DR 5.3: Expectation Management under Multi-Robot Collaboration

Ivana Kruijff-Korbayová[‡], Walter Kasper[‡], Sohini Sengupta[‡], Nanja Smets[*], Jurriaan van Diggelen[*], Bas Bootsma[*], Joachim de Greeff[¶], Chris Rozemuller[¶], Koen Hindriks[¶], Rainer Worst[§], Hartmut Surmann[§]

[‡]*DFKI, Saarbrücken, Germany*

[†]*Interactive Intelligence Group, Delft University of Technology, The Netherlands*

[*]*TNO, Soesterberg, The Netherlands*

[¶]*TUD, Delft, The Netherlands*

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

⟨ivana.kruijff@dfki.de⟩

Project, project Id:	EU FP7 TRADR / ICT-60963
Project start date:	Nov 1 2013 (50 months)
Due date of deliverable:	Month 38
Actual submission date:	March 14, 2017
Lead partner:	DFKI
Revision:	March 14, 2017
Dissemination level:	PU

We report Year 3 progress in the TRADR project WP5: *Persistent models for human-robot teaming.* The reported work includes further development of decision support tools for sharing and reviewing mission progress information for both single robot operation and multi-robot collaboration; the design of working agreements for robots to participate as team-members; new results on the effect of communication failure on agents solving a foraging task using Block World for Teams; an extended study of communication requirements for the completion of a single task in a multi-agent system taking into account more complex temporal goals; and an analysis of team performance gain depending on team size and communication.

Executive Summary

This report presents the progress achieved in Year 3 of the TRADR project in WP5: *Persistent models for human-robot teaming*, addressing Task 5.3: *Expectation Management Under Multi-Robot Collaboration* leading to Milestone MS5.3.

The main focus of work in Year 3 was on further developing the tools and frameworks to support the TRADR human-robot teams to establish "common ground" as shared mutual knowledge and understanding of the situation, the activities of team members and for planning the next steps. The tools aim to recognize and monitor activities of team members, their physical activity, the verbal communication of the human team members and robot states. User interfaces facilitate the access to gathered information in high-level form to all team members but especially to team leaders and other decision makers. Also documentation tasks get supported by the tools. The tools apply equally to both single robot operation and multi-robot collaboration.

The teamwork issues we have investigated this year included in particular the design of working agreements for robots to participate as team-members, the effect of communication failure on agents solving a foraging task, the communication requirements for the completion of a single task in a multi-agent system taking into account more complex temporal goals, and the influence of team size and amount of communication on team performance gain.

We have integrated our agent environment in the TRADR core, which means that the agents now can be employed in practice support situation awareness by showing user specific warnings, errors or notifications, based on (temporal) information available in the high level database.

The TRADR Joint Exercise in May 2016 in Prague and the TRADR End-User Evaluation in October 2016 at the Knepper power plant in Dortmund provided excellent opportunities for collecting development data for the evaluation of the above tools and their further improvemens. The data, interviews and discussions with end-users at these events also provided background for further analysis and investigation of coordination requirements and decision support in multi-agent teams.

Role of Human-Robot Teaming in TRADR

WP5 deals with the issue of how a human-robot team can operate, and grow over time through its experience of working together. Approaching this from the viewpoint of the robot as well as from a human perspective, WP5 aims at developing models and algorithms for determining and recognizing human as well as robot behaviour at the (social) team-level. This encompasses the analysis and modeling of team-level communication and coordination, reasoning with role-based social behaviour at a team level, learning how to adapt that reasoning to better anticipate social behaviour, and learning how to adapt (pre-defined) strategies for team-level interaction.

Contribution to the TRADR scenarios and prototypes

Issues of human-robot teaming are of central importance in the scenario chosen for TRADR, namely the response to an industrial accident consisting of multiple sorties over an extended period. The Year 3 use cases (cf. DR 7.3 of WP7) extend those of Year 2. They involve several teams consisting of a team leader, two UGV operators and UGVs and an UAV-operator with a (piloted) UAV in multiple sorties in a larger and dynamic environment. In addition to (simultaneous) operation of individual robots the Year 3 use cases also include multi-robot collaboration. The teams are performing an initial assessment of an accident site, followed by subsequent information gathering sorties. The use cases provide an abundance of opportunities for teamwork. Control as well as task and resource allocation become more challenging in the larger teams. An important issue with respect to team changes and multiple sorties is how information gathered by one team in one sortie can be transferred and used by new teams in other, later sorties. The work carried out in WP5 Year 3 improved the understanding of the issues involved in these challenges and developed supportive tools and methodologies to address them.

Persistence

Persistence in WP5 is addressed by monitoring events from on-going sorties in persistent databases. The stored information is exploited for creating interactive reports that allow users anytime and anywhere to get an overview of the progress of operations to survey their success and to provide decision support in preparing next steps and future sorties. The provided tools help users to establish common ground as shared information state about the mission. The work on working agreements starts addressing the issues of how a human-robot team can grow over time through experience of working with each other. This is achieved by adapting policies either automatically or by explicit feedback.

1 Tasks, objectives, results

1.1 Planned work

The plan for Year 3 had foreseen WP5 to address *Expectation Management Under Multi-Robot Collaboration* (Milestone MS5.3). The goal was to develop an account of what expectations, conflicts, and needs for alignment (typically) arise in a human-robot team, once robots can act in squads over a period of about 2.5-4 days. The focus in Year 3 was to be primarily on how expectations can be managed at team level during shared control between one or more humans and two or more robots (UGV/UAV) collaborating as a squad in performing mobile manipulation or observation tasks. The key issues to address were keeping the human in the loop and making the behaviour, goals and immediate results of squad operations clear to the human team members.

1.2 Actual work performed

Work in Year 1 had focused on exploring and modeling tasks and teamwork in USAR teams. In Year 2 we continued the theoretical work but also started to develop tools that can actually support teamwork during long-term missions. In Year 3 we again continued both efforts. We built on the experience gained during the Year 2 End User Evaluation (TEval 2015). The TRADR Joint Exercise (TJex) and the TRADR End User Evaluation (TEval) events in 2016 provided further opportunities to refine the concepts as well as to gather additional data for empirical grounding and evaluation of the concepts.

The work WP5 performed in Year 3 comprised the following:

- further development of tools for monitoring team activities and providing reports to build common ground among team members on long-term missions (Section 1.2.1)
- design of working agreements enabling robots to participate as team-members in a search task (Section 1.2.2)
- continued investigation of the effect of communication failure on agents solving a foraging task using Blocks World for Teams (Section 1.2.3)
- analysis of communication requirements for the completion of a single task in a multi-agent system taking into account more complex temporal goals (Section 1.2.4)
- analysis of the effect of team size and communication on team performance gain based on pareto-optimal agents (Section 1.2.5)

- integration of our agent environment in the TRADR core enabling the agents to support situation awareness by showing user specific warnings, errors or notifications, based on (temporal) information available in the high level database (Section 1.2.6)

Below we provide a summary on these subtasks. Section 2 contains abstracts of the papers and reports where this work is presented in more detail and which constitute the annexes of this report.

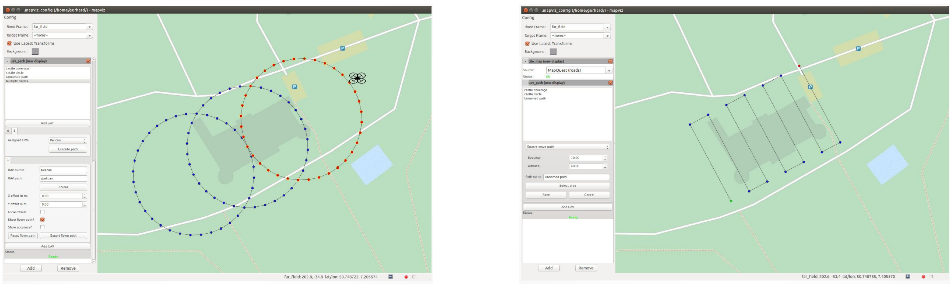
1.2.1 Mission Monitoring and Reporting

Work in year 3 focussed on extending the first monitoring and reporting prototype presented in D5.2 towards including additional information sources such as new sensors but also capturing user actions provided through other user interfaces such as the OCU and the TDS systems. A description of the current system can be found in [14]. Here we will only describe major changes with respect to the first prototype.

