DR 6.2: Multiple asynchronous sorties to assess a large-scale dynamic disaster area

Rainer Worst*, Dong-Uck Kong*, Erik Zimmermann*, Timea Bagosi†, Tomáš Svoboda‡, Michael Achtelik§ and the TRADR consortium

*Fraunhofer Institut für Intelligente Analyse- und Informationssysteme IAIS
†Technische Universiteit Delft
‡Czech Technical University in Prague
§Ascending Technologies
⟨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 2015
Actual submission date:	February 2016
Lead partner:	Fraunhofer
Revision:	final
Dissemination level:	PU

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

Figure 1: Location of Yr2 system setup at Phoenix site in Dortmund

Executive Summary

This report describes the final result of WP6 in Yr2 – an integrated cognitive robot system dealing with human-robot team building and situation awareness in dynamic environments. To achieve this goal, one single task was addressed:

Task T6.3 Multiple asynchronous sorties to assess a large-scale dynamic disaster area

The development of the second 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 upgraded ground robots, some new aerial robots by Ascending Technologies, and several PCs to run the control software of the TRADR system.

The task was to integrate a system for the project-wide scenario of Yr2 **Multiple asynchronous sorties to assess a large-scale dynamic disaster area**. Therefore, the functionality developed by WPs 1-5 was merged to an operable system for human-robot team operations. This integrated system was based on the user needs analysis and was used for the scenario-based evaluation, both performed by WP7 (cf. deliverable DR.7.2). The system evaluation took place at the Phoenix site in Dortmund (see Figure 1).

Role of system framework and integration in TRADR

In Yr2, the integrated robotic system has been evolved with respect to issues of dynamic environments. According to the scenario-based roadmap of TRADR the development process will be continued in the next two years by emphasizing multi-robot task adaptation and persistence in long-term human-robot teaming. The system's evolution process is guided by applying user-centric engineering practices, which play a central role in TRADR.

Contribution to the TRADR scenarios and prototypes

The results presented in this report contribute to the TRADR project by providing a second prototype of the integrated cognitive robot system, based on the technical framework developed in Yr1. 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.

Persistence

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, the two-layered architecture with a low-level and a high-level database showed to be an appropriate outset. A database API service was established and 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 was 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.

In WP1, local point clouds are computed (DR1.2), stored, and mutually registed in a bigger cloud (DR.1.2). The big clouds from different sorties and robots are also registered (DR1.2). In addition, WP1 uses local clouds, if available, for controlling the UGV morphology – adaptive traversal. WP4 uses the single or registered clouds for their path planning and autonomous driving (DR.4.2). A user can see the clouds and set a goal within the map which the robot should reach autonomously. Users can add points of interests (POIs) to the map like injured people or no-go-areas (DR.3.2.). These persistent POIs can be seen by other users and evaluated by the agents (DR.5.2.).

Furthermore the status of humanoid and robot team members is stored in the database as well as the status of important components. Thus, a monitor program can survey the current state of parts of the TRADR system and the agents can keep track of the current conditions. Human communication is recorded and processed by automatic speech recognition. The communication and its recognized content can be extracted and evaluated through the mission summary reports in the team reporting tool developed in WP5 ([28]..

Stardog recently released version 4.0 with added features in terms of geospatial query capabilities. Until now besides the storage of positional data no geospatial queries were possible in the high-level database like querying and comparing a robots position in relation to POIs or declared no-go-areas. For the future also the utilization of geospatial queries is planned.

1 Tasks, objectives, results

1.1 Planned work

The task addressed by WP6 in Yr2 was:

Task T6.3 Multiple asynchronous sorties to assess a large-scale dynamic disaster area

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 Yr2 "Multiple asynchronous sorties to assess a large-scale dynamic disaster area". Therefore, the functionality developed by WPs 1-5 had to be merged to an operable system for human-robot team operations. This integrated system was based on the user needs analysis and it was used for the scenario-based evaluation, both performed by WP7.

1.2 Addressing reviewers' comments

The comments on the Yr1 efforts of WP6 made by the reviewers were the following:

(A) Establish a common experimental setup shared by all partners to make the system integration easier

(B) Show network resilience to degradation in quality of connection

(C) Test for network resilience and report on it

(D) Can setup time be shortened?

Item A has been addressed by providing an experimental setup at Fraunhofer consisting of a central TRADR core system, one UGV, one UAV, and four TDS devices; this setup can be used at any time for integration testing. In addition, first steps were made to replace physical components by simulation.

Item B was considered in the development work of WP1-WP5, e.g., in WP2 some work has been done on network aware teleoperation of the UGV. We did also experiments with an industrial solution that allows dynamic creation of a mesh network [6]. If a UGV goes out of the signal, another can be sent in its direction - creating a mesh node - extending the wifi range.

A future/industrial rescue system could use it. Regarding item C, we now provide means to disturb the network [7] and to monitor the system (1.3.4).

To address item D, we changed the complete setup strategy. The measurement of setup time during TJEx (TRADR Joint Exercise) and TEval (TRADR Evaluation) showed a constantly decreasing setup time with the result of having an up-and-running system under an hour. Now we use only predefined hardware components and distinct software versions for the TRADR system which will further decrease the setup time. Furthermore we install as much software components as possible on the TRADR core system and improved the deployment process.

1.3 Actual work performed

This description starts with an introduction to the TRADR system architecture in section 1.3.1, which reflects the state of the ongoing discussion between all partners in Yr2. The following sections 1.3.2, 1.3.3, 1.3.4, 1.3.6, and 1.3.7 provide details about the setup of the TRADR system referring to the system setup in general, database design, network issues, ground and aerial robots. Finally, in section 1.3.8 the release of TRADR datasets for benchmarking purposes is addressed.

1.3.1 Principles of system architecture

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

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

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

Persistence TRADR has the objective to use data that has been collected across sorties to improve performance of the system and the team. These snapshots of the world model at different points in time can be used during the mission by the team-members, mainly to improve their planning activities.

After the mission, the collected data provide a detailed documentation of the entire event. 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 data layers, in which the corresponding data types are stored (see Figure 3).

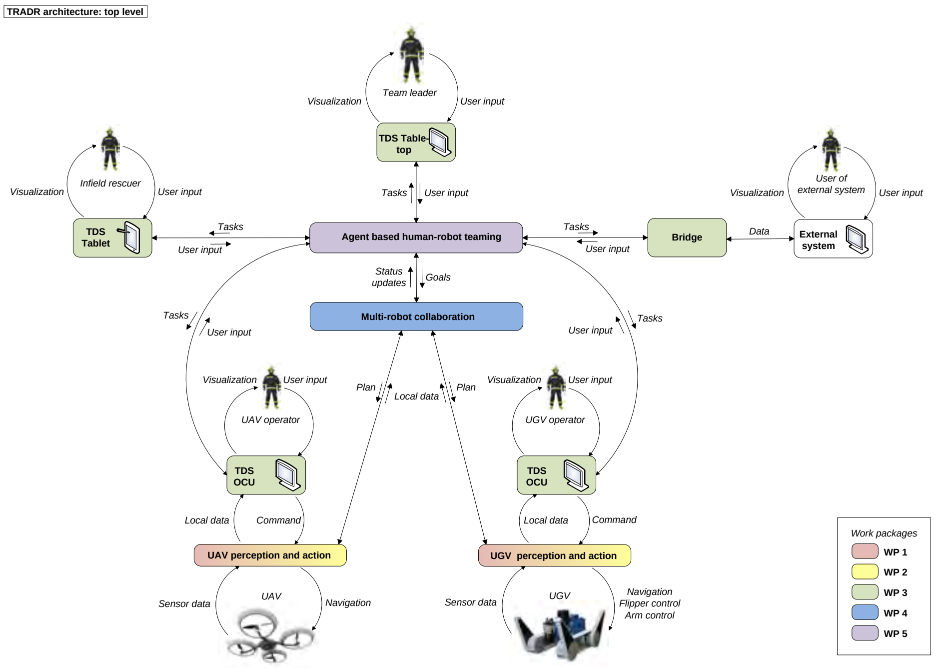


Figure 2: Top-level system architecture

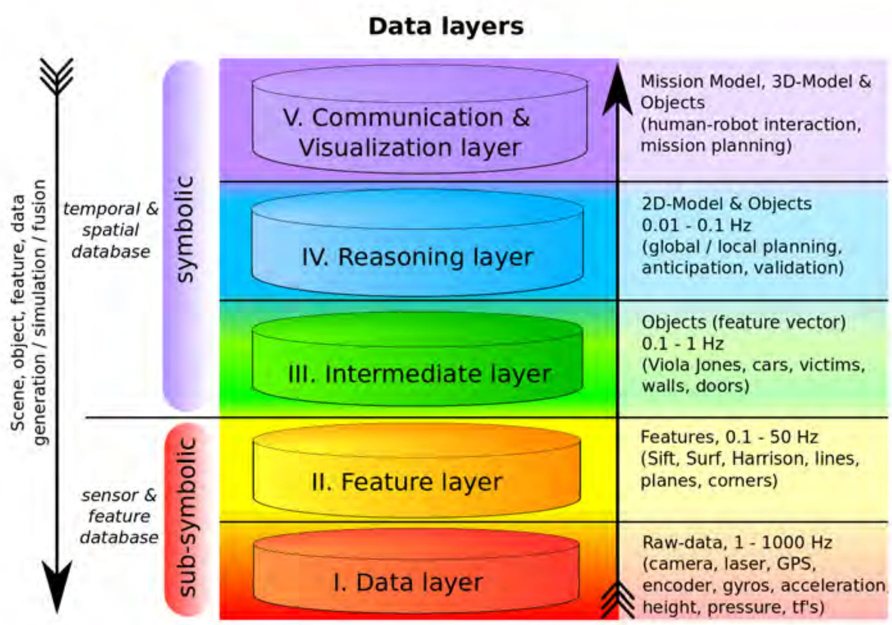


Figure 3: Basic DB-layers

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

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

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

As fully autonomous flights of UAVs are not allowed based on ICAO rules, we prefer the term semi-autonomous. This means that there is always a human pilot in the loop being able to intervene during the operation. Details on the control of UGVs and UAVs are presented in 1.3 of DR2.2.

