 3D) and collaborative context complexity, such as team size, its composition and geographical distribution (from 1 human/1 robot to a geographically distributed team consisting of multiple humans and robots). The robots used in NIFTi were an Unmanned Ground Vehicle (UGV, Fig. 20) and an Unmanned Aerial Vehicle (UAV microcopter, Fig. 21).

This scenario roadmap played a key role in providing an integrated conceptual picture for the project, to strongly drive integration of the various strands of R&D. Furthermore, by basing the scenarios directly in real-life situations in disaster response, we could ground R&D in real needs of Urban Search and Rescue (USAR) teams. The sections below describe the roadmap and the individual scenarios in more detail.

2.1 *Roadmap*

The NIFTi scenarios were designed in close cooperation between developers, and the USAR teams from the end user organizations involved in NIFTi as partners (Firebrigade Dortmund, Germany and National Firebrigade Corps, Italy). This was to ensure the scenarios would achieve a balance between practical relevance and feasibility, and necessary scientific progress. The result was a staged, iterative form of user-centric design cycle that addressed incrementally more complex situations.

On the one hand, the surroundings became incrementally harder for the robots to operate in. We went from flat, largely 2-dimensional terrain like the road surface in a tunnel, to semi-unstructured debris-strewn environments of an earthquake disaster. For robots, this necessitated

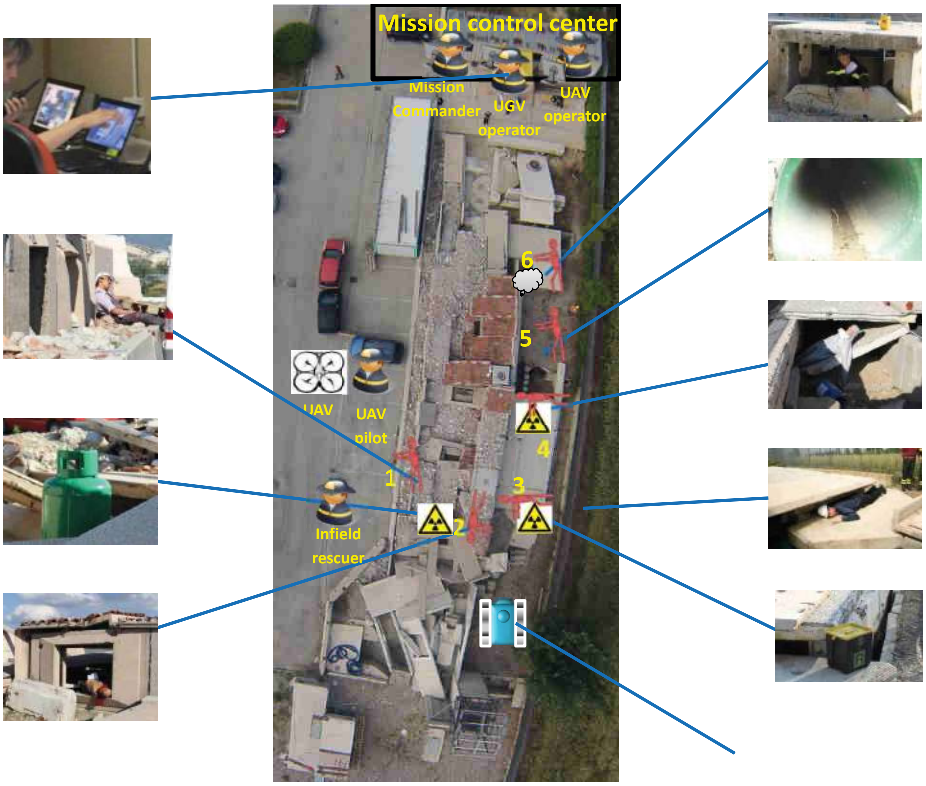


Figure 2. Year 4 scenario setup.

the development of increasingly more observational capabilities (from 2D to 3D), and progressively higher degrees of autonomy (3D path planning, adaptive morphology).

On the other hand, the organizational structure of the team become more realistic over the years. In Year 1, the organizational structure was non-existent, just a UGV and an operator. After that initial experience, we changed to a human-robot team setup. Eventually, the team included humans and robots, both UGV and UAV, working together in various locations – a genuinely geographically distributed team. By Year 4, robots were able to operate as team members in the sense of having the ability to build up situation awareness that was not immediately known by the operator and had to be shared.

2.2 *Scenario Design*

The joint scenarios that guided each year R&D were designed to evaluate all technical and operational requirements and to provide insight into the major determinants of human-robot team performance. To address important operational task demands, the environment provided realistic challenges, events and stressors like victim screams and (simulated) radioactive materials added at certain locations.

During the yearly scenario-based evaluation, the firefighter team-members (e.g., the commander, UGV operator, UAV operator or in-field rescuer) worked with the robots according to the role that was specified in the scenario. All the firefighters had an interface to show them (geographic) information [2]; the commander could contact both the UGV operator and the in-field rescuer, and vice-versa. Fig. 2 shows the final setting for the UGV-operator, where he can control the robot, get an overview of the situation and interact by walkie-talkie to the commander. Following an incremental R &D-approach, the complexity of the guiding scenario increased on three dimensions each year: scope of robot roles (i.e., level of autonomy and breadth of operations), team complexity (i.e., size and distribution), and terrain's complexity (i.e., accessibility and apparentness). Fig. 1 shows an impression of each year's setup.

The following is a brief chronological overview of how the NIFTi scenarios evolved and the convergence between R&D and real-life deployment came about.

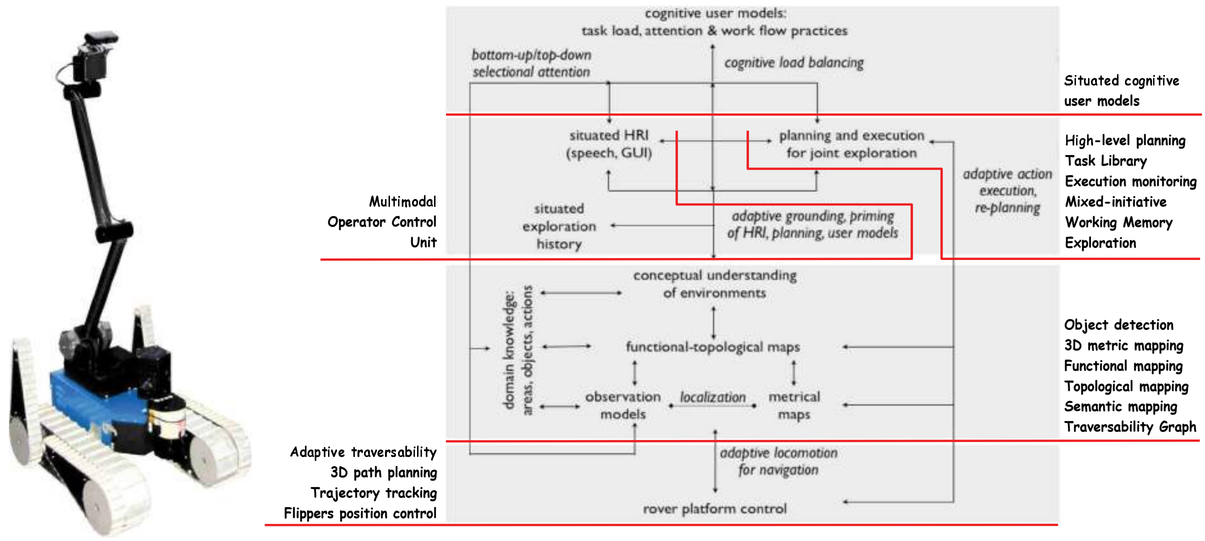
The **first year** scenario comprised a truck accident in a tunnel, which was dangerous to enter for humans. A single remotely located operator teleoperated a UGV in the tunnel to create a 2D-map populated with car objects recognised by the robot. We focused on bringing together the various pieces of individual robot functionality (control, mapping, vision) with a basic, end user oriented graphical user interface for teleoperating the robot. Admittedly, the first end user pilot study at the training area of the Firebrigade of Dortmund, Germany was less successful than we would have wished. We faced highly familiar problems such as network issues (too low bandwidth, bad connectivity), robot hardware issues (short battery operating time, blowing a fuse every few minutes), and a wide range of human factor issues. Consequently, the local firebrigade chief commented, when he was asked whether such robots would ever play a role in disaster response: "No ... At least not in my lifetime." (D. Aschenbrenner, Dortmunder Zeitung, January 14 2011). We concluded that we needed to move from the "single robot, single operator" setup to a full-scale human-robot team, and to get a better grip on technology.

In the **second year** we continued with the tunnel accident scenario, but now with a larger human team operating from a remote command post, and more difficult operational conditions, involving more smoke, flickering light and more debris. We introduced the new UGV platform developed in close collaboration with the end users involved in NIFTi (cf. Fig. 20). In addition, also a UAV microcopter was added to the rescue team (cf. Fig. 21). Furthermore, we moved to a human-robot team setup. The human team members took on various roles, such as a Mission Commander, UGV/UAV Operator and/or Mission Specialist.

During a joint exercise at the Firebrigade of Dortmund in summer 2011, end users and researchers teamed up, sitting in a real-life (large) command post to remotely operate the new UGV and to collaborate with an in-field UAV pilot, to explore a substantially more complex disaster site, a burning multi-story building. Robot control, vision, and mapping had significantly improved to move towards building up a robot-centric 3D understanding of the environment it was operating in. Human team members were provided access to this robot-centric situation awareness through an integrated user interface setup, facilitating multiple operational views (camera, map) and tactical views (team-level operations). All of this was simultaneously pushing the state-of-the art and showing how we were taking on board the lessons learnt from working with end users. When reporters once again asked the local firebrigade chief for his opinion on the practical feasibility of these robots, his response was: "The first deployment has moved to foreseeable future." (D. Aschenbrenner, Die Zeit, August 25 2011)

In the **third year** we moved to a scenario of an accident with a chemical freight train at a large terrain. Whereas the previous tunnel accident scenario was set in a relatively confined, enclosed 2D space cluttered with non-traversable objects, the train accident saw the introduction of large open space, a combination of static (bus, train, cars) and traversable 3D objects and structures (stairs, platforms, pallets), which made the terrain passable with difficulty. An in-field rescuer was added to the team, increasing team's size and geographical distribution. For R&D this put 3D robot-control and -mapping into the foreground, as well as the issue of building up and maintaining distributed situation awareness.

The real field-test of that system came during the summer of 2012: Not in the form of experiments at a training site, but during an actual deployment in Northern Italy during the aftermath of the July 2012 earthquakes. NIFTi deployed its then-actual system to support a human-robot team in exploring two large sites for structure damage assessment: A church, and a cathedral. During the missions in the cathedral, the robot was partly operating beyond line-of-sight. During this deployment, the enormous value of having worked with end users before became clear once more. Initial scepticism on the side of local rescue workers gradually changed into full-scale adoption as they came to realise what information our robots could really provide them with.



(a) (b)

Figure 3. (a) NIFTi ground vehicle and (b) system architecture for this single robot considered as a team member.

Furthermore, the difficult circumstances under which we were operating there (long days, intense heat, stress) confirmed the need for human-robot teaming: A team can manage operating in these situations, where a single operator can hardly be expected to endure.

The **fourth year** built on that experience, moving on the roadmap to the earthquake scenario with multiple levels to explore. Specific dynamic areas could only be explored by an UAV, UGV or a human rescuer. Continuous team coordination and communication was crucial for an adequate disaster response. The experiments with this multi-human, multi-robot team were run at two sites of the Italian firebrigade: a USAR training area in Prato and an abandoned, partly destroyed hospital near Pisa.

3. System description

The NIFTi system constitutes a complex ecology of robots, network communication infrastructure, and a multitude of graphical user interfaces (mobile, smart-table, monitor). All of these components share the same network. We use the ROS framework [3] as main middleware for communicating information between these different components.

In summary, for the system & network infrastructure, we used ROS for running processes on the robot. Data was streamed over WiFi to one or more operator control units (OCU) and other visualization tools (RViz), and for logging purposes (`rosbag`'s). Off-board computers were used for processing 3D laser range data (point clouds), and for the OCU and visualization. We used a 2.4GHz WiFi network, with an antenna nearby the entrance to the actual deployment area. The antenna was 50cm long, had 14dBi gain, and was extended with a Ubiquiti high power bullet enabling a transmission power of maximally 28dBm. Each robot (UGV and UAV alike) was also equipped with a bullet, and an omnidirectional rod antenna with a 9dBi gain. As we were mostly working in large open spaces, we did not experience substantial problems with network coverage.

3.1 *Overall system outline*

The NIFTi UGV has been developed in close collaboration between research partners and end users based on the system architecture shown in Fig. 3. The system uses the ROS framework [3] for lower-level control and the CAST middleware [4] for higher-level processing.

Besides the UGVs, light UAVs are used during USAR missions in NIFTi to get an overview. Because of their limited processing power, the UAVs are controlled by a significantly smaller software system, also based on ROS, such that they can be used mainly as flying cameras – either tele-operated or semi-autonomously.

The software system according to Fig. 3 is distributed between the on-board PC of the UGV and some stationary computers at the control center. Because of the message-passing features of ROS, this distribution can be very flexible and is defined by a set of launch files during run-time. However, the ROS stacks of the lowest layer, providing the drivers for the UGV and enabling the platform control, must definitely run on-board. In the layers above, there are several ROS stacks implemented to achieve particular capabilities needed for the USAR domain:

- object detection based on image processing, e.g. to detect victims automatically
- creation of metrical maps based on the LIDAR sensor mounted in front of the UGV
- 2D and 3D metric path planning
- multimodal (speech, GUI) human robot interaction based on the RViz package of ROS

These ROS stacks are connected to the CAST-based components by a bridge that transforms ROS messages to the CAST shared memory and vice versa. The higher layers, which are implemented as CAST components, include:

- conceptual and ontological understanding of the environment based on different kinds of maps (metrical, topological, functional)
- communicating the understanding from these concepts about the environment (including areas, objects and actions to be performed) to other high-level processes
- high-level planning and execution for joint exploration

These high-level components enable the UGV to communicate with a human through natural dialogue and to share its situation awareness within a mixed human-robot team. The following sections cover the essential contributions NIFTi achieved in its final year.

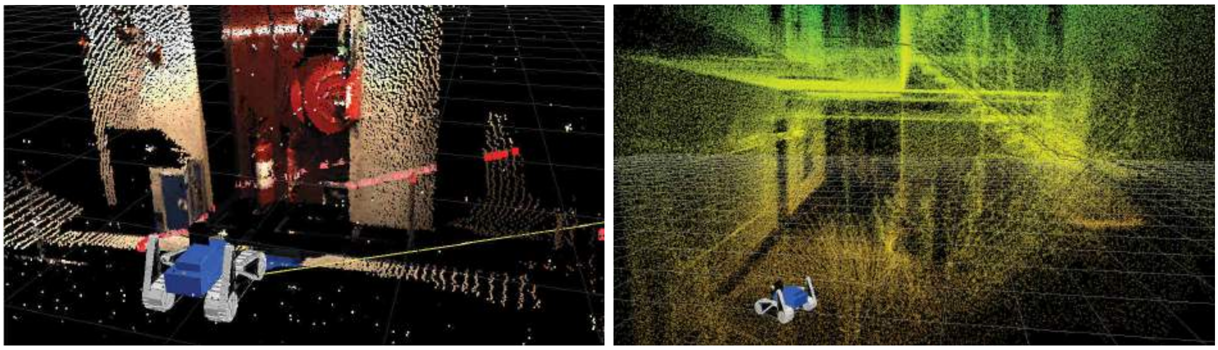
3.2 *Robot-centric situation awareness*

One of the ways to build robot-centric situation awareness is to design algorithms performing mapping of the robot sensory data into situation interpretation in the robot perspective (given for example by the interfaces of algorithms that the robot exploits). Such interpretation of the situation awareness vastly differs from the user perspective since its primary goal is the effectiveness of the used algorithms. In this field we contributed in two ways: first, we developed an algorithm for terrain perception—*the terrain-adaptive odometry* [5]; second, we implemented a robust *3D metric mapping* algorithm; both contributions improve the robot localization and are described below.

3.2.1 Terrain-adaptive Odometry

In our approach we exploited the concept of *sensing through body dynamics* [6] and combined state estimation techniques with machine learning. By using the same type of inertial sensors, the *Xsens MTi-G* unit, we even attempted to prove our concepts across platforms of vastly different morphologies. Initially, the proof of concept was tested and evaluated on a quadruped robot described in [7]. These results were then extended by a new methodology described in [8] and successfully applied to the NIFTi UGV platform [5], [9]. Furthermore, we developed algorithms necessary for processing the raw inertial data; see details in [10].

The *terrain adaptive odometry* algorithm was initially inspired by the research in robotic terrain classification (RTC) [11]. Contrary to the standard RTC approaches, we do not provide discrete terrain categories (usually labeled in the user perspective as sand, rubble, soil, etc.) but we classify the terrain directly by the values of coefficients correcting the robot odometry. These corrections make the odometry model naturally adaptable to the terrain due to inherent slip



(a) (b)

Figure 4. (a) Assembled point cloud with color information. (b) 3D Point cloud map of a staircase.

compensation. During experimental evaluation on rough outdoor terrain the overall improvement we achieved in root-mean-square error in position with respect to a state-of-the-art odometry model was approximately 68%.

3.2.2 3D metric mapping

Based on the terrain-adaptive odometry, we assemble individual laser scans into full 3D point clouds. Such a point cloud is a 3D representation of the contents of the field of view of the scanner at its current position. With proper calibration between the omnicamera and the laser, it is possible to add color information to each individual point (see Fig. 4(a)).

In order to reconstruct a consistent representation, we use `libpointmatcher`[1], our open-source implementation of the Iterative Closest Point algorithm [12, 13]. With this implementation, we are able to process all point clouds online [14] and therefore to build a metric representation of the environment during exploration. An example of a result is shown in Fig. 4(b).

This processing also corrects the pose estimate of the robot which allows for a more precise localization information. It can also be used in a more robust localization framework fusing odometry, inertial measurements, visual odometry, and this laser odometry [15].

3.3 *Bridging from robot-centric to human-centric situation awareness*

Robots use their various sensors to build up a 3-dimensional structural representation of an environment. The primary purpose of that representation is to facilitate robot operations. These include localization, path planning, and autonomous adaptation and -driving.