- Handling of continuously published sensor data. During Year 3 new environmental sensors such as gas and smoke detectors as well as diagnostic information about the robot status became available. To distinguish between "normal", uncritical information states from these sensors and critical states that users should be informed about and that are worth reporting, thresholding mechanisms were introduced in the monitoring system that would feed the reporting database only when the sensor values exceed some user provided threshold, such as what a critical gas concentration is for the various kinds of recognized gases. Instead of sending all the values to users, the reports will only include the start time and location of a critical phase and the end of the critical phase when the sensors go back to "normal" values. The location is inferred from the robot location if the sensor data do not include locational information themselves.

- Interface to the *Highlevel Database* (HLDB). The first prototype of the Reporting Tool was not using information from the HLDB as its content was limited or redundant with respect to the information we could gather by direct observation of the robots. With the further development of the TDS system, especially the functionality that allows users to provide additional information and mark points of interest on the TDS maps, this changed. Therefore, we developed an interface to the HLDB that allows us to retrieve that information from then HLDB and integrate it into the event reports. Since the HLDB does not provide a notification mechanism that informs clients when information is added or updated in the HLDB, the information currently is available on request only, that is in the asynchronous reports.[1] Whenever users request a report, the HLDB is queried for new information to be added to the already existing data in the reporting database (the *Ground-level database* GLDB).

- Plan reporting. With the progress of work in multi-robot collaboration including more autonomous behaviour of the robots in Year 3 some information about robot planning became available. Users can provide waypoints for the robots to navigate to. The waypoint settings that start a planning process on the robot are captured by the reporting monitor. So users could later see in the reports where the robots were directed to. Figure 1 shows how the planned trajectories are visualised via mapviz plug-in. Here, circle and grid forms are used to generate robot trajectories with optimal coverage of the area of interest for structure from motion. During plan execution the robots provide status information whether they are still ongoing or failed. We also capture this information but only report when the plan execution stops for some reason. The GPS path is stored in the database and is also viewed in the plug-in (see Figure 2).



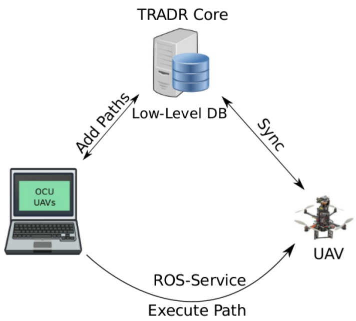
a) Planning three circles for the UAV. Circles are used to generate point clouds based on structure from motion.

b) Grid planning. Grids are used to generate stiched overview images.

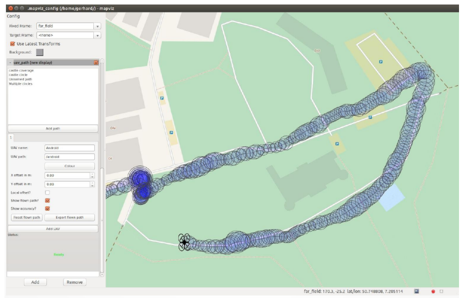
Figure 1: Mapviz planning plugin

We have also worked on analyzing the human-human communication in the team, with the goal to automatically process it in order to extract mission progress information, such as the reported status of the robots, keeping track of tasks assigned to various team members, and information about detected casualties and points of interest. This work is reported under the topic of speech interaction processing in DR.3.3

[1]Integration into the synchronous reports would require continuously polling on the HLDB. This is currently not implemented.



a) A GPS path for a UAV is stored in the database at the Tradr-Core. After getting an execute command the UAV synchronizes itself with the database to get the new flight path.



b) While executing its mission the UAV stores the GPS positions in the database and the plugin visualizes the flight path.

Figure 2: Mapviz plugin

1.2.2 Working Agreements for Robots as Team Members

One of the objectives within WP5 is to study how a human-robot team can operate, and grow over time through experience of working with each other. One way to ensure that robots act as expected is to set working agreements, or policies, beforehand. These policies can be adapted automatically or by explicit feedback. This is because we cannot expect a robot to behave "perfectly" immediately after purchase and also what is "perfect" differs per user preferences and context. Also humans need to have a training period to get to know each other and reach a higher level of team cohesion. The same holds for extending a team with robots.

To investigate the feasibility of using policies in combination with ontologies we took a small scenario that is relevant for the TRADR domain and took inspiration of the TRADR robot for describing the scenario. The scenario we chose was a house search scenario and we described it based on our knowledge of defense protocols. Firefighters evaluated these descriptions to see if they fit their house search protocol.

The scenario was then described in more formal rules from which a subset was chosen to be described in working agreements. An example of a set of working agreements or policies in the house search scenario is:

Policy 1 Robot is not authorized to perform enterRoom.

Policy 2 Robot is authorized to enterRoom when Leader performs enterRoom (overrules Policy 1)

Policy 3 Robot is authorized to enterRoom when enterRoom is requested by HigherAuthority enterRoom (overrules Policy 1)

A small part of the scenario was visualized using Unity. The change of policies, like the one above, was shown over sorties (with an imaginative debrief in which the human user could adapt the policies). Behavior trees were used to change the behavior of the robot within unity based on linked policies. The proof-of-concept showed that it was possible to derive relevant policies from a scenario and implement these using behavior trees that were visualized in Unity. More details are presented in [16] (Annex Overview 2.2) and [29] (Annex Overview 2.3).

1.2.3 Resilient Agents in BW4T

Communication failure is an important problem in rescue missions due to the harsh operating conditions. It is one of the focus points of TRADR to be able to deal with communication failure. In this ongoing experiment we investigate the effect of communication failure on agents that solve a foraging task. For this experiment we use the Blocks World for Teams (BW4T) simulated environment, which was designed to pose the same coordination challenges as a rescue mission [11]. The main research questions for this experiment are:

- What are the effects of communication failure of multi-agent team effectiveness?
- What design choices make agents resilient to communication failure?

To this end we have systematically designed five agents with incremental use of communication, and we additionally investigate 100 different agents designed by our students. BW4T was modified so we can control the probability of message loss. Furthermore we control task complexity, topology, resource redundancy and team size. We use a large computer cluster to test all agents under 252 different conditions with 10 repetitions. From the results we can conclude how communication influences performance, and which agent designs are resilient to communication failure. These insights will be used for the design of the agents in TRADR. This work contributes to the design of the TRADR team by clarifying which communication protocols are most effective in a multi-robot role-based team. More details are presented in [28] (Annex Overview 2.4).

