DR 6.4: Persistence in long-term human-robot teaming for robot assisted disaster response

Rainer Worst[*], Hartmut Surmann[*], Erik Zimmermann[*], Daniel Reuter[*], Tim Buschmann[*], Artur Leinweber[*], Alexander Schmitz[*], Gerhard Senkowski[*], Niklas Goddemeier[†], Luigi Freda[‡], Marco Tranzatto[§], and the TRADR consortium

[*]*Fraunhofer Institut für Intelligente Analyse- und Informationssysteme IAIS*

[†]*Smart Robotic Systems GmbH, Dortmund*

[‡]*Alcor Laboratory, Department of Computer, Control, and Management Engineering "Antonio Ruberti" - Sapienza University of Rome, Italy*

[§]*ETH Zürich - Autonomous Systems Lab, Zurich, Switzerland*

⟨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 2017
Actual submission date:	February 2018
Lead partner:	Fraunhofer
Revision:	version 1
Dissemination level:	PU

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

---

1 Tasks, objectives, results **5**
1.1 Planned work 5
1.2 Addressing reviewers' comments 5
1.3 Actual work performed 6
1.3.1 System development 6
1.3.2 System setup 7
1.3.3 Communication network 8
1.3.4 Flexible multi-robot configuration 9
1.3.5 Showcase TRADR mission control 12
1.3.6 Integration of UAV capabilities 14
1.3.7 Integration for ERL Emergency Robots 14
1.4 Relation to the state-of-the-art 17

References **19**

2 Annexes **23**
2.1 Surmann et al. (2017), "3D mapping with an Unmanned Aerial Vehicle (UAV)" 23
2.2 Worst et al. (2018), "Technical Report: TRADR UAV Integration" 23

A Poster: 3D mapping with an Unmanned Aerial Vehicle (UAV) **24**

Figure 1: TEval 2017 at the Deltalinqs training plant in Rotterdam.

Executive Summary

This report describes the final result of WP6 in Yr4 – an integrated system dealing with persistence in long-term human-robot teaming for robot assisted disaster response. To achieve this goal, one single task was addressed:

Task T6.5 Persistence in long-term human-robot teaming for robot assisted disaster response

The development of the final prototype of the TRADR system was based on the technical framework set up in the previous years, including ROS as middleware system, GIT repository for source code, TRAC with Redmine for issue tracking, a squad of five ground robots, some aerial robots, and several PCs to run the control software of the TRADR system. Particular attention was paid to the resilience of the communication network on the conceptual as well as on the practical level.

The task was to integrate incrementally a system for the project-wide scenario of Yr4 **Persistence in long-term human-robot teaming for robot assisted disaster response**. 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.7.4). The system evaluation took place at the Deltalinqs training plant in Rotterdam (see Figure 1).

Role of system framework and integration in TRADR

In Yr4, the integrated robotic system has been evolved with respect to persistence in long-term human-robot teaming for robot assisted disaster response. The system's evolution process was guided by applying user-centric engineering practices, which played a central role in TRADR during the entire course of the project. Task T6.5 merged the functionality developed by WPs 1-5 and integrated incrementally the final prototype of the TRADR system.

Persistence

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. 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* [1] 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* [2] high-level database, a python wrapper API is used to handle data sets.

Users are able to create images 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 the final prototype of the integrated cognitive robot system, based on the technical framework developed in the previous years. It enables us to showcase the architecture and the capabilities of the TRADR system for interested stakeholders.

1 Tasks, objectives, results

1.1 Planned work

The task addressed by WP6 in Yr4 was:

Task T6.5 Persistence in long-term human-robot teaming for robot assisted disaster response

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 Yr4 "Persistence in long-term human-robot teaming for robot assisted disaster response". 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 comments on the Yr3 efforts that are related to WP6:

(A) The leave of Ascending Technologies requires adaption and redistribution of work and resources from AscTec to other partners and in parts possibly to a subcontractor. The presented contingency plan is fully endorsed by the reviewers.

(B) Network resilience of the TRADR system is an indispensable prerequisite for any deployment subject to realistic conditions. Therefore, the reviewers re-emphasize their recommendation from last year to systematically perform network profiling and to report bandwidth constraints, in real-time during a mission as well as in an aftermath analysis. On this basis, it must be ensured that the TRADR system and its components are resilient to networking issues, such as reduced bandwidth, delays and outages. This still needs to be fully considered and demonstrated.