We have introduced further steps of sensory interpretation, to facilitate more user-centric forms of situation awareness. Given the scenarios we focused on, these include object detection including victim detection, and functional mapping. The sections below describe these in more detail. Like the structural representations, these interpretations can be made available to the users in the multi-modal GUIs.

3.3.1 Object detection

Victim detection in disaster sites proved to be very difficult by using only visual information. We therefore build a layered sensory data representation, see Fig. 5. Thermal image (2D array of temperature) and depth images (2D array of distances) are reprojected (computed) as if they were captured from the same frame (possibly with some data misses due to occlusions). As the RGB-D sensor does not work reliably outdoors we construct a fused depth image, taking inputs from RGB-D and laser line scans, see Fig. 6 (a). The present detector uses depth as a scale estimate for aggregation of temperature and skin color likelihoods, sample results are shown in Fig. 6 (b) and (c), respectively. Distribution of human temperature is modeled as a mixture of

[1] `https://github.com/ethz-asl/libpointmatcher`

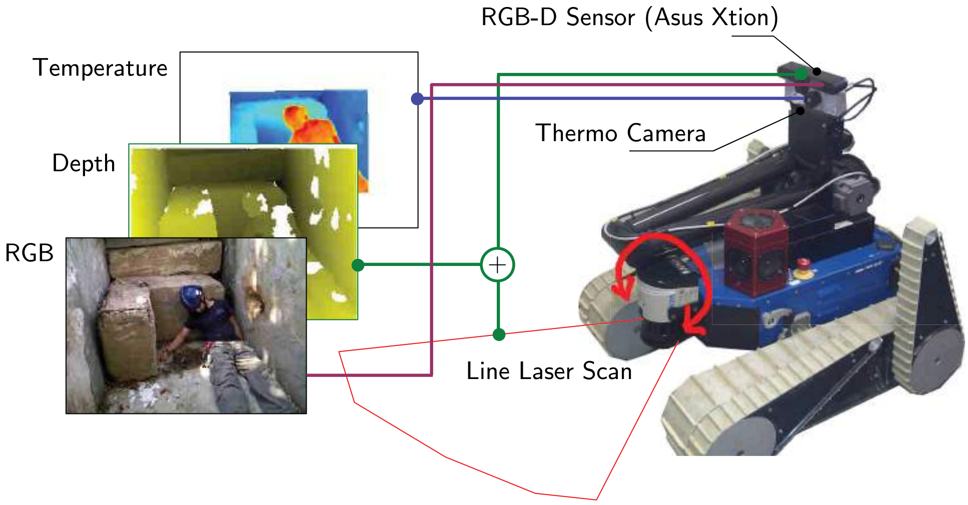
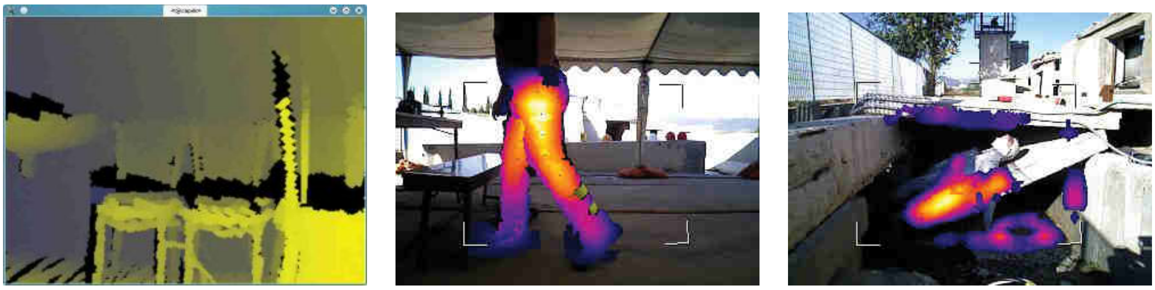


Figure 5. The scheme of multiple modalities (Color, depth, temperature) represented as a layered array computed from different sensors. The Lidar swivels from side to side, capturing full 3D.



(a) (b) (c)

Figure 6. (a) Semi-dense depth computed from laser line scans, while robot undergoes motion (in this case rotation). (b),(c) Examples of human detection. In (c) the false detections are caused by concrete blocks being heated by the sun to the temperature around 36 degrees.

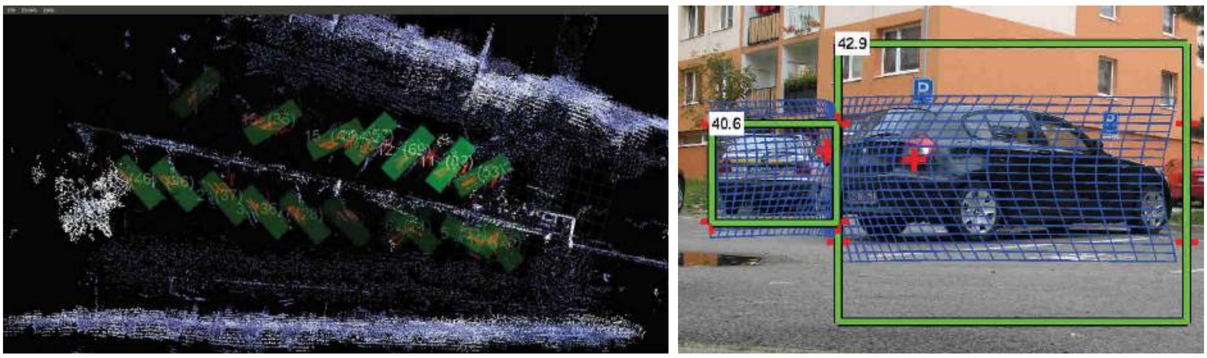


Figure 7. **Car detection and localization:** Left: colored point-cloud map with detected cars denoted by green rectangles. The number with the rectangle corresponds to the number of detections of one car accumulated over time. Right: Typical view point with car detections. Blue mesh corresponds to the estimated non-rigid alignment of the bounding box.

two components - skin and clothes. The parameters of the clothes' component are dynamically adapted according to the temperature of the background. In order to detect non-victim objects we developed a very efficient visual detector [16]. The detector discovers that the successively evaluated features in a sliding window detection process contribute not only to the confidence whether the object is present or not but also contain knowledge about object deformation. The standard sequential decision about the confidence is interleaved with feature warping. The same features are used for confidence and warp estimation. The interleaving process is an essential part of both learning and detection phase. A single detector can be thus used for deformed

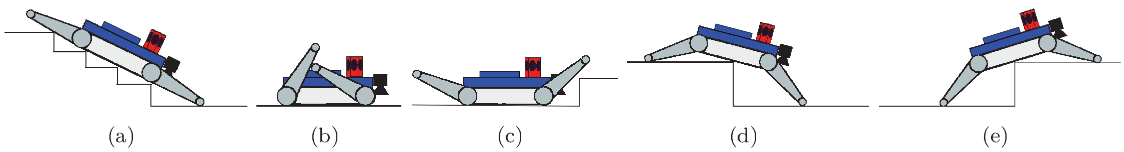


Figure 8. Five flipper modes corresponding to different morphological configurations of varying properties.

objects. The visual detections delineate relevant 3D point cloud data which are then used for distances and orientation estimation. Detections with 3D informations are sent to higher levels of the system. The detector has been applied mainly for car detection and localization, see Fig. 7.

3.3.2 Functional mapping

Robot-centric representations are, by definition, well-suited for robot tasks such as metric localization and navigation. However, in order to better communicate within the team, and especially with human operators, different representations are needed. The first user-centric representation we build is a topological decomposition of the environment into zones. We can perform this decomposition without any supervision [17] but we can also take hints from the users in order to seed the regions the user is interested [18].

These representations are also usually more compact and easier to reason with for high-level planning. Additionally to the topological representation, we also produce maps related to different functionalities the robot can perform; for example looking inside a car.

3.4 *Autonomy and planning*

Below we describe various levels of autonomous planning for different aspects of UGV locomotion. These include adaptive traversability, 3D metrical planning, and mixed initiative planning and dynamic control.

3.4.1 Adaptive Traversability

We define the *adaptive traversability* as means of motion control based on autonomous adaptation of robot morphology to traverse unknown complex terrain with obstacles. The aim is to reduce the degrees of freedom to be controlled (in our case the flippers) by the operator, therefore reducing the cognitive load. The main advantage of our approach is that no prior map or motion history is required, as only the latest incoming sensory data are processed—if some modalities provide data at lower frequency than others the latest measurement epoch is processed.

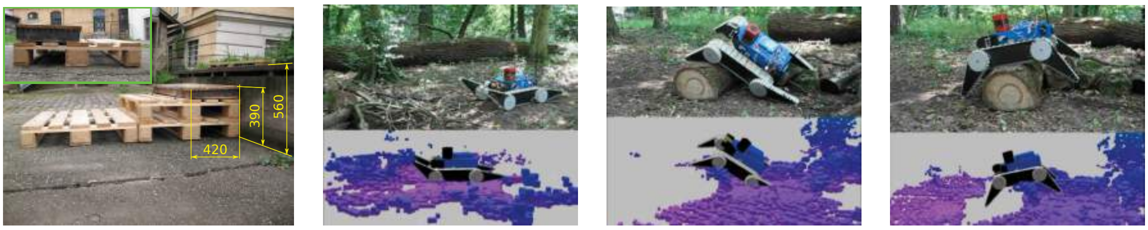
Input to the algorithm is laser data (expressed as Digital Elevation Map, DEM), motor torque signals, inertial data, and tracks odometry; the output is a robot control mode defined by a binary speed decision (stop or go) and a combination of discrete flipper configurations and their stiffness (hard or soft); see Fig. 8.

The control modes are selected automatically to maximize the *expected sum of discounted rewards defined by a reward function*; leading naturally to a reinforcement learning task. We defined the reward function as the weighted sum of (i) user denoted reward reflecting robot safety, (ii) high tilt angles penalty, i.e. negative reward, (iii) excessive flipper mode change penalty, (iv) robot forward speed reward (for making progress), and (v) motion roughness penalty.

To train the algorithm we created artificial obstacles using EUR pallets. To test the algorithm we were driving over natural complex obstacles in a forest environment and compared the performance to an expert operator using different criteria. For illustration a training obstacle and examples of testing obstacles are shown in Fig. 9. More details can be found in [9].

3.4.2 3D metric path planning

3D metric path planning for ground robot presents the difficulty to distinguish obstacles that need to be avoided from ground support. The standard approach is to have variants of elevation maps and do planning in 2D space [19]. This approach is clearly insufficient in case of multi-



(a) (b) (c) (d)

Figure 9. Examples of training (leftmost) and testing obstacles used for development of the adaptive traversability algorithm. The testing obstacles are shown with the corresponding DEM interpretation.

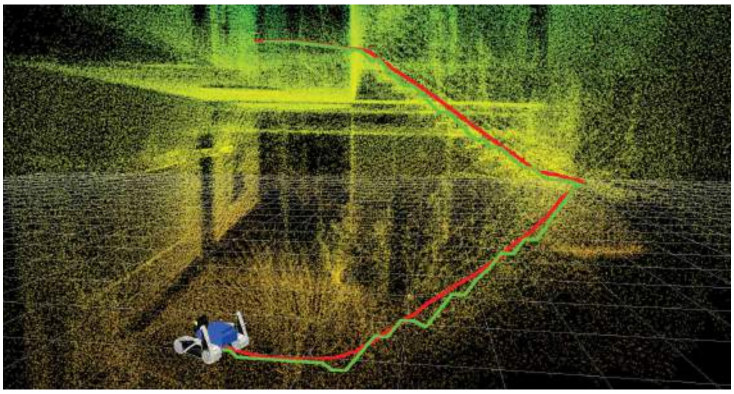


Figure 10. Autonomous descend of a complex staircase. Green line: the initial path planned by the robot. Red line: the actual robot path.

layered environments such as encountered in USAR applications.

Otherwise, planning is done on rich reconstructions of the environment such as polygon meshes. However this requires time-consuming processing and is not compatible with online planning in unknown environment [20, 21]. Instead of separating perception, path planning, and path execution, we propose to tightly integrate them in order to reduce latency.

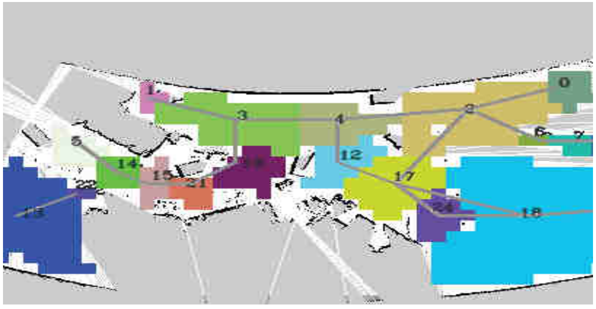
Our algorithm takes a 3D point cloud as input, that will be processed on-demand with tensor voting [22] and uses the D*-lite search [23] in order to plan and replan efficiently during the path. We also select the best flipper position out of a repertoire allowing the robot to overcome most obstacles.

As an example, Fig. 10 shows a trace of execution of our ground robot in a rounded staircase. After a teleoperated exploration of the environment, the robot was asked to autonomously climb back down.

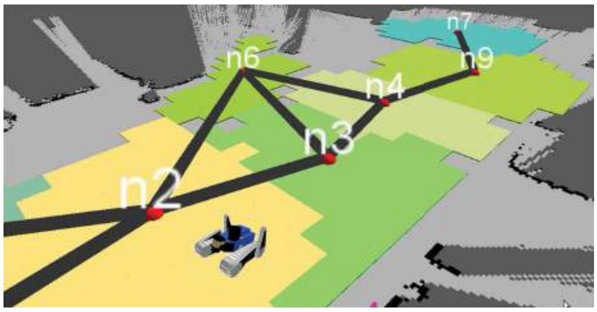
3.5 *Planning, control and execution for human-robot collaboration*

3.5.1 High-level representation of perception

To understand and reason about unstructured and dynamic environments, such as the USAR ones, high-level planning and execution have to primarily address a high level representation of the sensory inputs [24]. This requires to deal with the following problems: (i) extracting relevant features from raw data, gathered by the robot sensors; (ii) building a meaningful, higher level representation of this sensory information and (iii) mapping such a representation into a domain where both reasoning and decision making can take place. The above problems are faced by two complementary processes dealing with the topological and metric representation of the sensory data. The topological representation is defined by a graph of the environment (see Fig. 11). Nodes of this graph can be either objects, detected in the scene, by the visual detector system of the robot (see Section 3.3), or regions obtained by an unsupervised topological segmentation of the metric map (see the paragraph on *functional mapping* in Section 3.3, and



(a)



(b)

Figure 11. Graph-based representation of the topology of the firefighter training car accident scenario in Montelibretti (Italy)

also [17, 25]). Nodes are also annotated with properties related to the detected objects, wherein connections between nodes determine the traversability between the corresponding areas. This graph is directly translated into the domain knowledge supporting parametric planning [26].

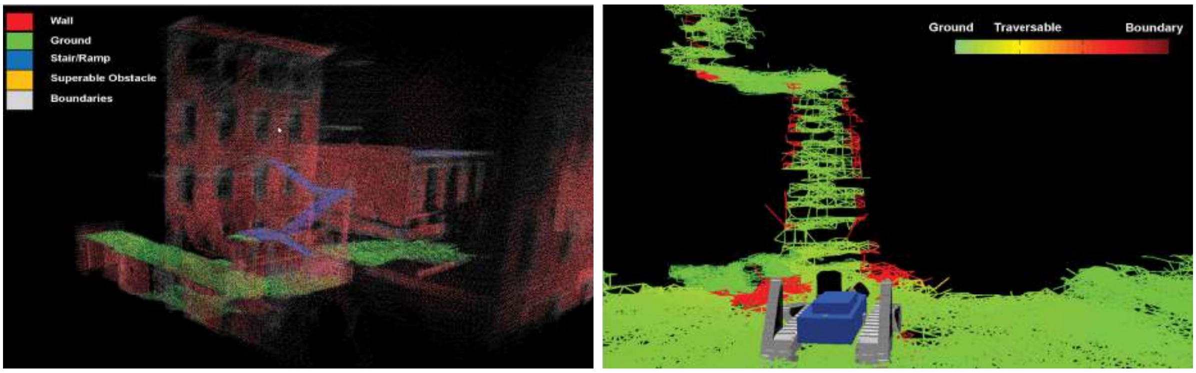
Autonomous navigation tasks in a USAR environment demand a precise representation of the environment in terms of what is traversable and what is not, and in terms of what can be reached and what cannot, for this reason the topological map for 3D planning is complemented by a suitable processing of the point cloud. The point cloud data is segmented and each segment is labeled to provide a basic categorization of the environment, specifically defined for navigation purposes, that is, walls, ground, stairs, ramps, and obstacles that can be overcome (like fences, barriers, blocks of a specified height). Segmentation and labeling are made by the following steps: (1) point cloud filtering; (2) estimation of normals to the surface and curvature and, finally, (3) clustering and merging of the filtered point cloud. Clusters are labeled according to the geometrical constraints applied to the surface normals, to the mean curvature and to the points 3D coordinates [27, 28]. This process results in a classification of the point cloud into walls, stairs or ramps, and ground and surmountable obstacles as illustrated in Fig. 12(a). The semantic labeling of the clustered point cloud is then mapped to a suitable logic representation for inclusion in the knowledge base of the robot system.

3.5.2 Complementary strategies for planning

The robot system makes use of the above defined representation to reason about the environment [29], more specifically, on what can be reached and what cannot, which determines the system navigation strategies. Navigation strategies amount to choosing the best planning method to execute a task. Indeed, if the robot is currently moving on a flat terrain, such as the floor of a room, the robot system can select a 2D path planning algorithm to generate paths toward a temporary goal location, without resorting to a more complex algorithm, such as the 3D metric path planning (see Section 3.4). Another advantage of choosing among alternative planning strategies is that the system can recover from failures in the generation of valid plans, due to the high degree of complexity of the environment [30].