1.2.4 Towards a More General Model on Coordination Requirements with Temporal Goals

In previous work reported in Year 2 we analysed communication requirements for the completion of a single task in a multi-agent system. As a follow-up of this work in Year 3 we investigated an extension that allows for more complex temporal goals (represented using LTL to extend our previous logical model). We have proposed a model that allows for modelling temporal constraints (that do not change anything) as well temporal goals (useful for modelling e.g. tasks that need to be executed in sequence or should be completed before a deadline). Temporal goals are important for setting priorities and expectations. For example, as a general rule a search and rescue team should locate victims first before doing anything else. This work focuses on identifying communication requirements for task completion of these temporal goals. For formal analysis, we have built on our earlier model of the search and rescue domain as a resource redistribution game, where agents can use various coordination mechanisms to jointly achieve their goal. This work contributes to identifying the need for communication, for distributed decision making, and thus for indirectly setting expectations within the TRADR team. More details are presented in [4] (Annex Overview 2.6)

1.2.5 The effect of communication and team-size on robot performance in exploration games

Performance in exploration games can be improved by adding more robots to the team or increasing the communication between robots in the team. However, increasing the robot or communication resources used comes at a cost. It thus is important to gain a better understanding of how much performance can be improved by adding more of these resources. In this paper, we analyse team size and communication in terms of performance gain. We study the performance of 16 different agent decision functions. To this end, we performed simulations taking important environment factors such as topology, map- and task-size and resource-redundancy into account. We show that, depending on team-size and number of messages exchanged, different decision functions are optimal, and we discuss how performance depends on environment factors and task size. More details are presented in [5] (Annex Overview 2.5)

1.2.6 Ontology and agents

This year we succesfully integrated our agent environment in the TRADR core. Agents are now able to support situation awareness by showing user specific warnings, errors or notifications, based on (temporal) information available in the high level database. Furthermore we continued the development of the agent environment with support for Stardog 4.2 as high level database, and additional reasoning and querying capabilities by supporting more SPARQL operators.

Part of the integration challenge has been to define an interface for integrating various KR into agents has been to ensure it facilitates and provides the support for defining different kinds of cognitive state of agents. The states of agents in some agent frameworks are basic and only consist of an agent's beliefs whereas in other frameworks more complex states need to be supported that consist of the agent's knowledge, beliefs, and goals with additional constraints on these components that need to be implemented.

The KR interface that we have designed provides a solution to the general problem of using and integrating any of a range of KRs in combination with any logic-based agent framework. But most of our practical work has focused on supporting the use of semantic web technologies in combination with the GOAL agent framework that we use for developing the TRADR agents. We have integrated support for Stardog 4.2. Amongst others we have added functionality for connecting to persistent databases to enable agents to connect to the TRADR high-level database.

The purpose of making semantic web technology accessible to agents has been to enable these agents to use search and rescue ontologies in their reasoning to support the TRADR S&R team. To achieve this, we have further developed the TRADR ontology with a focus on modelling teamwork and roles. More specifically, we have added four roles including the robot operator, infield team member, (autonomous) UGV and UAV, and the team leader. These roles are used to for example direct information to the appropriate roles, e.g. to inform the team leader of a robot that fails to complete its task.

1.3 Relation to the state-of-the-art

Mission Monitoring and Reporting Monitoring the operation and performance of technical systems is a standard task wherever such systems are used. Sensor systems are used in many domains to monitor e.g. environmental conditions or, in the medical domain, the state of patients. ROS itself provides a number of tools for monitoring and controlling robots, such as the RVIZ tool for visualizing and controlling robot activity and sensors (`http://wiki.ros.org/rviz`) and the *rosbag* tool (`http://wiki.ros.org/rosbag`) that allows to record messages for later reuse.

The generation of various types of reports from data collections and databases is standard practice in business domains. But in the robotic domain, monitoring missions with the goal of creating reports on joint human-robot team activity for non-specialist end-users has found little attention. Usually, the focus is on direct interaction with the robots by operators, that is, technical personnel. The use of natural language descriptions of events in single Autonomous Underwater Vehicle (AUV) missions has been proposed in [12]. These resemble the real-time running commentaries that the TRADR reporting tool provides during sorties. Extended natural language reports for single sorties are envisioned for AUV missions in [13]. The TRADR reporting tool goes beyond these proposals in integrating multiple data sources, in its options for selecting information, and in combining visual, textual and aural modes of presentation.

Working Agreements for Robots as Team Members Much of the research on hybrid human robot teams looks at how the task balancing can be improved. This year we extended this to include team work aspects next to the task work aspects. For this we looked at literature on human-human teaming (e.g. [18, 25]), but also on human-machine and human robot teaming (e.g. neerincx2003interacting, [8]). The use of policies to support the team work aspects within training is a step beyond the state of the art [27].

Resilient Agents in BW4T To provide for teamwork flexibility and reusability, [26] introduces the STEAM architecture. Failure detection is recognized as an important topic for future work, particularly in environments with unreliable communication. Our work is motivated by this work and focuses exactly on this problem and empirically established by means of simulation the impact of communication on team performance. [21] proposes a mobile robotic system (ALLIANCE) to tackle dangerous tasks to reduce the risk for humans. The robot system is designed to be fault tolerant, reliable and adaptive in nature. An explicit broadcast communication system was developed to form interactions between individual robots, but the communication medium is not guaranteed to be available. [17] also uses the BW4T simulation environment to show that so-called helper agents can improve team performance by facilitating communication between team members, thus indirectly showing that the absence of communication can decrease team performance. Our work provides new insights on the impact of communication failure on team performance in search and rescue and similar tasks.

Towards a More General Model on Coordination Requirements with Temporal Goals Many studies have shown [22] advantages of multi-robot systems over single robot systems because of: e.g. robustness, inherently distributed tasks, task complexity, efficiency or simpler robots. There are some formal approaches based on logic that provide coordination mechanisms for teams that guarantee task completion in foraging settings. For example, in [7] TeamLog was applied to guarantee effective teamwork in a rescue robot case. Most of these works, however, do not prove whether coordination mechanisms are needed nor do they show that a particular coordination mechanism is sufficient to ensure task completion. Work on strategic logic like ATL [2] is also concerned with teams of agents pursing a joint goal in an adversarial setting. However, the focus is on finding *joint* winning strategies, ignoring aspects related to (limited) communication and coordination. Also in resource bounded extensions of ATL such as [3, 1] where agents have to coordinate, in principle, on resource production and consumption to achieve a temporal task, questions related to coordination are avoided by focussing on the existence of joint winning strategies. In comparison to our work, the analysis remains at a comparable high abstraction level and no specific coordination nor communication mechanisms are discussed. Another distinguishing feature of our formal model, e.g. also to robotics setting such as [23, 24, 20], is that agents can alter the environment, by picking up, carrying, and dropping resources. As a consequence, our framework is much more grounded in real-world settings.

The effect of communication and team-size on robot performance in exploration games Various experiments performed for studying communication and performance have used the BW4T simulation environment, a tesbed for exploration games [11]. [9] use BW4T to investigate the impact of different types of communication on team performance. They examined agents with four different communication protocols: (i) agents that do not communicate, (ii) agents that only exchange information about the knowledge they have about the environment, i.e. the colors of blocks and their location, (iii) agents that only communicate their intentions, i.e. what they plan to do, and (iv) agents that both communicate about their knowledge and intentions. The main conclusion from this work is that communicating intentions is more effective than knowledge. In a similar setup, [30] also investigates the effects of four different communication protocols on performance: (i) agents do not communicate, (ii) agents only exchange their beliefs, (iii) agents only exchange their goals, and (iv) agents exchange both their beliefs and goals. The main conclusion is similar and this work also finds that communicating about goals is more effective than communicating about beliefs (only) but also shows that interference between robots can diminish the positive effects of communication. A big downside to using BW4T is the simulation speed, since it uses real-time agents. Therefore these experiments investigated only a very small set of agents and environments. In fact, both experiments used only one and the same environment lay-out. We use a simulation environment very similar to the BW4T environment, but in our experiments but go beyond [9] and [30] by evaluating more agent designs and, in contrast to these works, we also vary the topology of of the environment, and systematically vary and explore the impact of various other environment parameters. Moreover, we also study the use of randomization to prevent interference by randomizing the destinations of robots.

