DR 5.1: Expectation Management in Shared Control

Ivana Kruijff-Korbayová[‡] (editor), Timea Bagosi[†], Joachim de Greeff[†], Koen Hindriks[†], Walter Kasper[‡], Tina Mioch[*], Chris Rozemuller[†], Jurriaan van Diggelen[*], Mark Neerincx [*,†] and the TRADR consortium

[‡]*DFKI, Saarbrücken, Germany*
[†]*Interactive Intelligence Group, Delft University of Technology, the Netherlands*
[*]*TNO, Soesterberg, The Netherlands*

⟨`ivana.kruijff@dfki.de`⟩

Project, project Id:	EU FP7 TRADR / ICT-60963
Project start date:	Nov 1 2014 (50 months)
Due date of deliverable:	Month 14
Actual submission date:	Feb 23 2015
Lead partner:	DFKI
Revision:	Version: Final
Dissemination level:	PU

We report progress achieved in Year 1 of the TRADR project in WP5: *Persistent models for human-robot teaming.* The reported work concerns teamwork modeling, designing an agent-based team coordination framework and preparing teamwork simulation and experimentation tools, as well as measuring workload for dynamic task allocation and initial investigation of team activity reporting.

Executive Summary

This report presents the progress achieved in Year 1 of the TRADR project in WP5: *Persistent models for human-robot teaming*, addressing Task 5.1: *Expectation Management in Shared Control* leading to Milestone MS5.1.

As the first step in Year 1 we set out to establish a basis for modeling human-robot teaming in TRADR. To this end we designed an agent-based team coordination framework, developed an ontology for modeling human-robot teamwork and investigated the forms and uses of team activity reporting. We made progress on measuring workload for dynamic task allocation. We also prepared teamwork simulation and experimentation tools, that will allow us to investigate various aspects of teamwork in the future.

The proposed model of team level coordination, the teamwork ontology and the description of team activity reports are based on an analysis of the team organizational structure and reporting protocols within a search and rescue mission, as employed by the firefighters. Input for this analysis was obtained through discussions with the end-users and during the TRADR Joint Exercise at the Tremola hospital in Fall 2014.

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, WP5 develops models and algorithms for determining robot behaviour at the (social) team-level. This encompasses 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 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 1 use cases (cf. DR 7.1 of WP7) involve a team consisting of three humans team members (team leader, UGV operator and UAV operator) and two robots (UGV and UAV). The team is performing an initial assessment of an accident site, followed by subsequent information gathering sorties. The use cases provide an abundance of opportunities for teamwork, in particular w.r.t. shared control in navigation, search and/or manipulation. The work carried out in WP5 Year 1 contributes to a better understanding of the specific challenges at hand and starts to address some of them.

1 Tasks, objectives, results

1.1 Planned work

The plan for Year 1 had foreseen WP5 to address *Expectation Management in Shared Control* (Milestone MS5.1). The goal was to develop an account of what expectations, conflicts, and needs for alignment (typically) arise in a human-robot team, focusing on investigating how these occur during shared control between one or more humans, and a single UGV performing navigation and/or mobile manipulation tasks set within a "get&return-to-base" context over multiple sorties during 1 day.

1.2 Actual work performed

As the first step in Year 1 WP5 set out to establish a basis for modeling human-robot teaming in TRADR. This involved several lines of work and was necessary, because explicit modeling of and reasoning at the human-robot team level is a new aspect in TRADR, going beyond what was addressed in NIFTi. Although in NIFTi a team of several humans was using a UGV and a UAV in a collaborative fashion, and several team roles were distinguished, e.g., for the purposes of tactical information display, the robots were not involved in team-level reasoning: it was the human team leader and the operators who provided the team-level reasoning and the linkage between the tasks performed by the robots and by the in-field rescuer(s), respectively.

In order to let the robots become active team members in TRADR we thus first needed to put in place a framework for modeling and management of both taskwork and teamwork and specify the domain models and teamwork strategies. We also needed to equip ourselves with the means to run experiments that would allow us to investigate various aspects of teamwork.

