DR 6.3: Multi-robot task adaptation

Rainer Worst[*], Renaud Dubé[†], Tomáš Svoboda[‡], Luigi Freda[§], Mario Gianni[§], and the TRADR consortium

[*]*Fraunhofer Institut für Intelligente Analyse- und Informationssysteme IAIS*
[†]*Eidgenossische Technische Hochschule Zürich*
[‡]*Czech Technical University in Prague*
[§]*'Sapienza' University of Roma*
⟨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 2016
Actual submission date:	February 2017
Lead partner:	Fraunhofer
Revision:	version 1
Dissemination level:	PU

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

Figure 1: TRADR team at Knepper site in Dortmund during TEval 2016.

Executive Summary

This report describes the final result of WP6 in Yr3 – an integrated system dealing with multi-robot task adaptation. To achieve this goal, one single task was addressed:

Task T6.4 Multi-robot task adaptation

The development of the third prototype of the TRADR system was based on the technical framework set up in Yr1, including ROS [1] as middleware system, GIT [2] repository for source code, TRAC with Redmine [3] for issue tracking, a squad of five ground robots, some aerial robots by Ascending Technologies, and several PCs to run the control software of the TRADR system.

The task was to integrate incrementally a system for the project-wide scenario of Yr3 **Multi-robot task adaptation**. 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 (cf. deliverable DR.7.3). The system evaluation took place at the Knepper site in Dortmund (see Figure 1).

Role of system framework and integration in TRADR

In Yr3, the integrated robotic system has been evolved with respect to multi-robot task adaptation. According to the scenario-based roadmap of TRADR the development process will be continued in the next year by emphasizing 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.

Persistence

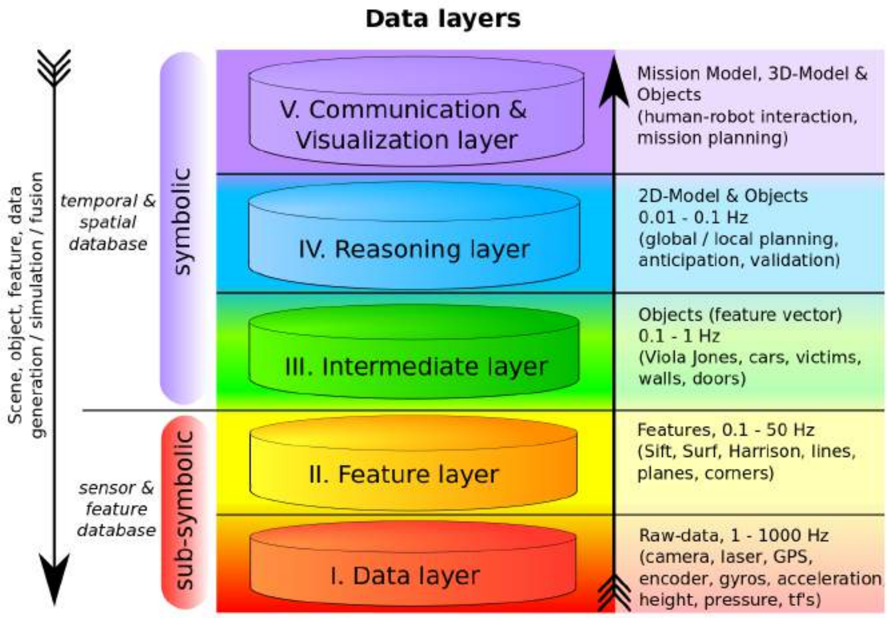


Figure 2: Basic DB-layers with semantic modelling from bottom to top.

One of the main TRADR objectives is to support the collection of data across sorties – which are performed sequentially in the course of a single mission – to improve the system's and the team's performance. 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.

The collected data provide a detailed documentation of the entire mission. Using the data for learning from experience is foreseen later on in the project, too. The structure of these data is a crucial feature of the whole system and is going to be elaborated in detail during the whole lifecycle of TRADR. Currently, we have designed five conceptual data layers, in which the corresponding data types are stored (see Figure 2).

To meet the requirements in TRADR to store and retrieve data that is updated across different sorties and extendable over a long period of time, a two-layered architecture with a low-level and a high-level database showed to be an appropriate outset. A database API service is used to enable ROS nodes to directly publish data to and receive data from the *MongoDB* [4] low-level database. From here, a collection of scripts summarized to a pool of functionalities called the "semantic modeler" handle the evaluation of the low-level data and the extraction of semantically usable high-level data. At the *Stardog* [5] high-level database, a python wrapper API is used to handle data sets.

Users are able to create still images from a video stream from the camera data of the robots. These images are stored and can be shared with other users - they are displayed at the GPS coordinates on the global map where they were created. Also the dynamic GPS location coordinates of the UGV and UAV robots are stored in the low-level database, transferred to the high-level database and shown on the TDS.

Contribution to the TRADR scenarios and prototypes

The results presented in this report contribute to the TRADR project by providing a third prototype of the integrated cognitive robot system, based on the technical framework developed in Yr1 and Yr2. It enables us to collect practical experiences and to identify necessary improvements of 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 task addressed by WP6 in Yr3 was:

Task T6.4 Multi-robot task adaptation

This task contributed to the overall objectives of WP6:

B Develop adaptive control on the system level

C Integrate WP components continuously into a single architecture

The result to be achieved was an integrated system for human-robot team operation, in particular for the project-wide scenario of Yr3 "Multi-robot task adaptation". 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 Addressing reviewers' comments

The reviewers made the following comment on the Yr2 efforts of WP6:

(A) Measuring available bandwidth and bandwidth usage in communication channels used by the system is recommended to assess more clearly how well the TRADR system copes with communication limitations.

This recommendation has been addressed by switching from *rocon* to *nimbro* nodes as base for the communication between different ROS masters as described in section 1.3.3. Hence, we were able to introduce a relay mechanism that avoids unnecessary wireless transmissions without losing performance. This mechanism allows us to measure the current bandwidth usage of system components as shown in Figure 3.