Ontology and agents The focus of our work has been on the use of ontologies for logic-based agent frameworks, as we use the GOAL framework for developing TRADR agents that use the TRADR ontology. The first step to enabling our agents to use the TRADR ontology has been to extend them with support for semantic web technologies (such as OWL). In contrast to our approach, most of the work on integrating and adding support for ontologies in agent technology has been rather pragmatic and quite different from the more generic approach that we have proposed. However, there is some related work on integrating semantic web technologies into agent-based frameworks. For example, JASDL is a version of Jason [15] that integrates OWL, and supports agents that incorporate OWL knowledge into their belief base. The work reported in [19] introduces a version of the programming language AgentSpeak based on description logic. The Java-based agent framework, JIAC [10], also uses OWL for representing agent knowledge. Facilitating the use of a specific KR technologies in various applications has been recognized as very useful in the literature. This has driven our efforts for designing an Application Programming Interface (API) for specific technologies. An important aspect of our work has been to also facilitate access to external data sources (in our case the persistent TRADR high-level database, or HLDB). The IMPACT agent framework [6] also offers support for this through an abstraction layer, dubbed *body of software code*, that specifies all data-types and functions of the underlying data source. Their approach, however, needs to be instantiated and requires development for each specific application.

References

[1] N. Alechina, B. Logan, N. H. Nga, and A. Rakib. Resource-bounded alternating-time temporal logic. In W. van der Hoek, G. Kaminka, Y. Lespérance, M. Luck, and S. Sen, editors, *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Toronto, Canada, May 2010. IFAAMAS. (to appear).

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[3] N. Bulling and B. Farwer. On the (Un-)Decidability of Model-Checking Resource-Bounded Agents. In H. Coelho and M. Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 567–572, Lisbon, Portugal, August 16-20 2010.

[4] K. H. Chris Rozemuller. Towards a More General Model on Coordination Requirements with Temporal Goals. Technical report, TU Delft, 2016. Unpublished.

[5] M. N. Chris Rozemuller, Koen Hindriks. The effect of communication and team-size on robot performance in exploration games, 2017. Submitted to IROS 2017.

[6] J. Dix and Y. Zhang. IMPACT: A multi-agent framework with declarative semantics. *Multi-Agent Programming*, pages 69–94, 2005.

[7] B. Dunin-Keplicz, R. Verbrugge, and M. Slizak. Teamlog in action: A case study in teamwork. *Computer Science and Information Systems*, 7(3):569–595, 2010.

[8] J. Garreau. Bots on The Ground. `http://www.washingtonpost.com/wp-dyn/content/article/2007/05/05/AR2007050501009.html`, 2007. [Online; accessed 07 March 2016].

[9] M. Harbers, C. Jonker, and B. Van Riemsdijk. Enhancing team performance through effective communication. In *Proceedings of the 4th Annual Human-Agent-Robot Teamwork Workshop*, pages 1–2, 2012.

[10] B. Hirsch, T. Konnerth, and A. Heßler. Merging agents and servicesthe JIAC agent platform. In *Multi-Agent Programming:*, pages 159–185. Springer, 2009.

[11] M. Johnson, C. Jonker, B. Van Riemsdijk, P. J. Feltovich, and J. M. Bradshaw. Joint activity testbed: Blocks world for teams (bw4t). In *Engineering Societies in the Agents World X*, pages 254–256. Springer, 2009.

[12] N. Johnson, P. Patron, and D. Lane. The importance of trust between operator and AUV: Crossing the human/computer language barrier. In *OCEANS 2007*, 2007.

[13] N. A. R. Johnson and D. M. Lane. Narrative Monologue as a First Step Towards Advanced Mission Debrief for AUV Operator Situational Awareness. In *ICAR 2011*, 2011.

[14] W. Kasper. Team Monitoring and Reporting for Robot-Assisted USAR Missions. In *Proceedings of the 2016 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR-2016)*, Lausanne, Oct. 2016. EPFL.

[15] T. Klapiscak and R. H. Bordini. JASDL: A practical programming approach combining agent and semantic web technologies. In *Declarative Agent Languages and Technologies VI*, pages 91–110. Springer, 2009.

[16] R. Looije, J. van Diggelen, A. Eikelboom, and J. van der Waa. HRT2016. Technical report, TNO, Soesterberg, 2016. Unpublished.

[17] M. D. Manner and M. Gini. Improving agent team performance through helper agents. In *International AAMAS Workshop on Cognitive Agents for Virtual Environments*, pages 89–105. Springer, 2012.

[18] J. E. Mathieu, M. R. Kukenberger, L. D?Innocenzo, and G. Reilly. Modeling reciprocal team cohesion–performance relationships, as impacted by shared leadership and members? competence. *Journal of Applied Psychology*, 100(3):713, 2015.

[19] A. F. Moreira, R. Vieira, R. H. Bordini, and J. F. Hübner. Agent-oriented programming with underlying ontological reasoning. In *Declarative Agent Languages and Technologies III*, pages 155–170. Springer, 2006.

[20] A. Murano, G. Perelli, and S. Rubin. Multi-agent path planning in known dynamic environments. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, pages 218–231. Springer, 2015.

[21] L. E. Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *IEEE transactions on robotics and automation*, 14(2):220–240, 1998.

[22] L. E. Parker. Multiple mobile robot systems. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 921–941. Springer Berlin Heidelberg, 2008.

[23] S. Rubin. Parameterised verification of autonomous mobile-agents in static but unknown environments. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 199–208. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[24] S. Rubin, F. Zuleger, A. Murano, and B. Aminof. Verification of asynchronous mobile-robots in partially-known environments. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, pages 185–200. Springer, 2015.

[25] E. Salas, R. Grossman, A. M. Hughes, and C. W. Coultas. Measuring team cohesion observations from the science. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 57(3):365–374, 2015.

[26] M. Tambe. Agent architectures for flexible. In *Proc. of the 14th National Conf. on AI, USA: AAAI press*, pages 22–28, 1997.

[27] A. Uszok, J. M. Bradshaw, J. Lott, M. Breedy, L. Bunch, P. Feltovich, M. Johnson, and H. Jung. New developments in ontology-based policy management: Increasing the practicality and comprehensiveness of kaos. In *Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop on*, pages 145–152. IEEE, 2008.

[28] J. Z. van den Oever. Multi-agent Communication and Decision making Strategies for Resilient Teamwork with Communication Failure, 2016. Preliminary draft of a Master Thesis in preparation.

[29] J. van Diggelen. Human robot team development: an operational and technical perspective. In *AHFE*, 2017.

[30] C. Wei, K. V. Hindriks, and C. M. Jonker. The role of communication in coordination protocols for cooperative robot teams. In *ICAART (2)*, pages 28–39, 2014.

2 Annexes

2.1 W. Kasper (2016), "Team Monitoring and Reporting for Robot-Assisted USAR Missions"

Bibliography W. Kasper (2016), "Team Monitoring and Reporting for Robot-Assisted USAR Missions". Proceedings of the 2016 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR-2016), October 2016.

Abstract The paper describes a monitoring and reporting system for robot-assisted long-term USAR missions. It is able to monitor robot activities and human verbal communication persistently and to make the information available to users in a structured multimodal interface integrating textual event descriptions, visualizations and audio playback as reports for briefing and debriefing activities as well as for creating situation awareness for new or outside participants. It is realized as a web application allowing it to be used anytime anywhere on any web enabled device. We also present results from an end-user evaluation of the system.

Relation to WP This paper presents the implementation of a reporting system to establish common ground and knowledge in teams for both single robot operation and multi-robot collaboration, extending the prototype presented in Deliverable D5.2. It directly contributes to T5.3.

Availability Public.

2.2 R. Looije, J. van Diggelen, A. Eikelboom, J. van der Waa (2016), "HRT 2016"

Bibliography R. Looije et al. (2016), "HRT 2016". Memo on work performed.

Abstract In this paper we will focus on a method for unmanned systems to learn preferences of individuals. For this we need to answer the question on how to develop an architecture that can adapt the behaviour of the robot in such a manner that it is understandable for human users and supports the feeling of team cohesion in the human team members.

Relation to WP This memo presents an approach, with working agreements, to support expectation management under robot human collaboration. It therefore contributes to T5.3.

