Towards a General Communication Concept for Human Supervision of Autonomous Robot Teams

Karen Petersen and Oskar von Stryk
Department of Computer Science
Simulation, Systems Optimization and Robotics Group
Technische Universität Darmstadt, Darmstadt, Germany
{petersen|stryk}@sim.tu-darmstadt.de

***Abstract*—Towards a general concept for human supervision of autonomous robot teams supporting the specific strengths of humans and robots, a communication concept between robots and a human supervisor is presented in this paper. The communication goal is to let the supervisor control a robot team with high-level commands, e. g., by adapting mission details and influencing task allocation in a manner that is applicable to different task allocation methods in general. For this purpose, the supervisor needs a high-level overview of the current state of mission and robots, which can be obtained with the presented approach. Relevant, important events are detected by the robots using complex event processing, and are labeled by topic and priority. A policy system controls the amount of messages that are sent to the supervisor. Notifications are used to inform the supervisor about the mission progress, unexpected events and errors. Queries are used to transfer decisions to the supervisor, to make use of implicit knowledge and experience in critical situations. The robots' level of autonomy can be adapted using policies, that require decisions to be either taken autonomously by the robots, or with support by the supervisor, using different query modes. Example scenarios from different applications including urban search and rescue will be used for validating the proposed concept.**

***Keywords*-human-robot team interaction; supervisory control; complex event processing; policies**

I. INTRODUCTION

Teams of autonomous robots have the potential to solve complex missions such as Urban Search And Rescue (USAR), but are not yet sufficiently reliable and powerful to operate without any human supervision. However, humans and robots have many complementary capabilities, which can contribute significantly to a successful and efficient mission achievement if utilized properly. For example, humans are very good at cognitive tasks such as visual perception, coping with unfamiliar or unexpected situations, and prediction of future states of the world based on incomplete knowledge of the current state. Robots, in contrast, have their strengths, for example, in fast execution of well-defined or repetitive tasks, evaluation and storage of large amounts of data, or operation in areas inaccessible to humans, like narrow spaces, in the air, or contaminated areas. This complementarity has already been observed when comparing humans and machines almost 60 years ago (c. f. Section III-A), which leads to the assumption that this situation will not change significantly in the near future. Therefore, in human supervision of autonomous robots efficient use should be made of these specific strengths.

A. Abilities and Scenarios for Interaction

The proposed concept addresses scenarios, where a team of autonomous robots can be supported by a human supervisor with high-level instructions. The main goal of the robots (called *mission*) can be subdivided into tasks, that may be known prior to the mission start or can emerge during the mission, where each can be fulfilled by a single robot. A task allocation method is used to decide which robot works on which task. It should be noted that the proposed approach can be applied to very different task allocation methods. This can be either a centralized planner for the whole team, or a distributed algorithm, where the robots negotiate the tasks among each other. The choice of an appropriate task allocation algorithm depends on the concrete mission setup and environmental conditions.

Common teleoperation interfaces require one operator per robot, which implies that having a team of robots also requires a team of operators. If a single human supervisor shall be enabled to control a whole robot team, a fundamentally different approach is needed.

The main difference between a supervisor and an operator as defined in [1] is, that the supervisor usually interacts by specifying goals and intentions, while the operator interacts at the action level. High-level commands from the supervisor in a USAR mission may be used, e. g., to confirm or discard a robot's object hypotheses (e. g., victims or fire), to classify terrain trafficability, or to specify regions that are to be searched first or for a second time by a specific robot. In a robot soccer scenario, supervisor interactions may include changing or adapting a team's tactic, or allocating specific roles to individual robots. Common for all applications is, that the supervisor should be enabled to modify the tasks' parameters and the allocation of tasks to robots, and to act as decision support for the robots, e. g., in case they are not granted sufficient authority or do not have sufficient information for good autonomous decisions.

 ISBN: 978-1-61208-117-5

B. General Concept

The goal of the presented concept is on the one hand to enable the supervisor to modify the mission's and tasks' details (including task allocation), and on the other hand to allow robots to transfer decisions to the supervisor, if the robots are not allowed or not able to decide autonomously. Mission and task allocation adaptations can be realized by introducing a layer, that modifies the cost calculated for executing a task, which is independent of the underlying task allocation algorithm. However, to enable the supervisor to take such decisions, he or she needs to be aware of the team's progress towards mission achievement and the current state of the world and the robots.