1.3 Actual work performed

This description starts with an overview on the TRADR system development in section 1.3.1, which reflects ongoing discussion between all partners in Yr3. Sections 1.3.2 and 1.3.3 are about the setup of the TRADR system and network issues. The following sections 1.3.4, 1.3.5, 1.3.6, and 1.3.7 provide details about the mobile robots – aerial robots, upgrades of the ground robots, integration of a Velodyne scanner and multi-robot modules.

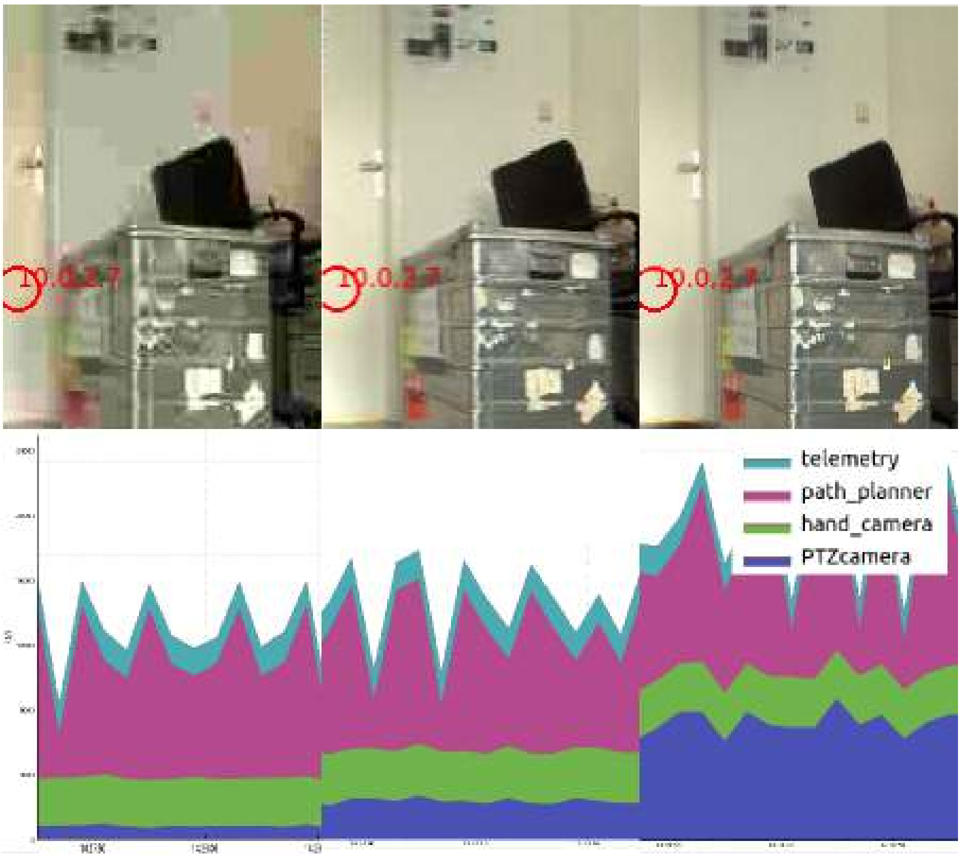


Figure 3: Visualization of bandwidth needed by several components, resolution of PTZcamera is set to 3 different values, the displayed image is shown at the top and the consumed bandwidth in the diagram at the bottom. Left: low resolution image. Middle: standard resolution image proves to be a good compromize. Right: high resolution image.

1.3.1 System development

The TRADR system is built in an incremental way by improving particular components according to the needs of the end-users as specified, e.g., during the experiments with the users at TJEx and TEval. This process is supported by a Software Development Environment based mainly on ROS as framework, GIT as repository for the source code, and Redmine as issue tracker system.

In Yr3, we cleaned up the repository according to the experiences with the build-process of the TRADR system collected so far. The focus was on the proper handling of dependencies, which had been a source of trouble previously. Therefore, it was necessary to create packages of all third-party software components that are included in the system, unless they are already available for Ubuntu. These packages are stored on a public GIT repository along with any software packages created and offered by the TRADR project itself. Hence, we avoid the situation that developers include slightly different versions of dependencies in their code, which would lead to conflicts in the integrated system.

The Jenkins server, which was provided for Continuous Integration of the TRADR system, did not prove as useful as expected and was shut down permanently. Although it had its value to detect integration issues during the system-build, it did not help us to validate the runtime-behaviour of the integrated system. Therefore, we replaced the concept of Continuous Integration by a snapshot-based system validation.

This means that we take each month a snapshot of the TRADR system from the GIT repository and perform a system test at Fraunhofer. The detected errors are documented and tickets are assigned in Redmine; a sample of such a report is shown in [38] (Annex Overview 2.2). Shortly after such a system test, a developers' teleconference takes place, where particular issues can be discussed in detail if necessary. Meanwhile, this scheme is well established and supports the development process as expected.

1.3.2 System setup and monitoring

The general setup of the TRADR system as introduced in Yr2 has proven to work well, as providing the following characteristics:

- System wide unified hardware
- Defined software versions
- Simplified setup

As in Yr2, the TRADR system contains of a central data storage, robot control and monitoring units as well as the robots themselves, see Figure 4.