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

Perception Perception takes place in all different levels of the TRADR system, leading from the sensor data on the robotic level to local data (egocentric map) of a single robot, then to global data (allocentric map) created by merging the local data within one sortie, and finally to a world model (environmental 3D details, traversability, points-of-interest, no-go-areas, positions of robots, etc.) on the mission level by collecting the global data of all sorties. Details on this part are presented in 1.3 of DR.1.2.

Planning Planning is needed on-board of each robot for the autonomous resp. semi-autonomous operation. The goals for this planning can either be given by an operator or by a superior planning component that deals with multi-robot collaboration. The multi-robot collaboration on

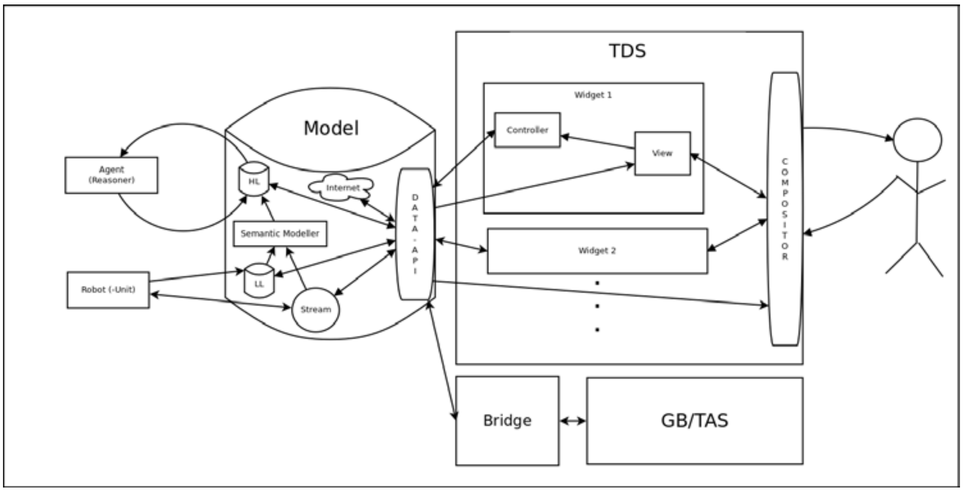


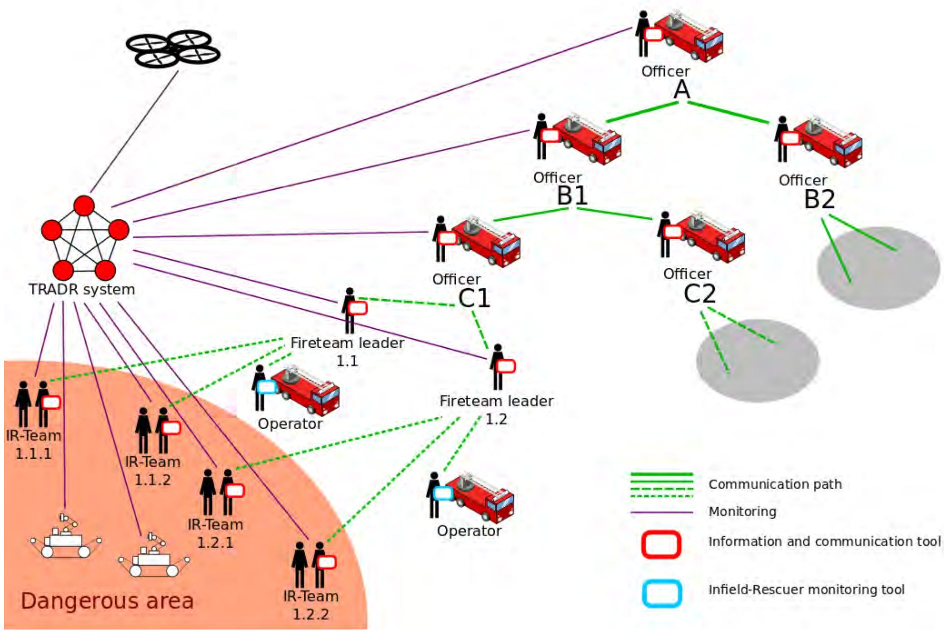
Figure 4: Architecture overview of the TDS

the other hand receives its goals by an agent-based human-robot teaming component at a higher level. Details on this part are presented in 1.3 of DR.3.2, 1.3 of DR.4.2, and 1.3 of DR.5.2.

Action Finally, all (sequences of) actions that are planned by the different planning components must lead to atomic actions, i.e., actions that actually affect the actuators of the robot. Details on this part are presented in 1.3 of DR.2.2.

TRADR Display System (TDS) In general, human team-members use interface devices for the interaction with other team-members incl. the robots, see 1.3 of DR.3.2. In particular, the OCU as the *Human-Robot Interface* offers the capability to survey the robots (UGV or UAV) and their sensoric data. Controlling the robots can either be done by assigning goals to the local planner of a robot or by classical tele-operation. In case of the UAV, there is always an additional pilot in the loop, who keeps the robot in line of sight and is able to intervene. Agents are responsible to accept user commands and to process data to be visualized on display devices (see Figure 4 and 1.3 of DR.5.2).

Communication As in Yr1 and 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 as well as infield rescuers depend on wireless connections with all the inherent drawbacks of WiFi. Apart from that wired LAN connections are used whereever possible. The human team-members are either talking directly to each other using



o

Figure 5: Communication structure

arbitrary communication technology (e.g., walkie-talkie) or their dedicated interface devices with support of Mumble (see Figure 5). 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.

1.3.2 System setup

Neither the software development environment (based on ROS, GIT, Redmine, Jenkins) nor the user-centric development paradigm was changed in comparison to Yr1; the approach was presented at an ICRA workshop on “Robotics & Automation Technologies for Humanitarian Applications: Where we are & Where we can be” [27] (Annex Overview 2.1).

During TJEx and TEval in Y2 the *old* setup with developer laptops as hardware platforms and with diverse software GIT branches was still causing a lot of unreliabilities. One other deficiency was the inconsistent usage of parameters in TRADR: some parameters, e.g., the basic location of the scenario, were solely set on single devices. By this, the position of the basic locations overview map showed inconsistencies amongst the different devices.

The setup of the TRADR system in comparison to Yr1 and during Y2 has changed in several ways: The utilization of consistent hardware makes the setup more reliable. Also the handling of the software versions that are installed on the TRADR system is treated more strictly. Furthermore a concept of the general usage of parameters was created. Thus the TRADR system setup (at the end of Yr2) provides the following characteristics:

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

Now the system is setup by well-defined components, no developer laptops are connected to the system anymore. Another demand was to have a system that is always usable (and presentable). To achieve this several improvements were made:

Hardware
New hardware was purchased and is used solely to set up the unified master system.

Devices
No developer laptops are connected directly to the master system any more.

Software
Software modules are deployed in virtual environments.

Parameters
System wide consistent usage of parameters prevents unforeseen effects on local devices.

Version control
Software changes on developer laptops are merged to the master branch and mirrored to a second, internal git server within the tradr master system.

Branches
No developer branches are used on the master system, only the git master branch.

Deployment
A deployment mechanism is used to propagate changes to the relevant components.

Speech communication
The usage of Mumble as the central speech communication software was enhanced during Yr2.