In research on teleoperation, the required knowledge about the robot's state and the environment is called Situation Awareness (SA), which is adopted from pilot situation awareness and measured with the same tools [2]. SA includes many details needed for teleoperation, that are not needed for high-level commands of a supervisor like the detailed pose of a robot, its detailed numerical distance to obstacles, full video streams of the cameras or live map-data. Therefore, we introduce the term *Situation Overview (SO)*, which includes more general knowledge about the world and the robots' status, e. g., the health of each robot (battery status, sensor and actuator functionality), the current and planned next task of each robot, and the overall mission progress. Maps and video data are only sent if needed for specific decisions, e. g., to decide if an image shows a potential victim or not. Details only needed for teleoperation are omitted.

The needed information is usually not fixed, instead, the communication should be adopted during runtime to the specific needs of different missions and supervisors. For this purpose, we propose to control the amount of information using policies, which define the events to be detected by the robots using complex event processing. These events are then classified according to their priority and are sent to the supervisor as notifications (to provide information) or queries (to ask for decision support).

A main advantage of SO over SA is the reduction of data that has to be communicated. This is an important factor in real-world applications where the available communication bandwidth is usually limited. SO, obtained with the presented methods, gives a human supervisor a basis for high-level team interactions, without overburdening the human with too specific information of single robots.

The rest of this paper is organized as follows: Related work is discussed in Section II. In Section III, first the interactions among humans in loosely coupled teamwork are observed. Second, inspired by these findings, the methods enabling the robots to send notifications and queries to the human supervisor are described. For detecting the important incidents, methods from complex event processing are applied. The messages are classified with different levels to allow filtering. The amount of messages can be controlled using policies, which can be adapted either manually or automatically, depending on the supervisor's workload. Some application examples, for general robot team applications and for concrete scenarios of USAR and robot soccer, are given in Section IV. The methods are discussed and future work is described in Section V.

II. Related Work

Especially in the USAR domain, much research has been done on teleoperation interfaces, e. g., [3], [4]. These strongly rely on video- and map-data, that need to be sent in real-time from the robot to the user interface. On the one hand, this allows to accurately control a robot even in unstructured and complicated environments, but on the other hand, those interfaces cannot be extended easily to control more than one robot simultaneously, and require high bandwidth, which is often not permanently available in real-world scenarios. Further, most teleoperation interfaces require extensive operator training and continuously demand maximum concentration of the operator, hence quickly leading to task overload and operator fatigue.

Approaches that allow a single supervisor to deal with robot teams and do not require continuous high bandwidth communication can be found in the area of sliding autonomy or mixed initiative. In [5], Markov models are used to decide whether a robot works on a task autonomously or is being teleoperated by an operator. This requires continuous communication connection only during the teleoperation phases. The mixed initiative system presented in [6] allows the operator to manually switch between autonomy modes, where the operator input varies from goal input to full teleoperation. Similarly, in [7], the operator can assign waypoints, move the camera, or completely teleoperate a robot. With the augmented autonomy approach used in [8], the robots in an exploration scenario can either select their next waypoints autonomously, or the operator can assign waypoints. Results show, that these methods are appropriate to deal with a larger number of robots and can produce much better results than purely autonomous or purely teleoperated systems. However, they still require periods of continuous communication connection between the operator and the robots, and most of them can hardly be extended to fundamentally different scenarios, where the main focus is not on search or exploration.

A completely different approach is described in [9], where the robots can ask questions to the human supervisor. Similarly, in [10], the human is treated as a source of information for the robots. The level of autonomy is controlled by adjusting the cost to contact the supervisor. The teleautonomous system presented in [11] enables the robots to detect situations where human intervention is helpful, which are in this context the states of robot stuck, robot lost or victim found. Human supported decision taking is presented in [12], here two variants are proposed: management-by-exception, where the operator can veto against an autonomous decision, and management-by-consent, where the operator needs to confirm an autonomous decision before execution. In [13], policies are used to restrict the autonomy bounds of the robots, in this context also rules are defined about which messages the robots are required to send to the human. These approaches are promising to be applicable to larger robot teams in real-world environments, because they do not require continuous human attention to a single robot and require less bandwidth as they do not rely on video streams. However, they are still not very flexible to be adapted to fundamentally different scenarios or for on-line adaption to different operator preferences. Furthermore, the events that require operator intervention are detected manually, and yet no method has been provided to flexibly detect complex events in arbitrary complex situations.

III. Control of Communication between Robots and a Supervisor

The concept presented in this section is inspired by teamwork among humans, which is described briefly. Afterwards, the specific strengths of humans and robots, that contribute to these kinds of scenarios and interactions are revised. Finally, the methods used to realize a flexible communication between the robots and the supervisor are presented and discussed.