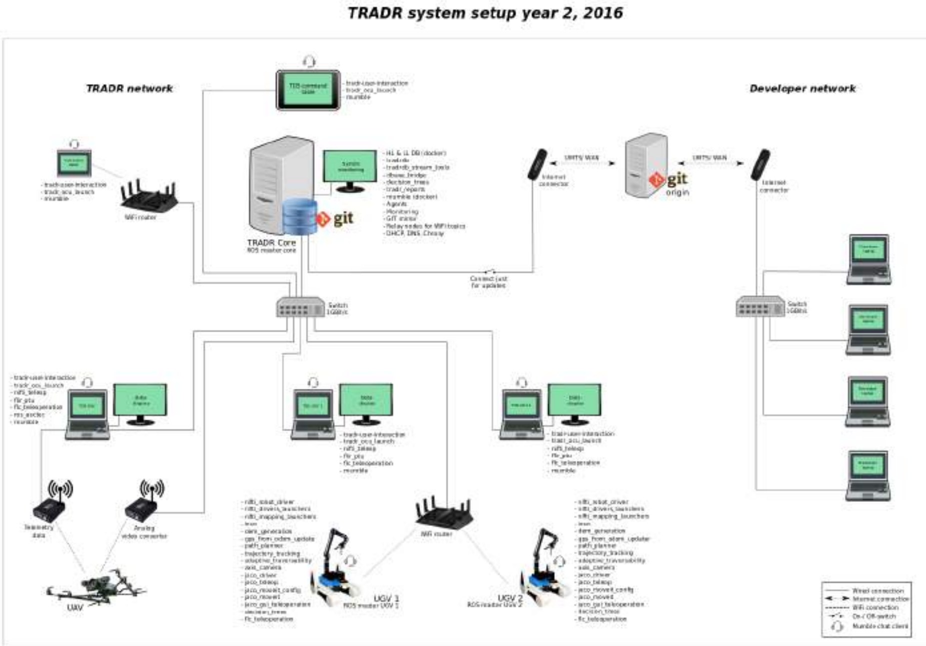


Figure 4: System setup overview with TRADR network on the left and Developer network on the right.

We added a powerful graphics adapter to the high-performance workstation named the *TRADR core system*, which is used as the main data storage and central reckoning unit. Hence the computational abilities for resource consuming processes like video data procession and map registration are improved because parallel processing based on CUDA is achievable now.

Furthermore the *TRADR core system* hosts the low- and high-level database, the agents, the relay nodes for WiFi topics, the mumble server, the reporting tool, the orchestration tool, and network services like DHCP, DNS and Chrony.

The partners continued to use the GUI framework RQT [6] and its plugin abilities as a basis for user interface development for the TDS (TRADR Display System), which is controlled by agents (see Figure 5 and DR.3.3 for details). RQT in addition to plain QT [7] offers a built-in support for ROS, which is also extensively used in TRADR. With this framework the ROS related data handling and visualization can be smoothly integrated into the TDS.

The TDS facilitates the joint situation awareness in TRADR by displaying maps and images, indicating positions of robots and actors, and highlighting points and areas of interests. The OCU is a constituent of the TDS focusing on robot data representation and robot control. The OCU thus is the interface between the robot and the robot operator.

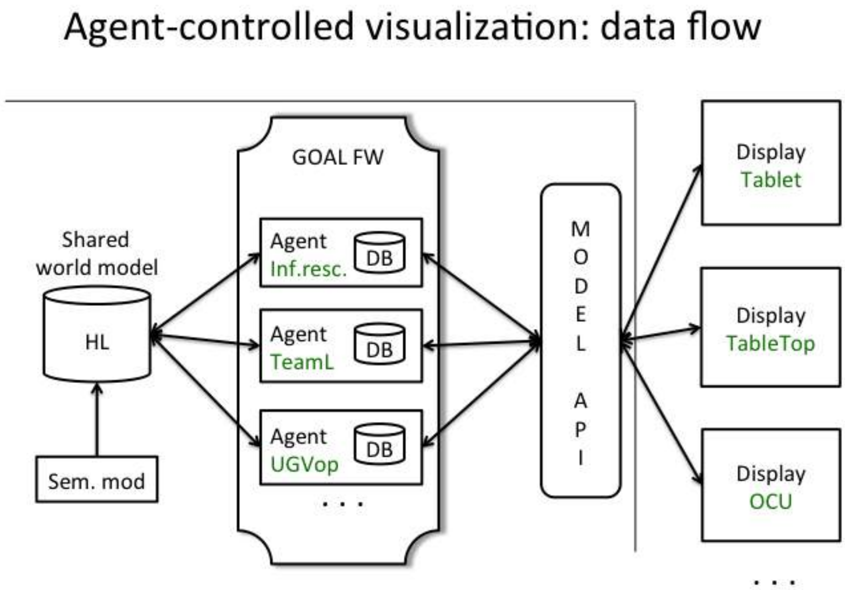


Figure 5: Agents control the views that are presented by the TDS on different display devices, e.g., if an operator inserts a POI, an agent creates an event that causes an alert on the team leader's display.

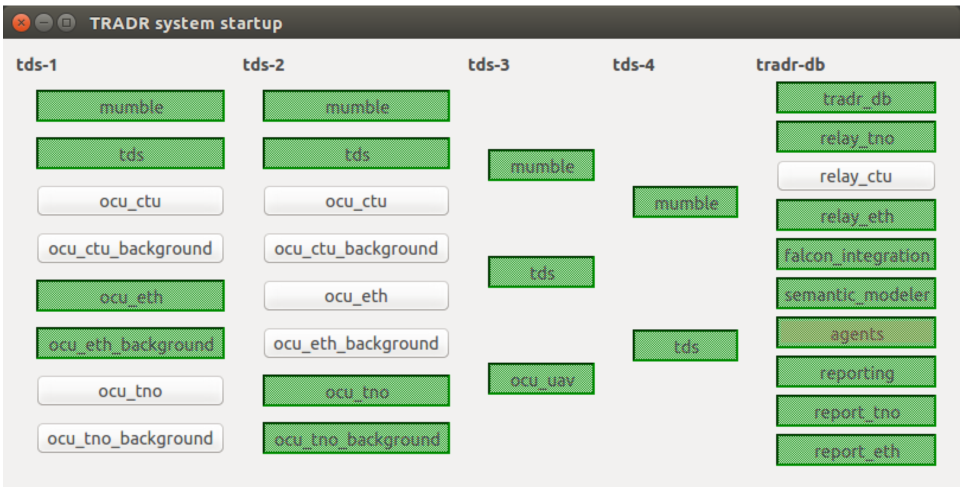


Figure 6: TRADR system startup for base station: green nodes are up and running.