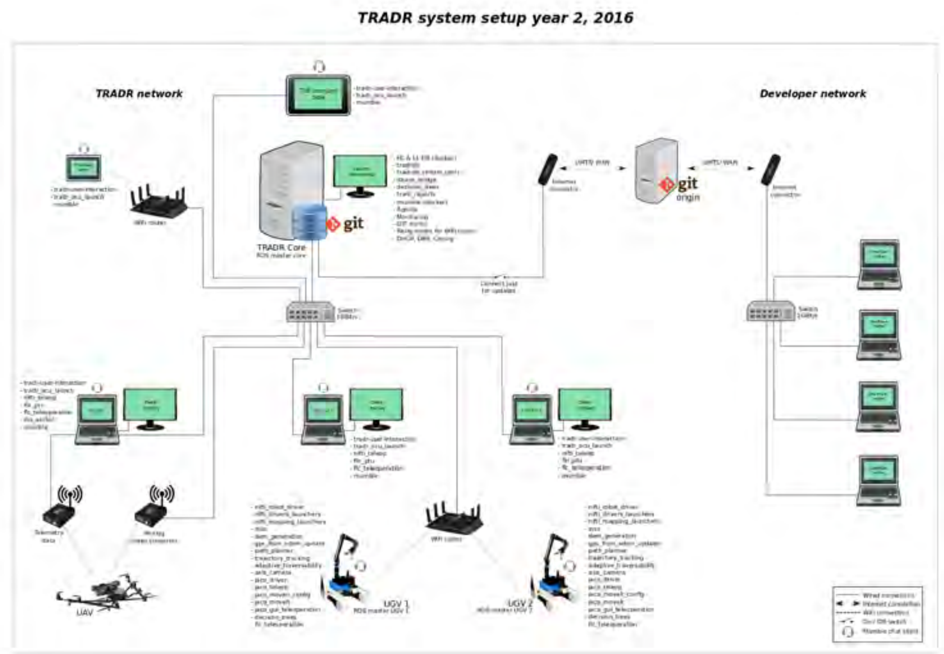


Figure 6: TRADR system setup overview

Monitoring

Central control and monitoring components are used to keep a clear overview of the current system status.

System Overview As in Yr1 the TRADR system contains of a central data storage, robot control and monitoring units as well as the robots themselves, see Figure 6.

The setup strategy was changed to tackle some major drawbacks: Instead of using partner devices with quite undefined software states, we use exclusive hardware running only software from the tested master branch passed through the Jenkins CI process. With this more straightforward procedure we gain a more stable and reliable system. These preconditions support the pretension that the whole system setup should be as easy as being accomplish by the firefighter themselves in a short time span.

To achieve this a high-performance workstation is now used as the main data storage and central reckoning unit. It offers high computational abilities for resource consuming processes like video data procession and map registration. This component is named the *TRADR core system*. Also running on the TRADR core system is an additional *GIT* server [2] which is a mirror of the project GIT server hosted in Prague. Hence, the project source code is reachable also without any internet connection. With this internal source repository an automated deployment of software to all TRADR devices is possible. 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 monitoring tool (1.3.2), and network services like DHCP, DNS and Chrony.

All operator control units are now consistently setup on convertible laptops, each combined with a 24 inch monitor. Hence, the firefighters can utilize a lot of display space and a familiar keyboard interface for data handling.

As for the infield rescuer device it is not finally decided which hardware to use. On one hand the position of the infield rescuer has to be tracked, on the other hand a robust verbal communication should be ensured. For the transmission of the verbal communication three different configurations are possible:

- Usage of a Mumble App via WiFi
- Additional firefighter walkie talkies
- Standard cellphone speech transmission

For the latter two strategies a second device of the same kind is necessary on the receiver side to transfer the language to the TRADR Mumble server running on the TRADR core system.

GUI Framework The partners agreed on using the GUI framework *RQT* [8] and its plugin abilities as a basis for user interface development for the *TDS* (TRADR Display System), refer to DR.3.2 for details. RQT in addition to plain *QT* [9] 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 (Refer to DR.3.2 for details). 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.

Role definitions Which content is visible and which functionality is offered by the TDS depends on the role of the signed in user. The following roles are defined within TRADR (see Figure 7):

Mission commander
Coordinates the collaboration of teams during a mission

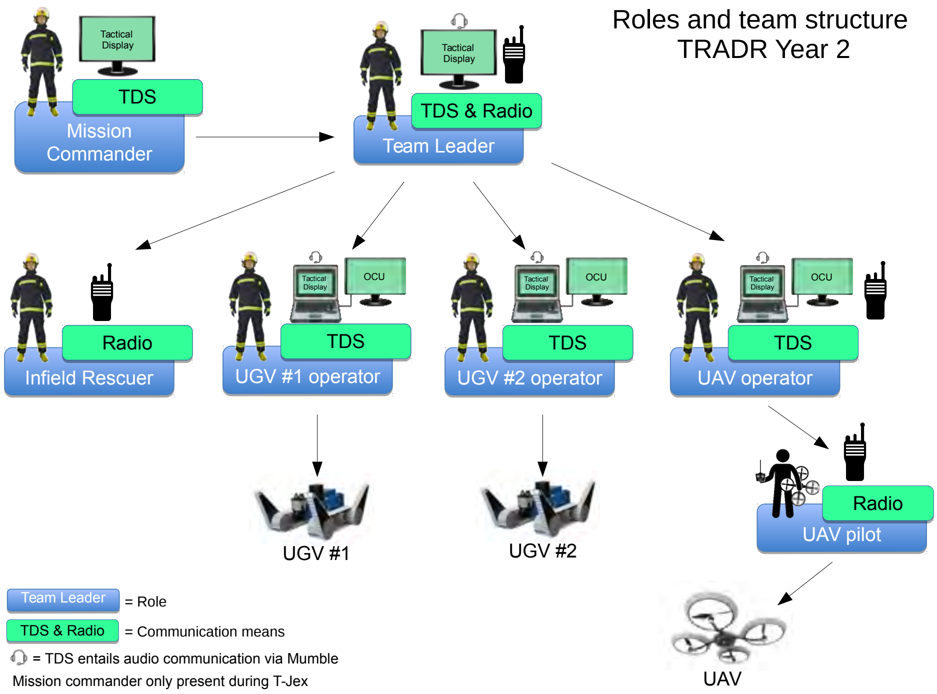


Figure 7: The TRADR roles and team structure in Y2

Team leader
Leads a team of actors and robots, uses the TDS for shared situation awareness

Infield rescuer
Provides additional information (images, descriptions, positions) from within the scenario area

Robot operator
Controls and monitors a robot (UGV or UAV) and selects the functionalities to be used, preselects details from the provided robot sensor data

The role of the UAV pilot is more transparent in comparison of the other roles as the pilot is just a human in the loop who is responsible for keeping the UAV in line of sight and to execute commands from the UAV operator.

Mumble communication tool In Yr1 there was only one channel used for the whole communication within TRADR and all participating users shared the same level of talking to everybody and hearing everything.

Now Mumble [10] 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.

As Mumble offers a wide range of connection configurations, e.g.,

a) one to one *b*) one to many *c*) many to one *d*) many to many and is able to select receivers with a key stroke, the capabilities can be adapted smoothly into the needs of the end-users. The communication channels additionally can be processed on-the-fly, e.g., by speech recognition (see Section 1.3.5 of Deliverable DR.3.2), or they can be recorded on harddisk for later analysis.

Reporting The reporting is also a new component in TRADR. It is able to report on arbitrary messages in ROS. Events can be filtered and the tool can report lively or in hindsight of a scenario. With a built-in Mumble connector and a speech recognition component it is able to transfer spoken language into textual language that can be shown on the reporting interface. By this an easier handover of collected information from one team to a subsequent team is possible. More details are presented in 1.3 of DR.5.2.

Controlling and monitoring As TRADR consists of many different connected components with many possible states, a proper monitoring of the current conditions is indispensable. To accomplish this, different tools were tested concerning network monitoring, component supervision and ROS surveillance.

TRADR status monitor The *TRADR status monitor* (see Figure 8) is a TRADR internal tool showing the status of involved hosts and robots. The data is taken from the high-level database.

rqt_robot_monitor The *rqt_robot_monitor* tool (see Figure 9) is a ROS related tool displaying the contents of the common *diagnostics_agg* topic. The utilization of this topic can be defined on the robots.

This tool is able to show important status information and status changes on robots. A time line at the bottom of the tool provides a dynamic retrospect status overview. The tool is only usable to visualize information one way, no feedback can be sent to the robot. This tool is written in RQT and thus is integrable into the TDS.

rosmon The *rosmon* bash tool (see Figure 10) can be used to bidirectionally handle ROS processes. Using this tool to start ROS components remotely provides the advantage that the current status of ROS processes