A. Interactions in Team Work Among Humans

When observing interactions in loosely coupled workgroups [14], some commonalities can be observed regardless of the scenario, e.g., home care, knowledge work, firemen in a search and rescue scenario, soccer players coordinating with each other and getting instructions from a coach, or people in an office preparing an exhibition at a fair: In all these situations, the overall mission is first subdivided into tasks, that are assigned to the individual team members. Every participant works on his or her tasks autonomously, and reports the progress to the teammates or the leader, either explicitly by verbal or written communication, or the progress can be directly observed by the others. Whenever an individual has problems in fulfilling a task, he or she can ask somebody else (who is expected to be more capable for this specific problem) for support.

To understand the benefits of supervisory control, it is important to be aware of some fundamental differences between humans and robots. In [15], the superiorities of humans among machines and vice versa are discussed. One of the main outcomes is that machines are good in fast routine work, computational power and data storage, while humans' strengths are perception, reasoning, and flexibility. These findings (although almost 60 years old!) are in most points still valid and can be transferred to a large extent from machines to robots. Especially the superiority of humans over robots in problem solving and situation overview is crucial, and does not seem to change in the near future. Further, although there are several sensors that allow robots to perceive data that humans cannot sense directly (e. g., distance sensors, infrared sensors), humans are much more capable in interpreting data, especially images.

As a conclusion, if a human supervisor is aware of the overall situation, but not necessarily of all details, it does make sense to leave some high-level decisions to the human, because he or she can be expected to decide based on implicit knowledge, situation overview and experience, that cannot easily be added to the robots' world model. Due to the complementary capabilities of robots and humans, it can be expected that humans can cope well with the problems that robots cannot solve autonomously.

If this model of human teamwork is applied to human robot interaction, with the human taking the role of a supervisor, the robots are required to report their progress and unforeseen events to the human, and ask for support if they cannot solve their tasks sufficiently well autonomously. This is enabled by using three methodologies: First, important or critical events are detected using complex event processing(Section III-B). Second, the detected events are classified to message classes, which are different levels of notifications and different query modes, according to their criticality (Section III-C). Third, the message flow is controlled by policies, that define which messages need (not) to be sent to the supervisor (Section III-D).

B. Complex Event Processing

The events to be detected by the robots can be very diverse to many aspects. Some are just special variables exceeding thresholds, others are patterns that have to be detected, or several occurrences of different events simultaneously. Certainly, the detection of every single event could be programmed manually, but this is very time consuming, can lead to many failures, and usually duplicates lots of code.

The research field of Complex Event Processing (CEP) deals exactly with this question, of how to detect events in communication systems [16], for example in databases or Wireless Sensor Networks (WSNs). In WSNs, the challenge is to use several hundreds of distributed sensor nodes to detect events, e. g., human presence or fire, and combine simpler events to detect complex events, that are aggregations or patterns of several events. *Simple events* are discrete events, that can be directly detected without aggregating more information, e. g., a variable exceeding a threshold, or a sensor (not) delivering data. *Complex events* are events that are composed of two or more (simple or complex) events, or events enhanced with external information. These compositions can be two events occurring simultaneously, an event chain, patterns, etc. To describe those aggregations, event algebras are used, e. g., HiPAC [17], SNOOP

[18], REACH [19]. Those algebras provide operators as conjunction, disjunction, sequence, etc., to combine two or more events to a complex event. They vary in complexity and versatility. Depending on the application, an appropriate algebra needs to be chosen, that satisfies all needs, but is not too complex, hence being more difficult to understand and leading to higher implementation efforts.

The analogy between CEP as used in WSNs and robotics is, that there are several sensors and pre-processed data available, based on this information certain events or states of the robot or the world have to be detected. The key differences are, that a robot has less, but more reliable sensors than in a WSN, and the "network" is static, apart from sensor failure. Further, a robot's sensors are not entirely distributed, they are all physically connected, therefore issues like time synchronization and timeliness can be disregarded for CEP on robots. If also events are considered that involve more than one robot, the team can be seen as a WSN. Overall, CEP provides good methodologies, that can be used efficiently not only in databases and WSNs, but also on robots.

To allow efficient filtering, events can be tagged. E. g., in a search and rescue mission, there could be events related to victim detection, events related to simultaneous localization and mapping (SLAM), or events related to the vehicle's health as the battery status.