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

2.3 J. van Diggelen (2017), "Human robot team development: an operational and technical perspective"

Bibliography Diggelen (2017), "Human robot team development: an operational and technical perspective". Abstract for AHFE conference 17-21 July 2017 Los Angeles.

Abstract Turning a robot into an effective team player cannot be completely realized at design time. This is because many of its behavior requirements only become apparent after the system has been deployed. To illustrate this point, a use case is presented in which robots assist humans during a house search for explosive materials. To successfully participate in this mission, the robot must possess a diverse set of communication skills, e.g. for deciding to whom it should report its findings, or whether it can pick up some object without permission. It is highly unlikely that these behaviors have been perfectly pre-programmed by the robot development firm when the robot was delivered. Therefore, they must be adaptable by the end user without the need of changing code.

In this paper, we propose a solution to this problem from an operational and technological perspective. The operational perspective is illustrated in a use case in which robots are allowed to participate in a debriefing of a mission. One of the goals of a debriefing is to allow the team to develop by sharing positive and negative experiences. In the case of the robotic participants, this means that explicit working agreements can be established between humans and robots which guide future behavior. In this way, the functioning of the human robot team is expected to improve as the team becomes more experienced.

The technological requirements are largely driven by this use case. To represent working agreements in a machine readable way, we have adopted a policy-based approach. Policies are a generic way to specify and govern an agent's behavior using rules for permissions and obligations. To make the policy engine applicable to our use case, we have built ontologies and a domain specific language (DSL). The ontologies define the domain specific terms that are needed to specify relevant working agreements. For example, they specify what qualifies as a dangerous object. The DSL functions as an extra layer above the policy language to make it easily understandable to non-expert users. For example, we have specified the DSL in such a way that sentences like "the robot is not allowed to pick up dangerous objects" is understood by the robot.

The main contributions of this paper are an operationally relevant scenario containing debriefings for human robot team (HRT) development, and

descriptions of two reusable technological components that can be used for HRT development: ontologies, and a DSL. We have tested our approach for HRT development with domain experts in the field using an implemented demonstrator. We have implemented the working agreements in a policy engine, and local agent behavior using behavior tree in a virtual environment. We implemented a test environment in which a human and robot jointly perform a house search and engage in a debriefing afterwards for HRT development.

Relation to WP This annex presents an approach, with working agreements, to support expectation management under robot human collaboration. It therefore contributes to T5.3.

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

2.4 Resilient Agents in BW4T

Bibliography Joris Z. van den Oever (2016). "Multi-agent Communication and Decision making Strategies for Resilient Teamwork with Communication Failure". Preliminary draft of a Master Thesis in preparation.

Abstract In rescue missions communication failure is an important problem due to the harsh operating conditions. It is one of the focus points of TRADR to be able to deal with communication failure. In this ongoing experiment we investigate the effect of communication failure on agents that solve a foraging task. For this experiment we use the Blocks World for Teams (BW4T) simulated environment, which was designed to pose the same coordination challenges as a rescue mission [11]. The main research questions for this experiment are:

- What are the effects of communication failure of multi-agent team effectiveness?
- What design choices make agents resilient to communication failure?

To this end we have systematically designed five agents with incremental use of communication, and we additionally investigate 100 different agents designed by our students. BW4T was modified so we can control the probability of message loss. Furthermore we control task complexity, topology, resource redundancy and team size. We use a large computer cluster to test all agents under 252 different conditions with 10 repetitions. From the results we can conclude how communication influences performance, and which agent designs are resilient to communication failure.

Relation to WP The work reported on here contributes to T5.3 by providing insight on the impact of communication failure on team performance, which facilitates managing team expectations in the context of such failures.

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

2.5 The effect of communication and team-size on robot performance in exploration games

Bibliography Chris Rozemuller, Koen Hindriks and Mark Neerincx. "The effect of communication and team-size on robot performance in exploration games". Submitted to IROS 2017

Abstract Performance in exploration games can be improved by adding more robots to the team or increasing the communication between robots in the team. However, increasing the robot or communication resources used comes at a cost. It thus is important to gain a better understanding of how much performance can be improved by adding more of these resources. In this paper, we analyse team size and communication in terms of performance gain. We study the performance of 16 different agent decision functions. To this end, we performed simulations taking important environment factors such as topology, map- and task-size and resource-redundancy into account. We show that, depending on team-size and number of messages exchanged, different decision functions are optimal, and we discuss how performance depends on environment factors and task size.

Relation to WP The work reported on here contributes to T5.3 by providing insight on the trade-off between performance, number of robots and communication load. Important environment factors for TRADR are considered, and the paper shows which decision functions are optimal.

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

2.6 Towards a More General Model on Coordination Requirements with Temporal Goals

Bibliography Chris Rozemuller, Koen Hindriks. "Towards a More General Model on Coordination Requirements with Temporal Goals". Memo on work performed.

Abstract In multi agent and robot design it is common practice to make use of coordination mechanisms such as communication or turn taking, however it is unclear when this is required for the task. For many tasks coordination is not required and using it could have unnecessary drawbacks on e.g. system complexity, system resources, or reliability. In this paper we formally and systematically proof under which circumstances it is absolutely necessary to use a coordination mechanism to complete a resource redistribution task. To this extend we propose a formal model of agents with beliefs and goals that can manipulate resources in an environment. We also introduce a range of coordination mechanisms that can be used by the agents to achieve their goal. We focus on resource redistribution tasks formulated using linear temporal logic (LTL).We aim to find a complete mapping from LTL formulas to required coordination mechanisms.

Relation to WP The work reported on here contributes to T5.3 by providing insight for which situations coordination, and specifically communication, is required.

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

Proceedings of the 2016 IEEE International Symposium on Safety, Security and Rescue Robotics, EPFL, Lausanne, Switzerland, October 23-27, 2016

Team Monitoring and Reporting for Robot-Assisted USAR Missions

Walter Kasper
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken
Email: kasper@dfki.de

Abstract—**This paper describes a monitoring and reporting system for robot-assisted long-term USAR missions. It is able to monitor robot activities and human verbal communication persistently and to make the information available to users in a structured multimodal interface integrating textual event descriptions, visualizations and audio playback as reports for briefing and debriefing activities as well as for creating situation awareness for new or outside participants. It is realized as a web application allowing it to be used anytime anywhere on any web enabled device. We also present results from an end-user evaluation of the system.**

I. Introduction

The TRADR project (*Long-Term Human-Robot Teaming for Robot-assisted Disaster Response*: [1]) develops novel technologies for human-robot teams to assist in disaster response efforts over multiple sorties during a mission. Various kinds of robots collaborate with human team members to explore the environment and gather physical samples. Throughout this collaborative effort, TRADR enables the team to gradually develop its understanding of the disaster area over multiple, possibly asynchronous sorties (persistent environment models), to improve team members' understanding of how to work in the area (persistent multi-robot action models), and to improve team-work (persistent human-robot teaming). The TRADR use cases involve response to a medium to large scale industrial accident by teams consisting of human rescuers and several robots (both ground and airborne). TRADR missions will ultimately stretch over several days in increasingly dynamic environments.

A sortie in TRADR represents an individual deployment of robots in the field. A sortie typically starts with a briefing session with the complete team, where an situation overview is given, the goals and global activity plan of the sortie are defined, and the team members are assigned their roles and tasks. This establishes the initial situation awareness of the team. A sortie ends with a debriefing session with the complete team. The *International Search and Rescue Advisory Group* (INSARAG) defined general guidelines for these tasks that are discussed in [2]. A team typically involves three levels of responsibilities: the mission commander as the overall leader of operations, team leaders that direct the infield personal and the robot operators and pilots. Team composition can change over time: new members join in, there are shift changes, robots with different capabilities might be required at various stages.