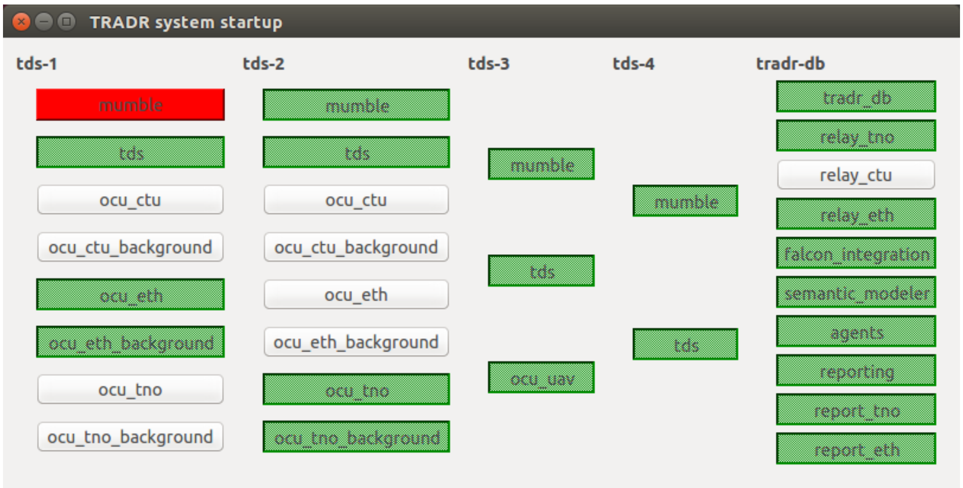


Figure 7: TRADR system startup indicating an error (mumble process died).

As TRADR consists of many different connected components with many possible states, a proper monitoring of the current conditions is indispensable. An orchestration tool is used to start and stop the components as well as to visualize their current status. Figure 6 shows the GUI of the orchestration tool for the base station, consisting of the *TRADR core system* and four TDS workstations.

The column on the right hand side is related to the *TRADR core system* with the tradr_db as its main component. The next three buttons are used to control the relay nodes that establish a connection with a specific UGV. In the example, the relay nodes for the tno-robot and eth-robot are started while the relay node for the ctu-robot is idle. The next button is attached to a node that provides the communication with the UAV AscTec Falcon 8 (cf. §1.3.4).

The next button controls the semantic_modeler, which consists of a set of processes to map the contents of the low-level database to the high-level database. The button below starts and stops the agents subsystem, and the three buttons at the bottom control the features of the reporting tool (cf. DR.5.3).

The first four columns are related to the four TDS workstations. The first two columns (tds-1, tds-2) control the startup for the OCU of the UGVs. This includes the mumble client, which is needed for speech communication, and the TDS node. In the example, tds-1 is connected to the eth-robot and tds-2 is connected to the tno-robot by activation the related OCU nodes.

The third column controls the startup for the OCU of the UAV, in addition to the mumble client and the TDS node. The fourth column controls the workstation of the team leader, where only the mumble client and the TDS node are needed.

Figure 7 shows how an error situation is indicated; in the example, the mumble client on tds-1 crashed and needs to be restarted. Without this visualization, it would be much harder to escape from such a failure.

An orchestration tool is running also on each UGV (see Figure 8) and helps a lot to startup the adequate software configuration and to monitor the system status. Either UGV or UGV_for_multi_robot needs to be started as the main node on each robot. The node record_bag is optional and would create bagfiles if started.

The button adaptive_traversability activates the same-named feature of the robot (cf. DR.1.3). The button gas_sensor is used to start the node that reads and publishes the values provided by the gas sensor if mounted. The next two buttons are related to the arm control resp. to the camera mounted at the arm (arm_cam).

The button flc toggles the feature "free look control" (cf. DR.2.3). The button below controls the relay node which is needed for the wireless communication between the UGV and the base station. The next button activates the node new_mapper, which creates a 3D model by registration of single point clouds (cf. DR1.3).

The remaining seven buttons at the bottom are related to the most complex component that runs on the UGV, namely the path_planner. Details on its method of operation can be found in DR.4.3.

1.3.3 Network communication

Based on the ROS publish-subscriber framework, communication between the robot team-members is achieved on one hand indirectly via shared memory (e.g., through the database), on the other hand directly by message exchange. Mobile robots depend on wireless connections with all the inherent drawbacks of WiFi. Apart from that, wired LAN connections are used wherever possible.

The human team-members are either talking directly to each other using arbitrary communication technology (e.g., walkie-talkie) or their dedicated interface devices with support of Mumble [8] (see Figure 9). The advantage of involving Mumble in the communication is, that the automatic speech recognigion can extract the contents of the communication for further processing in the system.

Mumble is used with distinct modes related to the communication needs: The TDS/OCU operators can choose between communication to the other human actors in TRADR or to use the communication facilities on a robot. With the second choice, an operator can talk through the loudspeaker of a robot to, e.g., victims and listen to responses from them or listen to environmental noises, incl. the robot. With this mode, the operator is disconnected from the other channel but can still hear what the team leader says to him.