(C) As Yr4 will be the final year of the project, it is high time for the TRADR consortium to prepare what will be the project's legacy. The partners are strongly recommended to prepare plans how the project could achieve a lasting impact well beyond its duration.

Item A was executed as presented in the contingency plan by redistributing AscTec's work and resources partially to Fraunhofer and DFKI (for subcontracting). The actual work done is described in section 1.3.6.

Item B had been addressed in Yr3 by switching from *rocon* to *nimbro* nodes as base for the communication between different ROS masters. In Yr4, we continued to work on this issue by using the built-in Forward Error Correction of *nimbro*, by the adaptation of the compression rate of images, and by the visualization of the current network quality for the users. The most outstanding contribution however is the design and implementation of a network-aware planning component as described in deliverable DR.2.4. More details on the network setup are given in section 1.3.3.

As a contribution to item C, the stationary components of the TRADR system have been installed in a standard container that was provided by the partner GB. This allows us to showcase the operation of the system in a more realistic way. This effort is described in section 1.3.5.

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 Yr4. Section 1.3.2 describes the general setup of the TRADR system and section 1.3.3 addresses network issues. Section 1.3.4 is about the new flexible orchestration tool that supports multi-robot exploration and patrolling. In section 1.3.5, the new showcase for TRADR mission control is described. Section 1.3.6 is about the UAV integration as specified in the recent amendment of the DoW. The final section 1.3.7 describes the specific setup for the competition ERL Emergency Robots.

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 [3] as framework, GIT [4] as repository for the source code, and Redmine [5] as issue tracker system.

In Yr4, 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.

We continued to take each month a snapshot of the TRADR system from the GIT repository and perform a system test at Fraunhofer. The detected errors were documented and tickets were assigned in Redmine. Shortly after such a system test, a developers' teleconference took place, where particular issues could be discussed in detail if necessary. The described scheme supported the development process very well, taken into account the inherent constraints of a joint research project.

1.3.2 System setup

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

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

As in Yr3, the TRADR system consists of a central data storage, monitoring units, robot control, and the robots themselves.

By adding 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, the computational abilities for resource consuming processes like video data procession and map registration were improved because parallel processing based on CUDA became achievable.

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 DR.3.4 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.

1.3.3 Communication network

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]. 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.

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.

In order for the robots to execute plans autonomously, while still providing feedback to the operator, we need a message passing method between the control room 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 control room also has a roscore. The different self-contained systems (robots and control room) 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 only slightly changed in Yr4 (see Figure 2). 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 no longer on the TRADR core system, but were moved to a dedicated router device with pfSense[10].

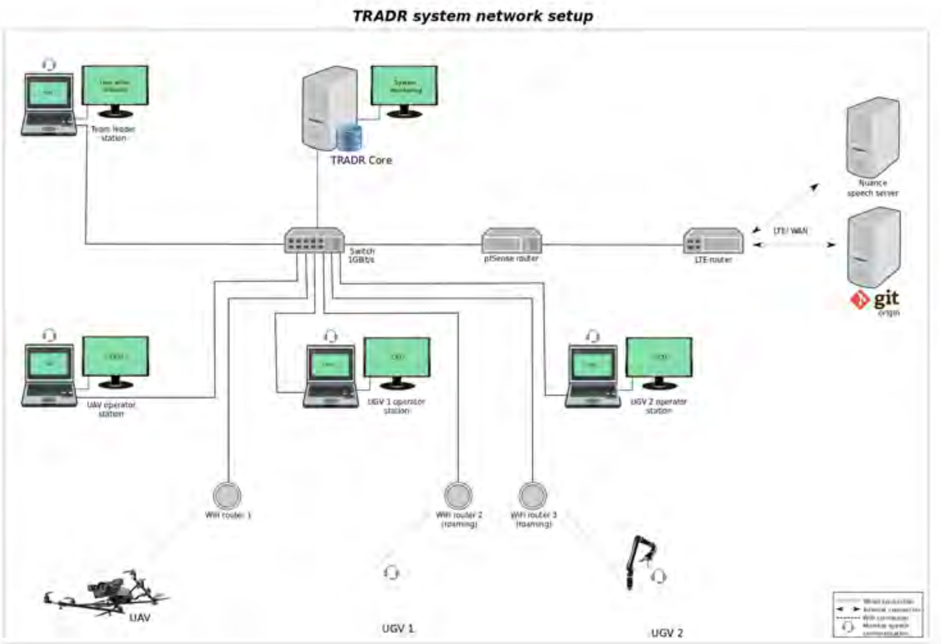


Figure 2: Network topology of the TRADR system with wired and wireless devices.