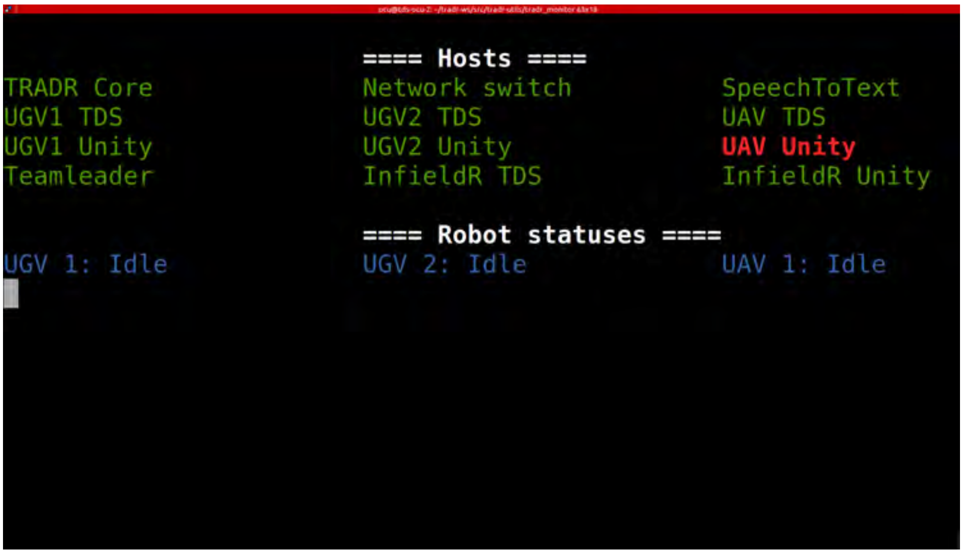


Figure 8: The TRADR monitor tool

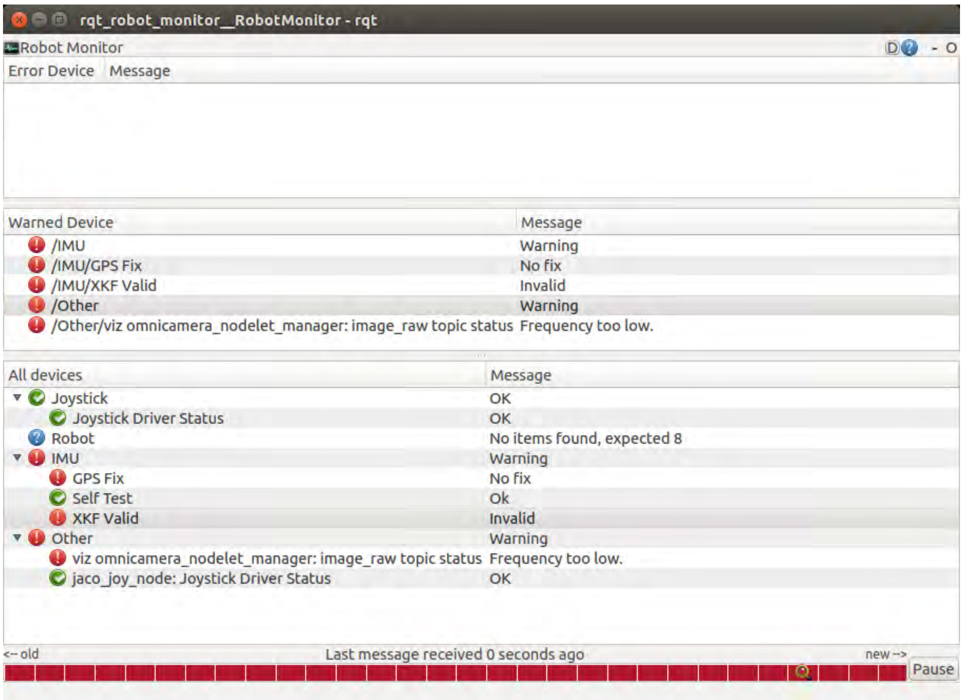


Figure 9: The rqt_robot_monitor tool

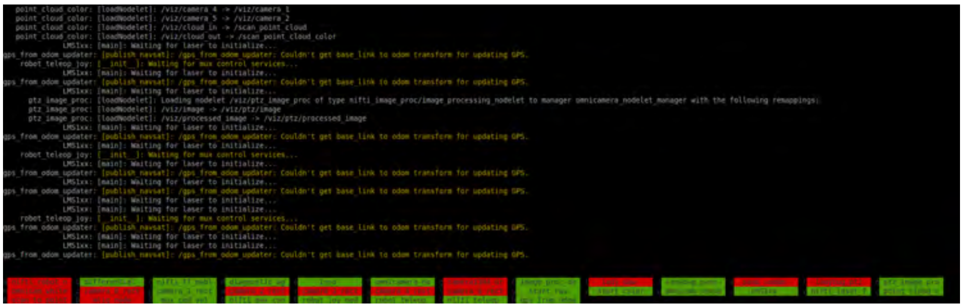


Figure 10: The rosmon tool

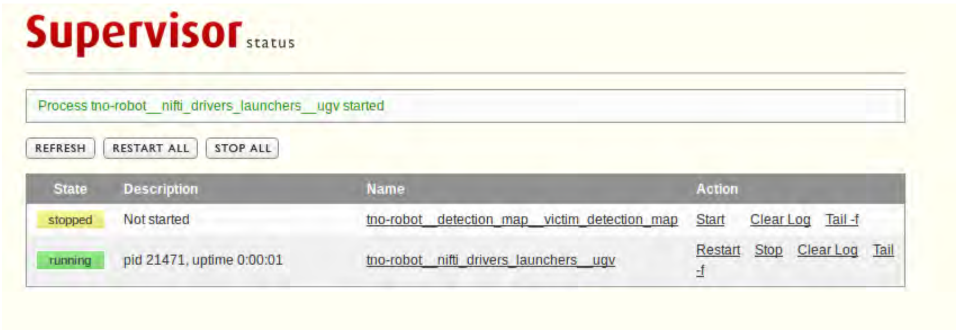


Figure 11: The supervisor tool

can be monitored and also be changed by keyboard shortcuts. Nevertheless this tool is not very user friendly to non-IT experienced end-users.

supervisor The *supervisor* tool (see Figure 11) offers a client-server architecture to control arbitrary processes on remote machines. For TRADR, the server has to be installed on all machines/robots that have to be controlled remotely from within a local browser.

On the remote machines any process can be fine-grained adjusted by the supervisor configuration scripts. The supervisor web application shows a basic status of included processes and offers basic control like start/restart/stop. Also the remote log file of the belonging process can be seen in the local browser which can simplify troubleshooting a lot.

With the *WebKit widgets* [11] this tool is also integrable into the RQT framework.

The TRADR status monitor summarizes different status data in a condensed view. This view is useful for end-users. The rqt_robot_monitor is meant to give information to technical staff maintaining the TRADR system - the information is too detailed for end-users. Rosmon and Supervisor are quite similar control tools: the latter one can also be used by skilled end-users to control robot components, e.g., switch on victim detection.

1.3.3 Database

The TRADR database which is installed on the *TRADR core system* is divided into two parts:

Low level database
The low level database, responsible for the storage of binary data, is realized by an instance of MongoDB

High level database
The high level database which provides reasoning functionality on RDF data is realized by an instance of Stardog. A more detailed report about the high-level database can be found in [17] (Annex Overview 2.3).

The high level database keeps references to binary data in the low level database. On startup both databases are started together and provide their own native API. Above these APIs an additional TRADR model API is provided to simplify TRADR specific database queries and operations. A TRADR model API library has to be integrated on each device using database functionality.

1.3.4 Network communication

The basic network topology was not changed in Yr2. All wired devices are connected via a 1 GBit/s LAN switch. However, a lot of tests were made with different devices and frequencies. At the TJEx we additionally tested military WiFi devices. They turned out to be able to build up an ad hoc mesh network for the price of halved bandwidth. Finally, an up-to-date WiFi 802.11ac triband router was purchased and is just about to be tested. Also new 802.11ac adapters for the robots are planned to complete the usage of this advanced WiFi technology including optional link aggregation and beam forming.

To analyze, which allocation of WiFi channels fits best for multiple WiFi devices, the built-in tool of the Ubiquity bullets was used at TJEx and TEval in Yr2. It offers a live data throughput overview and a live scan of all available and occupied frequencies for the fine-tuning of connection parameters, see Figure 12.

The network services like *DHCP*, *DNS*, and *NTP* running on the yellow boxes were shifted to the TRADR core system.

Network monitoring Tools like iftop and etherape were used to monitor the data throughput especially from the WiFi devices, see Figure 13 and 14. With these tools the network throughput on a selected network interface can be surveyed to detect potential transfer bottlenecks.