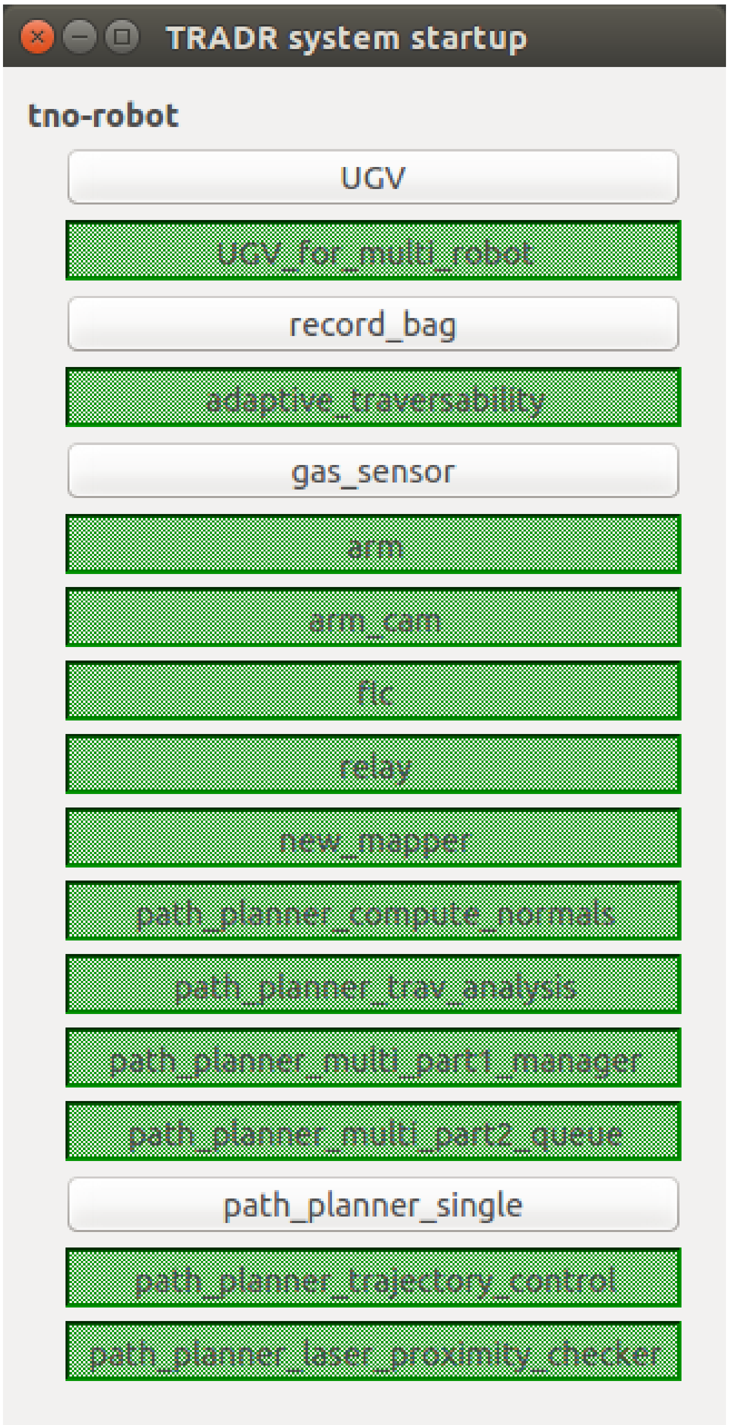


Figure 8: TRADR system startup for UGV: green nodes are up and running.

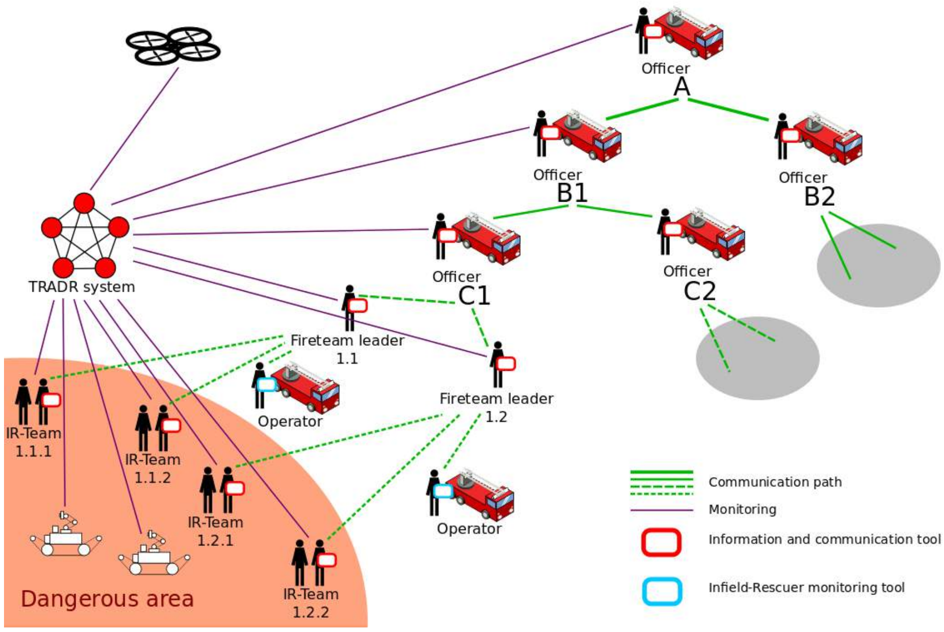


Figure 9: Embedding the TRADR system into the communication structure with human team members.

In order for the robots to execute plans autonomously, while still providing feedback to the operator, we need a message passing method between the base camp and the robots.

Each robot needs a so-called 'roscore' to be able to operate. The roscore manages the different channels (topics) on which messages can be send. In addition, the base camp also has a roscore. The different self-contained systems (robots and base camp) can then explicitly send messages to other actors by using relay nodes.

The relay nodes are implemented by the 'nimbro_network'[9] stack, which had been successfully used in the DLR SpaceBot Cup and in the DARPA Robotics Challenge. It contains a very simple but suitable multimaster model as well as UDP transmission, forward error correction, compression and rate limitation, which enhances the practical transmission quality. It also allows detailed monitoring of the network link quality, e.g. failure rates and available bandwith. Nevertheless wireless communication in dynamic environments stays unreliable and the resilience of the software against failures is vital.

The basic network topology was not changed in Yr3. All wired devices are connected via a 1 GBit/s LAN switch. An up-to-date WiFi 802.11ac triband router is used for the wireless communication. Network services like *DHCP*, *DNS*, and *NTP* run on the TRADR core system.

Figure 10: AscTec Falcon 8 presentation at Technical Day 2016 in Dortmund.

To test the failure safety of all components against network degradation, some random connection malfunctions were introduced at the TJEx and TEval. It showed that all components came back to normal function as soon as the network was up and running again.

1.3.4 UAV development

Typical applications of UAVs in Urban Search and Rescue environments include aerial photography, inspection tasks, and 3D modeling. To perform multiple sorties of UAVs during a mission, integration of information over time as well as fusion and integration of different sensors is needed. In Yr3, the work on these topics was continued and presented at the TRADR booths during ICRA and RoboCup as well as at the Technical Day [43] (Annex Overview 2.1) (see Figure 10).