The robot system, in fact, chooses among three different path planning strategies, basing the choice on the terrain surface and topology and on the possible sources of planning failures. The three strategies are: (1) 2D path planning; (2) 3D path planning and (3) 3D graph-based planning [28]. The first strategy relies on the `move_base` navigation stack, provided by ROS[1]. The second strategy is based on the 3D metric path planning, described in sub-section 3.4.2. The third strategy is performed on a graph-based representation of the environment (see Fig. 12(b)). This representation is obtained from the semantic labeling of the point cloud as follows. Points belonging to clusters, labeled as ground and stairs or ramps, are connected based on an iterative procedure taking into account both the model and the kinematic constraints of the robot, namely

[1] `https://github.com/ros-planning/navigation`



(a) (b)

Figure 12. (a) Point cloud segmentation and labeling and (b) weighted graph representation of a fire escape stairs scenario.

its morphology as well as its ability to overcome obstacles. The result of this procedure is a graph connecting the different regions of the point cloud, denoting areas accessible by the robot. In parallel, both boundary and inflated obstacle regions are estimated by projecting the points labeled as walls onto the planes tangent to the surfaces approximating ground, stairs or ramps. Upon the estimation of the boundary regions, the edges of the connectivity graph are weighted by a factor taking into account the distance of the graph vertexes from these boundaries, the density of the neighborhood of the vertexes and the arc length of the edge. This traversability structure is used by the graph-based planning strategy to find minimum cost feasible paths toward target goals [27, 28].

3.5.3 Complementary strategies for motion control

The robot motion control system is augmented with an adaptive low-level control module taking care of switching among three different motion controllers: (1) a Reinforcement Learning (RL) controller (2) a trajectory tracking controller and (3) a flipper position controller [31]. The RL controller is based on the algorithm described in Section 3.4. The trajectory tracking controller, on the other hand, implements a control strategy based on input-output linearization via feedback [31]. The controller takes as input the current pose of the robot, obtained by fusing laser data with odometry and inertial data, the pose of a virtual reference frame, on the desired trajectory, a velocity profile, and generates the linear and angular control commands, in order to asymptotically stabilize the trajectory error to zero.

The flipper position controller locally adapts the position of the flippers to the surfaces on which the path lies, namely to the planes tangent to each point of the path. The controller computes the position commands of the flippers as follows. Four points are identified on the surface on which the current segment of the path lies. These points are representative of the contact points of the flippers with the surface. The controller estimates the normals of each of these points and generates the position commands of the flippers, on the basis of the orientation of the normals with respect to the global reference frame of the robot. Note that the estimation of the normals is not accurate. Moreover, the flippers are neither endowed with contact sensors nor with proximity sensors. Therefore, it is quite hard to correct the estimation as well as to determine the contact between the flippers and the surface. To face this limitation, the flipper position controller relies on a model of contact sensor. This model is based on a learned function, assessing the touch and the detach of the flippers from the surface. The parameters of this function have been learned as follows.

From the measurements of both the actual angles of the flippers and the electrical currents of the flipper motors the following features are extracted: (1) the average of the absolute values of the electrical currents within a fixed time window, (2) the average of the absolute values of the angular velocity of the flippers within the same time window, (3) the sign resulting from the product between the average of the electrical currents and the average of the angular velocity of

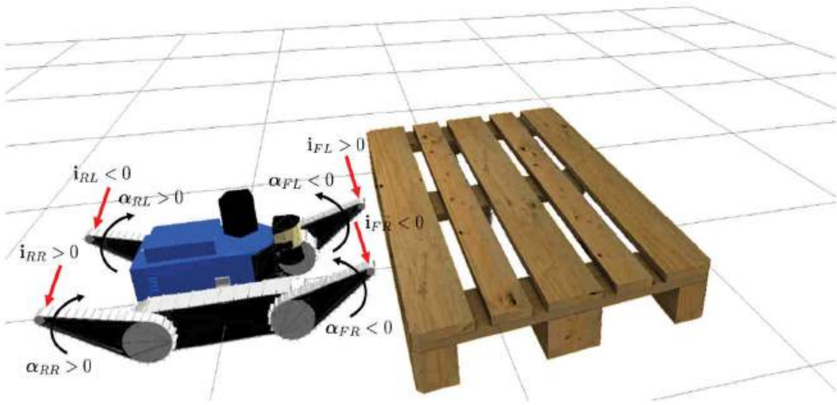


Figure 13. Relation between the values of the electrical currents and the directions of the forces applied to the flippers (red arrows). Relation between the values of the electrical currents and the values of the angles of the flippers (black arrows).

the flippers and, finally, (4) the average of the absolute values of the electrical current, filtered according to the Transposed-Direct-Form-II digital filter (see Fig. 13). The filter has been applied to reduce the oscillations of the signal current, during transient conditions of the servo motor, actuating the flippers. These features have been manually labeled to denote either the touch or the detach of the flippers from the surface. This data-set has been used to train a non-linear classifier, based on Support Vector Machine (SVM). A degree-d polynomial kernel has been chosen due to the non-linear separability of the data-set. The flipper position controller activate this contact sensor to correct the estimation of the position commands. The adaptive low-level control module can decide whether to activates the trajectory tracking and the flipper position controller on the basis of the values of the slip ratio of the tracks, to allow the robot to track a given 3D path and to ensure that the robot has a better traction on the harsh terrain [28].

3.5.4 Autonomous planning and mixed-initiative

The design of an autonomous planner taking into account mixed-initiative requires: (1) task sharing between the robot and the operator; (2) to provide the operator with a clear explanation of the robot behaviors, (3) to allow the operator to choose the level of autonomy of the robot during deployment, on the fly and, finally, (4) to generate goal strategies for information maximization. According to these requirements, we developed a control system for the UGV, with dynamic adjustment of the level of autonomy [26, 30]. The control system coordinates the interventions of the human operator and the low level robot activities, under a mixed-initiative planning perspective. More precisely, the control system is based on a declarative model of the activities of the robot, specified in the Temporal Flexible Situation Calculus (TFSC) [32–35]. The model explicitly represents the main components and activities of the robot, the cause-effect relationships as well as the temporal constraints between the activities. Further, the model integrates a representation of the activities of the human operator, enabling the control system to supervise his/her operations. A flexible planning engine (i) monitors the consistency of the robot and operator activities, with respect to the model, managing failures and (ii) incrementally generates plans, allowing the operator to locally assess the robot operations. Both the TFSC model and the flexible planning engine are implemented in ECLIPSE Prolog [36] which optimally combines the power of a constraint solver with logical inference, in order to generate plans. The model also ensures the continuous update of the system knowledge with incoming new information. A hybrid CAST subarchitecture has been designed to embed the TFSC model and the flexible planning engine, as well as the ROS nodes driving the communication tasks with the ROS layers (see Fig. 14). The choice of designing the hybrid CAST subarchitecture, effectively linking the ROS functionalities with the TFSC model is to overcome two main limitations of ROS: (1) the lack of a native mechanism for sub-typing and polymorphism to manage the data,

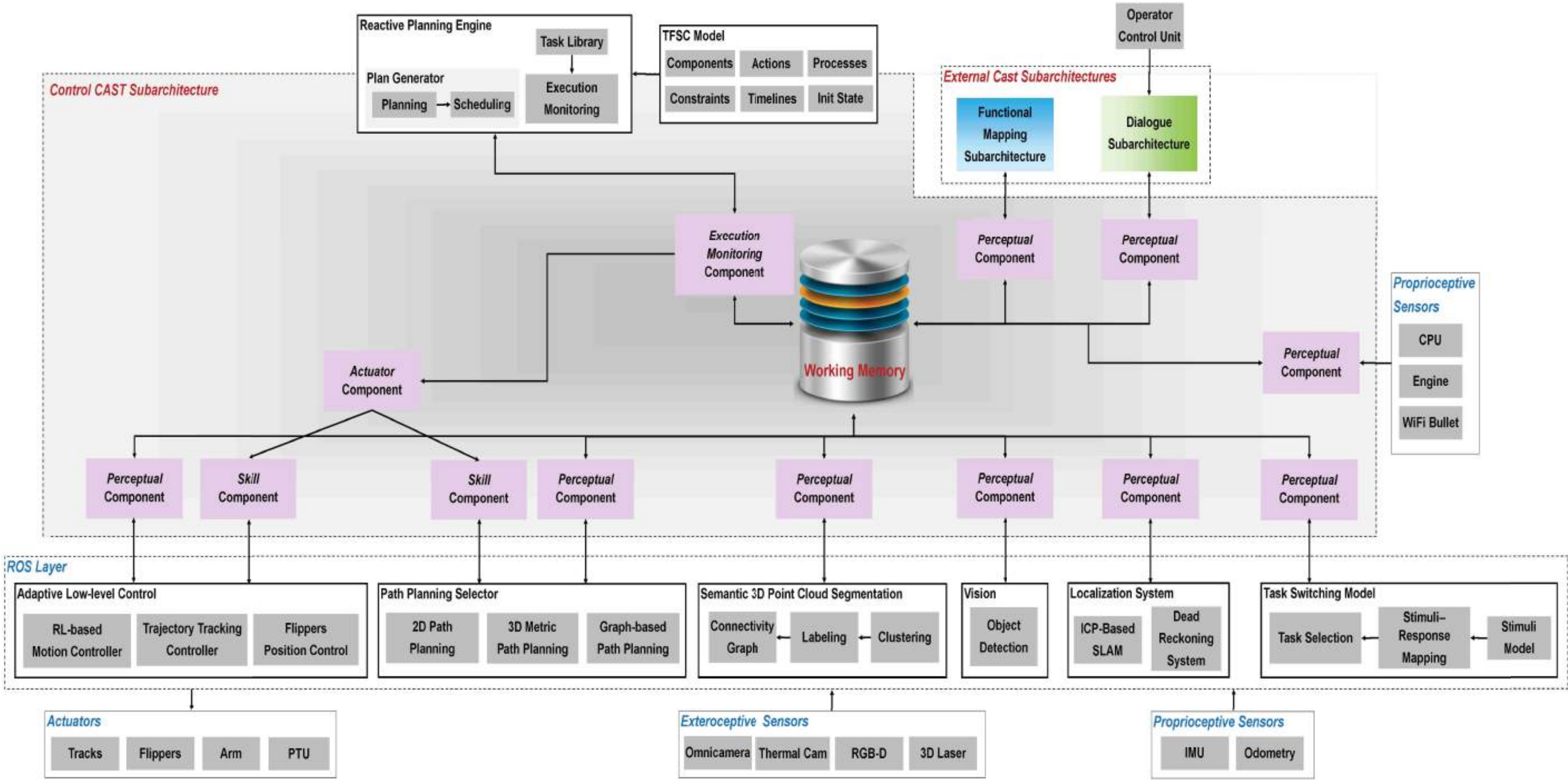
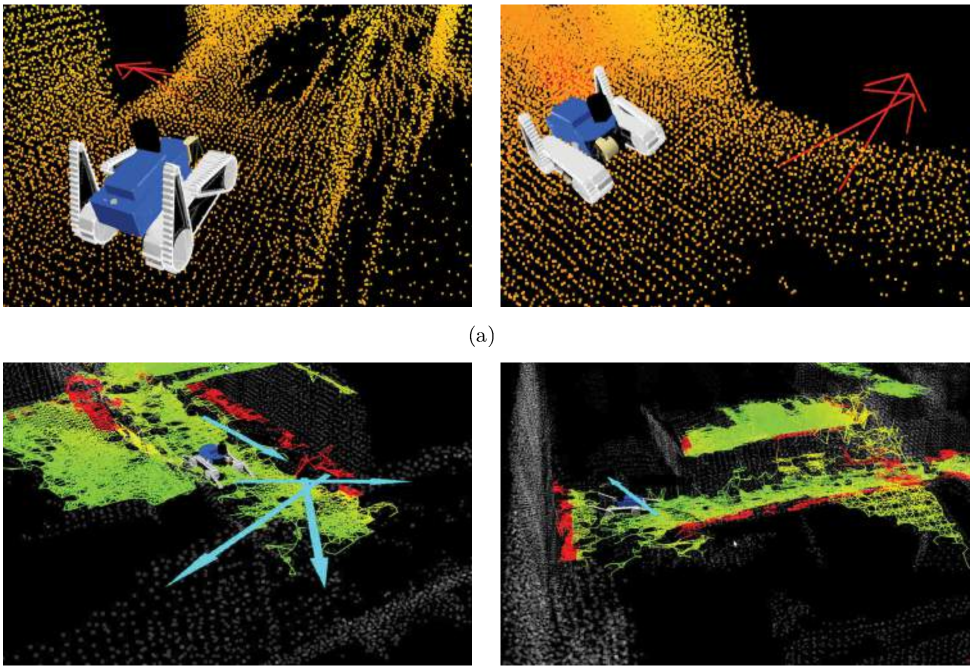


Figure 14. Subarchitecture CAST for semi-autonomous control.



(a)

(b)

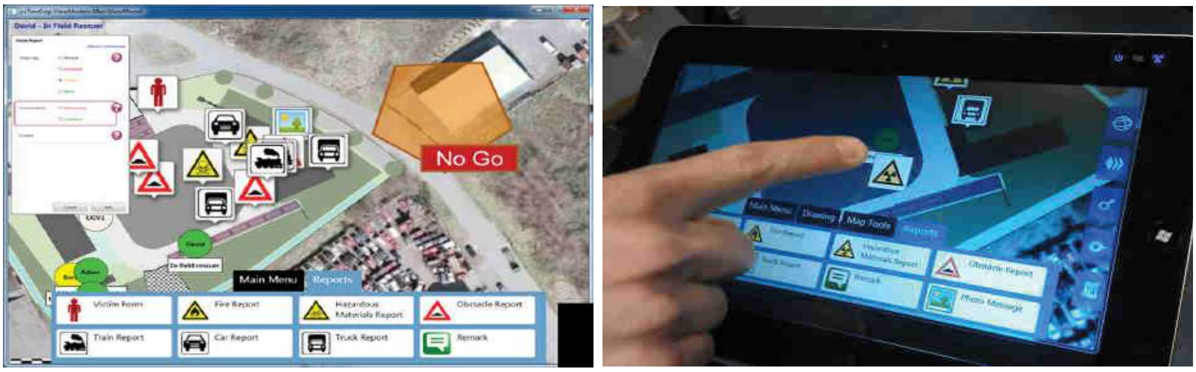
Figure 15. Next Best Pose generation: (a) the candidate poses (red arrows) suggest the robot to pass through a doorway of the corridor; (b) the candidate poses (cyan arrows) favor the acquisition of new information within the explored rescue area.

and (2) the lack of a structure to store the data, for planning and reasoning.

The core of the planning subarchitecture is implemented by the *Execution-Monitoring* (EM) component which plays a crucial role in orchestrating the other components of the subarchitecture. The EM both manages the communication with the human interfaces and maps the logical part of the control into the CAST subarchitecture. Furthermore, the EM manages the communication among all the components of the CAST subarchitectures, forming the robot system, and with the ROS modules implementing the sensory modalities of the UGV robot. The information gathered by the system is stored into the Working Memories (see Fig. 14), and mapped to the domain knowledge of the flexible planning engine. The EM sends task activation signals to the actuator components of the subarchitecture that finally execute the actions generated by the planning engine.

The overall control schema (see Fig. 14) implements several hybrid operative modalities ranging between autonomous and teleoperated modes, available during the execution of a task. The human operator can manually control some functional activities of the robot, scheduled by the flexible planning engine. For example, the operator can take control of the motion to explore an interesting location or escape from difficult environments, by suspending the robot autonomous navigation task. The operator can also modify the control sequence produced by the flexible planning engine, by skipping some tasks or inserting new operations. One of the most sought after operative modalities in navigation is to suggest exploration strategies for the rescue operators, based on information maximization. This modality requires to determine candidate poses, to be reached by the robot, from which most of the unknown space can potentially be observed by both the robot and the operator. Under this perspective, the problem of generating the next best observation positions of the robot can be formulated as a Next Best View (NBV) problem [37–39] where candidate poses are weighted by an utility function.

The utility function maximizes the information returned by a pose in space by estimating the volume of unknown space, falling into the viewing volume of the laser sensor or the robot. More concretely, this value is implemented as follows. For each of the rays cast from the origin of the laser reference frame toward all the possible directions a local utility is computed. The local



(a) Geo referenced information

(b) Mobile display

Figure 16. Content display.

utility takes into account the amount of free, occupied and unknown cells of the 3D occupancy map of the environment, hit by the ray.

Then the global utility associated with a pose is computed by integrating the local ones on the view cone domain. To speed-up the computation of the utility of a pose, rays are sampled within a solid angle, bounded by the limit angles of the laser sensor. Moreover, candidate poses are sampled from the connected component of the traversability graph (see Fig. 12(b)), on which the robot is moving, according to a Gaussian distribution, conditioned by the current position of the robot and the past position history. The next best view is the pose, among the sampled candidate poses, which maximizes the defined utility function. Fig. 15 shows how the robot infers a subset of next best views, on the basis of such exploration strategy.

The flexible planner, when in exploration modality, uses this strategy to generate goals and infer the adequate path to reach them, and evaluate with the EM the feasibility of the plan.

3.6 *User-centric/adaptive interaction and situation awareness*

Key to effective human robot teaming is proper communication of information between the different team members, to build up shared situation awareness. The human team members are ultimately the main stakeholders in this process. In keeping with the overall design methodology, we have also here adopted a strong (human) user-centric development. Below we describe the development of our interaction setup, and provide more details about two aspects that help bridging the gap between robot-centric environment models and human understanding.

We initially started by investigating and supporting multimodal one-to-one communication between a robot and an operator. After the experience of our initial pilot studies, we gradually shifted our focus to team-level communication. As the team was gradually extended, sharing and managing situation awareness became a crucial issue. To facilitate this, we designed and implemented ever more advanced multimodal user interfaces.

We initially focused on communication to support human-guided exploration by the robot. The human operator was remotely located, outside of visible range of the robot operating in the hot-zone. We designed and implemented an operator control unit (OCU) that facilitated multi-modal interaction (GUI, spoken dialogue) between a human and a robot. Based on the experience from the pilot study, we formulated a first approach to modelling the dynamics of the interdependent roles in a human-robot team. We focused primarily on determining how communication between roles in a team is affected when the team performs under varying stressful circumstances, so that a robot can adapt its multi-modal communication strategies given online human performance. This provided the setup for developing communication to support human-assisted exploration in the context of a human-robot *team*. This went beyond the originally envisioned setup of a single operator.