1.3.4 Flexible multi-robot configuration

As TRADR consists of many different connected components with many possible states, a proper monitoring of the current conditions is indispensable. An orchestration tool (aka *Orchestra*, see Figure 3) is used to start and stop the components as well as to visualize their current status.

Specifically, *Orchestra* allows the system administrator to launch, control, and monitor ROS nodes on the UGVs, TDS and on the TRADR core. In general, each system capability corresponds to a dedicated GUI button, which color informs the end-user about the state of the corresponding set of nodes (green $\mapsto$ nodes are alive; red $\mapsto$ nodes died). One of the main benefits of *Orchestra* is to drastically simplify the launching and monitoring of ROS nodes in the distributed and complex architecture of the TRADR system. Hence, the role of such a system administrator can be filled by an end-user.

In this regard, *Orchestra* can be flexibly configured for different purposes. In particular, different configurations are possible:

1. *TRADR core*: the tool is launched on the TRADR core, configured for the target TRADR deployment.

2. *Laptop core*: the tool is launched on a laptop, used as a minimal

Figure 3: *Orchestra* provides a visualization of the current system status and is used to start and stop single components.

TRADR core, to locally launch and control both visualization GUIs and required core nodes, and to remotely launch peripheral nodes on UGVs.

3. *TRADR core clone*: the tool is launched on a laptop or a different TRADR module, used as an external clone of the central orchestration tool running on the TRADR core. This allows to distribute the Orchestra management workload over different end-users.

Each of the aforementioned configurations corresponds to a different configuration file, which can be easily customized in order to include and launch a needed subset of system capabilities on each module of the TRADR system (TRADR core, UGVs and TDS). The configuration of the orchestration tool has been thoroughly documented in the README file of the corresponding ROS package, with the aim of making it readily usable by all the partners.

The laptop core configuration (laptop used as a minimal TRADR core) has been particularly useful during multi-robot software developments and testing. It was also used for performing separate live demos (e.g. multi-robot patrolling and exploration) with a single laptop used as a minimal TRADR core, without requiring the full deployment of the central TRADR core and TDS.

During the TRADR experience, the need of a flexible multi-robot configuration has emerged. The composition of the deployed TRADR multi-robot team is required to be readily changed at any time: different combinations of robots can be selected for multi-robot tasks. In the TRADR system, robots are mainly identified by short names such as *roma*, *eth*, *ctu*, *dfki*, *tno*, *sim1*, *sim2* and *sim3*[1]. On the other hand, a classical multi-robot system typically identifies each robot by using a unique ID number. Such a scheme allows to simplify the multi-robot software design and the implementation of the required coordination protocols, without recurring to complex tables and complex remapping mechanisms. This is particularly useful in the complex TRADR system architecture, where GUI configurations and ROS topic relays are preferred to be maintained fixed. To this aim, we provided *Orchestra* with a main global configuration file that allows easily to assign a standard multi-robot ID to the robots selected for the mission. In this context, each robot can be independently configured with a different set of robot capabilities according to the needed tasks.

It is worth mentioning that the flexible configuration of *Orchestra* (in particular the laptop core mode) allowed to deploy and test the patrolling framework on a team of three UGVs. This is a new achievement obtained in Year 4. More details and videos are provided on the web page [11].

Technically, these results were obtained by further developing and improving the multi-robot system communication architecture over nimbro. Due to the high complexity of the software system involving many different technologies, i.e. Python, Java, C++ and custom DSLs[2] and multiple different robot configurations (number of UGVs/UAVs, different sensors, different UAV vendors) managing all of the required software dependencies and configurations can be challenging. In order to mitigate this an idempotent deployment procedure was developed and maintained. It is based on the open source framework Ansible[12].

With this deployment procedure it is possible to recreate the current software stack on every machine or to update it, once a component or configuration has changed. The process is started on a single machine, which logs in to all hardware components and makes the necessary changes to adapt to the (newly) specified environment. This makes it also possible to swap components, i.e. the *TRADR core system* in case of for example a hardware failure.

Since every hardware component has to be configured slightly differently (the TDS for the teamleader and UGV operators have different views, the robots need to be aware of their sensors/capabilities), templates are used to reduce the amount of inconsistency between setups, while still being flexible in changing minor details. These differences are stored in a single configuration, which makes it easy to adapt to different situations.