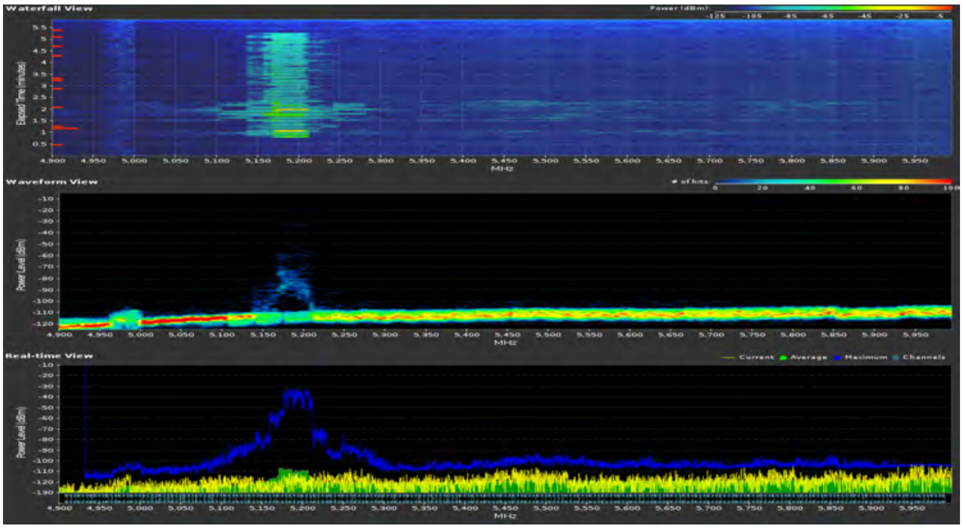


Figure 12: WiFi analysis tool

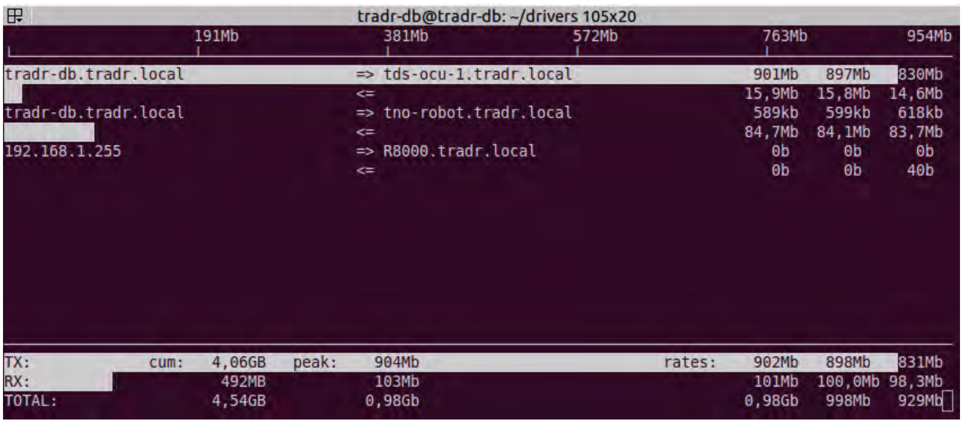


Figure 13: Monitoring network throughput with the iftop tool

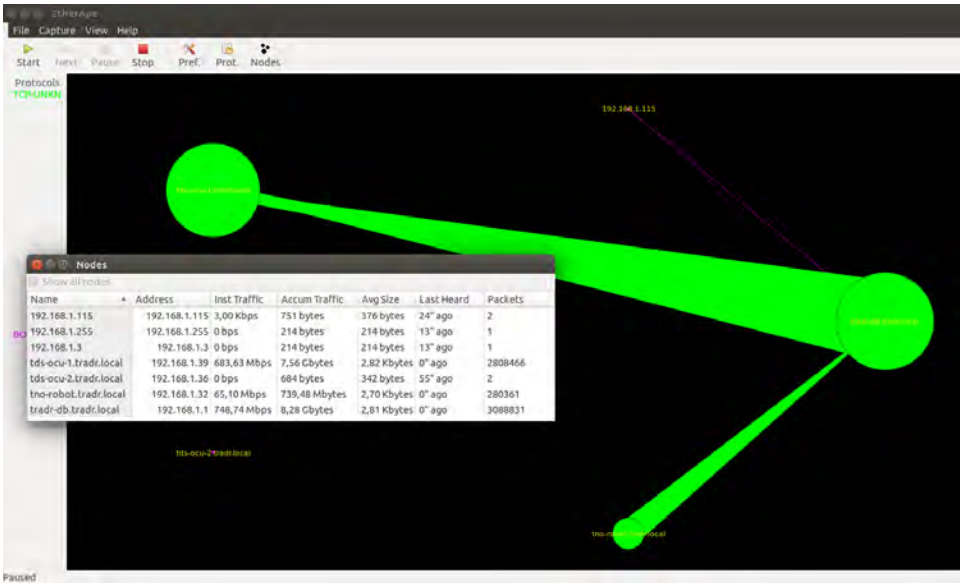


Figure 14: Monitoring network participants with the etherape tool

Network controlling 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 (see Annex 2.2 of DR.7.2).

In future scenarios, also tools like *Netem* [7] should be utilized to simulate adverse network conditions in a more fine-grained manner. To cope with temporal bad WiFi conditions during video transmission also the *ROS* built-in abilities of instant image resolution adjustments were used during TEval. The easy handling of this resolution adjustments via GUI sliders showed a good approach and was also usable by the firefighters.

1.3.5 Communication infrastructure for multi robot setups

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.

The robots need a so-called 'roscore' to be able to operate. The roscore manages the different channels (topics) on which messages can be send. In the setup of Yr1 we connected the OCUs directly to the roscore of the robot. This has several disadvantages, most notably the limitation of one robot per OCU, that communication between computers in the base camp could only be achieved via the database and that the standard ROS message passing system does not work well with unreliable, high-latency, low-bandwith networks.

Figure 15: UGV variants. Left: Additional pan-tilt-zoom camera mounted. Right: Kinova JACO arm mounted.

In order to improve this setup we decided to keep the idea of one roscore per robot, but also add a roscore for the base camp. The different self-contained systems can then explicitly send messages to other actors by using relay nodes.

At the beginning of Yr2, the utilization of *Rocon* [12] was planned as a ROS multi-master framework. It turned out that *Rocon* is oversized for TRADR with a vast amount of configurations for each WiFi topic and necessary adjustments throughout the existing source code. Also the abilities of *Rocon* to cope with WiFi connection losses was insufficient.

Finally, the relay nodes were implemented by the 'nimbro_network'[13] stack. In contrast to earlier approaches this provides metrics about the status and quality of the network link: 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.

1.3.6 UGV improvement

In Yr2, the upgraded UGVs were fully usable. Different deployment variants can be set up depending on the specific application planned during the mission. Examples are shown in Figure 15.

End-user evaluations motivated several UGV upgrades. We have recently upgraded flippers of one UGV, Figure 16. The flippers shall allow force sensing which increases usability in sensory deprived environments, think about dense smoke. See DR.1.2 for more detailed descriptions of the related algorithms. We are also experimenting with a ultrasonic sensor which should help to see through dense smoke. A simple smoke detector is also under development.

Figure 16: Upgrading flippers. Left: force sensors integrated into a thin metal construction. Right: Upgraded flippers integrates necessary electronics including a radio module.

The servomechanism which operates the lidar often stopped working during the demanding experiments. Restarting it had required rebooting the whole system including the CPU which was very time consuming and disturbing. We designed a hardware upgrade that allows a warm/soft reset of the servo.

For the network communication an up-to-date 802.11ac router is now used for the WiFi connection. The utilization of the arm was much improved in Yr2, see Figure 17 and DR.2.2. It is possible to control the arm via an *RQT* plugin, grasp and collect objects semi-autonomous and use the hand mounted camera as an enhancement in terms of a better situation overview. To control and monitor single software modules, e.g., victim detection or adaptive traversability, separated start scripts can be used according to the use-case to be fulfilled, see 1.3.2.

1.3.7 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 MAVs during a mission, integration of information over time as well as fusion and integration of different sensors is needed. In Yr2, the work on these topics was continued and presented at the TRADR booth during IROS [48] (Annex Overview 2.2).

Two different UAV platforms were used: the AscTec Pelican [14] as a research platform and the AscTec Falcon 8 [15] as the platform to be used for the exercises and evaluations. The challenge in TRADR is to combine the researchers' needs with the end-users' demands. The Pelican is a platform to be used, if experiments on navigation, planning and obstacle avoidance are made. The team can add sensors and the onboard computer can maintain CPUs up to Intel Core i7. In contrast, the Falcon is preferred by the end users – it can provide stabilized images and provides many comfortable features matured over the years based on real experience.

For the integration of the UAV into the TRADR system two different

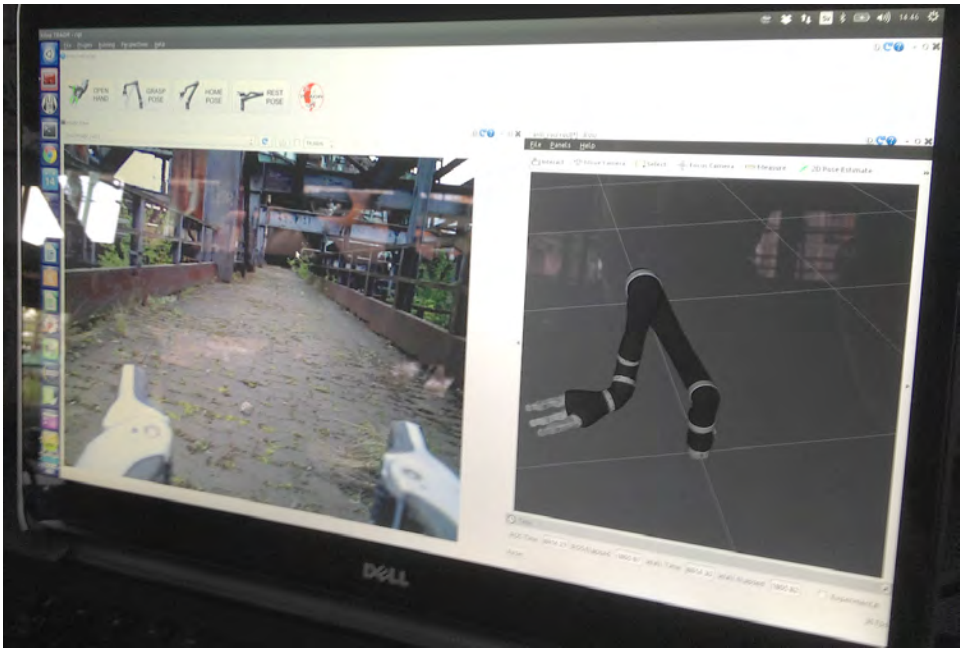


Figure 17: The arm control GUI

Figure 18: Falcon 8 by Ascending Technologies. Left: On ground. Right: Exploring the Phoenix site.

kinds of transmission channels are used:

- Analog transmission of the live video data
- Digital transmission of telemetry data

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 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 temporary additional laptop from *Ascending Technologies*. On this laptop the GPS data is published via *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 this laptop. At a later state of development all of this functionalities should be integrated into the OCU as an *RQT* plugin.