The AscTec Falcon 8 [10] was used as the default platform for TJEx and TEval in Yr3 as well as for the TRADR mission in Amatrice in September 2016 (see Figure 11). As a commercially available platform, it provides many comfortable features matured over the years based on real experience.

Figure 11: AscTec Falcon 8 equipped with optical camera at Amatrice.

It guarantees maximum safety by the up to 3 times redundant flight controller (AscTec Trinity) and provides useful automatic functions for image acquisition as well as direct control of the flight system and the camera settings.

Analog transmission of the live video data and digital transmission of telemetry data were used for the integration of the Falcon 8 into the TRADR system.

The analog video data is received by an analog receiver, converted by a frame grabber to a digital H264/MPEG-4 stream, published by a ROS bridge and displayed inside a QT-plugin within the OCU that subscribed the video topic. At the OCU the operator can observe the video stream and take single pictures of interesting aspects within the stream. All pictures are shown in an image bar, can be selected, evaluated and shared with other TDS/OCU instances.

The GPS and telemetry data is transmitted over the UAV remote control channels and transmitted to a laptop running the *Navigator* [11] software by Ascending Technologies. On this laptop the GPS data is published as a ROS topic and thus reaches the database. When a screenshot is taken at the OCU, the GPS position of the image is added to the image data. When the image is shared, it is displayed at the correct GPS position within the TDS. The control of the on-board camera movements (pan/tilt/zoom) is also accomplished via the separate laptop. However, it is planned to integrate these functionalities into the OCU as an RQT plugin.

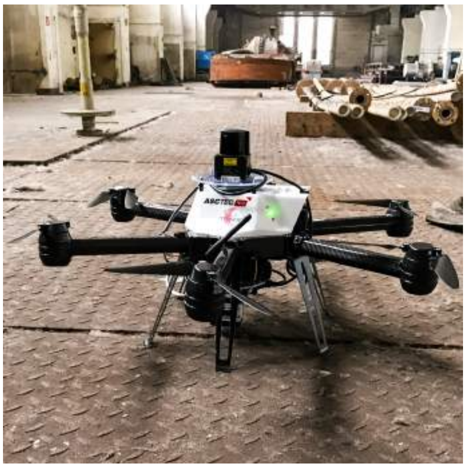


Figure 12: AscTec Neo. Left: on ground. Right: exploring the Knepper site.

Although the Falcon 8 works very well from the end-users' perspective, there is a need for a different platform from the researchers' perspective to make experiments on automatic navigation, planning and obstacle avoidance. The Falcon 8 neither provides the necessary payload for additional sensors nor a powerful onboard computer where ROS can be hosted. The AscTec Pelican [12], which had been used for this purpose until Yr2, was therefore replaced by the AscTec Neo [13] (see Figure 12) as a prototype of a TRADR specific UAV. The goal was to bring both platforms (Falcon 8 and Neo) together so that newly developed algorithms can be evaluated concurrently. However, this development had to be cancelled in Yr3 because the partner Ascending Technologies was not able to fulfill his tasks as a consequence of big organizational changes in the company. The common objective in the consortium is to finalize the integration of UAV technology in the TRADR system in spite of this situation, because it is of great value in USAR applications.

1.3.5 UGV upgrades

We equipped all four flippers with tactile sensing, designed a RealSense holder, see Figure 13. RealSense was integrated into the system and we are currently experimenting with using it for terrain perception, see Figure 14. UGVs were armed with additional electronics allowing remote hard-restart of motor controllers and better control of lighting subsystem. Also the pan-tilt unit has a newly designed power source. We experimented with a new long-range radio control with low band-width significantly going beyond the wifi range. A protype is under construction. We better integrated the gas/smoke detector.

A new sensor for tactile sensing/exploration by the Jaco robotic arm is described in the Deliverable 1.3 and the exploration algorithm in Deliverable 2.3. We strengthened the manipulation capabilities of the UGV by improving the control software for the arm, i.e., better integration in the OCU, providing preset default positions, and automatic object recognition using a small camera attached to the hand.

Figure 13: Left: All four flippers equipped with tactile senesing. Right: 3D printed RealSense holder mounted at the top of the Ladybug 3 camera.

Figure 14: Comparing Lidar (left) and RealSense (right) measurement from the Amatrice deployment. Lidar field of view (measurement) is much larger, yet, the RealSense provides detailed data for the close environment. More importantly, RealSense provides measurements in hight frame rates which makes it promising for real-time robot control (adaptive traversal).

1.3.6 Velodyne integration

In the process of upgrading the TRADR UGV platforms, we have considered the Velodyne PUCK sensor as a replacement for the rotating SICK sensor. To this end, we have integrated a Velodyne sensor to the TRADR UGV as

Figure 15: The Velodyne PUCK integrated to the TRADR UGV platform. It is mounted high in order to augment its field of view and localization capabilities.

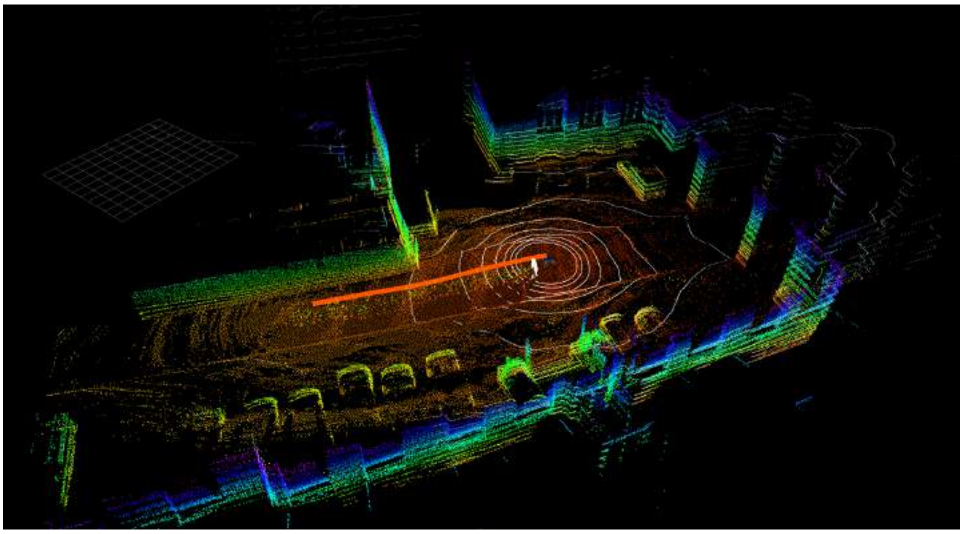


Figure 16: A 3D point cloud, colored by height and created from the Velodyne scans, displayed in white. The sensor trajectory, as estimated by the TRADR laser mapping framework, is shown in orange.