An important aspect in *Urban Search and Rescue* (USAR) operations especially for human team members is reporting: activities, technical problems, environmental conditions, findings, etc. Reports are the main tool used by USAR teams to document and share information about the progress of a mission as well as for instructing team members. Team leaders are required to create detailed log books about the sorties. Therefore, we investigate how TRADR technology can support human team members in their reporting tasks and information sharing, especially with respect to the deployment and use of robotic systems during their work.

[3] distinguish three central parameters for interdependent co-active human-robot teams: *observability*, *predictability* and *directability*. In TRADR, three kinds of user interfaces are involved to achieve these goals:

- The *Operation Control Unit (OCU)* is used by the robot operators and pilots for controlling the robots and to display sensor data.
- The *Tactical Display System (TDS)* is mainly used by the team leader, typically displaying maps of the disaster area and the team activities and findings in the field. It is The main tool for assessing information provided by the infield team members and directing their activities during a sortie.
- The *Reporting Tool* keeps and presents information about the complete mission not just the currently active sortie. Users can select in fine-grained manner which information and aspects they are most interested in.

These tools correspond closely to the distinction between operational and informational reporting ([4]) in information management. Operational reporting is reporting about details and reflects up-to-the-second information. It is typically used by the front-line operations personnel. Very short-term decisions are made from operational reports. Informational reporting is much more strategic in nature, looking at summarized data and extended time horizons, and targeting the management and analytical community. The Reporting tool targets mainly the informational reporting level while the OCU

978-1-5090-4349-1/16/$31.00 ©2016 IEEE

and TDS systems serve the operational reporting needs.

In the following the Reporting Tool will be described in detail. First some motivations for its general design and content will be presented, including requirements gathered from end-users of the system. Then the current implementation of the tool will be described in some detail, illustrated by screen-shots for an impression how it looks like. Finally, related work with impact on our setup will be discussed.

II. Natural Language Reporting

In [5], it is argued that reports in textual form about what is or had been going on in long term missions have many advantages compared to other modes of presenting and sharing information.

1) They are more concise and easy to understand and interpret by non-technical users.
2) They can be better adapted to various information needs.
3) They are easier to remember.
4) They can more easily be shared between (human) team members.

Because of such considerations and the requirement to support the briefing and debriefing task between sorties and also provide documentation support for the missions the core reporting system is based on generating natural language descriptions for events occurring during sorties. The main principle of such reports is that they should specify as much as possible the 5 Ws:

Who did What When Where Why

These corresponds well with what [5] call the 5 situational axis: Spatiality, Temporality, Protagonist, Motivation, Causality. The parameters and their relation to data sources available in TRADR scenarios can be characterized as follows:

Who: The agent/team member, with reference to the team model that defines his role in the team

What: Actions by the agent or other event

- Communication between team members: directives from team leaders, results, ...
- Robot control actions: Directing it to some place, pick up actions, ...
- (Robot) Perceptions: recognized objects, environmental conditions, ...
- Annotations for objects, regions etc.
- Movements of agents

When: Time of action or event (Timestamps)

Where: The event location, preferably descriptive labels for human users, if available, and GPS-based coordinates as general reference system shared between humans and robots

Why: The reasons for the action. These correspond to the motivation and the causality axis of [5].

These dimensions also suggest various report structures: a chronological report with ordering by time, spatial reports will focus on events at some region, motivational or causal reports will present events as connected by motivational or causal relations.

In [5] usability tests for reports organized along such dimensions are presented. They found that most of their users (robot operators) claim the chronological order to be their favorite order, but when asked to replay the mission based on the reports, were discovered to apply implicitly more the motivational order rather than the chronological order. The end users in TRADR similarly expressed a strong preference for chronological ordering as that is required anyway for log book reports.

Different types of report structuring based on the situation models are discussed in [5]. Since their target audience are only the robot operators, they do not consider that different types of users would need different types of reports. But USAR teams consist of various types of users with different responsibilities, capabilities and information needs at different times. The reporting tools reflects such differences by providing various report types but also by allowing users to request reports on specific aspects of the missions.

III. User Requirements on Reports

During an evaluation of the TRADR system with firemen as end users in October 2015 we presented a first prototype of the reporting system and collected their responses about its usefulness and what information they think important for inclusion in reports. The relevance estimations for supplying textual reports in addition to other information sources are shown in Table I. In general, the users regarded the textual mission reports as an important piece of information in prolonged missions. A bit surprisingly, they considered such reports as much less important for preparing new sorties.

TABLE I
Relevance Evaluation from 18 Questionnaires on a 5-point scale (1 = very important, 5 = unimportant)

Average	Description
1.78	Synchronous textual descriptions as running comments
1.72	Documentation
1.78	Debriefing
2.39	Briefing, Sortie Preparation
1.46	Team changes

We also asked the users for other types of information or presentation modes that should be captured by the reports. The most frequent answers were

- Pictures taken
- Display of robot movements and covered area
- Marks for dangerous spots
- Environmental conditions
- Technical problems

Some use cases for real-time synchronous textual reports during sorties as useful fallback channel were mentioned by users:

- Technical problems in transmitting visual or other sensor data from the robots
- Bad sight conditions
- Technical problems in general
- Goals set up for the robots

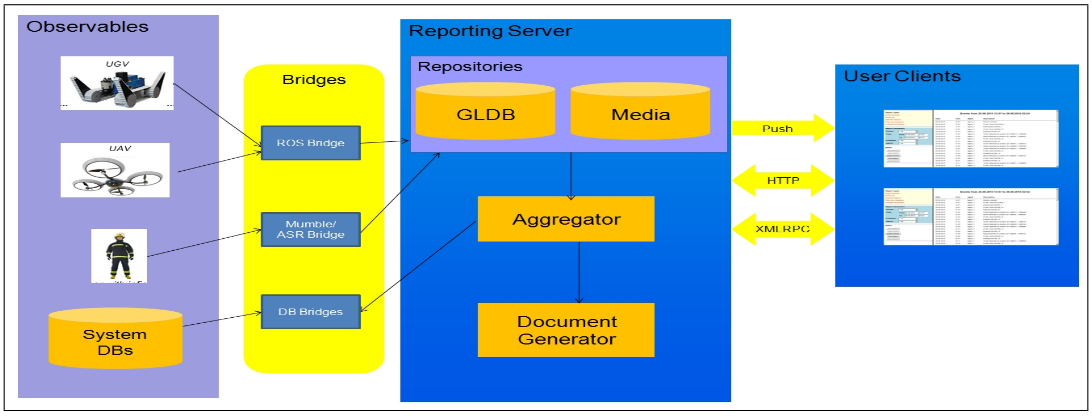


Fig. 1. Overview of the Reporting System

- Communication of additional situational information

The user feedback and requirements motivated to extend the Reporting Tool beyond textual event reports towards interactive visualization and display of multimedia content. Additionally, users can provide and share annotations and comments on all items any time.

IV. Overview of the Reporting System

An important consideration for the design and setup of the TRADR reporting system was that in actual situations TRADR robot teams usually will be part of larger teams that involve members not directly involved in working with robots but performing other tasks. Nevertheless, they should also be able to access information from the TRADR robot teams assuming

- Information from the robot teams should be available *anytime anywhere*.
- Customizable Views: users should be able to select what information they are mostly interested in.

Therefore, the report system is set up as a web application that can be used with any web enabled device (PC, laptop, tablet, smartphone, etc.) with access to the TRADR network without requiring installation of special software. The system consists of three layers:

- a monitoring system that observes *agents*. An agent here can be anything that is a possible source of events.
- the reporting server that receives the events captured by the monitoring system. Also, it answers requests by user clients.
- web clients that send requests to the server and display the results to the user.

In addition to reports requested by users, the system also provides a real-time synchronous mode that during sorties displays short textual descriptions of new events as running commentaries immediately.