According to the UAV roadmap, during Yr2 the Pelican was replaced in TRADR. This led to the development of the Ascending Neo, which is now available as a prototype within the TRADR consortium. In a next step, the goal is to bring both platforms (Falcon 8 and Neo) together so that newly developed algorithms can be evaluated concurrently with providing the comfort functions of the end-user platforms. This will lead to a first version of a TRADR specific UAV in Yr3 and a final version in Yr4.

1.3.8 Datasets for Benchmarking

Due to the high multimodality of TRADR robots but also because of the realistic and challenging task environment, data generated in TRADR exercises can provide an interesting and unique opportunity for benchmarking of mobile robotic algorithms. So far, diverse suitable sensor data were recorded during TJEx and TEval missions (e.g., see Figure 19). These data are all available as ROS bags.

To obtain an idea of how to publish benchmark datasets, well-known dataset repositories in mobile robotics domains were reviewed. The common form of dataset publication is building up websites providing dataset downloads and describing them. Some research groups present outstanding features on the download sites. Mentionable examples are online tools for automatic evaluation of algorithms, categorization of datasets in different

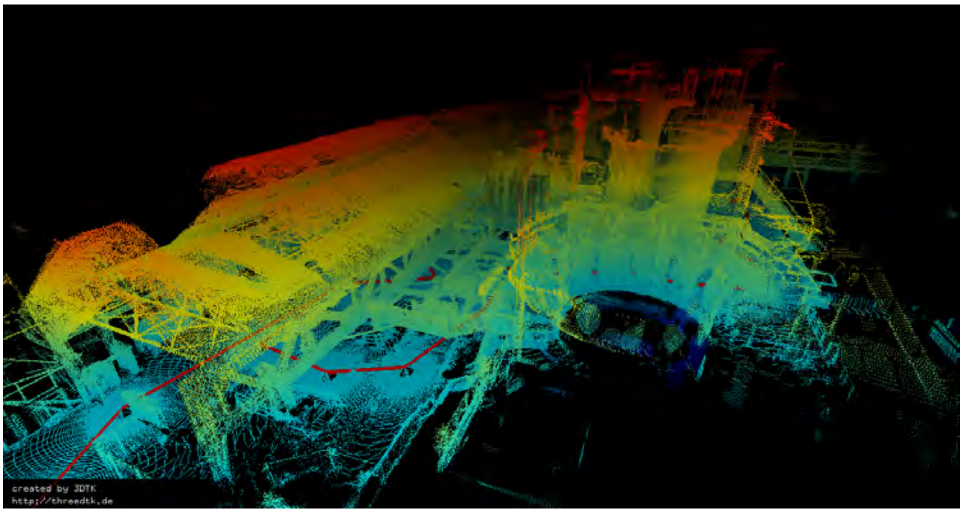


Figure 19: 3D point cloud of the Phoenix site based on data from TJEX

difficulty levels, a change log dealing with news and issues, a table to publish and compare own benchmark results with others, and the possibility to convert the table of results to LaTeX code. More detailed description of all researched websites can be found in the attached document [31] (Annex Overview 2.4).

To be user-friendly for the research community, the datasets should be provided in a well-structured easy-to-maintain website. It can contain rubrics like overview, hardware description, operation environment description, dataset description, and board for publishing results. Sensor data are to be categorized by task type e.g. 3D scan registration, SLAM, object detection, etc.. The full description of the concept as well as a list of required preparation works is also to be found in the attachment [31] (Annex Overview 2.4). A goal for Y3 is to decide in TRADR how to go about publishing the collected data.

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 [38], [36], [40], [43], [35], [37], [32], [25], [33]. 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 [22], [21], [39], [19], [42]. 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 [16].

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 [36], [45]. 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 [42], [50], [29], [41], [49], [51].

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 [18], [19] and has several advantages. In comparison to a raw video/sensor display and joystick-like control system like those presented in [21], [39], [32], 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 [44], [47].

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 [38], [36], [46]. 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 [24] and DARIUS [23] target the development of robotic tools that can assist during disaster response operations, focusing on autonomy. SHERPA [34] 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 [20], 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 [16], 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 [26], 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://www.persistentsystems.com/man-portable-unit-gen4.

[7] http://www.linuxfoundation.org/collaborate/workgroups/networking/netem.

[8] http://wiki.ros.org/rqt/.

[9] http://www.qt.io/.

[10] http://www.mumble.info.

[11] http://doc.qt.io/qt-5/qtwebkit-index.html.

[12] http://wiki.ros.org/rocon.

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

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

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

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

[17] Timea Bagosi. Tradr high-level database, year 2. Technical report, Delft University of Technology, Interactive Intelligence Group, 2015.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] Mario Gianni, Ivana Kruijff-Korbayová, Rainer Worst, and the NIFTi Project Team. Human-robot teaming in disaster response – a user-centric approach –. In *Proceedings of the ICRA 2015 Workshop on Robotics & Automation Technologies for Humanitarian Applications: Where we are & Where we can be*, Seattle, Washington, May 2015.

[28] Walter Kasper. Mission Monitoring and Reports for Shared Situation Awareness. In Walter Kasper, editor, *Expectation Management in Common Ground*. TRADR Deliverable DR5.2, jan 2016. Unpublished.

[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] Dong-Uck Kong. TRADR dataset dissemination strategy. Unpublished Technical Report.

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

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

[47] 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.

[48] Hartmut Surmann and Rainer Worst. How can MAVs assist human-robot teams in disaster response over multiple sorties? Poster from TRADR booth at IROS 2015.

[49] 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.

[50] 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.

[51] 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 Gianni et al. (2015), "Human-robot teaming in disaster response – a user-centric approach –."

Bibliography Mario Gianni, Ivana Kruijff-Korbayová, Rainer Worst and the NIFTi Project Team. "Human-robot teaming in disaster response – a user-centric approach –." Invited talk at ICRA 2015 RATHA Workshop

Abstract In the last decades, several research efforts have been made to develop Human Robotic Systems (HRSs) for supporting people in aerial, terrestrial and maritime Search and Rescue (SAR) operations, in real-world hostile environments like earthquakes, volcano eruptions or avalanches. HRSs have further been developed for industrial, domestic and entertainment domain applications. Unlike autonomous systems, HRSs are primarily designed to support people and robots working together in teams in close and continuous human-robot interaction. This interaction requires that the robotic technologies meet the stakeholders needs in order to develop HRSs which are really effective in amplifying human productivity as well as in reducing mission risks. These requirements can be achieved only if end-users are tightly involved in all the phases of the developmental process of the HRS, and each of these phases provides a close coupling to real-life experimentation.

In this presentation, the HRS developed under the EU FP7 ICT project NIFTi is exposed. The core of the design methodology is a user-centric approach. The user-centric methodology is a general approach for both iteratively and incrementally developing HRSs. Its cyclical nature provides for an end-users involvement, which gradually increases in the development process. Moreover, its incremental nature is such that the complexities of both use cases and scenario evaluations increases, at each cycle, in order to meet project objectives, user requirements and technology. The presentation aims to give a brief technical overview of the complex system, which has been developed employing this methodology, also discussing the benefits of this approach in particular with respect to situation awareness and human-robot collaboration.

Relation to WP This paper describes the design approach developed in the project NIFTi that is also used in TRADR, e.g., in T6.3.

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

2.2 Surmann, Worst (2015), "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 booth at IROS 2015.

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 T6.3.

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

2.3 Bagosi, Timea (2015), "TRADR High-level database"

Bibliography Bagosi, Timea. "TRADR High-level database, Year 2" Unpublished Technical Report, 2015.

Abstract In this report we describe the high-level database for TRADR Year 2. The evolution of the web reached its semantic era, data on the semantic web is organized according to graph-like ontologies, and stored in triple store repositories. A short overview of the existing triple store technologies and their benefits is provided. A more detailed description shows the technology chosen and used for the TRADR project.

Relation to WP This report discusses the existing semantic web triple store technologies for the TRADR high-level database, hence it is part of the database and high-level (team) data integration of WP6.

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

2.4 Kong (2015), "TRADR Dataset Dissemination Strategy"

Bibliography Dong-Uck Kong. "TRADR Dataset Dissemination Strategy" Unpublished Technical Report, 2015.

Abstract Benchmarking is an objective to quantify and compare the performance of own processes to the others. Often in that way, measuring the success becomes possible in the research community. A lot of robotics research groups over the world publish the recorded sensor data from their experiments and provide benchmarks. One of the main goals of TRADR-project is to achieve the best possible modeling and understanding of the disaster scene. To enable this by comprehensive environment perception, multiple types of sensors on multiple robot platforms are used at the same time. Due to the multimodality and the realistic and challenging experiment environment, data generated in TRADR exercises can provide an interesting and unique opportunity for benchmarking of mobile robotic algorithms.