Some examples of important events in a USAR mission are of course if a robot has detected a potential victim or a fire, but also reports about the status of the exploration, e. g., if a room has been explored completely without finding a victim. In a humanoid robot soccer match, a robot can monitor the frequency of falling when walking or kicking, taking into account disturbances by teammates or opponents (e.g., by pushing), and can deduce if it is still capable of playing efficiently. The goalkeeper can monitor its benefit to the match, if it observes the frequency of jumping to catch the ball, compared to the number of goals scored by the opponent, i. e., if the goalkeeper jumps for the ball, and no goal is scored directly afterwards by the opponents, the team presumably benefits from the goalkeeper. If the opponents score, regardless of the goalkeeper jumping or not, the robot can potentially contribute more to the team's success when acting as a further field player.

C. Message Classification

The supervisor shall be supported – and not confused – by the messages from the robots. To enable the user interface to prominently present critical messages and show other information when needed to obtain SO, the messages are classified according to their importance and criticality. The queries are graded with different modes of action selection, depending on the desired degree of robot autonomy.

Proposed Levels of Notifications: Usually, logging systems for software development use five stages: debug, information, warning, error, and fatal. The concept provided here targets users unfamiliar with the implementation details of the robot control software, hence the debug-level can be omitted here, as these notifications would confuse the supervisor, instead of advancing the SO. Fatal are usually those errors, that cannot be handled properly and lead to program termination. Because these notifications cannot be communicated, also the fatal-level is omitted here.

In summary, there remain three notification levels, to be used by the robots: *information*, representing regular events (e. g., start or termination of execution of a task), *warning*, representing unexpected but noncritical events (e. g., task execution takes longer than expected), and *error*, representing critical events (e. g., sensor failure).

As examples for notifications, an information can be sent by a USAR robot, informing the supervisor that it has finished exploring a room without finding any victims. A warning can be sent by a soccer robot, that detects that it falls frequently without external influence and therefore cannot play properly. An error should be sent by a robot that detects that an important sensor, e. g., the camera or the laser range finder, does not deliver any or sufficiently meaningful data.

Types of queries: Depending on the desired degree of robot autonomy, there are several possibilities to take decisions. Besides deciding and executing everything autonomously, also the supervisor can be integrated for confirming or vetoing decisions, or even for selecting the appropriate answer. Decisions that allow or require supervisor intervention are formulated as queries.

Three query classes are proposed:

(1) *Autonomous decision with veto:* The robot selects among several solutions, and does not start execution before a specific time t_{exec} has elapsed. The supervisor is given a time t_{veto} to contradict this decision. t_{exec} and t_{veto} are independent of each other, which means, if $t_{exec} < t_{veto}$, the supervisor can veto a decision even after the robot started execution, if $t_{exec} > t_{veto}$, the supervisor cannot abort execution after it started.

(2) *Autonomous decision with confirmation:* The robot selects among several solutions and presents the selected solution and the alternatives to the supervisor. Execution does not start before the supervisor confirms or contradicts the selection.

(3) *Supervisor decision:* The robot provides several solutions to the supervisor, but does not preselect a solution. Execution starts after the supervisor selects and confirms a solution.

The robots are granted more autonomy in the first class, and less autonomy if confirmation by the supervisor is required. The second and third query classes make no difference for the robots, but for the human there is a psychological difference if a selection is proposed or not.

As an example, the humanoid soccer goalkeeper of the previous paragraph is considered. If this robot detects that it is either not needed (because the opponents do not shoot on the goal) or is not beneficial (because it cannot block the goal shots), the robot could instead act as an additional field player, to potentially contribute more to the team's success. Depending on how much autonomy is granted to the robot, this tactic change could either be autonomous with veto, or (to give the human more control) autonomous with confirmation.

D. Control of Message Flow

The amount of messages that are sent to the supervisor needs to be controlled carefully. On the one hand, too many messages can result in information overflow and supervisor stress, or in complacency if most of the robot decisions are trivial, which brings the danger of overseeing wrong decisions. On the other hand, too few messages lead to a loss of SO. In general, there should not be any static rules about which events shall be communicated to the supervisor, and which decisions the robot should take autonomously or with some support by the supervisor. Rather, this is highly dependent on the current mission, the supervisor's preferences, and the supervisor's trust in the system.

In [13], policies are used to define the bounds of an agent's autonomy. Policies are positive and negative authorizations, that define what an agent is (not) allowed to do, and positive and negative obligations, that define what an agent is (not) required to do. Policies are applied to actions as well as to communication, e. g., sending acknowledgments when receiving new instructions. In the scope of this paper, the only bound on autonomy is decision taking, therefore it is sufficient to apply similar rules to regulate the amount of notifications and queries.