The work WP5 performed in Year 1 therefore comprised the following:

- developing a formal model that allows us to investigate which conditions require coordination of agents to ensure task completion in a team setting
- preparation of a simulation environment for carrying out team-coordination experiments
- developing an ontology for modeling human-robot teamwork in the search and rescue domain
- designing an agent-based framework for coordination of human-robot teaming to manage the roles, objectives, responsibilities and expectations for members of the team

- developing a workflow modeling tool for specifying search and rescue tasks and testing the combinations of level of automation, task-division, user interfaces, communication means, etc. early in the design process
- evaluating and further developing the dynamic task allocation model for adaptive automation introduced in NIFTi
- exploring the forms and uses of synchronous and asynchronous team-activity reporting in search and rescue teams

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

1.2.1 Coordination Requirements of Cooperative Teams

We have started working on the team level coordination by analyzing the team organizational structure within a search and rescue mission, based on the practices used by the firefighters and discussed with them. This discussion has laid the basis for the design of an initial ontology for reasoning about team structure, roles and organization. This facilitates reasoning with role-based social behavior at a team level, is a first step towards reasoning to anticipate behavior within the team, and strategies for team-level interaction. The ontology also supports reasoning about the domain which further enhances prediction of behavior and expectation reasoning about behavior within a search and rescue mission. We also have included the modeling and representation of capabilities of team members and plan to include cognitive task load as an important factor for task allocation and goal setting and reasoning about what can be expected from a team member. We have also started looking into communication and coordination requirements within teamwork. In particular, the work of, e.g. [8] and [34], provides a useful abstract basis for identifying essential assumptions that can be made within the context of a team working on a search and rescue mission. Examples are assumptions about mutual responsiveness and being helpful in a team setting. Requirements for coordinating effectively, however, depend on the specific task setting and complexity which we have analyzed from purely qualitative point of view. Identifying structural coordination requirements and assumptions provides a basis for effective expectation management where team members are able to predict progress during a mission.

The work presented in [45] (Annex Overview 2.1) establishes the ground for investigating how agents can reason about cooperative teamwork in a search and rescue task. Search and rescue tasks are a good abstraction of the coordination problems encountered in search and rescue. The main contribution of the paper is a formal model that also covers some other important aspects of a search and rescue setting such as: online coordination, a partly unobservable environment, irreversible actions and autonomous agents with different roles and abilities. The model is purposely designed to be complementary to the BW4T simulation environment presented in the next section §1.2.2, that we plan to use for future experiments on teamwork in WP5.

The methodology we present in [45] (Annex Overview 2.1) allows us to distinguish between multiple levels of coordination (implicit and explicit) and identify when these are required. For example: an interesting finding is that search and rescue tasks without resource ordering constrains can be solved without communication by using implicit coordination. In future work will use this formal model and methodology in combination with simulation results from BW4T to understand important topics for WP5 such as team resilience and communication failure.

1.2.2 BW4T Simulations

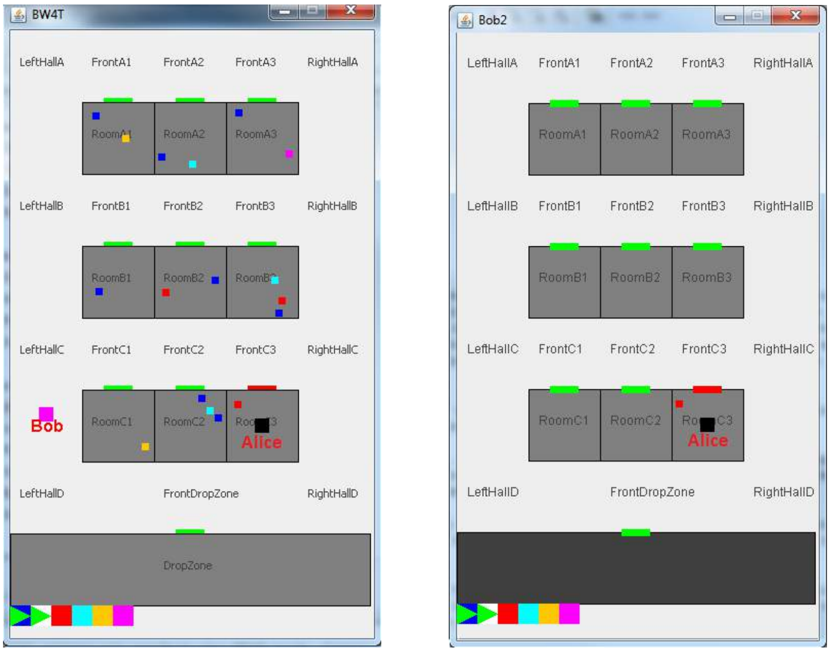
We propose to use the Blocks World for Teams (BW4T) simulation environment [26] for empirical experiments on teamwork strategies in WP5.