Relation to WP This report presents examples of available datasets for benchmarking in mobile robotics and provides a suggestion how to disseminate the datasets collected in TRADR, e.g., in the context of T6.3.

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

Human-robot teaming in disaster response – a user-centric approach –

Mario Gianni*, Ivana Kruijff-Korbayová‡, Rainer Worst§ and the NIFTi Project Team
*ALCOR Laboratory, DIAG A. Ruberti, Sapienza University of Rome, Italy
‡Language Technology Lab, DFKI, Saarbrücken, Germany
§Fraunhofer IAIS, Sankt Augustin, Germany

I. Introduction

"Whereas early research on teamwork focused mainly on interaction within groups of autonomous agents or robots, there is a growing interest in better accounting for the human dimension. Unlike autonomous systems designed primarily to take humans out of the loop, the future lies in supporting people, agents, and robots working together in teams in close and continuous human-robot interaction" [1], [2].

We have been pursuing this goal in the EU FP7 ICT project NIFTi [3].[1] The core of our design methodology is a user-centric approach [4], [5]. We have applied this methodology to design and develop a complex system supporting human-robot collaboration in search and rescue (SAR) scenarios. Below we first describe the components of the user-centric development cycle, discussing the benefits of this approach in particular with respect to situation awareness and human-robot collaboration. Then, we summarize its application in NIFTi and the lessons we learned along the way about human-robot teaming for SAR. To complete the picture we give a brief technical overview of the complex system we developed employing this methodology.

II. The user-centric design methodology in Humanitarian applications

The user-centric methodology is a general approach for both iteratively and incrementally developing Human Robotic Systems (HRSs). The term *Human Robotic* refers to advanced robotic technologies to amplify human productivity and reduce mission risk by improving the effectiveness of human-robot teams. The methodology comprises four main phases: (1) end-user insights and requirements analysis; (2) definition, development and benchmarking of the component technologies of the system; (3) system-level integration and experimentation and, finally, (4) end-user system evaluation (see Figure 1). A preliminary phase precedes the begin of the iterative cycle. This phase defines the humanitarian domain application, the objectives, the milestones for achieving the objectives, the real-life scenarios for both system experimentation and end-user evaluation and, finally the roadmap. The milestones focus on questions which arise from operationalization of human-robot cooperation. The operationalization identifies a naturalness loop to achieve a balance between operational and cooperational demands. The roadmap specifies the use cases to test specific hypotheses. Use cases gradually increase

[1]NIFTi was funded within the EU FP7 ICT programme, Jan 2010 – Dec 2013, grant No. 247870.

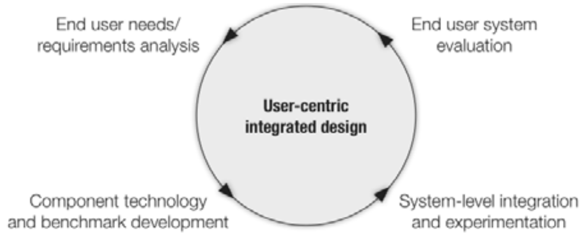


Fig. 1: Iterative and incremental cycle of the user-centric design methodology.

both task and scenario complexity. Each use case investigates forms of mixed-initiative human-robot cooperation, in the humanitarian domain application. At each development cycle, the first phase identifies the human factors of the HRS, for both human-like interaction and collaboration. On the basis of the hypotheses, formulated according to the analysis of the human factors, all the required components of the HRS prototype are developed and integrated. Then, in the fourth phase, the claims are systematically tested through cognitive walkthroughs with human experts. This evaluation highlights the discrepancies between the claims and the capabilities of the HRS prototype. It also provides further insight into the user needs, which will be further addressed in the subsequent development process. In this methodology, humans are involved in every phase of the cycle. Every phase provides a close coupling to real-life experimentation. At the end of each cycle, the evaluation results feed back into research, thus possibly adapting the roadmap.

The user-centric methodology allows researchers to study the human behaviors in different environmental circumstances,to determine what humans pay attention to and how their cognitive task load can vary, depending by the environment model presented by the HRS. The close cooperation with end-users also provides researchers with important insights regarding *what* and *how* information about the environment has to be communicated to the users. This prevents information overload as well as filters out irrelevant information.

The methodology results to be very effective to validate the potential, the usability, the effectiveness and the reliability of the HRS. During the first phase of the cycle, the end-users specify, on the basis of both their expertise and the complexity of the in-field scenario, which capabilities the HRS should provide them as operational support. The end-users also collaborate with researchers in order to specify both the use cases and the scenario evaluation. After both the development

(a) Firefighter's training area, Montelibretti, Italy

(b) Firefighter's rescue training area, Prato, Italy

(c) Mirandola, Northern Italy, after the earthquake in May 2012

(d) Ex-American collapsed hospital, Calambrone, Italy

Fig. 2: Case studies

and integration phases, the end-users test the usability of the HRS, as well as validate its effectiveness in the in-field scenarios. At the end of this phase of the cycle, feedback from both developers and end-users is used to update and refine the specification requirements, thus closing the loop. Finally, the process foresees collecting additional end-user insights and extending the functionalities of the prototypes to fulfil both the claims and the prerequisites. The cyclic nature of the methodology intrinsically tends to increase the degree of trustworthiness of the HRS.

III. Case studies

We applied the user-centric methodology to develop the HRS, under the EU FP7 ICT project NIFTi [3]. NIFTi adopted the goal to bring the human factor into cognitive architectures while developing robots capable of collaborating with human team members under the complex outdoor circumstances of a disaster response. The rescue organizations included as partners in the NIFTi consortium enabled close involvement of end-users, throughout the entire R&D cycle. They provided input to system specifications, participated in yearly exercises and evaluations, and provided feedback for further iteration cycles of the development process. NIFTi organised its R&D around a sequence of scenarios that gradually increased in complexity, including operational context complexity and collaborative context complexity, such as team size, its composition and geographical distribution. The scenarios were designed in close cooperation between developers and the USAR teams from the end-user organizations. This was to ensure the scenarios would achieve a balance between practical relevance and feasibility, and necessary scientific progress. The robots used in the HRS were an Unmanned Ground Vehicle (UGV) and an Unmanned Aerial Vehicle

Fig. 3: Robotic platforms of the HRS.

(UAV) (see Figure 3). We started the development process facing the complexity of a flat, largely 2-dimensional terrain of a tunnel accident scenario (see Figure 2(a)). The process gradually increased the complexity of the scenario up to a semi-unstructured debris-strewn environment of an earthquake disaster (see Figure 2(b)). The increase in complexity led to the need to develop increasingly more observational capabilities (from 2D to 3D), and to progressively increase the degrees of autonomy (3D path planning, adaptive morphology). The increase in complexity also led to significant changes within the organizational structure of the human-robot collaboration. In fact, over the cycle, this structure has become more realistic. Initially, the organizational structure was non-existent. In the tunnel accident scenario, a single remotely located operator teleoperated the UGV to create a 2D-map populated with car objects, recognised by the robot. In this scenario, we focused on bringing together the various components of the robot functionalities (control, mapping, vision) with a basic, end-user oriented graphical user interface for teleoperation. We also faced highly familiar problems such as network and robot hardware issues (e.g., low bandwidth, bad connection, short battery operating time), as well as a wide range of human factor issues. At this stage of the iterative development cycle, the HRS was very far from being used in real-life applications. The results obtained during the in-field evaluation suggested to move from the *single robot, single operator* setup to a full-scale human-robot team, and to get a better grip on technology. Therefore, we extended the HRS with a larger human team operating from a remote command post. We also studied more difficult operational conditions, such as the presence of smoke, flickering light and debris. Moreover, we added the UAV microcopter to the HRS. Finally, we moved to a *human-robot team* setup. The human team members took on various roles, such as a Mission Commander, UGV/UAV Operator and/or Mission Specialist. Robot control, vision, and mapping had significantly improved to move towards building up a robot-centric 3D understanding of the environment it was operating in. Access to this robot-centric situation awareness was provided to human team members through an integrated user interface setup, facilitating multiple operational views and tactical views. After the experimental evaluation of this setup, we extended the HRS by adding an in-field rescuer to the team, increasing team's size and geographical distribution. This improvement raised the issue of building up and maintaining distributed situation awareness. In July 2012, we deployed the HRS in the red-area of the ancient city of Mirandola, in Northern Italy, hit by an earthquake in May 2012, to support the Italian National Fire Corps during damage assessment of historical buildings and cultural artifacts, located in the two main churches of the city center (see Figure 2(c)) [6]. The crucial insight from this deployment was the need for integrated persistent situation awareness. Multiple robots need to be sent into the area, together or one after another. Different kinds of robots play complementary roles in this process. They need to build integrated persistent situation awareness gradually over multiple sorties, to allow the team to coordinate its efforts, and learn to best execute its tasks. After this in-field experience we moved on the roadmap to the earthquake scenario with multiple levels to explore. Continuous team coordination and communication was crucial for an adequate disaster response. The experiments with this multi-human, multi-robot team were run at two sites of the Italian firebrigade: a USAR training area in Prato (see Figure 2(b)) and an abandoned, partly destroyed hospital, near Pisa, in Italy (see Figure 2(d)).