illustrated in Fig. 15. This sensor was also integrated in the TRADR laser mapping framework as shown in Fig. 16.

We discovered that this sensor favors very good localization and mapping accuracy, thanks to its true 360 degrees field of view. However, we have decided not to go forward with this sensor as a global TRADR solution because of its high price and further payload on the UGVs obstructing the field of view, reducing the localization capabilities. Finally, the sparseness of the single Velodyne scans would have required significant changes in many TRADR subsystems which we decided not to address at this stage of the project.

1.3.7 Multi-robot modules

V-REP multi-robot simulator. In order to ease and test the integration of the multi-robot ROS modules in the nimbro network architecture a suitable V-REP simulation framework was designed and implemented. To this end, we have prepared a VREP robot simulator which emits the same topics as an actual robot. Many instances of this simulator can be launched, each one running on a distinct computer and each one acting (from a ROS network perspective) as a real robot. Laser mapper, traversability and path planner can be normally launched as on the actual robots.

In this integration process, the first challenge was to build a multi-roscore V-REP simulator. To this end, the *single-roscore* V-REP simulator already existing in TRADR framework was used as a base. In this starting version, each multi-robot scenario is populated by different robots with names ugv$i$, where $i \in \{1, 2, 3\}$. In each of these single-core scenarios, each robot emits ROS topics with convenient prefixes "/vrep/ugv$i$" (e.g. "/vrep/ugv$i$/robot_status", "/vrep/ugv$i$/odom", etc). This clearly allows ROS topics coming from different robots to live in a single roscore context within the same namespace without conflicts.

On the other hand, in the real TRADR system, each robot has its own roscore and emits ROS topics with standard non-prefixed names (e.g. "/robot_status", "/odom", etc). In order to emulate the real *multi-roscore* architecture, a multi-roscore V-REP simulator has been prepared for each scenario. In particular, each generic V-REP scenario X populated by n robots has been cloned n times. In this new framework, in order to perform a multi-robot simulation, one has to launch n V-REP instances, each one with its own roscore and running the i-th clone of the scenario X. In particular, the i-th scenario clone hosts the robot ugv$i$ as the *main actor*, which emits the same ROS topics of an actual robot, and the other robots as passive *replicants*, i.e. just passively replicating the positions of the corresponding main robot actors in the other running V-REP instances. All the launched V-REP instances are obviously connected within a same network and exchanges the robot positions of the different ugv$i$.

With such a multi-roscore V-REP framework, we are able to use the different instances of the V-REP simulator as actual robots. These can be connected in the nimbro network and to the tradr-core in the same way as real robots. Moreover, as already stated, each roscore can host the same laser mapper, traversability and path planner nodes that can be normally launched on the actual robots.

More detailed information about the V-REP simulator and how it was used within the TRADR project can be found in Task 4.3 Deliverable 4.3.

Multi-robot transformation tree management. A second important challenge was to design and configure a multi-robot RVIZ interface as a single-point GUI where to visualize and control all the running robots and the available maps of the environment. It is worth noting that each roscore running on a real robot hosts the same transformation tree topology. In order to suitably "merge" together the distinct transformation trees coming from the different robots without conflicts, we devised and implemented some dedicated ROS nodes that actually run on the tradr-core, receive the standard non-prefixed frames and convert them into new multi-robot ones by adding a new convenient prefix "/<robot-name>" to the robot links.

Multi-robot maps management. A further challenge was to collocate the distinct robot maps in a single "global" frame without disrupting the software code and architecture. In the previous TRADR years, each robots had its own "/map" and "/odom" frames which were connected together by the frame transformation estimated by the laser SLAM module. In the new multi-robot framework, we obtained the possibility to combine together the distinct maps by adding an intermediate "/<robot-name>/map" frame in each robot, between the main "/map" frame and the "/odom" frame. Now, the new laser SLAM module estimates the transformation between the "/<robot-name>/map" and "/odom" frames, while the user can specify as a starting initial guess (or adjust) the transformation between each "/<robot-name>/map" frame and the "/map" frame, where the latter now assumes the role of a new global frame shared by all the robots.

1.4 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 [32], [30], [34], [37], [29], [31], [26], [22], [27]. 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 [19], [18], [33], [16], [36]. Since 2012, the DARPA Robotics Challenge tries to promote the development of disaster reponse robots; few participants from Europe are currently involved [25]. Furthermore at Eurathlon, a land-air-water challenge, a realistic disaster scenario is targeted [14].

The TRADR system addresses some of these open questions. Omnidirectional cameras mounted on the robots 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 [30], [40]. In addition to raw camera views, a mapping system helps to keep track of the current position and state of the robot. By using well established laser range finders (2D and 3D) together with state-of-the-art mapping technologies, the robot is autonomously recording a representation of the environment and presenting itself correctly located and aligned in this representation to the user [36], [45], [24], [35], [44], [46].

To keep the human operator in the loop, we use a graphical user interface to present the preprocessed information and to receive commands from the user. This technology is well elaborated [15], [16] and has several advantages. In comparison to a raw video/sensor display and joystick-like control system like those presented in [18], [33], [26], 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. Moreover such an interface can contribute to the shared understanding on the team level and present users information about what everybody else is doing providing the right information to the right actors [39], [42].