The BW4T simulation environment can be used to represent a search and rescue task for agents. It is designed to test online planning and teamwork capabilities of cognitive agents in a dynamical, partly unobservable, environment where actions can be irreversible. This environment has been proposed by Johnson et al. as a testbed for joint activities [26]. BW4T is not limited to virtual agents; it can also be used to test human teamwork abilities. BW4T allows for simulation of teamwork that involves both human participants and robots (as virtual agents). As such it can be used for experiments in which agents act at different levels of autonomy and interact correspondingly with human participants, thus enabling us to study the expectation management.

In BW4T it is the task of agents to collect resources of different types in a certain order. Figure 1a shows this environment with two agents and resources distributed in the rooms. Initially the resources are hidden in rooms and they are observable only for the agent that enters the room. This limited perspective of an agent is shown in figure 1b). When a resource has been found it can be carried to the dropzone where, once correctly dropped, it contributes to complete the task. To efficiently complete the task agents need to communicate about (sub)goals, beliefs and temporal planning. In some cases task completion cannot be guaranteed without communication, because planning mistakes can lead to lost resources.

The results obtained after a simulation can be:

- Successful or unsuccessful task completion as a measure of reliability.
- Time-to-complete, meters traveled or number of plan changes as a measure of efficiency.



(a) Complete information view (b) Agent perspective

Figure 1: The BW4T environment with two agents (Alice and Bob), resources located in rooms (small colored squares), and a partially finished task (sequence on the bottom, triangles indicate task completion). Fig a shows complete information with Bob carrying a pink resource and fig. b shows Alice's limited perspective on this world

- Amount of knowledge collected per agent to measure sharedness of information.
- Number of messages sent/received as a measure of communication efficiency.

Maria Gini's experiments [29, 30, 36] are a good example of the use of BW4T to evaluate teamwork. In early experiments with Jonker et al. she uses BW4T to empirically show the positive effect of sharedness of team mental models on team performance. In a later experiment with Manner she evaluated different strategies to overcome communication limitations.

We propose to do experiments in a similar fashion. Currently we are preparing an experiment to evaluate resilience of agent strategies against communication failure. In this experiment we sabotage communication between agents and we search for reasoning strategies that can help the team discover the errors and recover from them. Results obtained in this environment can be backed up by the formal model we propose in [45] (Annex Overview 2.1) (discussed above in section 1.2.1). In later experiments we would also like to use the human interface in BW4T to test the resulting teamwork models with human participants.

1.2.3 Ontology for Teamwork Modeling

An ontology is a formal representation of knowledge, including concepts, properties and relationships between them. Hence it is a conceptualization of a domain, an area of interest. In the TRADR project, this domain is the real world search and rescue operation, with combined human-robot teamwork. The TRADR ontology consists of three types of entities: concepts, object properties and data properties, and objects or individuals of these entities, such as instances of a class. The ontology structure can be viewed as a graph, with nodes being concepts, individuals or data instances, and properties being the arcs between the nodes.

Communication between agents benefit from sharing the same ontology to represent common knowledge, as the ontological terms constitute the vocabulary to be used in exchanging messages. This way both parties (sender and receiver) will know how to interpret every part of the message, relating it to the common ontology they both are aware of. Thus, all agents, both human and robot, will operate on the shared vocabulary, ensuring semantic interoperability [5]. Furthermore, the use of an ontology allows for the employment of a reasoner which deduces implicit knowledge from the set of axioms and asserted facts of the ontology. Thus, given a set of concepts and properties, inferences can be made about individuals. A full report describing the TRADR Ontology can be found in annex [2] (Annex Overview 2.2).