By means of the tagged events defined in Section III-B and the different messages classes defined in Section III-C, policies can be defined for groups of messages, according to their importance, or according to a topic, or for single event types.

Sets of policies can be loaded, dependent on the current mission, or even situation dependent. Further policies can be defined by the supervisor. Finally, policies can be adapted automatically, depending on the supervisor's current workload. For example, if there is a number of pending queries, only queries with supervisor selection or supervisor confirmation should be sent, because queries with supervisor veto can be expected to expire before the supervisor notices them.

As an example in the USAR scenario, a supervisor without trust in the robots' autonomous victim detection might want to get informed every time human-like temperature is detected with a thermal sensor, while a supervisor with more trust might be satisfied getting just the hypotheses that are positively verified by the robots. For the soccer scenario, if the supervisor is occupied with notifications about malfunctioning sensors or instable walking abilities, the queries about tactics changes can be omitted, because they are just of secondary importance if so many other problems have to be handled.

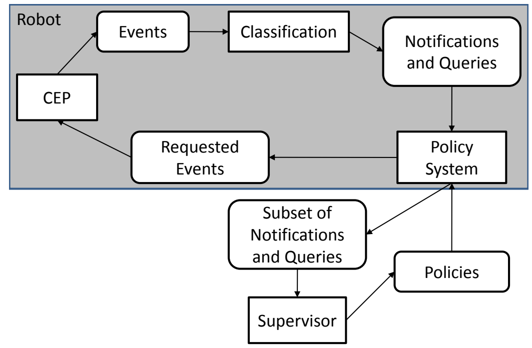


Figure 1. Visualization of the interactions among the different components of the proposed general communication concept

E. Discussion

All three methods applied here have been well established in entirely different fields. The new concept is, to combine them, and to use them to enable a human supervisor to obtain situation overview on a high level.

Overall, the three components are connected in a loop with external feedback from the supervisor, as shown in Figure 1: CEP detects important events, which are then classified to notification levels or query types. The policy system then decides which of those messages are sent to the supervisor. For closing the loop, the policies can be adapted during runtime, either manually by the human or automatically, and therefore it changes dynamically, which events have to be detected by the CEP system. Compared to other approaches, the presented concept allows a more flexible communication, that allows to control on the one hand the supervisor's workload, and on the other hand also the use of network capacity, if low bandwidth is an issue.

IV. Application Examples

In this section it is demonstrated, how the proposed approach can be applied to different scenarios from different applications. First, some general examples are given, that apply to arbitrary robot missions in general. Second, first steps of the integration of the concept into our USAR robot and our humanoid soccer robots are outlined.

A. Mission-independent Examples

With most robot user interfaces, the supervisor needs to be familiar with the system for deciding which system functions need to be monitored, and how they can be monitored. With the methods proposed in this paper, the robots provide methods to monitor themselves and can on the one hand inform the human about the status, and on the other hand send warnings if the status changes or is critical.

Before the start of a mission, all important sensors have to be checked for functionality. Instead of doing every check by hand – which is often omitted or only done for some samples to save time – this can be done automatically using CEP. A successful check results in an event of the type ”successful sensor check”, which is sent as information message. If a check fails, an event of the type ”sensor failure” is sent as error message, accompanied with an error description.

The battery status of every robot should be monitored continuously, to prevent malfunctions because of too low voltage or damaged batteries. Battery displays for each robot can be overlooked, especially if a single human has to monitor the battery status of many robots in parallel to several other monitoring or coordination tasks. With the methods presented in this paper, each robot can monitor its battery status individually, and can send a warning notification before the battery runs empty. The methods of complex event processing further allow to warn not before the voltage is constantly below a threshold for some seconds, and therefore is able to filter voltage peaks or faulty measurements.

As a general proposal, an unexperienced supervisor should start with no restricting policies, and then gradually constrain the messages, if they are not needed. On the one hand, this leads to lots of messages at the beginning, but on the other hand, the supervisor learns, which types of events are provided by the robots and can decide on this basis which messages are important for the current setup.

B. Example: Urban Search and Rescue

In the USAR setup, a team of heterogeneous, autonomous robots has to search for trapped victims in a partially collapsed building, e. g., after an earthquake, and to locate potential hazards like gas leaks or fire. The methodologies proposed in this paper apply to robot behavior that can be observed for example at RoboCup rescue, whereas in current real-world deployments the robots are not yet autonomous at all, and the proposed concept requires robot autonomy as a starting point.