The physical setting for the interaction remained such that most of the human team is located

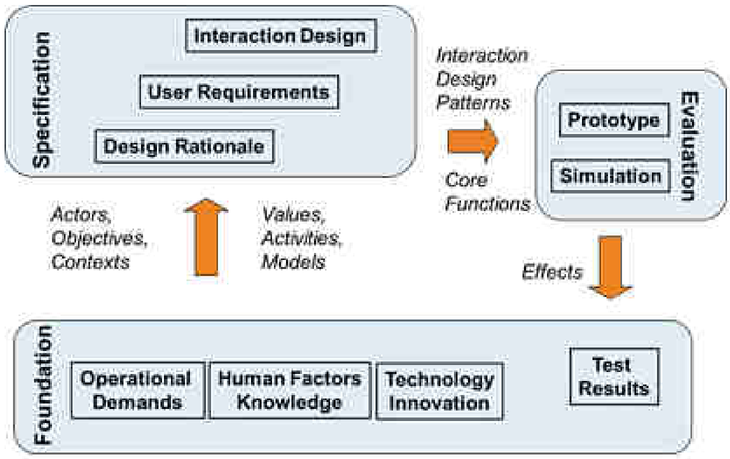


Figure 17. Situated Cognitive Engineering (sCE) design and evaluation process [42].

at a remote command post, outside of visible range of the robot operating in the hotzone. We developed a multi-view user interface to facilitate different views on information in the human-robot team, to help support different roles in the team. The interface provides multiple modes of communication, including touch and spoken dialogue. Views (possibly shown on multiple monitors) include the visualization of information from the various robots (UGV, UAV), and team situation awareness.

Beyond the remote command, we have extended the human-robot team setup to include an in-field human rescuer. This resulted in a further geographic distribution of both robot and human team members. We developed and investigated different versions of mobile interfaces to facilitate multi-modal communication between the in-field human rescuer, and the rest of the remotely located human team members. We followed the approach to support different team roles with different views and functionality for human-robot team interaction. In this context we also developed a novel method for content adaptation, i.e. presenting the "right material at the right time, in the right modality" [40, 41]. As information from an in-field rescuer typically comes "asynchronously" relative to when a robot is in the observed area, we have developed new means of storing and presenting such (geo-referenced) information at the operational and tactical levels of communication in the human-robot team (see Fig. 16).

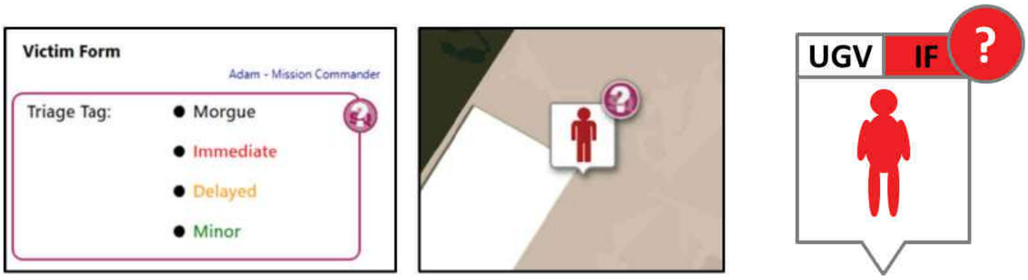
3.7 *Teaming*

In order to support the team effort, the system builds up and maintains knowledge on the users, supporting them to stay in a continual workflow, by attuning the information processing and sharing to the task at hand. First, this involves making sense of what the users are doing, in terms of their current task, their cognitive task load (CTL) and emotional state. Second, the team and user context can be exploited by suggesting the appropriate level of autonomy for the task at hand, or notifying other team members to help their colleague.

The NIFTi system gives us a rich environment for collecting data on the users' behavior, and for the design and implementation of such team support functionality. Based on the data gathered in our end user experimentation, we have tailored the CTL and emotional state models, and integrated them into the prototype system, where they were incorporated into a formal framework for dynamic task allocation and adaptive dialogues. We also collected a data-set for conducting 3D eye-tracking experiments to further develop computational visual attention models for top-down search tasks. Over the course of the project, we have also refined the methods for balancing the information transfers by means of policies for information exchange of team members with specific roles and capabilities. These working agreement policies, designed in collaboration with our end users and evaluated during the end user experiments, aim at establishing coherent and

	USAR robot	
Communication level	Interaction design pattern: for look and feel of smart question	Premise: smart questions are easy to use questions
Task level	Requirement: Provide automatic object recognition	Claim: Improved object detection
Group level	Requirement: Provide information on location and recognized objects teammates	Claim: Improved object detection

Table 1. Design patterns, premises, requirements and claims on different levels.



(a) (b)

Figure 18. (a) Explicit unknown design and (b) re-design.

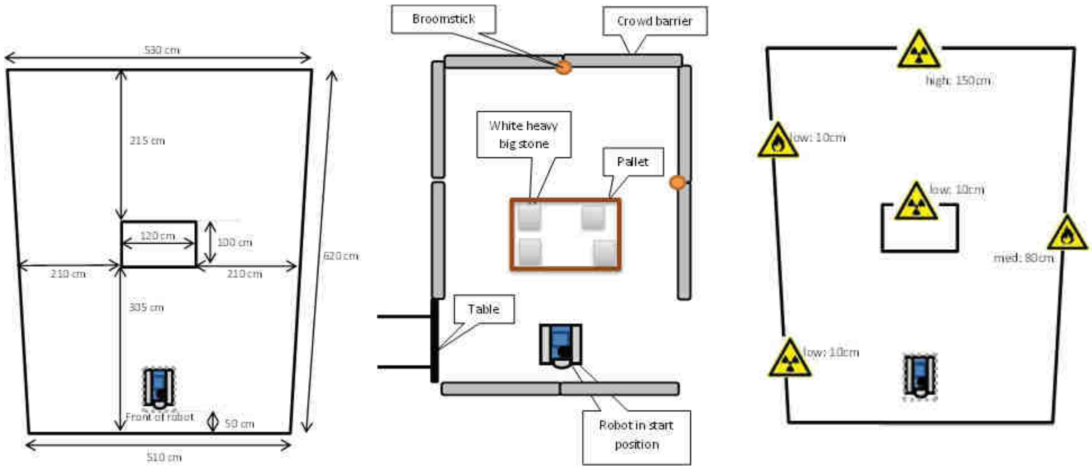
re-usable specifications of human-automation interaction at the communication level.

4. Experimentation

Each year the project went through a development cycle of a description of scenarios and use-cases, derivation of requirements and hypotheses, implementation of the system components, integration, and finally an evaluation. Fig. 17 shows the three components of this situated Cognitive Engineering (sCE) methodology: the foundation entails operational, human factors, and technological analyses to derive a sound and practical design rationale, the specification and maintenance of the requirements baseline, and the evaluation by means of simulation or a prototype, to validate and refine the requirements baseline [43], [42]. The requirements and hypotheses were described on three levels, the communication level, the task level and the group level [44]. An example of each level in the context of a robot for USAR is given in Table 1. The task level and group level can both be evaluated with small tasks (Task battery) or within a scenario.

4.1 *Design patterns*

The manner of information presentation influences the performance of the user enormously. To approach the evaluation of this presentation in a structured manner and to support development and reuse of good design solutions we use design patterns [45]. Table 1 provides an example of a design pattern that has been evaluated in the fourth year of the project and Fig. 18 provides the evaluated design and the suggested design improvement based on an evaluation. In the evaluation, 18 subjects evaluated the design using a questionnaire and they noted some problems in understanding who was responsible for answering the information request. Therefore information about who should react on the information request was added (the UGV (operator) in the case of the improved design figure).



(a) Dimensions (b) Materials (c) Scenario

Figure 19. Set-up of the detect objects task

4.2 *Task battery*

To benchmark the progress in the yearly changing scenarios (see Section 2.2) we developed small tasks that evaluated some basic functionality (e.g. stopping the UGV before collision). This set of tasks was extended over the years to incorporate the growing capabilities of the systems. A first year task was for instance 'detect objects' in a maze like environment (see for example Fig. 19) to look at the performance of the combination UGV, UGV control interface, situation awareness display and operator. This task was kept over the years to see if changes in the combination had an (positive) effect on operator performance (task level). Because in the fourth year there was a complete team of two operators and a commander the same task was performed but with both a UAV and UGV (group level). This made it possible to not only compare the results over years, but also over different team compositions. The complete system with UAV and UGV and their operators and a commander provided in the fourth year of course involved opportunities which entailed an extra test where the complete team had to detect objects in an area.

4.2.0.1 Measures. Each task had its own set of measures. For the object detection task these were: performance (How many objects are detected) and situation awareness (participant has to draw how they drove through the area and draw the UGV's start and end position). To measure the performance on a task like 'explore area', the performance was measured as amount of area covered, the efficiency was measured by taken path (ratio length of path and explored area) and the same measure for situation awareness was taken.

4.3 *Scenario*

Next to the task battery, the scenario as described in Section 2.2 was performed by multiple firemen each year. The setup of the fourth year (see Fig. 1) shows that a capability such as objects detection is also relevant here. The scenario is less structured than the task battery tasks but the same requirements can be tested. Another difference from the task battery is that multiple tasks are to be executed and the participants always work in a team and are free to use the operation mode of the UGV they prefer.

4.3.0.2 Results. One of the things we are interested in is if performance on the task battery is predictive for the scenario performance. In [46] it is shown that the "detect object" task does partially predict the scenario performance.

5. Deployment

In May 2012, two major earthquakes occurred in the Emilia-Romagna region, Northern Italy, followed by further aftershocks and earthquakes in June 2012. This sequence of earthquakes and shocks caused multiple casualties, and widespread damage to numerous historical buildings in the region. The Italian National Fire Corps (CNNF) deployed disaster response and recovery of people and buildings.

In June 2012, they requested the aid of NIFTi, to assess damage to historical buildings, and cultural artefacts located therein. To this end, NIFTi deployed a team of humans and robots (UGV, UAV) in the red-area of Mirandola, Emilia-Romagna, from Tuesday July 24 until Friday July 27, 2012. The team worked closely together with the members of the CNVVF involved in the red area. Below we briefly summarise our experience; see [47] for more detail.

In Mirandola we deployed a subset of the available NIFTi functionalities, described above. We focused on robust functionalities for robot control, video streaming from various omni-directional and monocular cameras (UGV,UAV), and laser-based 3D reconstruction of the environment (UGV), coupled to the NIFTi multi-modal OCU.

We deployed two NIFTi UGV platforms in Mirandola: One as the main system, and one in reserve should something go wrong. Fig. 20 shows the UGV platform used. In addition to the usual sensor suite, we mounted a $25cm$-tall static mast on the battery compartment of the robot. On top of the mast was a pan-tilt unit with a Kinect camera. This provides a chase-style view of the robot, which is highly useful when navigating (tele-operating) the robot in tight or complex spaces – cf. also the recent experience with Quince reported in [48]. During the deployment we had the robot running all day long, under outside temperatures of $35 - -40\,^{\circ}\mathrm{C}$ and operating temperatures inside the robot up to $95\,^{\circ}\mathrm{C}$. Batteries only needed a recharge in the evening.

Figure 20. NIFTi UGV with a rotating SICK-Laser (LMS100), a LadyBug3 omnicam, active flippers and active/passive bogeys, IMU, GPS, and a static mast mounting a PTU with a Kinect sensor.

Two different types of UAVs were prepared for the mission (Fig. 21). One was a NIFTi UAV microcopter platform, the other a research platform which we could flexibly outfit with a variety of cameras, e.g. a high-definition camera or an ASUS Xtion Pro.

The human-robot team included operators for the UGV and UAV, UGV and UAV mission specialists, and a Mission Commander. Both the UGV Operator and the UAV Operator suffered from cognitive overload. UGV missions typically lasted about half an hour, and were characterized by interleaving driving, and observing. This interleaving made it possible for the UGV Operator to relax, momentarily; a luxury the UAV Operator did not have. The UAV did have some degree of autonomous flight control, but circumstances demanded that the UAV Opera-

(a) (b)

Figure 21. Standard NIFTi UAV octocopter with a standard configuration (a) and NIFTi UAV research octocopter (b) with a mounted on top camera, and a camera in a tilt unit under the main body. Another configuration flown includes a PC and a Kinect-style sensor mounted on top of the research UAV.

Figure 22. 3D reconstruction based on NIFTi UAV data

tor continuously attended to the UAV. This provides a first insight in possible roles of "robot autonomy." In human-robot teams, humans and robots are (inherently) interdependent [49]. Robots can go where humans need to but cannot, whereas humans can aid robots in better understanding and operating in the environment. Both humans and robots are problem-holders – with the obvious "but" though that *the human users are the stake-holders*. Robot autonomy is ultimately to be in service of the human user, to reduce cognitive load (improved autonomous navigation, sensor data interpretation, collaboration) and to improve the possibility for the hu-

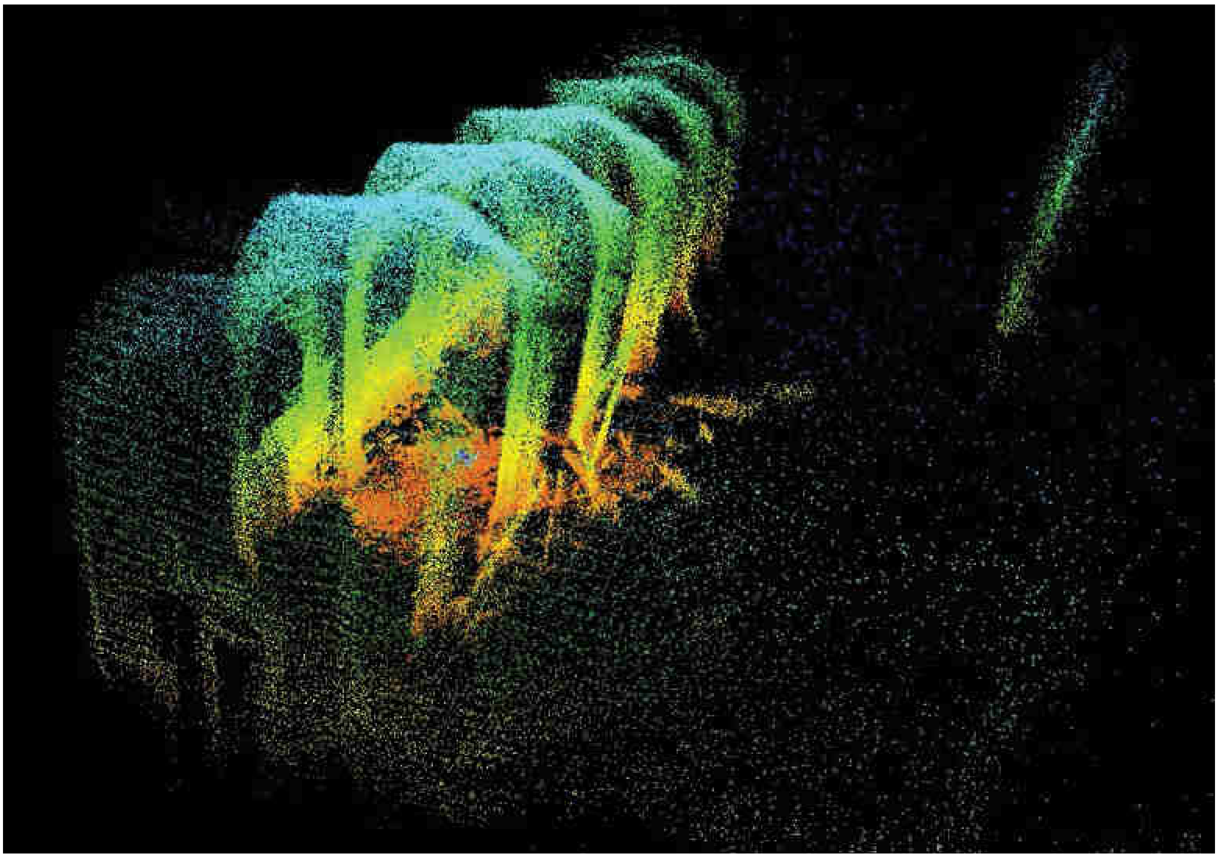


Figure 23. 3D map constructed by the NIFTi UGV.

man to collaborate with the robot as if "operating the world rather than the robot" [50]. We saw this over and again during the deployment: Autonomy is to make life easier for the human to understand the environment.

The UAV serves as a good example here. The UAV mission specialist used augmented reality eyewear (*Vuzix WRAP 920AR+*) to watch the video stream from the camera mounted in a tilt-unit under the UAV. This quickly led to a pseudo-immersive experience, and the desire to look left-and-right and have the UAV and/or the tilt-unit follow suit. More (and better) flight control autonomy, enabling the UAV to simply hover and turn on the spot, would have facilitated this.

Further insights concern the flow of information between the UAV team and the UGV team, in terms of tactical (team-level) situation awareness (tacSA) and mission planning. During the entire deployment, the UAV team and the UGV team never operated in the same area simultaneously. Partly, the reasons were technical (network) and environmental (dust). Another reason regarded the *use*, the workflow which emerged in using information from the different teams in establishing further missions. Based on in-field line-of-sight observations of the area to be deployed in, and a first set of recon missions by the UAV team, we would establish a first sketch of the environment. Most importantly, we would identify important landmarks to navigate by, establishing explicit names for them (e.g. "column 4"), and determining targets for future missions. These targets typically included areas and objects to be observed, and how these observations were to be made. Targets were discussed together with members of the CNVVF.

Follow-up missions then helped detail out situation awareness and revise mission targets. Since awareness was coming from the different teams, we occasionally found mismatches in expectations which then required further missions; (as was to be expected, cf. [51]). For example, video from initial UAV recon missions in San Francesco church gave the impression that the top of the nave would be reachable from the western aisle, either from between the fourth and fifth columns, or the opening behind that. This would then make it possible for the UGV to drive close to the altar, and provide close-up video. As it turned out at the end of the second UGV mission, what seemed accessible terrain from the viewpoint of the UAV, was not so in UGV-reality. The UGV did manage to take video of the altar, but an additional mission was then planned for the UAV to fly in over the main nave and record video from that viewpoint.

The UAV and the UGV thus supported each other, but indirectly so. It did result in the required situation awareness for the team, and the other stake-holders. At the same time, it also opened new questions as for how to optimally transfer data from one mission to the next, to make consolidated awareness available online. Before the deployment, we had developed a basic viewer for post-mission analysis. During a mission, a Mission Specialist could take snapshots in an OCU, annotate them with a description. Snapshots were stored with the text annotation and robot position information. For post-mission analysis, the viewer could then load snapshots and a 2D map, mark the snapshots on the map, and enable the user to browse snapshots. We did use some of this functionality, particularly to get high-definition snapshots of cultural artifacts, but what was missing was the possibility to correlate *geo-referenced video* from one mission, and show this during another mission in a *context-aware fashion*, i.e. show previously recorded video of the environment in which the robot in the current mission is located. This is a form of information fusion to provide continuous situation awareness across different missions within a single area. We made similar observations about map information. The UAV could be deployed to gather a 3D reconstruction of the environment. This map would not need to be so detailed as to enable the UGV to localize itself in it. All the map would need to make possible is a form of *forward mapping/scouting* for the UGV team to determine the optimal path amongst different alternatives. While operating in a harsh environment like the ones in Mirandola we would have greatly benefited from such functionality, as it could have saved time, or have indicated paths where none were obvious (like a traversal from the western aisle to the nave in San Francesco church). See Fig. 22 and 23: Coupling the UAV 3D information to the dense 3D metrical representation for the UGV could improve situation awareness for the Operator as well as the robot.