The basic architecture of the reporting system is shown in Figure 1. In the following we will describe the components as used in TRADR in more detail.

A. *Monitoring System*

The monitoring system is responsible for gathering data from agents in the system. We call these agents “observables” as we do not really interact with them but they are just regarded as sources of events that we want to report on later. The observation of the agents is managed by *bridges* for specific types of agents. The bridges are responsible for capturing events from the agents and an initial evaluation whether they would be interesting for reporting. E.g., diagnostic information from the robots is only regarded as interesting, if it indicates the start or end of some critical or problem state.[1] If the event looks interesting, it is sent to the reporting server for further handling. The server also receives metadata about the event, such as time stamps, message topics and types as well as the source (agent) of the event.

For different types of observables different types bridges are employed.

1) Robots: TRADR uses ROS for interacting with the robots. For observing the robots we rely on the ROS messaging system. The observer bridge consists of a node that registers with the ROS master of the robot and subscribes to a configurable set of (ROS) topics of interest. The topic sets might differ for robots with different capabilities or tasks. The sets are defined in a configuration file that also can specify threshold values and conditions for published messages to become interesting for the reports. Currently, the following information is retrieved from the robots:

- photos and image snapshots requested by users
- GPS locations of the robot positions for display on a map
- Identified objects with their coordinates
- Diagnostic information about critical battery states, network problems, etc.
- Navigation goals for the robots set by the users and the progress of the navigation plan execution with respect to success or break
- Environmental observations such as dangerous gas concentrations, smoke, etc.

[1]For all continuously published sensor data we try to identify the start and end of interesting sequences.

Fig. 2. Full event log of a sortie

2) Humans: From human team members, the verbal team communication is observed. The monitor captures the audio stream and sends it to an automatic speech recognition system (ASR) that attempts to transcribe the utterance into text. Recognition results and audio data are passed to the server. The audio data are used to replay the original utterance in case the ASR transcript looks wrong. TRADR uses MUMBLE for audio communication. This allows to identify the speakers.

3) TRADR System Database: The TRADR user interfaces allow users to mark and annotate interesting locations and objects detected on images. The reporting server uses this database as additional data source to include user provided information.

B. Reporting Server

The reporting server provides several functionalities:

- It receives the event data form the observables as described above and maintains a repository for the data used for creating mission and sortie reports.
- It provides a push service for clients that register for immediate synchronous notification on incoming events.
- It provides the reporting web application for the user clients.

1) The Reporting Repositories: The reporting server maintains its own repositories for creating reports requested by the clients. There are two repositories:

- a file-based multimedia database for storing images and audio recordings received from the monitoring system.
- a Mongo database (GLDB) as object-oriented database for storing the event data as JSON objects. This allows to store easily any kind of structured data and provides flexible access to the data. Since the robotic event data are based on ROS messages, the ROS message definitions also provide the basic report ontology.

On top of the GLDB we also implemented reasoners as database scripts that augment the database by deriving from the data sets additional meta-information for use in the user interfaces.

2) Push Service: The standard user interface presented in Section IV-C is request-based, that is users have to initiate a request to the reporting server by pressing a button etc. to get some information. An alternative way is provided by a push service that automatically sends information to the user clients whenever some new information comes in. This push service realizes the synchronous reporting mode: Users get information instantly and automatically instead of having to trigger requests. When users register for the push service each time the server receives an event, it will generate a textual description of the event and send it to the client where it will be displayed as a running comment.

3) Report Generation: The main task of the reporting system is the generation of reports about the ongoing mission. The reports are requested by the users through a web interface as described in Section IV-C. As discussed there, several types of reports are possible and users can provide some parameters for fine-tuning what the report should contain. To answer the requests, the major steps to answer the request are

- **Data aggregation**: The server collects the data from its databases that are relevant for the selected type of report and matching the user provided custom parameters. The retrieved raw data get further processed to build an integrated data model for the generator that produces the report document.
- **Document Generation**: Document generation is based on document templates. Each report type corresponds to

a document template that receives the aggregated data model. The document template selects from the aggregate the information items it wants to use, generates their linguistic form, formats them and casts them into a suitable form, such as HTML for the web interface.

C. The Web Client

The user interface of the reporting system is realized as a web application that can be used with any modern web browser. The only obvious restriction is that the user has access to the network where the reporting server is running, such as the TRADR network.

The top-level user-interface as illustrated in Figure 2 is organized in tabbed panels that provide different views.

- the "Reports" tab provides the main interface.
- the "Live Events" tab provides the interface to the push service where users can register themselves for the push service The event descriptions will be continuously displayed also in that view, even if the users in between switches to some other view.
- the "Map" tab allows to display a map of the area and to display the paths of the robots in it and the positions of objects, images, points of interest, etc. An example from a TRADR evaluation is shown in 3. Users can add items and comment items.

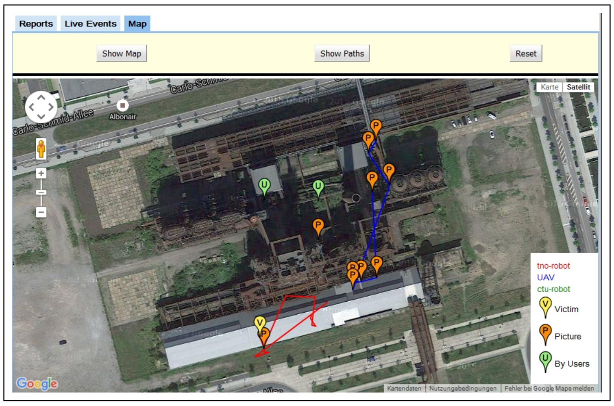
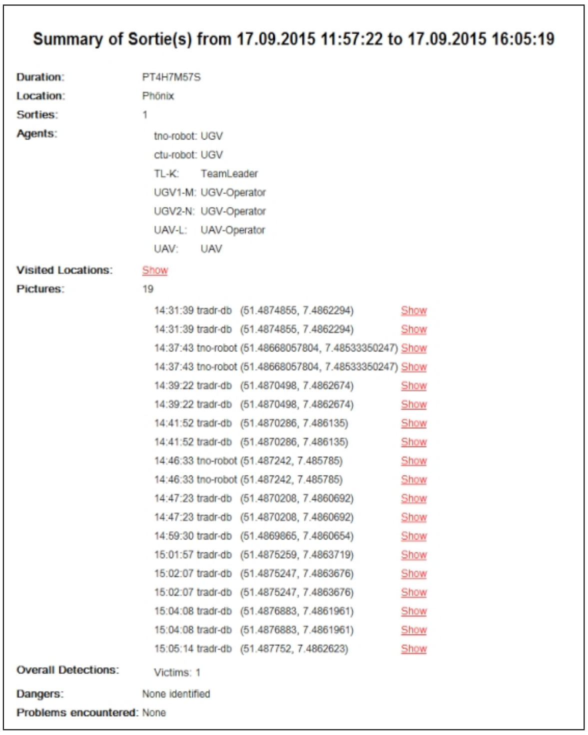


Fig. 3. Map view with paths and object markers

The "Reports" tab illustrated in Figure 2 provides the main user interface. The left-hand side shows available options to the users, while the right-hand side displays the reports requested. The options allow users to select among various report types. Fine tuning of the reports can be achieved by setting parameters for time range, select a specific sortie, focus on activities by selected agents or happenings at a specific location. The "Admin" box provides a number of other operations, some that should be available only to team leaders or command officers, such as marking the start or end of missions or sorties.

Currently, we provide the following predefined report types:

- **Sortie Summary** presents a comprehensive tabular overview of all or selected sorties as illustrated in Figure 4. Locations will be shown as map.