The results are discussed for the Unmanned Ground Vehicle (UGV) of Team Hector Darmstadt [20] (Figure 2(a)). This robot can autonomously explore an environment, build a map and search for potential victims and markers that indicate hazardous material (hazmat signs, see Figure 2(b)). Although at RoboCup the simulated victims are not yet too sophisticated (baby dolls with electric blankets, as can be seen in Figure 2(c)), and can be detected by only using a thermal sensor (see Figure 2(d)), we use a sensor fusion algorithm, that combines victim hypotheses from the daylight camera and the thermal camera with information from the laser range finder to build up a semantic map [21]. This algorithm is also suitable for more realistic conditions than RoboCup, i. e., to reliably find real people in environments that also contain other heat sources or shapes similar to humans.

With most user interfaces, the supervisor requests the information he or she believes to be of interest. The only information that usually pops up automatically is, when the robot states to have found a victim. In this case, some important data is potentially not monitored, or much information is sent continuously, even if it is not needed. For example, the operator requests the map generated by the robot, the robot's position in the map, and the camera images, but does not have a look at the output of the thermal sensor. If this sensor has a malfunction, it is potentially never noticed. We propose instead, to automatically provide the operator with relevant information.

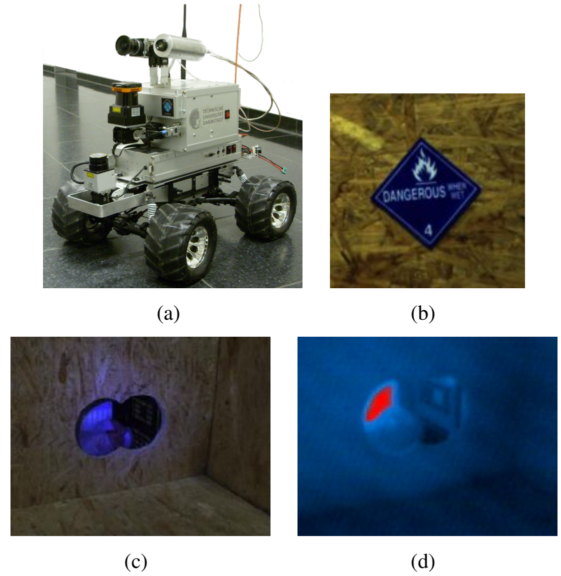
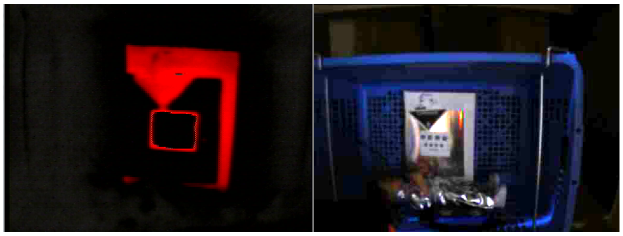


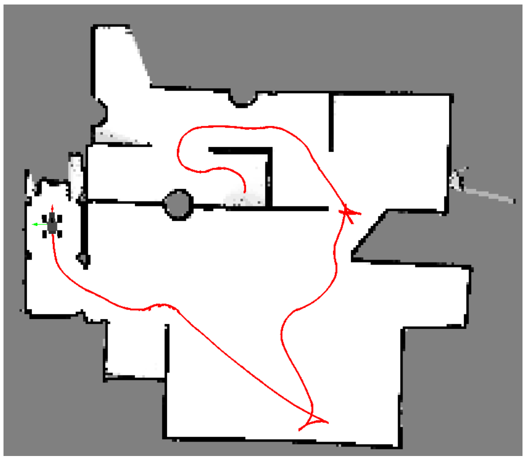
Figure 2. (a) Hector UGV. (b) Example of a hazmat sign. (c) Example of a simulated victim at RoboCup. (d) Thermal image of the simulated victim.

Mission progress is usually monitored by looking at the camera images and the map in real-time. However, as the robots usually do not proceed very fast, not all information is needed all the time. With the methods provided here, it is possible to send an image of the map every time a progress is observed. Progress can be, for example, every time the robot traveled more than 3 meters, or every time a robot enters or leaves a room, which results in an event of, e. g., the type ”entering room”, labeled to the general topic ”progress”, and is published as information message. In addition, every time a robot starts exploring a new victim hypothesis, this can be communicated, possibly together with attached sensor data that motivated the victim hypothesis. This results in an event of the type ”explore victim hypothesis”, labeled as related to ”victim” and ”progress”, and is published as information message. In turn, this method also allows to detect a lack of progress (by observing that no progress events are detected for a predefined time, although the robot intends to move), which can indicate a malfunctioning or disoriented robot. This is an event of the type ”no progress”, labeled with the topic ”progress” and should be published as a warning message. A supervisor who trusts in the robot's

 ISBN: 978-1-61208-117-5



(a)



(b)

Figure 3. (a) Images showing a simulated victim in the thermal image and the camera image. (b) The current map learned by the robot.

capabilities may not want to see all progress messages, but only those that refer to non-progress. To achieve this, he or she can define two policies: a negative authorization for sending notifications labeled with ”progress”, and a positive obligation for sending notifications of the type ”no progress”.