With this configuration at deployment time approach it is possible to in-

[1]The last three robots *simi* correspond to simulated robots in the V-REP robot simulator.

[2]Domain Specific Languages

tegrate and configure even third party and proprietary software (i.e. mumble or stardog) and otherwise hard to integrate software components.

Configurations that are expected to change between or within one sortie are kept in the orchestration tool and are loaded at runtime, through the linux environment. This means, that if a change must occur only the orchestration tool and the effected components need to be restarted.

1.3.5 Showcase TRADR mission control

Figure 4: Partner GB delivers roll-off container at Fraunhofer campus.

One important issue of the integration activities in the final year of TRADR was the creation of a hardware/software system that is able to persist after the end of the project. This is a reqirement for all follow-up dissemination actions that include demos of the system functionality. While the configuration of the robots always was straight-forward, the setup of the control components (TRADR core, TDS, OCU) and network infrastructure needed a lot of discipline to keep all necessary parts together and to assemble them in the correct way.

This situation could be improved significantly when the partner GB provided a roll-off container for the TRADR consortium, which had been used as a break room before. Figure 4 shows the arrival of this container at the Fraunhofer campus.

The idea was to refurbish the interior of the container and to create a control room for TRADR. This would allow us to showcase the TRADR mission control in a more realistic way than before. Therefore, the guiding objective was formulated in this way:

- The container should be usable for a system prototype demonstration in operational environment

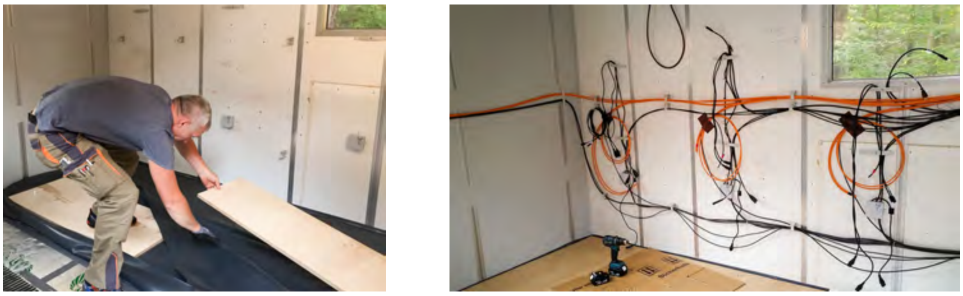


Figure 5: New floor and wiring for the TRADR control room.

Given the small size of the container (5 m x 2.4 m), compromises were unavoidable, regarding the ergonomics of the work stations as well as the available space for the audience of the demonstrations. Within few months between the birth of the idea and the date of the TEval, we removed all unusable furniture, renovated the floor and the walls (Figure 5, left), and did the wiring for electricty and data communication (Figure 5, right).

Figure 6: Four workstations for operators inside the TRADR control room.

Four computer workstations, each with two screens, can be used by UGV and UAV operators (Figure 6). Additionally, a TDS workstation is provided for the team leader. The TRADR core and the network equipment is located in a separate smaller room, which can be also used as a storage compartment during the transport of the container.

During TEval, the new control room was used successfully for the first time (see deliverable DR.7.4). The pre-built setup of the entire TRADR system allows us to preserve the results of TRADR in a compact way and to showcase the main functionalities on any demand.

1.3.6 Integration of UAV capabilities

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 Yr4, the work on these topics was continued and presented during the Technical Day [44] (Annex Overview 2.1).

The AscTec Falcon 8 [13] was used as the default platform for TJEx and TEval in Yr3 as well as for the TRADR mission in Amatrice in September 2016. As a commercially available platform, it provides many comfortable features matured over the years based on real experience. Although the Falcon 8 works very well from the end-users' perspective, there was 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 Neo [14] had been chosen as the prototype of a TRADR specific UAV. The goal was to bring both platforms (Falcon 8 and Neo) together so that newly developed algorithms could be evaluated concurrently. However, this development had to be cancelled in Yr3 because the partner Ascending Technologies was not able to fulfill its tasks as a consequence of big organizational changes in the company. Therefore, an amendment of the DoW was necessary and executed in Yr4 as described in [46] (Annex Overview 2.2).

1.3.7 Integration for ERL Emergency Robots

The TRADR consortium competed in a robotics challenge, named European Robotics League (ERL), in Summer 2017 in Piombino (Italy). ERL is a civilian, outdoor robotics competition, with a focus on realistic, multi-domain emergency response scenarios, inspired by the 2011 Fukushima accident.