In summary, we observed several issues regarding the operations of a geographically distributed human-robot team, with team members operating both in-field and at a remote command post. As the UGV and UAV teams operated asynchronously, maintaining and transferring situation awareness between missions was an issue to the extent that system automatization could help (in the future) to make aspects of operational situation awareness from one team available to the next in an operational context-aware fashion. We thus need to address persistence of information.

6. Conclusions

We presented an overview of what we achieved in the NIFTi project. We have robust models of 3D dynamic environments, fusing information from a wide variety of sensors. We have highly robust and adaptive robot platforms (UGV, UAV) that use these models to operate in the complex environments typical for disaster response. And we have embedded all of that information, all these platforms, into the use context of a human-robot team. Humans have access to information at different operational levels, to form an assessment of the overall situation, and collaborate with robots as team members to guide further operations. The way information is presented takes into account that these contexts are stressful, with people working under varying cognitive load. What is displayed, how, and when, is adapted to fit the current load and usage. While further improvements are possible and needed at the level of the individual functionalities and system components, the important global achievement of NIFTi is that robots, and the information they provide, have been made useful to people. NIFTi achieved this through closely integrating research and development with a scenario- driven roadmap, end users, and real-life experiments. Reality, and real end user demand, drove the NIFTi R&D – and the NIFTi R&D showed that, despite the fact that robot-assisted disaster response is a complex and difficult task, we can make substantial advances towards real-life deployments of these systems. The NIFTi Mirandola deployment is an example of that.

Through our practical experiences we have learnt that it is important to go beyond the single-robot single-operator paradigm and consider the operation of the response team as a whole with the robots as team members who collaborate with humans as well as among each other. Clearly, more research is needed in this area, to further improve distributed situation awareness as well as dynamic levels of autonomy, adapted to task, the situation and the team-members capabilities and needs.

Another aspect the importance of which we have identified through our practical experience is that such missions take time. In Mirandola we ran multiple missions over one or more days to explore a site. A robot does not just drive in and out, and the mission is done. In order to support this, a system needs to maintain and continue to update information over the course of missions. Information, and human-robot team experience as such, must become persistent. Based on our experience we see persistence as a key challenge for further progress in human-robot disaster response: we need persistent models for perception, of acting, of distributed joint situation awareness, of collaboration and human-robot teaming. We are addressing the challenges involved in taking such *long-term ecological perspective on robot-assisted disaster response* in our next project, TRADR [52].

Acnowledgments

The work reported in this article was supported by the EU FP7-ICT-Project NIFTi, Contract No. 247870.

Notes on contributors

Geert-Jan Kruijff is a Senior NLP Research Manager at Nuance Communications Germany, and a Research Fellow at the German Research Centre for Artificial Intelligence. From 2004 until 2013 he was active in the field of Human-Robot Interaction, focusing on situated dialogue processing. He led the HRI efforts at DFKI Saarbrücken, and was coordinator of the NIFTi project for the period 2010-2013. In 2013 he joined Nuance, where he leads the development of language resources for cloud-based natural language understanding, for a variety of consumer projects in the mobile and automotive domains.

Francis Colas is currently senior researcher at Inria Nancy Grand Est. After his Ph.D. in Computer Science from the INPG in 2006, he was postdoctoral fellow at Collège de France in Paris in the *Laboratoire de Physiologie de la Perception et de l'Action* (2007–2008). From 2009 til 2014 he was a postdoctoral fellow at the Autonomous Systems Lab of Prof. Siegwart (ETH Zurich). His research interests include Bayesian modeling of perception and action applied from cognitive sciences to robotics.

Mark Neerincx Mark Neerincx is full professor in Human-Centered Computing at the Delft University of Technology, and senior research scientist at TNO Perceptual and Cognitive Systems (Netherlands). He has extensive experience in fundamental and applied research on human-automation collaboration, among other things in the domains of health, security, defense, and space; as initiator and project leader, in large national and international research programs. Recent projects focus on the situated cognitive engineering of electronic partners (ePartners) that enhance the social, cognitive and affective processes in human-automation (robot) teams.

Fiora Pirri is currently full professor and head of the ALCOR Laboratory at the DIAG, Department of Computer, Control, and Management Engineering, "A. Ruberti" at Sapienza, University of Rome. She founded ALCOR Lab in 1998 and since then the Lab has widely to the theory and practice of Cognitive Robotics applied to rescue environments, winning several awards among which the AAAI-2002 Robot Exhibition, third award and finalist at the RESCUE RoboCup competitions of 2004, 2005 and 2006. Best paper award at ISVC 2010, finalist at SSRR2013. Her research area is Cognitive Robotics, and in particular Perception, Attention and theories of Actions, on which she contributed with more than 130 peer reviewed papers which can be consulted on Google Scholar.

Dipl.-Inform. Rainer Worst: Senior researcher at Fraunhofer IAIS. Diploma in Computer Science (Univ. Bonn, 1981). IT-Expert at a consulting company in Wiesbaden. In 1984 joined the former GMD (National Research Center for Information Technology) in Sankt Augustin, which merged 1999 with Fraunhofer. Experience as a consultant, researcher and project leader in the areas Software Engineering, Quality Management, Autonomous Systems, and Virtual Environments.

Tomáš Svoboda: Ph.D. in artificial intelligence and biocybernetics from the Czech Technical University in Prague in 2000. Post-doc with the Computer Vision Group at the ETH Zurich (Swiss Federal Institute of Technology) for 3 years. Currently associate professor and deputy head of the department of Cybernetics at the Czech Technical University. Research interests include omnidirectional cameras, image based retrieval, and machine perception for robots in general.

References

[1] NIFTi (Natural Human-robot Cooperation in Dynamic Environments). 2013. Available from: `http://www.nifti.eu`.

[2] van Diggelen J, van Drimmelen K, Heuvelink A, Kerbusch PJ, Neerincx MA, van Trijp S, Ubink EM, van der Vecht B. Mutual empowerment in mobile soldier support. Journal of Battlefield Technology. 2012;15(1):11.

[3] ROS home page. 2014 March. Available from: `http://www.ros.org/`.

[4] Hawes N, Wyatt J. Engineering intelligent information-processing systems with CAST. Advanced Engineering Infomatics. 2010;24:27 – 39. Available from: `http://www.cs.bham.ac.uk/~nah/bibtex/papers/haweswyatt10aei.pdf`.

[5] Reinstein M, Kubelka V, Zimmermann K. Terrain adaptive odometry for mobile skid-steer robots. In: Robotics and automation (icra), 2013 ieee international conference on. 2013. p. 4706–4711.

[6] Iida F, Pfeifer R. Sensing through body dynamics. Robotics and Autonomous Systems. 2006; 54(8):631 – 640. morphology, Control and Passive Dynamics.

[7] Reinstein M, Hoffmann M. Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer. In: Proc. ieee int robotics and automation (icra) conf. 2011. p. 617–624.

[8] Reinstein M, Hoffmann M. Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information. IEEE Transactions on Robotics. 2013 April;29(2):563–571.

[9] Zimmermann K, Zuzanek P, Reinstein M, Hlavac V. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In: Proc. ieee int robotics and automation (icra) conf. 2014. Accepted for publication.
[10] Kubelka V, Reinstein M. Complementary filtering approach to orientation estimation using inertial sensors only. In: Proc. ieee int robotics and automation (icra) conf. 2012. p. 599–605.
[11] Tick D, Rahman T, Busso C, Gans N. Indoor robotic terrain classification via angular velocity based hierarchical classifier selection. In: Proc. ieee int robotics and automation (icra) conf. 2012. p. 3594–3600.
[12] Chen Y, Medioni G. Object modeling by registration of multiple range images. In: IEEE International Conference on Robotics and Automation (ICRA). 1991. p. 2724–2729.
[13] Pomerleau F, Colas F, Siegwart R, Magnenat S. Comparing ICP variants on real-world data sets. Autonomous Robots. 2013;34(3):133–148.
[14] Pomerleau F, Magnenat S, Colas F, Liu M, Siegwart R. Tracking a depth camera: parameter exploration for fast ICP. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE. 2011. p. 3824–3829.
[15] Kubelka V, Oswald L, Pomerleau F, Colas F, Svoboda T, Reinstein M. Robust data fusion of multimodal sensory information for mobile robots. Journal of Field Robotics. in press;.
[16] Zimmermann K, Hurych D, Svoboda T. Non-rigid object detection with local interleaved sequential alignment (LISA). IEEE Transactions on Pattern Analysis and Machine Intelligence. 2014 April; 36(4):731–743.
[17] Liu M, Colas F, Siegwart R. Regional topological segmentation based on mutual information graphs. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE. 2011. p. 3269–3274.
[18] Liu M, Colas F, Pomerleau F, Siegwart R. A Markov semi-supervised clustering approach and its application in topological map extraction. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE. 2012. p. 4743–4748.
[19] Tompkins P, Stentz A, Wettergreen D. Global path planning for Mars rover exploration. In: IEEE Aerospace Conference. 2004. p. 801–815.
[20] Singh S, Kelly A. Robot planning in the space of feasible actions: Two examples. In: IEEE International Conference on Robotics and Automation (ICRA). Vol. 4. 1996. p. 3309–3316.
[21] Wettergreen D, Moreland S, Skonieczny K, Jonak D, Kohanbash D, Teza J. Design and field experimentation of a prototype lunar prospector. The International Journal of Robotics Research. 2010; 29(12):1550–1564.
[22] Medioni G, Tang CK, Lee MS. Tensor voting: Theory and applications. In: Reconnaissance des formes et Intelligence Artificielle, 2000. Proceedings of the Conference on. 2000.
[23] Koenig S, Likhachev M. Fast replanning for navigation in unkown terrain. Transactions on Robotics. 2005;21(3):354–363.
[24] Pirri F. The well-designed logical robot: Learning and experience from observations to the situation calculus. Artificial Intelligence. 2011;175(1):378–415.
[25] Papadakis P, Gianni M, Pizzoli M, Pirri F. Constraint-free topological mapping and path planning by maxima detection of the kernel spatial clearance density. In: Proceedings of the international conference on pattern recognition applications and methods. 2012. p. 71–79.
[26] Gianni M, Papadakis P, Pirri F, Liu M, Pomerleau F, Colas F, Zimmermann K, Svoboda T, Petricek T, Kruijff GJM, Khambhaita H, Zender H. A unified framework for planning and execution-monitoring of mobile robots. In: Automated action planning for autonomous mobile robots. Vol. WS-11-09. AAAI. 2011.
[27] Ferri F, Gianni M, Menna M, Pirri F. Point cloud segmentation and 3D path planning for tracked vehicles in cluttered and dynamic environments. In: Proceedings of the 3rd IROS workshop on robots in clutter: Perception and interaction in clutter. 2014.
[28] Menna M, Gianni M, Ferri F, Pirri F. Real-time autonomous 3D navigation for tracked vehicles in rescue environments. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. 2014.
[29] Khambhaita H, Kruijff G, Mancas M, Gianni M, Papadakis P, Pirri F, Pizzoli M. Help me to help you: how to learn intentions, actions and plans. In: AAAI 2011 spring symposium "help me help you: Bridging the gaps in human-agent collaboration". 2011. p. 30–35.
[30] Gianni M, Papadakis P, Pirri F, Pizzoli M. Awareness in mixed initiative planning. In: AAAI fall symposium series. 2011.

[31] Gianni M, Gonnelli G, Sinha A, Menna M, Pirri F. An augmented reality approach for trajectory planning and control of tracked vehicles in rescue environments. In: Proceedings of the IEEE international symposium on safety, security, and rescue robotics. 2013. p. 1–6.
[32] Carbone A, Finzi A, Orlandini A, Pirri F. Model-based control architecture for attentive robots in rescue scenarios. Autonomous Robots. 2008;24(1):87–120.
[33] Finzi A, Pirri F. Switching tasks and flexible reasoning in the situation calculus. DIAG. 2010. Tech Rep 7.
[34] Finzi A, Pirri F. Representing flexible temporal behaviors in the situation calculus. In: Proceedings of the international joint conference on artificial intelligence. 2005. p. 436–441.
[35] Carbone A, Finzi A, Orlandini A, Pirri F, Ugazio G. Augmenting situation awareness via model-based control in rescue robots. In: Proceedings ot the IEEE/RSJ international conference on intelligent robots and systems. 2005. p. 3699–3705.
[36] Apt KR, Wallace MG. Constraint logic programming using eclipse. Cambridge University Press. 2006.
[37] Surmann H, Nüchter A, Hertzberg J. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. Robotics and Autonomous Systems. 2003; 45(3):181–198.
[38] Blodow N, Goron LC, Marton Z, Pangercic D, Rühr T, Tenorth M, Beetz M. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. 2011. p. 4263–4270.
[39] Dornhege C, Kleiner A. A frontier-void-based approach for autonomous exploration in 3D. Advanced Robotics. 2013;27(6):459–468.
[40] van Diggelen J, Grootjen M, Ubink EM, van Zomeren M, Smets NJ. Content-based design and implementation of ambient intelligence applications. In: Ambient intelligence-software and applications. Springer International Publishing. 2013. p. 1–8.
[41] Streefkerk JW, Heuvelink A. Smart questions context-dependent mobile information exchange for military operations. In: Proceedings of the 6th international conference on knowledge and smart technology (kst). 2014.
[42] Neerincx MA. Situated cognitive engineering for crew support in space. Personal Ubiquitous Computing. 2011 Jun;15(5):445–456.
[43] Neerincx M, Lindenberg J. Situated cognitive engineering for complex task environments. In: Schraagen JMC, Militello L, Ormerod T, , Lipshitz R, editors. Naturalistic decision making and macrocognition. Aldershot, UK: Ashgate. 2008.
[44] Mioch T, Ledegang W, Paulissen R, Neerincx M, Van Diggelen J. Interaction design patterns for coherent and re-usable shape specifications of human-robot collaboration. In: Eics 2014. Rome, Italy. ACM. 2014.
[45] Dearden A, Finlay J. Pattern languages in hci: A critical review. Human–computer interaction. 2006; 21(1):49–102.
[46] Mioch T, Smets N, Neerincx M. Predicting performance and situation awareness of robot operators in complex situations by unit task tests. In: Achi 2012, the fifth international conference on advances in computer-human interactions. 2012. p. 241–246.
[47] Kruijff G, Tretyakov V, Linder T, Pirri F, Gianni M, Papadakis P, Pizzoli M, Sinha A, E P, Corrao S, Priori F, Febrini S, Angeletti S. Rescue robots at earthquake-hit mirandola, italy: a field report. In: Proceedings of the 10th ieee international symposium on safety, security, and rescue robotics. ieee international symposium on safety, security, and rescue robotics (ssrr-2012). College Station, TX, USA: IEEE Press. 2012 November 5-8.
[48] Yoshida T, Nagatani K, Tadokoro S, Nishimura T, Koyanagi E. Improvements to the rescue robot quince -toward future indoor surveillance missions in the fukushima daiichi nuclear power plant. In: Proceedings of field & service robotics (fsr) 2012. Matsushima/Sendai, Japan. 2012.
[49] Johnson M, Bradshaw J, Feltovich P, Hoffman R, Jonker C, van Riemsdijk B, Sierhuis M. Beyond cooperative robotics: The central role of interdependence in coactive design. IEEE Intelligent Systems. 2011 May/June;:81–88.
[50] Olsen DR, Goodrich MA. Metrics for evaluating human-robot interactions. In: Proceedings of permis 2003. 2003.
[51] Salmon P, Stanton N, Walker G, Jenkins D. Distributed situation awareness: Theory, measurement, and application to teamwork. Human Factors in Defence. Ashgate. 2009.
[52] TRADR (Long-Term Human-Robot Teaming for Robot-Assisted Disaster Response). 2014. Avail-

able from: http://www.tradr-project.eu.

German Journal of AI manuscript No.
(will be inserted by the editor)

TRADR Project: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response

Ivana Kruijff-Korbayová · Francis Colas · Joachim de Greeff · Koen Hindriks · Mark Neerincx · Petter Ögren · Mario Gianni · Fiora Pirri · Tomáš Svoboda · Rainer Worst

Received: date / Accepted: date

Abstract This paper describes the project *TRADR: Long-Term Human-Robot Teaming for Robot Assisted Disaster Response*. Experience shows that any incident serious enough to require robot involvement will most likely involve a sequence of sorties over several hours, days and even months. TRADR focuses on the challenges that thus arise for the persistence of environment models, multi-robot action models, and human-robot teaming, in order to allow incremental capability improvement over the duration of a mission. TRADR applies a user centric design approach to disaster response robotics, with use cases involving the response to a medium to large scale industrial accident by teams consisting of human rescuers and several robots (both ground and airborne). This paper describes the fundamentals of the project: the motivation, objectives and approach in contrast to related work.

Keywords disaster response robotics · persistent environment models · persistent multi-robot action models · persistent multi-robot collaboration models · persistent human-robot teaming · user-centric design

---

TRADR is an EU-funded Integrated Project in the FP7 ICT Programme, grant nr. 609763, Nov. 2013 – Dec. 2017. URL: www.tradr-project.eu

I. Kruijff-Korbayová *(corresponding author)*
Language Technology Lab, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
Tel.: +49-681-85775.5356 Fax: +49-681-85775.5338, E-mail: ivana.kruijff@dfki.de

F. Colas, Eidgenössische Technische Hochschule Zürich (ETH)
J. de Greeff and K. Hindriks, Technische Universiteit Delft (TUD)
M. Neerincx, Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (TNO)
P. Ögren, Kungliga Tekniska Hoegskolan (KTH)
M. Gianni, Università degli Studi di Roma "La Sapienza" (ROMA)
F. Pirri, Università degli Studi di Roma "La Sapienza" (ROMA)
T. Svoboda, Czech Technical University in Prague (CTU)
R. Worst, Fraunhofer Institute for Intelligent Analysis and Information Systems

Fig. 1 NIFTi deployment in Emilia Romagna. Top: Structural damage on Duomo in Mirandola. Bottom (left-to-right): UGV, UAV and mobile command post.