If a robot detects a victim, the resulting event is a supervisor decision query, where the supervisor can decide to (a) confirm the victim, (b) discard the victim, or (c) try to collect more information. The message also contains images from the cameras (see Figure 3(a)), and an image of the current map to display the location of the victim (see Figure 3(b)). The red line in Figure 3(b) shows the robot's traveled path. It can be seen that continuous monitoring of the map does not give more information to the supervisor than an image of the map every time a progress is observed or if the robot got stuck for a while, as it was the case in the upper right corner. Therefore, much communication overhead can be saved by omitting data transmissions that do not advance the supervisor's SO.

Queries can not only be used to let the supervisor confirm or discard potential victims or hazards, but also for decision support regarding path planning, e. g., an autonomous decision with veto can be sent, if the terrain classification is not confident enough and the supervisor should decide if a robot can negotiate an area or not.

C. Example: Humanoid Robot Soccer

At RoboCup soccer matches, usually monitoring is done by visually observing the match, at the team Darmstadt Dribblers [22] also the team messages sent between the robots are monitored. As direct human intervention is not allowed by the rules, the proposed concept can be used either for tuning the robots in practice games, or for changing details during game breaks.

Monitoring the health of each robot could also be done visually, but with three or more robots on the field it is difficult to keep track of each robot's performance. With CEP, it is possible to monitor the falling frequency of each robot for different motions like walking or kicking, and its correlation with other factors like the vicinity of opponents or teammates (which could indicate that the fall was due to a collision), or motor temperature and battery status, which could be a reason to switch to a behavior that consumes less energy, e. g., dribbling instead of kicking the ball.

Further, the benefit of the specific roles can be monitored, like already proposed for the goalie in Section III. This allows on the one hand to tune the parameters during tests to maximize each role's benefit, and on the other hand to quickly change tactics or preserve hardware during a match.

V. Conclusion and Outlook

The communication concept presented in this paper is designed for interactions between a human supervisor and a team of autonomous robots. To make use of the specific complimentary strengths of humans and robots, supervisor interactions are focusing on high-level commands. As a basis for high-level decisions, the supervisor needs SO, which can be flexibly achieved for several fundamentally different scenarios using the presented methods, which are complex event processing, message classification, and policies. This communication concept is inspired by loosely coupled human teamwork and requires a low communication overhead compared to standard teleoperation methods, because only data needed for SO are sent, while omitting details only used for exact teleoperation. The methods enable a human supervisor to gain a general SO of a whole robot team, without requiring the supervisor to be familiar with implementation details. The performance of the team can be enhanced by transferring critical decisions to the supervisor, because in this case the decision is based on SO, human experience and implicit knowledge, and is therefore expected to be more reliable and efficient for achieving the mission goal.

In general, an interface that is based on this new communication concept can provide a higher SO than standard interfaces, because the robots can send information that the supervisor would probably not request, hence problems and errors can potentially be noticed earlier. SO gives the supervisor a basis to take high-level decisions, e. g., for adapting task allocation or mission details. Preliminary results in USAR and robot soccer indicate the potential of the

 ISBN: 978-1-61208-117-5

developed concept. These two fundamentally different setups demonstrate, that the proposed concept can be applied to a large variety of problem classes. It is furthermore planned to implement the whole concept, including the communication concept as well as high-level commands by the supervisor, for different scenarios with heterogeneous robot teams.

Future work includes experiments in simulation and with real robots to support the hypotheses of this paper. It is planned to conduct user studies in different application scenarios to show the wide applicability of the proposed methods. Further, the possibilities of the supervisor to coordinate robot teams based on the proposed situation overview will be examined. For dealing with larger robot teams, a basis for a large-scale interface is provided by the presented concept, as it offers the data for SO and supports efficient filtering.

ACKNOWLEDGMENTS

This research has been supported by the German Research Foundation (DFG) within the Research Training Group 1362 “Cooperative, adaptive and responsive monitoring in mixed mode environments”.

REFERENCES

[1] J. Scholtz. Theory and evaluation of human robot interactions. In *Proc. of the 36th Annual Hawaii Intl. Conference on System Sciences*, 2003.