In this competition, different kinds of robots, land and aerial, interacted with each other in a shared environment to perform a common task in a search and rescue scenario. To enable the TRADR UGV to cooperate with other robots, a preliminary phase of system integration was required, where new hardware and software components were integrated into the existing system. NIFTI took part in a larger team competing in the ERL Challenge, composed by an Unmanned Aerial Vehicle (UAV), named Jay, and another Unmanned Ground Vehicle (UGV) called ANYmal (see Figure 7).

Integration of Piksi - RTK GPS Receiver These three robots were equipped with a Real-Time Kinematic (RTK) GPS receiver. An RTK receiver is a device able to provide absolute position measurements with a

Figure 7: European Robotics League Challenge - Land and Air Robots: TRADR and ANYmal (UGVs), and Jay (UAV).

centimeter accuracy. Such precise measurements require that another RTK receiver is used as a “base station”. The base station is a fixed and steady RTK device, with a known global position, that sends position corrections to other RTK receivers, which then compute their absolute position very precisely. Piksi Multi RTK receiver by Swift Navigation [15] was installed on the robots. This device provides global position measurements with an accuracy of 2-3 cm in the horizontal plane and 8-10 cm in the vertical plane.

The new component required the development of a custom ROS driver to be utilised in the existing software architecture of the ground and aerial robots. The ROS driver was open sourced and is available on-line [16]. The main advantage of this driver is to support a redundant-two link communication for position corrections: radio link and WiFi. By using a redundant link it is possible to increase the robustness of the whole system, which still keeps on working if one of the two link fails, and increment the fix convergence speed up to a factor of 2.

By employing the same model of RTK receiver, the three robots were able to localize themselves in a common environment, in an absolute reference frame. This feature is very convenient in case the robots have to share information, such as a common map.

Integration of the UAV Platform A custom UAV was developed specifically to compete in the ERL Challenge. The Challenge description for aerial vehicles [17] required the usage of special hardware, such as Long Range Systems (LRS) radiolinks operating in the 433 MHz band, and wireless emergency stop, to ensure safeness during operation. These components are not usually present in off-the-shelf drones, that is why the team had to develop a new system by employing hardware components from different brands. The developed UAV was composed by a DJI frame, a Pixhawk flight controller unit, and a NUC-i7 as main on-board PC. The UAV was equipped with a Piksi Multi RTK GPS receiver which was integrated in its software stack. With the addition of this component the UAV could build a precise and accurate 3D map using an open source volumetric mapping library [18]. The combination of these two tools resulted in a more accurate 3D map, as compared to when solely using the visual inertial odometry state estimator of the UAV. Moreover, as already mentioned, the built map was shared with the two UGVs that could use it to compute a traversable path of the area that was previously explored by the flying robot.

Integration of TRADR UGV The TRADR UGV was equipped with a Piksi Multi RTK GPS receiver, a payload delivery system, and a wireless safety switch (see Figure 8). The payload delivery system enabled the robot to carry a payload, specifically a med-kit, and to release it close to a victim in the search and rescue scenario of ERL. The wireless safety switch allowed an operator to stop the robot in any moment, in case of dangerous situations. This addition was required to address the safety requirements and compete in the ERL Challenge [19].

Figure 8: European Robotics League Challenge - TRADR UGV equipped with RTK GPS receiver, payload releaser, and wireless safety switch (left), exploring the competition area together with Jay (right).

Network Solution The three robots were connected to the same WiFi network and were able to share core information such as RTK position corrections (sent from the common base station), and a map containing the reconstructed terrain of the competition area. To avoid bandwidth overload the three robots had to throttle feedback information they were sending back to their operators.

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 [40], [34], [36], [31], [27], [32]. 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], [37], [22], [39]. Since 2012, the DARPA Robotics Challenge tries to promote the development of disaster reponse robots; few participants from Europe are currently involved [30]. Furthermore at Eurathlon, a land-air-water challenge, a realistic disaster scenario is targeted [20].

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 [35], [42]. 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 [39], [47], [29], [38], [45], [48].

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 [21], [22] and has several advantages. In comparison to a raw video/sensor display and joystick-like control system like those presented in [24], [37], [31], 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 [41], [43].

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 [33] 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 [23], 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 [20], 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 [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. 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://www.mongodb.org/.

[2] http://stardog.com.

[3] http://wiki.ros.org/indigo.

[4] http://git-scm.com/.

[5] http://www.redmine.org/.