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 [32], [30], [41]. For the TRADR system, we continue to use the highly adaptive UGV, which is a track-based platform and able to traverse complex terrain. In addition, we apply 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 [21] and

DARIUS [20] target the development of robotic tools that can assist during disaster response operations, focusing on autonomy. SHERPA [28] 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 [17], 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. In the Eurathlon (air+land+sea) competition [14], the teams are asked to deliver a representation of the paths travelled, point of interests found, etc., but using this information in subsequent sorties is not part of the task. Additionally, the EU project STRANDS [23], 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. The TRADR concept of persistent situation awareness goes beyond this in various respects dealing with persistence in multiple sorties in an unstructured outdoor environment carried out by human-robot teams.

References

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

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

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

[4] `http://www.mongodb.org/`.

[5] `http://stardog.com`.

[6] `http://wiki.ros.org/rqt/`.

[7] `http://www.qt.io/`.

[8] `http://www.mumble.info`.

[9] `https://github.com/AIS-Bonn/nimbro_network`.

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

[11] `http://www.asctec.de/en/deutsch-asctec-navigator-1-0-jetzt-verfuegbar/`.

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

[13] `http://www.asctec.de/downloads/public/AscTec-Neo-flight-robot-research-uav-uas-drone-hexacopter-intel-realsense-iros-aeroworks.pdf`.

[14] `http://www.eurathlon.eu/index.php/compete2/eurathlon2015`.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] Daniel Reuter and Erik Zimmermann. Report on integration test december 2016. Unpublished.

[39] PM Salmon, NA Stanton, GH Walker, Daniel Jenkins, C Baber, and Richard McMaster. Representing situation awareness in collaborative systems: A case study in the energy distribution domain. *Ergonomics*, 51(3):367–384, 2008.

[40] 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.

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

[42] Neville A Stanton, Rebecca Stewart, Don Harris, Robert J Houghton, Chris Baber, Richard McMaster, Paul Salmon, Geoff Hoyle, Guy Walker, Mark S Young, et al. Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics*, 49(12-13):1288–1311, 2006.

[43] Hartmut Surmann and Rainer Worst. How can MAVs assist human-robot teams in disaster response over multiple sorties? Poster from TRADR Technical Day 2016.

[44] 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.

[45] 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.

[46] 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.

2 Annexes

2.1 Surmann, Worst (2016), "How can MAVs assist human-robot teams in disaster response over multiple sorties?"

Bibliography Hartmut Surmann, Rainer Worst. "How can MAVs assist human-robot teams in disaster response over multiple sorties?" Poster from TRADR Technical Day 2016 in Dortmund.

Abstract Typical applications of Micro Aerial Vehicles (MAVs) in Urban Search and Rescue environments include aerial photography, inspection tasks, and 3D modeling. To perform multiple sorties of MAVs during a mission, integration of information over time as well as fusion and integration of different sensors is needed.

Relation to WP This poster describes essential MAV actions and approaches to use them for the creation of persistent models, which were developed in the context of T6.4.

Availablity Unrestricted. Included in the public version of this deliverable.

2.2 Reuter, Zimmermann (2016), "Report on integration test December 2016"

Bibliography Daniel Reuter, Erik Zimmermann. "Report on integration test December 2016." Unpublished report, December 2016.

Abstract Every month a snapshot of the TRADR repository is taken and Fraunhofer performs a test focusing on integration issues according to a predefined list of system features. A report is created to document the results of the test, including links to the resp. tickets created in Redmine.

Relation to WP As a sample, this report describes the results of one of the integration tests that were performed monthly in the context of T6.4.

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

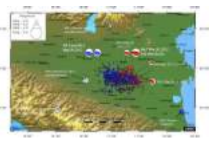
Long-Term Human-Robot Teaming for Robot Assisted Disaster Response

How can MAVs assist human-robot teams in disaster response over multiple sorties?

Hartmut Surmann, Rainer Worst and the TRADR consortium*

Typical applications for Micro Aerial Vehicles (MAVs) in Urban Search and Rescue environments:

- **Aerial photography, Inspection tasks, 3D Modeling**

Disaster response is not just “in-and-out”.

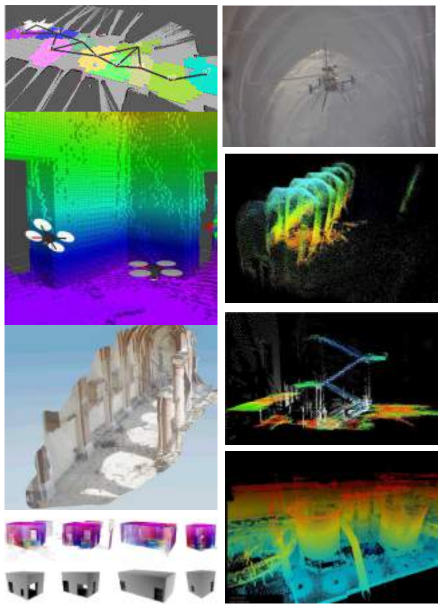
MAVs perform multiple sorties during missions.

<u>Needed:</u>

- **Integration** of information, to create persistent situation awareness.
- **Fusion** and **integration** of different sensors i.e. Mono / Stereo / Omni Cameras, 2D / 3D Laser scanners, Radar, GPS, Gyros, Compass ...

MAV actions

1. Localization and Planning
2. Construction of 3D models for dynamic environments, from observations obtained **over time** across **multiple sorties**
3. Persistent models for MAVs acting in environments with or without GPS.
4. Persistent models for human-robot teaming




* DFKI Saarbrücken Germany, TU Delft Netherlands, Fraunhofer IAIS Germany, KTH Schweden, ETH Switzerland, CTU Czech Republic, Uni Roma Italy, Ascending Tech. Germany, Inst. für Feuerwehr und Rettungstechnologie Dortmund Germany, Corpo Nazionale Vigili del Fuoco Italy, Unified Industrial & Harbour Fire Department Netherlands, TNO Netherlands
This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no. *609763.*