Summary of Sortie(s) from 17.09.2015 11:57:22 to 17.09.2015 16:05:19

Duration:	PT4H7M57S		
Location:	Phönix		
Sorties:	1		
Agents:	tno-robot: UGV		
	ctu-robot: UGV		
	TL-K: TeamLeader		
	UGV1-M: UGV-Operator		
	UGV2-N: UGV-Operator		
	UAV-L: UAV-Operator		
	UAV: UAV		
Visited Locations:	Show		
Pictures:	19		
	14:31:39 tradr-db	(51.4874855, 7.4862294)	Show
	14:31:39 tradr-db	(51.4874855, 7.4862294)	Show
	14:37:43 tno-robot	(51.48668057804, 7.48533350247)	Show
	14:37:43 tno-robot	(51.48668057804, 7.48533350247)	Show
	14:39:22 tradr-db	(51.4870498, 7.4862674)	Show
	14:39:22 tradr-db	(51.4870498, 7.4862674)	Show
	14:41:52 tradr-db	(51.4870286, 7.486135)	Show
	14:41:52 tradr-db	(51.4870286, 7.486135)	Show
	14:46:33 tno-robot	(51.487242, 7.485785)	Show
	14:46:33 tno-robot	(51.487242, 7.485785)	Show
	14:47:23 tradr-db	(51.4870208, 7.4860692)	Show
	14:47:23 tradr-db	(51.4870208, 7.4860692)	Show
	14:59:30 tradr-db	(51.4869865, 7.4860654)	Show
	15:01:57 tradr-db	(51.4875259, 7.4863719)	Show
	15:02:07 tradr-db	(51.4875247, 7.4863676)	Show
	15:02:07 tradr-db	(51.4875247, 7.4863676)	Show
	15:04:08 tradr-db	(51.4876883, 7.4861961)	Show
	15:04:08 tradr-db	(51.4876883, 7.4861961)	Show
	15:05:14 tradr-db	(51.487752, 7.4862623)	Show
Overall Detections:	Victims: 1		
Dangers:	None identified		
Problems encountered:	None		

Fig. 4. Summary Report of a Sortie

- **Event Log** presents a chronological description of events and activities in tabular form. Figure 2 illustrates such an event report, including the communication activities with the transcripts from speech recognition (ASR). The coloring of the agents correspond to the role of the team member, e.g. team leader or operator. Optionally, the original audio recording can be replayed if the ASR result looks wrong. Images can be inspected as popups as illustrated in Figure 5 that shows an event report for one of the agents only (the "tno-robot").
- **Communication Protocol** is similar to the *Event Log* reports but containing only the human communication.
- **Detected Objects** gives a summary of objects detected by the robots with their type and position.
- **Movement Map** displays a map with the paths of the robots as in the map tab.
- **Plans and Obstacles** presents a report focusing on goals and tasks for the robots to be solved autonomously.
- **Problems and Breaks** presents reports on problems encountered by the robots.

Each report reflects the state of the reporting database when the report is requested by users. Thus, users can retrieve anytime actual information as well as past information, depending on the parameter settings.

V. Related Work

Monitoring the operation and performance of technical systems is a standard task wherever such systems are used. Sensor systems are used in many domains to monitor e.g. environmen-

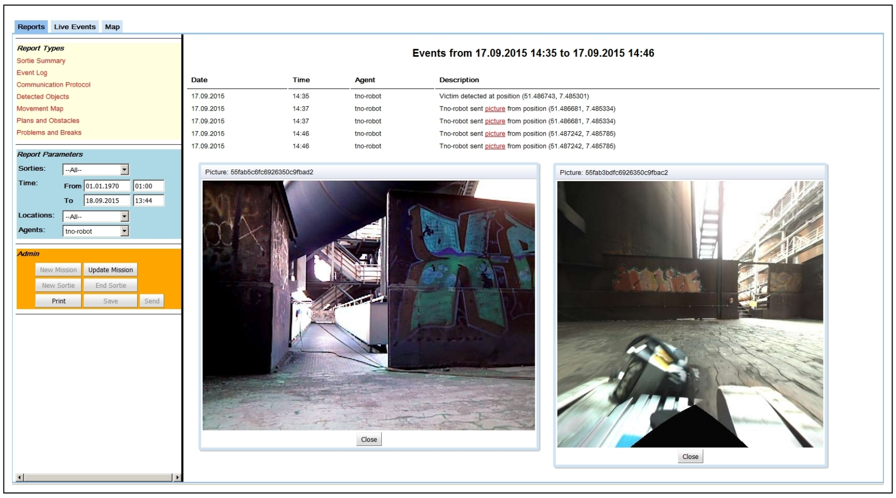


Fig. 5. Event Log Filters: One agent

tal conditions or, in the medical domain, the state of patients. ROS itself provides a number of tools for monitoring and controlling robots, such as the RVIZ tool for visualizing and controlling robot activity and sensors (http://wiki.ros.org/rviz) and the *rosbag* tool (http://wiki.ros.org/rosbag) that allows to record messages for later reuse.

The generation of various types of reports from data collections and databases is standard practice in business domains. But in the robotic domain, monitoring missions with the goal of creating reports on joint human-robot team activity for non-specialist end-users has found little attention. Usually, the focus is on direct interaction with the robots by operators, that is, technical personnel. The use of natural language descriptions of events in single Autonomous Underwater Vehicle (AUV) missions has been proposed in [6]. These resemble the real-time running commentaries that the TRADR reporting tool provides during sorties. Extended natural language reports for single sorties are envisioned for AUV missions in [5]. The TRADR reporting tool goes beyond these proposals in integrating multiple data sources, in its options for selecting information, and in combining visual, textual and aural modes of presentation.

VI. Conclusion

The reporting tool for TRADR provides a flexible and modular framework for reporting activities of team members in USAR teams, with a focus on robot activities and human verbal communication. We provide a persistent monitoring system for agent activities. Reports on these activities can be retrieved anytime. The reports are multimodal in combining concise textual descriptions of activities with visual and audio presentations. Thereby reports are easy to interpret without special technical knowledge. The design as web application allows it to be used with mobile and stationary devices whenever and wherever it is desirable.

The presented system is able to support the management of USAR operations in documenting their progress and findings. Due to its anytime capabilities it also supports shared situation awareness by tracking ongoing activities in the whole team.

Current work focuses on reports about the planning processes in autonomously operating robots. This should enable users to better foresee, understand and control the behavior of autonomous robot activities. Also, new information types related to new robot functionalities are under investigation.

VII. Acknowledgment

The reported work was funded within the EU FP7 ICT programme under grant No. 60963.

References

[1] I. Kruijff-Korbayov, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerincx, P. gren, T. Svoboda, and R. Worst, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI - Knstliche Intelligenz, German Journal on Artificial Intelligence - Organ des Fachbereiches "Knstliche Intelligenz" der Gesellschaft fr Informatik e.V.*, vol. 29, no. 2, pp. 193–201, 2 2015.

[2] G. Foo, *Emergency Response & Search and Rescue*. Black Badge, Jade Consulting Ltd, 2009. [Online]. Available: https://sites.google.com/site/sarbook1

[3] M. Johnson, J. M. Bradshaw, P. J. Feltovich, C. M. Jonker, M. B. van Riemsdijk, and M. Sierhuis, "Coactive design: Designing support for interdependence in joint activity," *Journal of Human-Robot Interaction*, vol. 3, no. 1, pp. 43–69, 2014. [Online]. Available: http://humanrobotinteraction.org/journal/index.php/HRI/article/view/173

[4] B. Inmon, "Operational and informational reporting," *Information Management*, 7 2000. [Online]. Available: http://www.information-management.com/issues/20000701/2349-1.html

[5] N. A. R. Johnson and D. M. Lane, "Narrative Monologue as a First Step Towards Advanced Mission Debrief for AUV Operator Situational Awareness," in *ICAR 2011*, 2011.

[6] N. Johnson, P. Patron, and D. Lane, "The importance of trust between operator and AUV: Crossing the human/computer language barrier," in *OCEANS 2007*, 2007.