IV. Results

The HRS builds up a robot-centric situation awareness of the environment, from raw data coming from the different robot sensors. This situation interpretation is based on a 3D metrical mapping of the environment [7]. In order to bridge the gap between robot-centric and human-centric situation awareness, this representation is extended with visual perception [8], [9], unsupervised and user-driven topological decomposition [10], [11], functional mapping [12], point cloud categorization, based on segmentation [13] and traversability analysis [14]. The HRS deploys on top of these representations various levels of reasoning and autonomous planning. The lower levels include morphological adaptation [15], trajectory planning and tracking control [16], for complex terrain traversal tasks, and three different strategies for 3D path planning, based on the 3D map, the segmented map and the traversability map representations of the environment, respectively. The HRS chooses among these three different strategies, basing the choice on the terrain surface, topology and possible sources of planning failures. All these low-level autonomous functionalities are managed by a high-level cognitive control. The control coordinates the interventions of the human operator and the low level robot activities, under a mixed-initiative planning perspective. Indeed, it implements several hybrid operative functionalities lying between autonomous and teleoperated modes, available during the execution of a task. The human operator can manually control some functional activities of the robot, for example, the control of the motion to explore an interesting location or escape from difficult environments, by suspending the robot's autonomous navigation. The operator can also modify the control sequence by skipping some tasks or inserting new operations. The control integrates a model of task switching [17]. This model has been learned from observations of human operators identifying stimuli, selecting the best task choices or inhibiting inappropriate urges, while controlling the robot, embodied in a rescue-like scenario. The main advantage of such a model is to allow the system to suitably manage, for example, possible breakdowns of the network communication. In such a case, the system flexibly decides either to focus on the task at hand or to switch to the recovery WiFi connection task.

The HRS provides a multi-view user interface to facilitate different views on information in the human-robot team, to help support different roles in the team. The interface provides multiple modes of communication, including touch and spoken dialogue. Views include the visualization of information from the various robots (UGV, UAV), and team situation awareness. An operator control unit (OCU) is implemented to support human-guided exploration tasks as well as to facilitate communication [18]. Beyond the remote command, the system extends the human-robot team setup to include an in-field human rescuer. This additional team member is endowed with an interface to both store and present, in an asynchronous way, geo-referenced information at the operational and tactical levels of communication in the human-robot team (see Figure 4).

The HRS builds up and maintains knowledge on the users, supporting them to stay in a continual work flow, by attuning the information processing and sharing to the task at hand, in order to support the team effort. This team support functionality is based on cognitive task load and emotional state models. These models have been designed on the basis of the huge amount of data about user's behavior, gathered in our end-user experimentations. The system makes use of these models for dynamic task allocation and adaptive dialogues. The HRS further integrates computational visual attention models for top-down search tasks. These models have been learned from data collected with the Gaze Machine, a wearable device gathering and conveying visual and audio input from end-users while executing a task [19].

The HRS uses the ROS framework [20] as main middleware for communicating information between all the components of the system. This framework is a good choice for running processes on the robots. It allows message-passing

(a) Geo-referenced information

(b) Mobile display

Fig. 4: Content Display

between the robots and the control center over WiFi. The ROS stack is connected to a working-memory-based middleware, developed in CAST [21]. The CAST middleware is mainly responsible of both the human-robot communication and the shared situation awareness within a mixed human-robot team.

V. Ongoing work

We are currently adopting this proven-in-practice user-centric design methodology in the context of the EU FP7 ICT project TRADR [22]. [2] Our in-field experience has highlighted that any incident serious enough to require the deployment of a HRS most likely involves a sequence of sorties over several hours, days and even months. Therefore, building on the research and experience of the NIFTi project, TRADR's aim is to develop a HRS which enables the human-robot team to gradually develop its understanding of the disaster area over multiple synchronous and asynchronous sorties (persistent environment models), to improve team members' understanding of how to work in the area (persistent single- and multi-robot action models), and to improve team-work (persistent human-robot teaming) [23].

References

[1] M. Sierhuis *et al.*, "Human-agent teamwork and adjustable autonomy in practice," in *Proceedings of the International Symposium on AI, Robotics and Automation in Space*, 2003.

[2] M. Johnson, J. Bradshaw, P. Feltovich, R. Hoffman, C. Jonker, B. van Riemsdijk, and M. Sierhuis, "Beyond cooperative robotics: The central role of interdependence in coactive design," *IEEE Intelligent Systems*, pp. 81–88, May/June 2011.

[3] "NIFTi (Natural Human-robot Cooperation in Dynamic Environments)," 2013. [Online]. Available: http://www.nifti.eu

[4] G. Kruijff *et al.*, "Designing, developing, and deploying systems to support human-robot teams in disaster response," *Advanced Robotics*, vol. 28, no. 23, pp. 1547–1570, 2014.

[5] ——, "Experience in system design for human-robot teaming in urban search and rescue," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, K. Yoshida and S. Tadokoro, Eds. Springer Berlin Heidelberg, 2014, vol. 92, pp. 111–125.

[6] G. J. M. Kruijff *et al.*, "Rescue robots at earthquake-hit mirandola, italy: a field report," in *SSRR*, 2012, pp. 1–8.

[7] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[8] D. Hurych, K. Zimmermann, and T. Svoboda, "Fast learnable object tracking and detection in high-resolution omnidirectional images," in *VISAPP*, March 2011.

[9] K. Zimmermann, D. Hurych, and T. Svoboda, "Non-rigid object detection with local interleaved sequential alignment (LISA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 731–743, April 2014.

[10] M. Liu, F. Colas, and R. Siegwart, "Regional topological segmentation based on mutual information graphs," in *ICRA*, 2011, pp. 3269–3274.

[11] M. Liu, F. Colas, F. Pomerleau, and R. Siegwart, "A Markov semi-supervised clustering approach and its application in topological map extraction," in *IROS*, 2012, pp. 4743–4748.

[12] N. Goerke and S. Braun, "Building semantic annotated maps by mobile robots," in *TAROS*, 2009, pp. 149–156.

[13] M. Menna, M. Gianni, F. Ferri, and F. Pirri, "Real-time autonomous 3D navigation for tracked vehicles in rescue environments," in *IROS*, Chicago, USA, 2014.

[14] F. Ferri, M. Gianni, M. Menna, and F. Pirri, "Point cloud segmentation and 3D path planning for tracked vehicles in cluttered and dynamic environments," in *Proceedings of the 3rd IROS Workshop on Robots in Clutter: Perception and Interaction in Clutter*, Chicago, USA, 2014.

[15] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac, "Adaptive traversability of unknown complex terrain with obstacles for mobile robots," in *ICRA*, 2014.

[16] M. Gianni, F. Ferri, M. Menna, and F. Pirri, "Adaptive robust 3d trajectory tracking for actively articulated tracked vehicles AATVs," *Journal of Field Robotics*, pp. –, 2014, accepted for publication.

[17] M. Gianni, G. Kruijff, and F. Pirri, "A stimulus-response framework for robot control," *ACM Transaction on Interactive Intelligent Systems*, vol. 4, no. 4, 2014.

[18] B. Larochelle and G. Kruijff, "Multi-view operator control unit to improve situation awareness in usar missions," in *RO-MAN*, 2012, pp. 1103–1108.

[19] H. Khambhaita *et al.*, "Help me to help you: how to learn intentions, actions and plans," in *AAAI 2011 Spring Symposium "Help Me Help You: Bridging the Gaps in Human-Agent Collaboration"*, 2011, pp. 30–35.

[20] "ROS home page," January 2015. [Online]. Available: http://www.ros.org/

[21] N. Hawes and J. Wyatt, "Engineering intelligent information-processing systems with cast," *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 27 – 39, 2010.

[22] "TRADR (Long-Term Human-Robot Teaming for Robot-Assisted Disaster Response)," 2014. [Online]. Available: http://www.tradr-project.eu

[23] I. Kruijff-Korbayov *et al.*, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *in press in the Zeitschrift fuer Kuenstliche Intelligenz*, pp. –, 2015.

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

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:

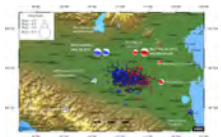
- **Aerial photography, Inspection tasks, 3D Modeling**

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

MAVs perform multiple sorties during missions.

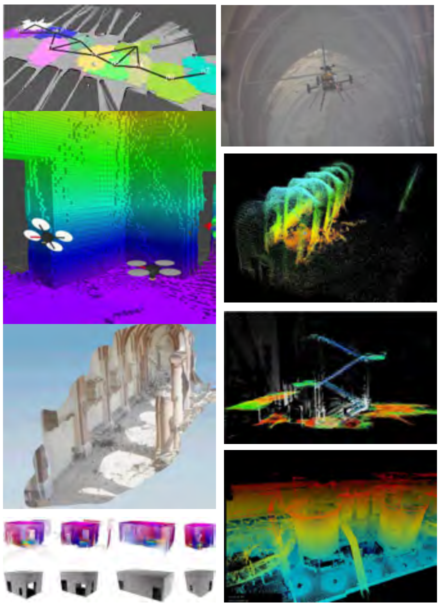
Needed:

- **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*.

TRADR / http://www.tradr-project.eu / EU ICT FP7 Cognitive Systems