1.2.4 Agent architecture

The TRADR project envisions the use of robots not as mere tools, but instead aims at developing robots capable of acting as members of flexible teams who collaborate with people [41] in a search and rescue environment [39]. To realize this, TRADR is developing a framework for coordination of human-robot teaming, which is built on agent-based technology [23]. This framework manages the different roles, objectives, responsibilities and expectation for members of the team (which consists of both robots and humans and which may change over different sorties) and allows for conflict resolution and dynamical task-allocation depending on capabilities, task-load and chances of success.

In the TRADR scenario, every member of the team (both humans and robots) will be represented and mediated by an agent, where humans interact with the system via the TRADR Display System (TDS) (cf. DR 3.1 of WP3). Since intelligent agents mediate information processes and the corresponding views of the user interface, they are responsible of visualizing relevant information at a relevant time to the relevant person. Information about the team (i.e. members, members' status, each member's role, commands, eventual messages or warnings) helps to achieve a good collaboration on the team level. A possible addition in this area would be a system that does reasoning about the type of information, and the way information is presented to the user. The intended goal is a smart graphical user interface controlled by the agent, that, taking into account the user's cognitive load, task, the robot's autonomy level, and other relevant factors, can determine the necessary and sufficient amount and type of data to be displayed.

Furthermore, the agents architecture extends towards high-level coordination of the robot team members, tying in with the topics described in section 1.2.2 and section 1.2.1. The full report can be found in annex [1] (Annex Overview 2.3).

1.2.5 Teamwork Workflow Modeling and Testing

Complex systems consisting of humans, robots, tools and organization are often referred to as socio-technical systems. We have been developing ST2, a socio-technical simulation tool that allows non-expert users to test the combination of level of automation, task-division (between humans and machines), user interfaces, communication means, early in the design process. The basis for such an evaluation is a concept of operation, specified in workflow models. Applied to TRADR, this means specifying common USAR tasks such as reconnaissance, victim rescuing, and resource management. ST2 enables the execution of these workflows and in this way simulate the assignment and execution of tasks within a group of task-performers (humans, robots, or agents in TRADR), within a predefined scenario. ST2

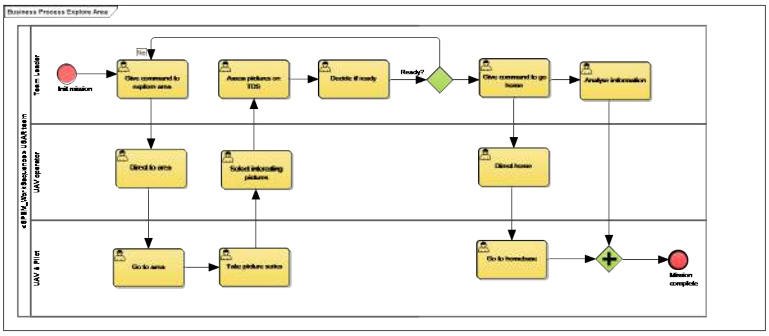


Figure 2: TRADR scenario in BPMN 2.0.

supports both the automatic simulation of a concept of operation, as well as a human-in-the-loop experimentation. Human-subject experimentation may use low-fidelity automatically generated user interfaces, or specifically designed high-fidelity user interfaces. Of particular concern is process flexibility. Because the task environment may change due to unexpected events or unexpected resource availability, ST2 allows a task performer (human or machine) to change the planned tasks or resources at runtime. In a typical human-robot team, not every actor will have the authority to change tasks or perform resource management. In some systems, only the team leader will have these capabilities, and all other actors are instructed to give feedback to the team leader who can then change the team structure if needed. Such systems possess a very limited degree of shared control. In other systems, all team-members can change plans or change resource-task allocations. These systems exhibit a very high degree of shared control. Both extremes will probably not be practical in real world USAR applications. ST2 allows the experimenter to specify teams with different levels of shared control, and autonomy, which can be tested with real users in an early phase of system development.

Example. As an example, we have modelled the workflow of the TRADR scenario in BPMN 2.0 (see Figure 2). This workflow can be executed in ST2 which will then automatically generate low-fidelity interfaces that correspond to the input-variables that are specified in each task. An example of such an interface is presented in Figure 3.

The person who receives this task (in this case "crew member" can execute the task by filling in the requested information and by pressing the complete button. The system will automatically generate the next task for the relevant person who is scheduled next in the workflow. ST2 also allows real time introspection of workflows. For example, by visualizing the currently active process using a special color, see Figure 4.