1 Introduction

A real disaster response takes longer than a single sortie into the area. As witnessed recently for example in Japan (Fukushima) and in Northern Italy (Emilia Romagna) deployments can last days, weeks, months, if not years.

TRADR builds on the research and experience of the NIFTi project [21]. In July 2012 NIFTi assisted in structure damage assessment in Emilia Romagna, after it was hit by over 250 seismic events in May–June 2012, causing widespread damage to an area rich in cultural heritage (Fig. 1). Together with the Vigili del Fuoco, the Italian national rescue organisation responsible for disaster response, NIFTi fielded a human-robot team with a mobile command post, two unmanned ground vehicles (UGVs), and two quadcopter unmanned aerial vehicles (UAVs). The crucial insight from this deployment was the need for *integrated persistent situation awareness* [22]. Multiple robots need to be sent into the area, together (*synchronous operation*) or one after another (*asynchronous operations*). Different kinds of robots play complementary roles in this process. They need to build integrated persistent situation awareness gradually over multiple sorties, to allow the team to coordinate its efforts (*team-level*), and learn to best execute its tasks (*task-level*).

TRADR addresses the ensuing challenge of making the experience of a human-robot disaster response team persistent over multiple sorties during a prolonged mission. We employ proven-in-practice user-centric design methodology (Fig. 2, left), involving tight cooperation with end users and tight integration of technology. The TRADR use cases involve response to a medium to large scale industrial accident by teams consisting of human rescuers and several ground and airborne robots (Fig. 2, right). The team collaborates to explore the environment and gather measurements and physical samples. TRADR's goal is to enable the team to gradually develop its understanding of the disaster area over multiple synchronous and asynchronous sorties (persistent environment models), to improve team members' understanding of how

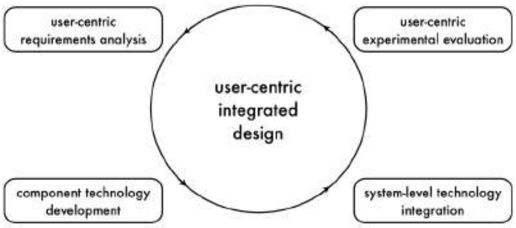


Fig. 2 Left: TRADR one-year-round development cycle. Right: TRADR UGV and UAV

to work in the area (persistent single- and multi-robot action models), and to improve team-work (persistent human-robot teaming). TRADR missions will ultimately stretch over several days in increasingly dynamic environments.

Project Partners The TRADR consortium consists of 12 partners,[1] including 3 research institutes: DFKI (coordinator), Fraunhofer, TNO; 5 universities: ETH, KTH, CTU, ROMA and TUD; one industry partner: Ascending Technologies; and 3 end-user organizations, representatives of the fire-brigades from Germany (Stadt Dortmund Institut für Feuerwehr und Rettungstechnologie), Italy (Vigili del Fuoco directed by the Ministero Dell'interno) and the Netherlands (Gezamenlijke Brandweer). 8 of the partners have already collaborated very successfully in the NIFTi project.

2 The TRADR Concept

In this section we present the research challenges addressed in TRADR in more detail and contrast the TRADR approach with related work.

2.1 Persistent Environment Models

Low-level situation awareness of the TRADR system requires sensory data from all involved robots registered in space and time, to keep creating and updating robot centric representations, and ground them into the world coordinate frame. The obtained representations are furnished to other parts of the TRADR system, which maintain higher level situation awareness. Persistent multi-robot environment models are grounded in two different aspects: *environment representation* and *adaptive action*.

Regarding environment representation, 3D mapping has so far essentially been studied for a single robot starting from an empty map. In TRADR we need to develop new data structures similar to octrees [42] and multi-resolution surfel maps [37] but with the added capabilities to integrate different sensor

[1] Cf. the authors' list for full names of the institutes listed here only by an abbreviation. For more information on the partners, please visit the project website: www.tradr-project.eu

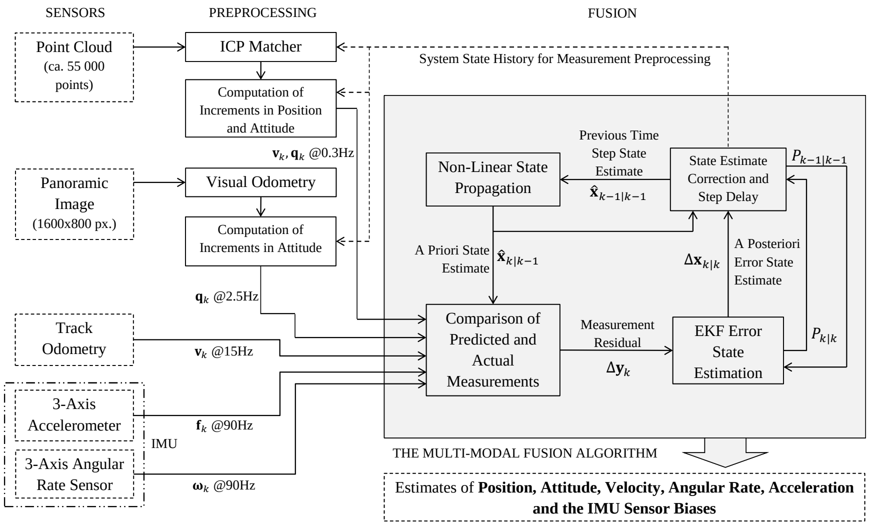
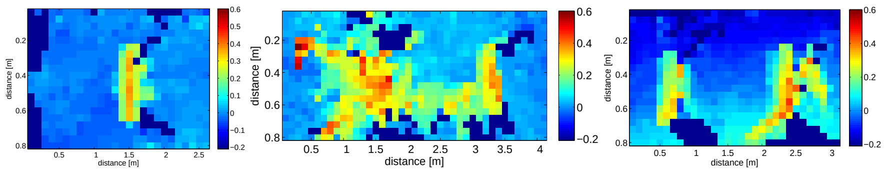


Fig. 3 Data fusion for grounding robot and maps. Figure from [23]

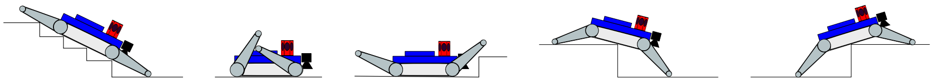


(1) simple obstacle (2) soft-terrain-sideroll obstacle (3) double obstacle

Fig. 4 Perception of obstacles, functional recognition. Overview of the Digital Elevation Maps (DEM) for given type of obstacles. Figure from [44]

modalities from different robots, to scale to arbitrary environment sizes, and to cope with dynamic obstacles [34]. In order to achieve robust grounding we fuse all available modalities (Fig. 3).

Regarding adaptive action, impressive demonstrations of aggressive manoeuvres have shown the capabilities of UAVs but always in a closed environment with high-precision external tracking systems [25,27]. To replicate these results in field experiments, it is necessary to improve the performance of current state estimation techniques relying on vision or laser sensors to complement IMU measurements [1,41]. While for UAVs the difficulty lies often more in control since they are unstable systems, UGVs research is more focused on path planning. A plethora of algorithms allow robots on flat ground to find optimal paths using robot constraints [20,35] but few approaches investigate moving in a rough terrain by using flippers [31,8] and these are not yet ready for large-scale or dynamic environments. To this end, we develop algorithms to recognize different terrains in front of the robot and changing the morphology by adjusting the flippers (Fig. 5) for smooth traversal (Fig. 4).



I-shape V-shape L shape U-shape soft U-shape hard

Fig. 5 Robot modes Robot morphology, Figure from [44].

2.2 Persistent Models for Acting

Building persistent models for action in TRADR basically corresponds to the human-robot team *learning on the job*. The models for acting will obviously rely heavily on the world models described above, but also learn from experiences generated in human-robot interaction on *different autonomy levels*.

Consider the following example. A UGV is sent to explore a given part of a building and retrieve some samples. It starts off in fully autonomous mode and successfully passes some difficult terrain and obstacles. Then it comes to an even more difficult area that is judged to be beyond its current capabilities. It stops and requests human support on a lower autonomy level. The human then guides the UGV across the terrain in an intelligent teleoperation mode. The choices made by the human during the traversal are stored and made accessible to the system. The path chosen by the human will be a preferred option in the next autonomous traversal attempt. Similarly, when a door needs to be opened or a sample of a possibly toxic liquid needs to be collected, the autonomous mode can request help by a human and then learn from that experience.

To achieve the above, we build upon state of the art approaches such as the intelligent teleoperation described in [30], the Click and Grab functionality of [2], the augmented virtual reality interface of [3], and the flipper position control of [31]. But the ambition of learning action models on the job on a team level goes beyond those approaches. Also the ambition of developing these persistent models will influence the design of the algorithms, leading to new results across all autonomy levels.

2.3 Persistent Models for Multi-Robot Collaboration

Multi-robot collaboration presupposes intention to collaborate, awareness of roles, partial knowledge, distinct beliefs, desires, capabilities and goals [15,5, 39,29,11,4]. Although significant research results have been achieved in the last thirty years, the concept of persistent collaboration is new in TRADR, as it requires persistence to be verified through sorties where an enormous amount of data is collected by the robot team. The challenge is to model how the information content of the data collected is preserved, and it is lifted to knowledge, while changing the team, changing the ways of communication and changing the experience gathered. Persistence asks for strong communication structures at different layers for role assignment, for distributed task inference and for sharing the team members current state. Persistence also demands consistent continuous information sharing which is especially hard in damaged environments and has never been experienced before.

We aim to develop a statistical-logical model for flexible collaborative planning. This model exploits the powerful language of the Flexible Temporal Situation Calculus (FTSC) [13], extended with constraints specifying dependencies between robot' abilities and their spatial distribution, also accommodating statistical inference [33]. The model includes a knowledge and memory structure which is used, through sorties, to manage information sharing, common plan generation and dynamic role allocation. Both role and task allocation is based on a cost assigned to resources, robot groups capabilities, tasks and contexts [16, 24]. A learning schema, based on a Bayesian approach to tensor factorization is proposed to build a relation between group composition and costs [43]. Group reconfiguration exploits the stimulus-response framework, proposed in [17], modeling the human inspired mechanism of task switching in robot cognitive control. Finally, an extension of the ACL communication language is proposed for modeling the information flow between robots, in order to support collaboration [14]. This language is also used for knowledge retrieval and updating, via OWL [40].

2.4 Persistent Models for Human-Robot Teaming

As robots become more sophisticated a tendency has arisen within HRI to perceive them as teammates rather than tools [19, 32]; also in the context of disaster response robotics the importance of robots capable of operating as a (social) team-member has been acknowledged and addressed [12, 28]. Even though in NIFTi multiple robots were employed, they did not necessarily partake on the team-level; each robot was controlled by an individual operator taking orders from the human commander. This is similar in a number of other projects, where teams of heterogeneous robots are employed in a collaborative fashion, but it is human operators who provide the linkage between the robots and the human rescue workers, e.g., [9, 7]. A stronger notion of human-robot collaboration is developed in the alpine rescue project SHERPA [26], employing a metaphor of the human as "busy genius" who collaborates with a group of robots with different capabilities (the "SHERPA animals") towards a common goal. TRADR will also go beyond an approach in which robots are mere tools, instead aiming at robots with an adaptive level of autonomy (e.g. semi-autonomous navigation, data gathering etc.) as members of flexible teams improving their collaboration over time. To realize this, TRADR is developing a framework for coordination of human-robot teaming, which is built on agent-based technology [18]. This framework manages the different roles, objectives, responsibilities and expectation for members of the team (which consists of both robots and humans and which may change over different sorties) and allows for conflict resolution and dynamical task-allocation depending on capabilities, task-load and chances of success.

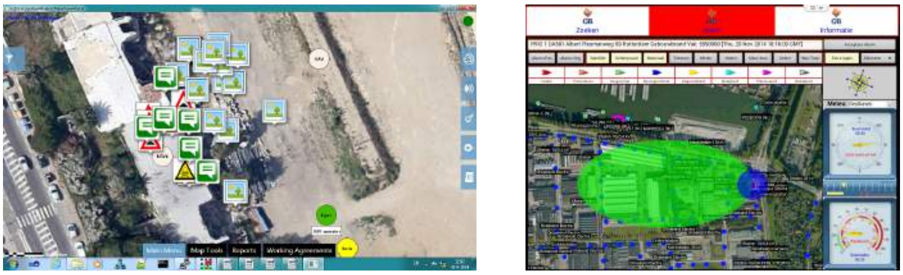


Fig. 6 Left: screen capture from the NIFTi TREX system showing a base map of the disaster area and various icons depicting location of rescue workers, robots, warnings, notes etc. Right: screen capture from the GB fire-brigade system showing various tactical information.

2.5 Persistent Models for Distributed Joint Situation Awareness

Situation Awareness (SA) is paramount for a team to work effectively in disaster response missions [36]. To achieve robust SA on a team-level in TRADR, we are designing a *Tactical Display System* (TDS) that builds on the experiences gathered with the system developed to support distributed joint SA in NIFTi (Fig. 6, left) and existing end-users systems (e.g. the system employed by the GB fire-brigade, Fig. 6, right). The TDS will provide trustworthy and relevant tactical information about the physical environment and give access to a hierarchical representation of experiences to support tactical decision making (e.g., task allocation, (re)planning and coordination). It will be designed to support guided (a)synchronous information exchange between distributed or co-located actors through multi-modal interaction (graphical UI and spoken dialogue). This guidance needs to be personalized and context-tailored. A survey [38] found that in many cases adaptivity towards the user is realized through a customizable interface that does not significantly affect the behavior or interaction patterns of the systems. Following in NIFTi footsteps, TRADR aims to push adaptivity beyond simple widget placement, concretely adapting the system's behavior to different use contexts.

2.6 User-centric design and development

TRADR adopts a scenario-based roadmap to guide iterative development of the persistent models described in the previous subsections, to drive continuous integration of the development results into a technical system, and to allow evaluation of the integrated system with end-users in yearly cycles (Fig. 2).

The roadmap defines a large-scale industrial disaster scenario. This is a kind of disaster where persistence is key to a successful mission. We need multiple robots to investigate the disaster from different angles (literally), and we need to use them over a number of sorties to gradually build up and maintain situation assessment, e.g., through observation and sample gathering. Within the industrial accident scenario, the roadmap then defines yearly use

Fig. 7 TRADR use cases set up at training sites in Germany, Italy, and the Netherlands.

cases which deal with situation assessment under increasingly more complex circumstances, as described in Tab. 1. In Fig. 7 various use case setups at the TRADR end user training facilities are illustrated.

End users are closely involved in TRADR: Each year of the development cycle in Fig. 2 starts by a deep domain analysis with end-users, followed by the development and integration of the components. The development cycle is rounded off by evaluating the developed components on system-level and performing end-user evaluations of the integrated system.

Integration takes place in a continuous process. An (as far as possible) automated procedure combines periodically the latest component versions, performs a static analysis of the code, and executes run-time tests. Reports of successes and failures are reported to the responsible developers, who can take the necessary actions. The components are mainly based on the ROS framework; however, since in TRADR more than one mobile robot is involved in the mission, we must set up a multi-master mechanism, which is necessary for the cooperation of multiple ROS-based systems.

Year 1	A fixed human-robot team with 1 UGV and 1 UAV gradually builds up situation awareness of a *static* disaster site over multiple *asynchronous* sorties.
Year 2	A fixed human-robot team with 1 UGV and 1 UAV gradually builds up situation awareness of a *dynamic* disaster site over multiple *synchronous or asynchronous* sorties.
Year 3	A human-robot team with multiple UGVs/UAVs carries out multiple sorties. Focus is on how environment models get fused, and may be used. Task adaptation moves from a strictly individual focus, to a multi-robot setting: How could a robot learn from its use of information provided by others, to adapt its own tasks as well as anticipate requests for such collaboration in (future) plans?
Year 4	A human-robot team with multiple UGVs/UAVs collaborates in various ways, synchronously and asynchronously. Persistence in modeling the environment covers ever-increasing complexity in local and global dynamic events, appearing on an ever-larger spatiotemporal scale. Team competence gradually improves based on experience.

Table 1 TRADR roadmap: year-by-year use cases within the industrial disaster scenario

2.7 Related European Projects

Several other European projects address the deployment of (teams of) UGVs and UAVs in various disaster response scenarios. ICARUS [9] and DARIUS [7] target the development of robotic tools that can assist during disaster response operations, focusing on autonomy. SHERPA [26] is focused on the development of ground and aerial robots to support human-robot team response in an alpine scenario. None of these projects addresses the persistence issues. In TIRAMISU [6], a toolbox is developed for removal of anti-personnel mines, submunitions, and Unexploded Ordnance (UXO). It includes a component called TIRAMISU Repository Service, which provides a centralized data-sharing platform that contains the locations of detected landmines and UXOs. The TRADR concept of persistent situation awareness goes beyond this in various respects as we described above. On the other hand, the EU project STRANDS [10], aims at modeling the spatio-temporal dynamics in human indoor 3D environments in order for a single robot to adapt to and exploit long-term experience in months-long autonomous operation. In contrast, TRADR deals with multiple sorties into an unstructured outdoor environment carried out by a human-robot team.

3 Conclusions

We presented an overview of the TRADR aims and approach. TRADR advances the use of the user-centric methodology established in the NIFTi project, and builds on the experience and insights obtained through the deployment of the NIFTi system, that there is a need for persistent, integrated situation awareness gathered over multiple sorties during a mission, and that different kinds of robots each play complementary roles in this process. To this end TRADR develops the capacity for persistent environment models, persistent multi-robot action models and persistent human-robot teaming.