[2] M. R. Endsley. Situation awareness global assessment technique (sagat). In *Proc. of the IEEE 1988 National Aerospace and Electronics Conference*, volume 3, pages 789 – 795, 1988.

[3] M. W. Kadous, R. Sheh, and C. Sammut. Effective user interface design for rescue robotics. In *Proc. of the ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 250–257, 2006.

[4] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics and Automation*, 23(5):927–941, October 2007.

[5] B. P. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. *Proc. of the IEEE - Special Issue on Multi-Robot Systems*, 94(7):1425 – 1444, July 2006.

[6] Jacob W. Crandall and Michael A. Goodrich. Experiments in adjustable autonomy. In *IEEE Intl. Conference on Systems, Man, and Cybernetics*, volume 3, pages 1624 –1629, 2001.

[7] J. Wang and M. Lewis. Human control for cooperating robot teams. In *Proc. of the ACM/IEEE intl. conference on Human-robot interaction*, pages 9 – 16, 2007.

[8] Y. Nevatia, T. Stoyanov, R. Rathnam, M. Pfingsthorn, S. Markov, R. Ambrus, and A. Birk. Augmented autonomy: Improving human-robot team performance in urban search and rescue. In *Proc. of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 2103 – 2108, Nice, France, Sept, 22-26 2008.

[9] T. Fong, C. Thorpe, and C. Baur. Robot, asker of questions. *Robotics and Autonomous Systems*, 42:235 – 243, 2003.

[10] T. Kaupp and A. Makarenko. Measuring human-robot team effectiveness to determine an appropriate autonomy level. In *Proc. of the IEEE Intl. Conference on Robotics and Automation*, pages 2146 – 2151, Pasadena, CA, USA, 2008.

[11] R. Wegner and J. Anderson. Balancing robotic teleoperation and autonomy for urban search and rescue environments. In *17th Conference of the Canadian Society for Computational Studies of Intelligence*, Lecture Notes in Computer Science, pages 16–30. Springer, 2004.

[12] M. L. Cummings, S. Bruni, S. Mercier, and P. J. Mitchell. Automation architecture for single operator-multiple uav command and control. *Intl. Command and Control Journal*, 1(2):1 – 24, 2007.

[13] M. Johnson, P. J. Feltovich, J. M. Bradshaw, and L. Bunch. Human-robot coordination through dynamic regulation. In *Proc. of the IEEE Intl. Conference on Robotics and Automation*, pages 2159 – 2164, Pasadena, CA, 2008.

[14] D. Pinelle and C. Gutwin. A groupware design framework for loosely coupled workgroups. In *Proc. of the Ninth European Conference on Computer-Supported Cooperative Work*, pages 65 – 82, Paris, France, 18-22 September 2005.

[15] P. M. Fitts, editor. *Human engineering for an effective air-navigation and traffic control system*. Washington, DC: National Academy of Sciences Archives, 1951.

[16] A. Hinze, K. Sachs, and A. Buchmann. Event-based applications and enabling technologies. In *Proc. of the Third ACM Intl. Conference on Distributed Event-Based Systems*, pages 1:1–1:15, 2009.

[17] U. Dayal, A. Buchmann, and D. McCarthy. Rules are objects too: A knowledge model for an active, object-oriented database system. In Klaus Dittrich, editor, *Advances in Object-Oriented Database Systems*, volume 334 of *Lecture Notes in Computer Science*, pages 129–143. Springer, 1988.

[18] S. Chakravarthy and D. Mishra. Snoop: an expressive event specification language for active databases. *IEEE Data and Knowledge Engineering*, 14(10):1 – 26, 1994.

[19] H. Branding, A. Buchmann, T. Kudrass, and J. Zimmermann. Rules in an open system: the REACH Rule System. In *Proc. of the 1st Intl. Workshop on Rules in Database Systems*, pages 111 – 126, Edinburgh, Scotland, September 1993.

[20] Team Hector Darmstadt website: http://www.gkmm.tu-darmstadt.de/rescue/, December 2010.

[21] J. Meyer, P. Schnitzspan, S. Kohlbrecher, K. Petersen, O. Schwahn, M. Andriluka, U. Klingauf, S. Roth, B. Schiele, and O. von Stryk. A semantic world model for urban search and rescue based on heterogeneous sensors. In *RoboCup Symposium, Singapore*, 2010.

[22] Team Darmstadt Dribblers website: http://www.dribblers.de/, December 2010.

 ISBN: 978-1-61208-117-5