The person who monitors this workflow (typically the team leader (called

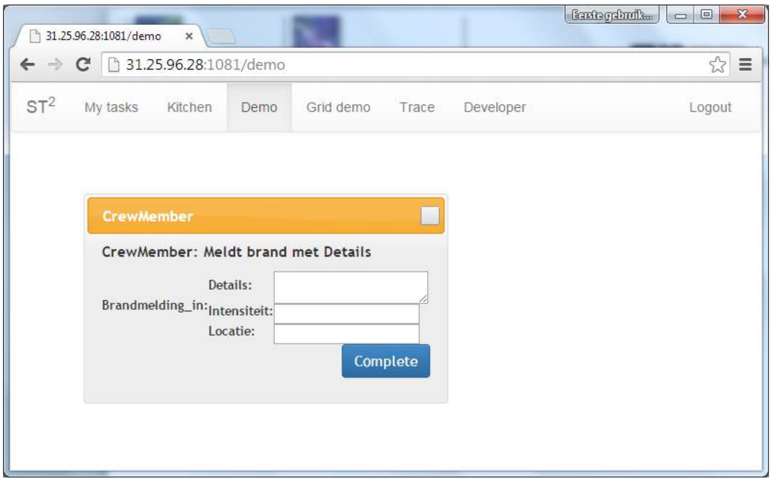


Figure 3: TRADR scenario in BPMN 2.0.

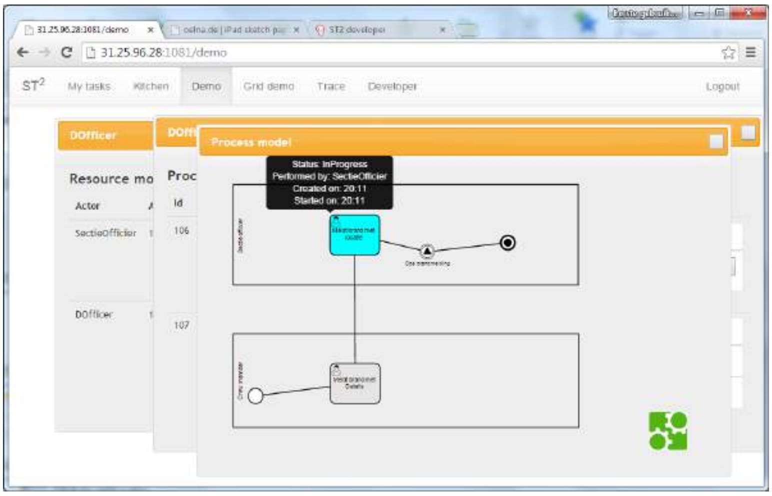


Figure 4: TRADR scenario in BPMN 2.0.

mission commander in NIFTi) can choose the change the workflow real time to deal with some disturbance.

Current state of development. ST2 is available as a web-based tool and is accessible over the internet. It is based on the JBPM workflow tool suite (http://www.jbpm.org/). On top, we have implemented functionality that is needed to perform the desired form of human-in-the-loop simulations.

The current implementation contains the following functionality:

- Workflow editing and execution
- External forms
- Basic Team Observability displays (process and resource)
- Basic Team Directability functionality
- Demonstration screen
- Event logging

Future work. We wish to apply the ST2 tool in TRADR to validate Human-Robot-Teamwork concepts and interfaces with real domain experts, the end-users. Our focus will be on resilient workflow patterns, i.e., reusable patterns of human-machine interaction that can be applied to make a fixed workflow dynamically adjustable. For example, the team leader manually monitors processes and adapts them if necessary (as illustrated in the example earlier). Another resilient workflow pattern would be the use of some automated intelligent process that can recognize suboptimal workflows, and proposes changes if necessary. By modeling these patterns in ST2 we can obtain valuable insights in

- Which user interfaces are needed to make the system work?
- What is a good task division between humans, robots and machines?
- Which way of directing the process flow is required (who can do it, and how)?
- How can robots be incorporated into an existing USAR concept of operation?