References

1. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In: SPIE Unmanned Systems Technology XI (2009)
2. Anderson, D., Howard, T., Apfelbaum, D., Herman, H., Kelly, A.: Coordinated control and range imaging for mobile manipulation. In: Experimental Robotics, pp. 547–556. Springer (2009)
3. Atherton, J., Goodrich, M.: Supporting remote manipulation with an ecological augmented virtuality interface. In: Artificial Intelligence and Simulation of Behaviour (AISB) (2009)
4. Brenner, M., Nebel, B.: Continual planning and acting in dynamic multiagent environments. Autonomous Agents and Multi-Agent Systems **19**(3), 297–331 (2009)
5. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Transactions on Robotics **21**(3), 376–386 (2005)

6. Cantelli, L., Mangiameli, M., Melita, C., Muscato, G.: Uav/ugv cooperation for surveying operations in humanitarian demining. In: Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, pp. 1–6 (2013). DOI 10.1109/SSRR.2013.6719363
7. Chrobocinski, P., Zotos, N., Makri, E., Stergiopoulos, C., Bogdos, G.: Darius project: Deployable sar integrated chain with unmanned systems. In: Telecommunications and Multimedia (TEMU), 2012 International Conference on, pp. 220–226 (2012). DOI 10.1109/TEMU.2012.6294722
8. Colas, F., Mahesh, S., Pomerleau, F., Liu, M., Siegwart, R.: 3D path planning and execution for search and rescue ground robots. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE (2013)
9. De Cubber, G., Doroftei, D., Serrano, D., Chintamani, K., Sabino, R., Ourevitch, S.: The eu-icarus project: Developing assistive robotic tools for search and rescue operations. In: Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, pp. 1–4 (2013). DOI 10.1109/SSRR.2013.6719323
10. Duckett, T., Hanheide, M., Krajnik, T., Fentanes, J.P., Dondrup, C.: Spatio-temporal representations for cognitive control in long-term scenarios. In: International IEEE Workshop on Autonomous Cognitive Robotics (2014)
11. Duncan, B., Ulam, P., Arkin, R.: Lek behavior as a model for multi-robot systems. In: Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on, pp. 25 –32 (2009)
12. Fincannon, T., Barnes, L.E., Murphy, R.R., Riddle, D.L.: Evidence of the need for social intelligence in rescue robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 2, pp. 1089–1095. IEEE (2004)
13. Finzi, A., Pirri, F.: Representing flexible temporal behaviors in the situation calculus. In: Proceedings of the 19th international joint conference on Artificial intelligence, pp. 436–441 (2005)
14. FIPA: FIPA ACL message structure specification (2002). URL http://www.fipa.org/specs/fipa00061
15. Gerkey, B.P., Mataric, M.J.: Murdoch: Publish/subscribe task allocation for heterogeneous agents. In: AAAI/IAAI, p. 1070 (2000)
16. Gerkey, B.P., Mataric, M.J.: Multi-robot task allocation: analyzing the complexity and optimality of key architectures. In: ICRA, pp. 3862–3868 (2003)
17. Gianni, M., Kruijff, G.J.M., Pirri, F.: A stimulus-response framework for robot control. ACM Transaction on Interactive Intelligent Systems (2014)
18. Hindriks, K.V.: Programming rational agents in goal. Multi-Agent Programming: Languages, Tools and Applications pp. 119–157 (2009)
19. Hoffman, G., Breazeal, C.: Collaboration in human-robot teams. In: Proc. of the AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, USA (2004)
20. Koenig, S., Likhachev, M.: Fast replanning for navigation in unknown terrain. IEEE Transactions on Robotics **16**(3), 354–363 (2005)
21. Kruijff, G., Colas, F., Svoboda, T., van Diggelen, J., Balmer, P., Pirri, F., Worst, R.: Designing intelligent robots for human-robot teaming in urban search & rescue. In: Proceedings of the AAAI 2012 Spring Symposium on Designing Intelligent Robots (2012)
22. Kruijff, G., Tretyakov, V., Linder, T., Pirri, F., Gianni, M., Papadakis, P., Pizzoli, M., Sinha, A., E, P., Corrao, S., Priori, F., Febrini, S., Angeletti, S.: Rescue robots at earthquake-hit Mirandola, Italy. In: Proc. of the 10th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR-2012). IEEE Press, College Station, TX, USA (2012)
23. Kubelka, V., Oswald, L., Pomerleau, F., Colas, F., Svoboda, T., Reinstein, M.: Robust data fusion of multi-modal sensory information for mobile robots. Journal of Field Robotics (2014)
24. Lerman, K., Jones, C., Galstyan, A., Mataric, M.J.: Analysis of dynamic task allocation in multi-robot systems. I. J. Robotic Res. **25**(3), 225–241 (2006)
25. Lupashin, S., Schoellig, A., Sherback, M., D'Andrea, R.: A simple learning strategy for high-speed quadrocopter multi-flips. In: IEEE International Conference on Robotics and Automation (ICRA) (2010)

26. Marconi, L., Leutenegger, S., Lynen, S., Burri, M., Naldi, R., Melchiorri, C.: Ground and aerial robots as an aid to alpine search and rescue: Initial sherpa outcomes. In: Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, pp. 1–2 (2013). DOI 10.1109/SSRR.2013.6719381
27. Michael, N., Fink, J., Kumar, V.: Cooperative manipulation and transportation with aerial robots. Autonomous Robots (2010)
28. Murphy, R.R.: Human-robot interaction in rescue robotics. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **34**(2), 138–153 (2004)
29. Nair, R., Tambe, M., Marsella, S.: Role allocation and reallocation in multiagent teams: towards a practical analysis. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, AAMAS '03, pp. 552–559. ACM, New York, NY, USA (2003)
30. Ögren, P., Svenmarck, P., Lif, P.: Enhanced Teleoperation Control of Differential Drive UGVs using a Computer Game Control Mode. submitted to Journal of Autonomous Robots (2012)
31. Okada, Y., Nagatani, K., Yoshida, K., Tadokoro, S., Yoshida, T., Koyanagi, E.: Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning. Journal of Field Robotics **26**(6), 875–893 (2011)
32. Ososky, S., Schuster, D., Jentsch, F., Fiore, S., Shumaker, R., Lebiere, C., Kurup, U., Oh, J., Stentz, A.: The importance of shared mental models and shared situation awareness for transforming robots from tools to teammates. In: SPIE Defense, Security, and Sensing, pp. 838,710–838,710. International Society for Optics and Photonics (2012)
33. Pirri, F.: The well-designed logical robot: Learning and experience from observations to the situation calculus. Artif. Intell. **175**(1), 378–415 (2011)
34. Pomerleau, F., Krüsi, P., Colas, F., Furgale, P., Siegwart, R.: Long-term 3D map maintenance in dynamic environments. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE (2014)
35. Rufli, M., Ferguson, D., Siegwart, R.: Smooth path planning in constrained environments. In: IEEE International Conference on Robotics and Automation (ICRA) (2009)
36. Salmon, P., Stanton, N., Walker, G., Jenkins, D.: Distributed Situation Awareness: Theory, Measurement, and Application to Teamwork. Human Factors in Defence. Ashgate (2009)
37. Stueckler, J., Behnke, S.: Multi-resolution surfel maps for efficient dense 3D modeling and tracking. J. of Visual Communication and Image Representation (2013). To appear
38. Van Velsen, L., Van Der Geest, T., Klaassen, R., Steehouder, M.: User-centered evaluation of adaptive and adaptable systems: a literature review. The knowledge engineering review **23**(03), 261–281 (2008). DOI 10.1017/S0269888908001379
39. Vig, L., Adams, J.A.: A framework for multi-robot coalition formation. In: IICAI, pp. 347–363 (2005)
40. Wang, X., Zhang, D., Gu, T., Pung, H.: Ontology based context modeling and reasoning using owl. In: Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, pp. 18 – 22 (2004)
41. Weiss, S., Achtelik, M.W., Lynen, S., Chli, M., Siegwart, R.: Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In: IEEE International Conference on Robotics and Automation (ICRA) (2012)
42. Wurm, K., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In: Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation, vol. 2 (2010)
43. Xiong, L., Chen, X., Huang, T.K., Schneider, J.G., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: SDM, pp. 211–222 (2010)
44. Zimmermann, K., Zuzánek, P., Reinstein, M., Hlaváč, V.: Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In: L. Parker (ed.) ICRA2014: Proceedings of 2014 IEEE International Conference on Robotics and Automation, pp. 5177–5182. IEEE Robotics and Automation Society, IEEE, Piscataway, USA (2014)

Authors' Profiles

Ivana Kruijff-Korbayová: Graduation in informatics, 1992, Czech Technical University; Ph.D. in mathematical linguistics, 1998, Charles University, Prague; 1999–2000 British Academy and Royal Society/NATO Fellowships, University of Edinburgh; 2001–2010 Saarland University; since 2008 DFKI. Research in various areas of discourse and dialogue processing.

Francis Colas Ph.D. in Computer Science from the INPG in 2006. 2007–2008 postdoctoral fellow at Collège de France in Paris in the *Laboratoire de Physiologie de la Perception et de l'Action*. Since 2009 postdoctoral fellow at the Autonomous Systems Lab (ETH Zurich). Research interests include Bayesian modeling of perception and action applied from cognitive sciences to robotics.

Mario Gianni: Ph.D in Computer Engineering from DIAG - Department of Computer, Control and Management Engineering A. Ruberti at Sapienza University of Rome. Currently, post-doctoral researcher at ALCOR Laboratory. Research interests include statistics and logic, applied to robotics, autonomous navigation and adaptation for self-reconfigurable robots in cluttered environments, low and high-level control in multi-robot collaboration.

Fiora Pirri: is currently full professor and head of the ALCOR Laboratory at the DIAG, Department of Computer, Control, and Management Engineering, "A. Ruberti" at Sapienza, University of Rome. She founded ALCOR Lab in 1998 and since then the Lab has widely to the theory and practice of Cognitive Robotics applied to rescue environments. Her research area is Cognitive Robotics, and in particular Perception, Attention and theories of Actions.

Joachim de Greeff: Ph.D. in concept learning and human-robot interaction from Plymouth University, United Kingdom in 2013. In 2013-2014 employed as research fellow on the EU FP7 ALIZ-E project which was about the development of a robotic companion for children with diabetes. Currently working at Delft University of Technology on the TRADR project as a postdoctoral researcher. Research interests include cognitive modelling, human-robot interaction and teamwork, evolutionary robotics and general AI.

Koen Hindriks: MSc degree from the University of Groningen; PhD degree from Utrecht University; part of PhD candidature at the University of Toronto. Currently Assistant Professor at the Delft University of Technology, Department of Electrical Engineering, Mathematics and Computer Science. Research in multi-agent technology.

Mark Neerincx After M.Sc. worked at the TNO Institute of Preventive Health Care, the University of Leiden and the University of Amsterdam. After Ph.D. became research scientist at TNO Human Factors and, from 2000, headed the Usability Engineering group. Also (part-time) full professor in Human-Centred Computing at Delft University of Technology. Extensive experience in applied and fundamental research. Conducted empirical studies in diverse domains, such as process control, defence, aerospace, health, public safety and e-commerce.

Petter Ögren: MSc and PhD degree from KTH - Royal Institute of Technology, Sweden; part of PhD candidature at Princeton University. Currently Associate Professor at KTH, Department of Computer Vision and Active Perception (CVAP), School of Computer Science and Communication. Research interests include multi-agent coordination, formations, obstacle avoidance, intelligent teleoperation and applying ideas from computer games to robot control problems.

Tomáš Svoboda: Ph.D. in artificial intelligence and biocybernetics from the Czech Technical University in Prague in 2000. Post-doc with the Computer Vision Group at the ETH Zurich (Swiss Federal Institute of Technology) for 3 years. Currently assistant professor and deputy head of the department of Cybernetics at the Czech Technical University. Research interests include multicamera systems for scene understanding and telepresence, tracking and motion analysis.

Rainer Worst: Senior researcher at Fraunhofer IAIS. Diploma in Computer Science (Univ. Bonn, 1981). IT-Expert at a consulting company in Wiesbaden. In 1984 joined the former GMD (National Research Center for Information Technology) in Sankt Augustin, which merged 1999 with Fraunhofer. Experience as a consultant, researcher and project leader in the areas Software Engineering, Quality Management, Virtual Environments, and Autonomous Systems.

Simple Mobile Robots and Self-Adaptive Wireless Networks

Hartmut Surmann, Fraunhofer IAIS, hartmut.surmann@iais.fraunhofer.de, Germany
Rainer Worst, Fraunhofer IAIS, rainer.worst@iais.fraunhofer.de, Germany
Erik Zimmermann, Fraunhofer IAIS, erik.zimmermann@iais.fraunhofer.de, Germany
Stefan Wilkes, University of Applied Science Gelsenkirchen, stefan.wilkes@gmail.com, Germany
Tom-Marvin Liebelt, University of Applied Science Gelsenkirchen, fh@tom-liebelt.de, Germany
Christopher Eulering, University of Applied Science Gelsenkirchen, christopher.eulering@gmx.de, Germany

Abstract

Disaster areas require mobile robots with extreme capabilities. This paper presents an approach for setting up a network infrastructure to operate such mobile robots. We present a waterproof netserver box as a main component and mobile router robots (RC cars) to extend the network capabilities. The simple mobile robot consists of cheap standard RC components. It provides a view into the disaster area with its sensors, e.g., cameras, and in addition and very important, it sets up a persistent communication between all mobile robots and rescuers. All components use ROS as a middleware and can be integrated in the overall system. In addition to infrastructure networks, mesh networks are also supported to replacce the destroyed network infrastructure. Furthermore, the network and robots are prepared for cloud computing.

1 Introduction

1.1 Motivation

Urban search and rescue (USAR) missions need support from modern computer technologies to fulfill their task to help people. In the past decade robotic researchers and computer scientists enhanced robots especially to help in rescue missions, i.e., funded by the EU and DARPA [1, 3, 17]. These projects lead to complex and expensive robots, e.g., tracked robots, humanoids or drones, but some problems are still open, e.g., how to ensure communication ([18]) and what happens with robot suicide missions. Cheap, disposable robots would make the decision for suicide robot missions much easier for the mission commander. Furthermore, simple robots could help to gather information as sensor nodes to clear up rescue scenarios and to help the commander to make good decisions. For example, the collapse of the historical archive of the city of Cologne in march 2009 lead to hundreds of cracks, which had to be observed manually during the first five days of the mission[20]. Changes of the cracks were not well documented since, e.g., short time interval photos were not available. Beside the use of tracked robots (UGV) and drones (UAV) the NIFTi and TRADR projects [2, 3] have also to cover the communication problem and therefore to build robust network nodes to set up a network for the robots. Simple communication robots extend the adaptaibility of the network structure and will also be presented in this paper. The paper is structured as follows;

After a short state of the art in the next section we present the network components in chapter 2 followed by chapter 3, which presents use cases and performance measurements.

Figure 1: Outdoor resistant WLAN box. Hardware components: RaspberryPI, battery pack, Bullet M2.

1.2 State of the art

Nowadays computer networks consist mainly of ethernet and WLAN networks. WLAN networks can be used in infrastructure or in ad-hoc mode. This allows several network architectures [19, 4]. For each of the network topologies thousands of network adapters and routers exist e.g. [8, 9]. Furthermore ,cheap and small PC boards are available, e.g., the RaspberryPI with a huge amount of commercial and open source software. Also RC cars are very common, cheap, and widely used e.g controlled by special hardware and a router [12]. In general sensor networks and nodes is a wide area [5]. The integration of these components into network nodes suitable for rescue missions is a new approach and looks very promising.

2 Network design

2.1 The Netserver Box

The waterproof netserver box is designed to provide LAN and WLAN capabilities in field operations and in a lab environment (**Figure 1**). It can be configured by a hardware switch to act as a server or as a client in the network and is protected against overvoltage and overcurrent. Usally one box works as a server in the network. The client mode can be set quickly with the switch, when it is required to connect several boxes together. The box can be powered by a LiPo battery (Vislero LiPo 6600, 14,8V, 4S1P flat pack), wall socket (230VAC, AC/DC converter) or automobile plug (12VDC); it contains a battery low indicator and a battery hot swap over a second T-plug. Usually, in field missions, the LAN1 waterproof socket (type PX0834) of the server is connected to a Hub in the command center to connect all stationary computers to the field, i.e., mobile robots or in-field rescuers. It offers DNS, NTP and DHCP services, which are necessary for all robots and drones that run Ubuntu and ROS. This setup establishes a complete independent network in the field. The additional LAN2 socket can be used to connect the box to an external network ,e.g., internet if avaliable in the field of operation. This network adapter is automatically configured via external DHCP. If both networks are connected, computers from the internal network can also access the external network (maybe internet), which is very helpful for the maintenance, administration and setup of the complete system. WLAN can be provided by different kinds of bullets, e.g., at 2.4 GHz and 5 GHz, which are connected to a RaspberryPI. The whole network consists of the following components:

- RaspberryPI (running Raspbian image of the dedicated netserver for DHCP, DNS, NTP, ...)
- Bridge node at ground station, e.g., Ubiquiti 2,4 GHz Bullet M2 802.11N or Ubiquiti 5 GHz Bullet M5 802.11 and Omni-directional antenna
- Cables and power supply for each Ubiquiti Bullet
- Switch at ground station,extension cables, and power bars

The use of bridge nodes for the WLAN communication has the advantage that no WLAN device drivers for Linux are needed, which in general could have reliabilty problems over a longer period of time. A crucial issue for ROS-based ([16, 6]) communication is exact synchronization of the nodes involved. To provide a reliable time base for all PCs in the network to synchronize with a NTP (Chrony) server is running on a RaspberryPI ([7]) inside the netserver box, which also provides DHCP and DNS services. Tests of the network configuration are given in section 3.2.

Figure 2: The Rock Crawler in action. Attached components: RaspberryPI, battery pack, ALFA WLAN Adapter and a wide angle Logitech webcam with a microphone. A video can be found at: http://www.youtube.com/watch?v=St8QtC1kut0.

2.2 Simple Mobile Robots act as mobile access points

To adapt the network to different configurations, we set up two mobile robots to operate as movable WLAN bridges as seen in **Figure 2** and **Figure 3**. The robots should be as simple and cheap as possible to interact in so called suicide robot missions. In some cases the robot has to be fast to travel long distances and in other cases the robot has to be powerful to crawl through difficult terrain like small rocks or sidewalks.

For that reason, we have designed a computational unit that is 100% compatible with the network box introduced in the previous section and which is able to control several PWM channels. This computational unit can be attached to almost every commercial RC model. The result is a disposable robot which can be configured for specific scenarios like ground, aerial or water operations.

2.2.1 Hardware Design

We have build two variations of a mobile robot using RC model cars. The first robot is a fast mover based on a Reely Truck, e.g., [10] for about 170,- Euro. The car can reach a velocity up to 70 km/h. The second robot is a rock crawler based on an Axial AX10 Scorpion, e.g., [11] for about 200,- Euro. This robot (5 km/h) is not that fast as the the first one, but it is able to drive through difficult terrain and climb up obstacles with a slope up to 70Â°. The only modification of these RC models is the detachment of the RC receiver unit, so the computational unit can be connected to the PWM signals directly. Usually a RC model has two PWM channels for full operation. One channel is to control the steering servo. This channel needs PWM pulses with a period of 20 Hz. The duty cycle of the signal sets the position of the servo motor, e.g., 1,5 ms interval is the central setting while 1 ms and 2 ms duty cycles are standing for the maximal deflections. The other channel is to control the linear velocity. There can be two operation modes. One is exactly like the servo steering method, which means that the model can be in a neutral position or in full forward / reverse mode. The second operation mode is trivial. In this case the duty cyle controls the power which flows through the motor immediately.