[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] https://www.pfsense.org/.

[11] https://sites.google.com/a/dis.uniroma1.it/3d-cc-patrolling/.

[12] https://www.ansible.com/.

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

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

[15] https://www.swiftnav.com/piksi-multi.

[16] https://github.com/ethz-asl/ethz_piksi_ros.

[17] https://www.eu-robotics.net/robotics_league/upload/documents-2017/ERL_ER_Rule_book_v1_5.pdf. Section 5.1.5.

[18] https://github.com/ethz-asl/voxblox.

[19] https://www.eu-robotics.net/robotics_league/upload/documents-2017/ERL_ER_Rule_book_v1_5.pdf. Section 5.1.3.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] Hartmut Surmann, Rainer Worst, Erik Zimmermann, Kalle Knipp, Daniel Reuter, Tim Buschmann, Artur Leinweber, Alexander Schmitz, and Gerhard Senkowski. 3D mapping with an Unmanned Aerial Vehicle (UAV). Poster from TRADR Technical Day 2017.

[45] 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 Safety, Security and Rescue Robotics*, 2010.

[46] Rainer Worst, Hartmut Surmann, Erik Zimmermann, Daniel Reuter, Tim Buschmann, Artur Leinweber, Alexander Schmitz, Gerhard Senkowski, and Niklas Goddemeier. TRADR UAV Integration. Unpublished Technical Report, 2018.

[47] 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.

[48] 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 et al. (2017), "3D mapping with an Unmanned Aerial Vehicle (UAV)"

Bibliography Hartmut Surmann, Rainer Worst, Erik Zimmermann, Kalle Knipp, Daniel Reuter, Tim Buschmann, Artur Leinweber, Alexander Schmitz, Gerhard Senkowski. "3D mapping with an Unmanned Aerial Vehicle (UAV)". Poster from TRADR Technical Day 2017 in Rotterdam.

Abstract Inspection and mapping are typical applications for UAVs in Urban Search and Rescue environments. However, disaster response is not just in-and-out, but UAVs perform multiple sorties during a mission. Fusion of different sensors and integration of information is needed to create persistent situation awareness.

Relation to WP This poster describes the work flow of UAV actions, including their pre- and post-processing, for the creation of persistent models, which was developed in the context of T6.5.

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

2.2 Worst et al. (2018), "Technical Report: TRADR UAV Integration"

Bibliography Rainer Worst, Hartmut Surmann, Erik Zimmermann, Daniel Reuter, Tim Buschmann, Artur Leinweber, Alexander Schmitz, Gerhard Senkowski, Niklas Goddemeier. "Technical Report: TRADR UAV Integration." Unpublished technical report, February 2018.

Abstract This report describes the integration of different UAV platforms into the TRADR system – based on a common UAV framework – and the related workflow for the 3D mapping with UAVs.

Relation to WP This report presents the results of the UAV-related amendment of the DoW related to T6.5.

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

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

Fraunhofer IAIS

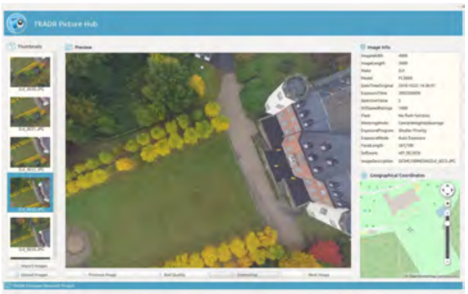
3D mapping with an Unmanned Aerial Vehicle (UAV)

Hartmut Surmann, Rainer Worst, Erik Zimmermann, Kalle Knipp, Daniel Reuter,
Tim Buschmann, Artur Leinweber, Alexander Schmitz, and Gerhard Senkowski

EU FP7 Grant 609763

3D mapping is a typical application for UAVs in Urban Search and Rescue Environments. However, disaster response is not just "in-and-out", but UAVs perform multiple sorties during a mission. Fusion of different sensors and integration of information is needed to create persistent situation awareness.

TRADR UAV Workflow



DFKI Saarbrücken Germany, TU Delft Netherlands, Fraunhofer IAIS Germany, KTH Sweden, ETHZurich Switzerland, CTU Prague Czech Republic, La Sapienza University Rome Italy, Ascending Technologies (Intel) Münich Germany, Institute für Feuerwehr und Rettungstechnologie Dortmund Germany, Corpo Nazionale Vigili del Fuoco Italy, Unified Industrial & Harbour Fire Department Netherlands, TNO Netherlands