1.2.6 Task allocation

In the NIFTi project, we introduced a high-level framework for dynamic task allocation. The framework details how context information can be used to find possible role assignments for actors and to evaluate these role assignments. It also describes the important concepts in context information that influence team performance and can be used to dynamically allocate tasks. The framework was used as a basis for designing a model for adaptive automation, taking into account the cognitive task load of a robot operator and the coordination costs of switching to a new task allocation. In the TRADR project, the model for real-time Cognitive Task Load (CTL) monitoring was refined, implemented and tested. This model distinguishes three load factors that affect operator performance and mental effort: time occupied, level of information processing, and number of task switches. Long duration tests with robot operators demonstrated that the CTL model can contribute, in a non-invasive manner, to estimating an operators cognitive state (see [11] (Annex Overview 2.4).

Based on the two context factors *cognitive task load* and *coordination costs*, the previously introduced dynamic task allocation model for adaptive automation finds the optimal level of autonomy of a robot, separately for all tasks that need to be executed. We evaluated this model in a small experiment. In the TRADR project, this work has been further continued by working out the model in more detail and finalizing the analysis and results of the experiment, as reported in [15] (Annex Overview 2.5).

The dynamic task allocation framework foresees different preference factors that give an indication of how well a task set can be executed by an actor. Possible preference factors are for example cognitive task load, but also emotional state determination. In the following, we will give a short description over our work on both these aspects.

Activity recognition for determination of cognitive task load Based on [40], a cognitive task load model has been developed in the NIFTi project for real-time monitoring and, subsequently, balancing of workload on three factors that affect operator performance and mental effort: time occupied, level of information processing, and number of task set switches. To know the number of task set switches, the model needs to know the current activities and tasks of the actor. In this reporting period, we have started to work on a model for better activity recognition of human team-members (i.e., both the activities of robot operators in the control centre and the activities of the team-members in the field).

At the moment no research specifically on the topic of activity recognition in human-robot rescue teams in urban search and rescue environments exists. Even though activity recognition is already a challenging task in simple environments [50, 33], the characteristics of the urban search and rescue domain provide additional challenges for state-of-the-art activity recognition techniques.

We are developing an activity recognition system (see Figure 5) for the recognition of human activities in a human-robot rescue team. The activity recognition system senses human behavior composed of physical motion,

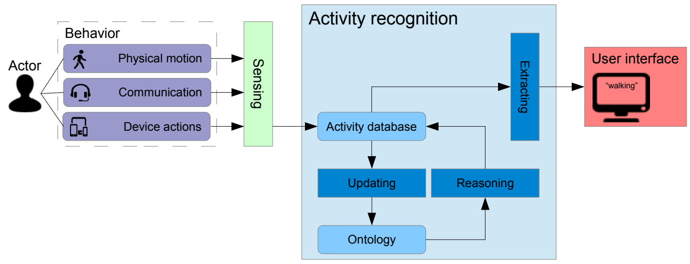


Figure 5: Activity recognition system.

communication acts, and device actions. The behavior is transformed into basic activities and stored in an activity database, with which an ontology, concerning human-robot rescue teams, is updated periodically. Reasoning on the ontology is performed in order to infer complex activities using temporal, and relational connections between basic activities.

Since human-robot rescue teams are hierarchically structured according to role, a large variety of different activities is present. Also, due to the nature of human-robot rescue teams, individual as well as team activities are considered. We will first focus on individual and team activities of the role of robot operator. In the next months, the activity recognition system will be evaluated in realistic scenarios in a virtual environment.

Emotional state determination The appraisal of the cognitive task load of a robot operator is of utmost importance for the current model for adaptive automation. The approach chosen in the NIFTi project computes a cognitive taskload (CTL), which maps the cognitive resources required for a set of tasks, to a numeric value. To improve this approach, we extended it with a component modeling the emotional state (ES) in terms of arousal and valence, from the operator's physiological measurements [31] (Annex Overview 2.6). This approach was evaluated by two experiments. One experiment was set up with young children in a relaxed setting, where we measured heart rate, analyzed video footage with facial activity detecting software and manually annotated smiling and frowning as a type of ideal sensor. Another experiment was set up with adults performing cognitively demanding tasks, where we measured galvanic skin response, heart rate and activity of the corrugator (frowning) and orbicularis oculi (smiling) muscles.