Figure 3: The Fast Mover. Attached components: RaspberryPI, battery pack, Bullet2M Router. A video with an attached network camera can be found at: http://www.youtube.com/watch?v=qVALTltiF6o

The computational unit is based on a RaspberryPI ([7]), like in the netserver box above. A big advantage is that the Debian operating system, installed on the PI, allows the usage of the robot operating system ROS Hydro [16, 6]. A USB power box, attached to the RC models, provides the necessary operation power of 5V to the PI. The main reason, why we are using a separate power pack, is that the power supply for the computer system is independent from the power supply of the mobile base. Therefore, we can reach an online time up to 10 hours, depending on connected hardware. For example, the mobile network is still accessible and provides sensor information even if the robot's base has run out of power. Furthermore, several ROS compatible sensors like USB cameras or infrared cameras can be attached individually to the PI. In our setup the Rock Crawler has a wide angle Logitech webcam attached, which provides visual information over the network. To teleoperate the Crawler in real time we provide an image stream with a resolution of 160x120 pixels at 20 frames per second and transmit the stream over the network. The image stream generates a CPU load of about 50% at the RaspberryPI. Both robots can be adapted with further sensors, e.g., an action cam like the GoPro Hero, to retrieve high resolution wide angle pictures after a mission. Also the network interface can be changed. While the Fast Mover has still the Bullet2M Router attached, we are using an ALFA AWU036NH, well known in the Wardriving community, for the Rock Crawler. This allows more compact hardware setups and thus smaller and cheaper robots.

2.2.2 Software Design

On the base of ROS and the GPIO unit of the RaspberryPI, the PWM signals can be created straight in software. No additional hardware is needed to communicate with the RC models. To achieve this functionality, a module called PIBlaster is installed on the PI [14]. This module provides 8 PWM channels that can be directly attached to the corresponding GPIO pins. We have modified the PIBlaster module so that we obtain a stable 20 Hz PWM signal to control the servo motors with a processor load on the PI of less than 5%.

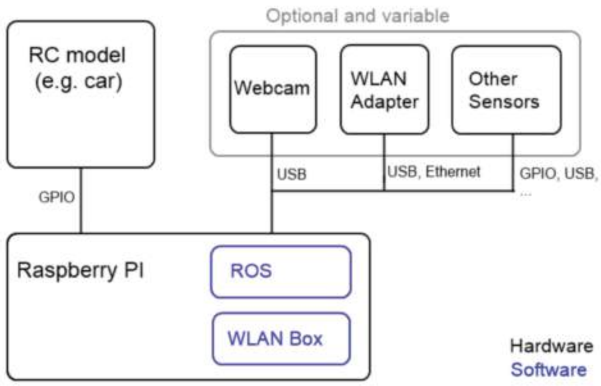


Figure 4: Communication between hard- and software components for individual RC robots

The core of the software is a ROS node called rc_bot. In this node, we have implemented a conversion of standardized ROS messages for velocity to the corresponding PWM duty cyles. The type of operation mode for the velocity PWM and the connected channels to the mobile base can be set in the node. The minimum and maximum duty cycles for the servo deflections can be adapted, because some servo motors interact more accurate than other ones. In our case the Rock Crawler steering servo has a deflection space between 0,9 ms and 2,1 ms. We have also implemented a watchdog functionality which immediately stops the robot, if no more command messages are received. The timeout for the watchdog can be dynamically set via a single parameter. This is an essential safety feature. Based on the teleoperation interface, that publishes new speed commands in a fixed interval, and the velocity of the robot, the watchdog should act properly to detect operation timeouts and to stop the robot as fast as possible, especially for the Fast Mover. Based on this core functionality the RC robot can be controlled by all ROS supported human machine interfaces, like graphical user interfaces, web browsers, tablets / smart phones or simple joystick devices. Optional sensor functionality for webcams or anything else can be included via their corresponding ROS nodes. We have written a launch file, which is also part of the rc_bot node, that starts the driver and also loads a connected webcam. The source code is avialable in our git hub at: https://github.com/roblab-wh-ge/roblab-whge-ros-pkg.

As mentioned above, the RC cars are only one example for using this core component. The modular framework, see **Figure 4**, provides the opportunity to attach the computational sensor to any RC model. For example the PI can be mounted on a RC boat for water operations. This can provide video information from a water flooded area, which is unaccessible for humans, or it can extend the network between two riverbanks by driving the boat to the center of the river.

There are several videos, which show the mobile robots in action. The videos can be found at the YouTube channel of the University of Applied Science Gelsenkirchen (http://www.youtube.com/user/RoblabFhGe)

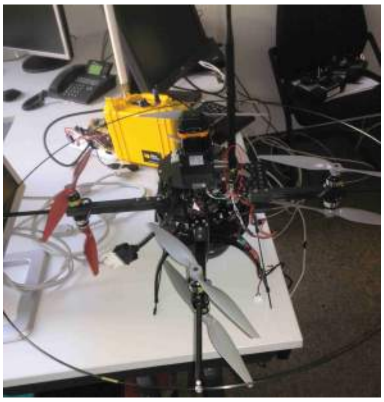


Figure 5: First test scenario: UAV with a bullet connected via WLAN to the netserver box.

3 Results: Network evaluation

3.1 Performance test of the netserver box

What is the performance of the network infrastructure based on the netserver box? To answer this question, we define typical mission scenarios. The first scenario is a mission with a drone where no network infrastructure is available. The drone is connected via WLAN with the netserver box and the box via ethernet with the command center (**Figure 5**). This reflects the situation at the earthquake area in Mirandola 2012 [2]. For the test we transfer ten times a 500MB file in both directions and measure the throughput in Mbit/s. For the second and third scenario, we connect an external network to the LAN2 socket, i.e., internet is available at the side. This scenario is needed, e.g., if an infield rescuer has a tablet and uploads pictures to a cloud in the internet or a robot send its sensor data to the cloud. Furthermore (third scenario), the operator control unit has access to the internet and can download / upload data. Since the PI has to route the packets, it generates some load. These scenarios are also a typical developers' scenario, when the code on all robots/computers has to be updated from an external repository. The next three scenarios simulate connections if additional services, e.g., a local cloud is running on the PI. All scenarios show that the PI is nearly completely loaded with this task. Our tests have also shown that the speed of the SD card (class 10 and better) is critical since all the data has to be stored on the card and the PI stops transmitting data over the network when it writes data to the card. **Table 1** show the results of the different tests for the scenarios.

Table 1: Performance evaluation. A 500MB file is transfered 10x via scp. The numbers are mean values. The last column shows the load of the RaspberryPI (standard deviation 5%). The maximal deviation for the data rate was +/- 5 MBit/s. The WLAN connections was optimal. In real situation the WLAN connection depends on the distance to the robot and will be lower.

Scenario	data rate (Mbit/s)	CPU load (%)
Scenario 1: robot -> operator control unit		
LAN1 to WLAN	55	0
WLAN to LAN1	54	0
Scenario 2: infield rescuer -> internet		
LAN2 to WLAN	35	60
WLAN to LAN2	39	43
Scenario 3: infield rescuer -> internet		
LAN2 to WLAN	39	55
WLAN to LAN2	42	74
Scenario 4: operator control unit -> box (local cloud)		
LAN1 to WLAN	23	90
WLAN to LAN1	23	98
Scenario 5: internet -> box (local cloud)		
LAN1 to WLAN	26	95
WLAN to LAN1	22	98
Scenario 6: robot -> box (local cloud)		
LAN1 to WLAN	23	90
WLAN to LAN1	24	98

3.2 Network topology

Another challenging question for USAR environments is to handle scenarios where communication is either lost or degraded during a mission. On one hand the robot should move autonomously, on the other hand we need communication with the robot to get the data from the field. So, one task is to test new network architectures, e.g., mesh networks instead of infrastructure networks with repeaters, which are currently used to enhance communication capabilities.

For the first test, we extend scenario 1 from above and install also the mesh firmware OpenWrt with olsr at the bullet routers (at 2.4 GHz). Furthermore, we connect the command computer also via WLAN to the robot. **Figure 6**:top shows the basic test setting at the Fraunhofer Campus at Sankt Augustin. AP (blue) is the netserver box in the field. C1 is the computer of the robot in the field and C2 the computer of the operator. The distance between the robot and the netserver box is about 35 m and from the box to the command computer about 50 m. The scenario is equivalent to the set up at the NIFTi project review in April 2012. It has to be noticed that the computers C1 and C2 have no direct connection to each other. The blue bars show the performance of the up-to-date version of the original router firmware (airOS) and the red bars the patched router firmware with OpenWrt [13, 15]. For this experiment, airOS shows the better performance (**Figure 6**:middle). For the case that computer C1 and computer C2 have a direct connection, OpenWrt shows a little better performance than airOS because the communication over the netserver box (AP) reduces the transfer rate (**Figure 6**:bottom). This leads to the result: if possible put the AP in the center of the scenario.

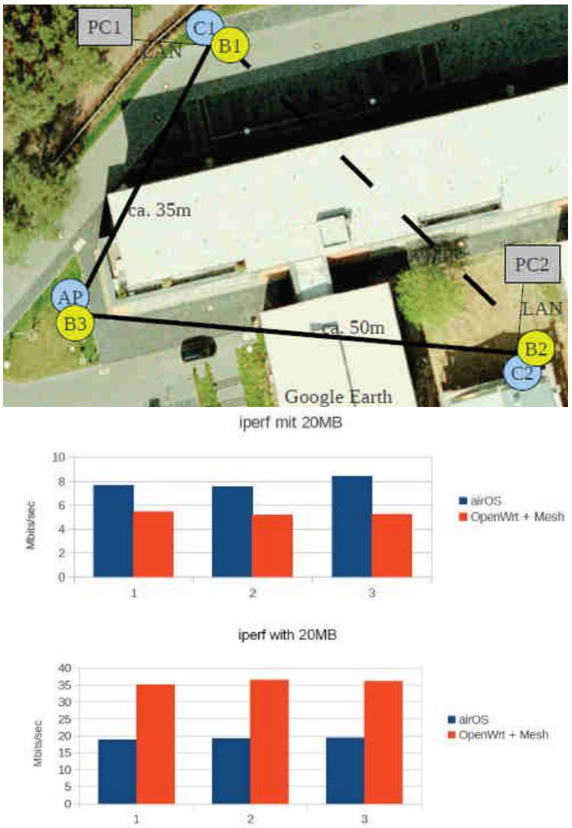


Figure 6: Top, scenario 1: AP (blue) is the netserver box in the field. C1 is the computer of the robot and C2 the computer of the operator. Both are connected via WLAN. Middle: C1 and C2 do not have a direct connection. For the test we have transferred one hundred times a 20MB file at three different days to test the influence of other networks in the environment. The blue bars show the performance of the up-to-date version of the original router firmware (airOS) and the red bars the patched router firmware with OpenWRT. The standard deviation is 5%. Bottom: Computer C1 and C2 have a direct connection.

The optimal scenario for mesh networks is defined in the last test given in **Figure 7**. The computer C1 and C2 indicate two robots in the field, e.g., C2 the drone and C1 the Fast Mover or Rock Crawler. One of the simple robots C1 has a connection to the netserver box but C2 is to far away from it. As a result, C2 is lost in the infrastructure mode and has no connection to the netserver box AP and can not transmit data to the operator. The mesh network (or also a repeater) architecture has nearly the same performance as in the scenario above but the bandwidth has to be shared between both robots. As a result, mesh networks with their reduced throughput are more suitable here.

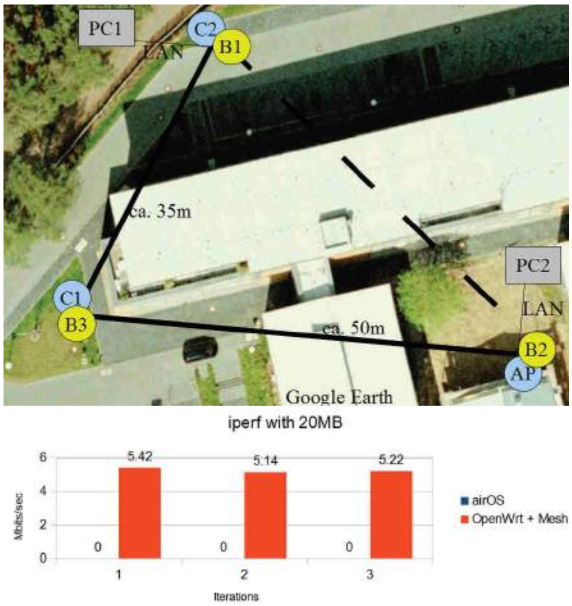


Figure 7: Top, scenario 2: Two robots are in the field (indicated as C1 and C2). C2 has no direct connection to the netserver box AP but C1 has a connection. Bottom: The performance of the mesh network is similar to the first scenario but C2 is lost and has no connection. Also 20MB files are transferred hundred times at 3 different days with a similar standard deviation of 6%

4 Conclusion

What kind of network architecture is suitable for rescue missions? This paper shows one approach to answer this

question. We presented a waterproof netserver box to provide LAN and WLAN capabilities in field, consisting of a bullet M(2/5) and a RasberryPI as a static access point with several options and connectors. Furthermore, we extended the network nodes and made them mobile by adding RC cars. The servo motors of the RC cars are controlled by the RasberryPI over the GPIOs. The bullet routers convert the cars to mobile access points, which can serve as bridges or mesh nodes to extend the network infrastructure. We defined typical in-field scenarios and evaluated the performance of the network architecture.
Future work will concentrate on the integration of new hardware, e.g., more powerful computer boards than the PI (RIoTboards) and to the improvement of the capabilities of the RC cars, e.g., adding autonmous features and on-board computer vision. The use of drones for example, the AR Drone from Parrot would also be an interesting option to adjust the network infrastructure to the realities in the field. Furthermore, we would like to add a low bandwith safety and maintenance channel between the netserver box and the robots, e.g., by using citizens' band radio or other frequencies.

Acknowledge
This work has been partly funded by the European Commission under contract number FP7-247870-NIFTI and FP7-609763-TRADR. Thanks to all colleagues from Fraunhofer and the other partners in the consortiums for their support.

References

[1] NIFTi project website: *www.nifti.eu*, 2014.

[2] Kruijff, G.-J.; Janicek, M.; Keshavda, S.; Larochelle, B.; Zender, H.; Smets, N.J.J.M; Mioch, T.; Neerincx, M.; van Diggelen, J.; Colas, F.; Liu, M.; Pomerleau, F.; Siegwart, R.; Hlavac, V.; Svoboda, T.; Petricek, T.; Reinstein, M.; Zimmermann, K.; Pirri, F.; Gianni, M.; Papadakis, P.; Sinha, A.; Balmer, P.; Tomatis, N.; Worst, R.; Linder, T.; Surmann, H.; Tretyakov, V.; Corrao, S.; Pratzler-Wanczura, S.; Sulk, M: *Experience in System Design for Human-Robot Teaming in Urban Search and Rescue*. In: Proceedings of 8th International Conference on Field and Service Robotics. July 16-19, Japan, STAR, Springer Verlag, 2012.

[3] TRADR project website: *http://www.tradr-project.eu/*, 2014.

[4] Dong, Jiawei; Kim, Won-jong: *Experimental analysis and implementation of a multiscale wireless/wired networked control system*, International Journal of Control, Automation and Systems, pp 102-110, 2014.

[5] Jennifer Yick, Jennifer; Mukherjee, Biswanath; Ghosal, Dipak: *Wireless sensor network survey*. In Computer Networks, vol. 52, no. 12, pp. 2292-2330, 2008.

[6] ROS documentation website: *http://wiki.ros.org/*, 2014.

[7] Raspberry Pi documentation website: *http://www.raspberrypi.org/*, 2014.

[8] UBiQUiTi bullet routers / documentation: *http://www.ubnt.com/airrouter*, 2014.

[9] ALFA network: WLAN adapter/documentation: *http://www.alfa.com.tw/products.php*, 2014.

[10] Monstertruck website: *http://www.conrad.de/ce/de/product/237300/Reely-110-Elektro-Monstertruck-Titan-Brushless-4WD-EM-04MT-RtR-40-MHz-FM*, 2014.

[11] Rock-Crawler website: *http://www.conrad.de/ce/de/product/235973/Reely-18-Elektro-Rock-Crawler-4WD-CR-300-RtR-40-MHz-FM*, 2014.

[12] Jon Bennett: The Wifi Robot project. *http://www.jbprojects.net/projects/wifirobot/*, 2008.

[13] OpenWrt website: *https://openwrt.org/*, 2014.

[14] Pi-blaster github repository *https://github.com/sarfata/pi-blaster/*, 2014.

[15] olsrd: an adhoc wireless mesh routing daemon: *https://olsr.org/*, 2014.

[16] Quigley, Morgan; Gerkey, Brian; Conley, Ken; Faust, Josh; Foote, Tully: *ROS: an open-source Robot Operating System*. In: ICRA workshop on open source software, 2009.

[17] DARPA Robotics Challenge website: *www.darpa. mil/Our_Work/TTO/Programs/DARPA _Robotics_Challenge.aspx*, 2014.

[18] Murphy, Robin; Tadokoro, Satoshi; Nardi, Daniele; Joacoff, Adam; Fiorini, Paolo; Choset, Howie; Erkmen, Aydan: *Chap. 50.6 Search and Rescue Robotics, Fundamental Problems and Open Issues* in Handbook of Robotics, eds. Siciliano, Bruno; Khatib, Oussama, Springer, 2008.

[19] Ogren, Petter; Fiorelli, Edward; Leonard, Naomi Ehrich: *Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment*. Automatic Control, IEEE Transactions on, 2004, 49. Jg., Nr. 8, S. 1292-1302.

[20] Linder, Thorsten; Tretyakov, Viatcheslav; Blumenthal, Sebastian; Molitor, Peter; Holz, Dirk; Murphy, Robin; Tadokoro, Satoshi. and Surmann, Hartmut: *Safety Rescue robots at the Collapse of the municipal archive of Cologne City: A field report*, 8. IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR2010), 2010.