None of the collected measurements were sufficiently accurate to detect a difference between the most and least cognitively demanding or exciting aspects of the sessions. However, when using the manual annotations of smiling, the fuzzy logic model computed valence with an average accuracy of 95%, and a minimal accuracy of 80%. We also discovered a Pierson's correlation between frowning and cognitive taskload of 0.9. Assuming that the future will bring an improvement in accuracy of physiological measuring devices, fuzzy logic offers a simple, transparent, fast way of modeling an emotional state.

Constraint presentation for task allocation support The general framework for task allocation is a general framework that describes possible inputs about the situational state of all team members, their capabilities, and amongst others, the cognitive task load of the team members to allocate tasks as optimal as possible. However, to actually develop an implemented model that takes all these aspects into account, the model has to be further specified in more detail. To provide task allocation desicion support to the team leader in the meantime, we have started working on a system that makes current constraints visible. Constraints might include the current position and tasks of the team members, their capabilities, and other current constraints that might play a role in the decision to allocate a task to them. For now, we are concentrating on the constraints of the robotic team members. Questions we would like to answer regarding this support system are, amongst others, how to present this information to the team leader to best support his decision making process, and which kind of information actually is needed and helpful to make the decisions. In addition, we will further investigate in how far the decisions can be automated (and still be accepted by the human team members).

1.2.7 Team Activity Reporting

USAR teams have a strong hierarchical structure and clearly defined communication ways and methods that are to a high degree also formalized and standardized. Especially briefings (before start of sortie) and debriefings (at the end of a sortie) are important team management activities for establishing shared awareness of the situation and tasks. During the sorties also log book reports are created to document team activities, findings, directives, etc. In the report presented in [32] (Annex Overview 2.7) we investigate the communication and reporting tasks in USAR teams. We develop concepts for supporting such tasks by natural language tools and technologies based on the TRADR architecture.

1.3 Relation to the state-of-the-art

Technical progress of robots for search and rescue is happening, and there is a growing body of relevant research, e.g., the Human-Agent Robot Teamwork (HART) workshops. However theoretical and empirical foundations are lacking for real-world design proposals for integrated, context-sensitive cognitive systems.

Human-Robot Teaming As robots become more sophisticated a tendency has arisen within HRI to perceive them as teammates rather than tools [25, 41]; also in the context of disaster response robotics the importance of robots capable of operating as a (social) team-member has been acknowledged and addressed [14, 39].

Even though in NIFTi multiple robots were employed, they did not necessarily partake on the team-level; each robot was controlled by an individual operator taking orders from the human mission commander. This is similar in a number of other projects, where teams of heterogeneous robots are employed in a collaborative fashion, but it is human operators who provide the linkage between the robots and the human rescue workers, e.g., ICARUS [12], DARIUS [10]. A stronger notion of human-robot collaboration is developed in the alpine rescue project SHERPA [37], employing a metaphor of the human as "busy genius" who collaborates with a group of robots with different capabilities (the "SHERPA animals") towards a common goal.

TRADR is also going beyond an approach in which robots are mere tools, instead aiming at robots with an adaptive level of autonomy (e.g. semi-autonomous navigation, data gathering etc.) as members of flexible teams improving their collaboration over time. To realize this, TRADR is developing a framework for coordination of human-robot teaming, which is built on agent-based technology [23], as described above.

Coordination Requirements of Cooperative Teams Most work on coordination for multi-agent and multi-robot systems deals with improving the performance of these systems. For example, many works focus on improving multi-robot path planning [22]. Although our model is inspired by these problems and incorporates a notion of location and movement, it abstracts from most aspects related to (local) spatial coordination. The model that we have adopted is similar to the "gothru" navigation model of [44] and we refer to this paper for related coordination mechanisms.

Tasks that require cooperation are different from task allocation approaches in multi-agent or multi-robot teams that deal with the assignment of tasks that can be accomplished independently by a single agent. [49] calls the latter tasks weakly-cooperative. We are mainly interested here in what [9, 49] call strongly-cooperative tasks, which require agents to act in concert to achieve the goal, executing tasks that are not trivially serializable [43].

There are many formal approaches many of which are based on logic [13, 17, 18, 53]. Our work differs from this work in providing a model that allows to more explicitly specify decision functions and to investigate related coordination problems. Our model differs from Dec-MDPs [6] and

DCOPs [24] mainly by adopting a more specific notion of state, and a more qualitative approach with a focus on robustness rather than optimization.

Finally, [54] studies coordination requirements in centralized offline multi-agent planning problems. Our model is more general, and, also allows to study incomplete information settings, online decision-making, and decentralized teamwork.

Ontologies for Human-Robot Teaming in USAR Existing ontologies exist for a vast variety of domains, and they are designed and made available for sharing and reuse. The conducted literature review tries to explore the use of ontologies from multiple perspectives that concerns the TRADR project, described in the following.

Modeling the disaster management domain was described by Othman et al. in [42], where they create a unified knowledge metamodel in UML format, based on the analysis of thirty existing models. The authors identify disaster response phases, associated concepts and their semantical meaning. Girardi et al. in [16] introduce GRAMO and ONTODUM, a technique to create and represent domain and user models in ontologies, for multi-agent domain engineering. They break down the domain model into a concept, a goal, a role and an interaction model.

The survey of upper ontologies for situation awareness (SAW) by Baumgartner and Retschitzegger [4] describes an evaluation framework for upper ontologies, taking three different perspectives on the modelling requirements: top-level concepts, SAW-specific concepts and design principles for upper ontologies. The conducted comparison favors the Situation Awareness Assistant (SAWA) [38] ontology in most measured criterias, lacking in, for example, an appropriate approach to represent spatio-temporal information. The BeAware! ontology for situation awareness by Baumgartner et al. [3] takes the SAWA ontology to the next step by addressing the identified lack for spatio-temporal relations, and shows an implemented prototype in the road traffic domain. They identify performance problems that querying temporalized information introduces, and argue that this should be supported by Semantic Web reasoning technologies, such as AllegroGraph. Another situation awareness ontology is developed by Kokar et al. in [35], by capturing Barwise's situation theory into the Situation Theory Ontology (STO). Their purpose is to play the role of a basis for a unifying theory of computer-based situation awareness.

The use of ontologies for autonomous vehicle is described in several articles, from various perspectives. Schlenoff et al. in [47] use ontologies to understand intelligent behavior of both humans and machines used as combat vehicles. They develop an Intelligent Systems (IS) Ontology that provides a set of standard domain concepts for the intelligent vehicle community, and models tactical behaviors expected from an army soldier. The size of the IS Ontology is 489 classes, 213 properties and 2674 instances, and claim to be in growth. Uschold et al. in [51] model an environment of obstacles for supporting autonomous vehicle navigation. One of the major issues they faced was the integration of an obstacle ontology with the ontology of objects in the environment. Schlenoff and Messina present a robot ontology for Urban Search and Rescue (USAR) in [46]. After conducting a thorough requirements analysis for structural, functional and operational capabilities, they use OWL-S to create a service ontology, that describes service requirements from the user (profile), how a service works (model) and how it is used (grounding).

As much as we would like to reuse existing ontologies, based on the presented related works, we can only take inspiration, metamodels or developed standards from multiple different sources, but can not fully adopt a specific one. When creating the TRADR ontology, we have to keep the following properties as high priority in mind: small size, as abstract as it can be, as expressive as it needs to be, problem-oriented and performant.

Workflow modeling Over the last decade, workflow management systems have become mainstream in Information Technology. Many commercial and open source solutions exist which enable specification, management and execution of workflows (among which JBPM which forms the basis of ST2). To use workflow systems to specify human robot teaming, a degree of flexibility and adaptability is needed which is not straightforwardly offered by industry systems. A hot topic in the research community is how to visualize complex workflows in a comprehensive way. Since 2012, a series of scientific workshops has started on this topic, i.e. International Workshop on Theory and Applications of Process Visualization (TAProViz). A system which is designed for process visualization is proviado [7][1], which allows configurable and personalized business process visualizations, to make workflows insightful even when they are very large and complex. Another approach to reduce the visual complexity of large workflows is to use abstractions which describe sequences of tasks at a functional level. We have experimented with this this approach in earlier work [52]. This research provides a useful way to develop our process monitoring interfaces in ST2 further. Another topic which is relevant to ST2 is to realize *flexible* workflows. A lot of attention is currently being given to this topic in the research community. In [48], different types of approaches to realize process flexibility are described. In the future, we intend to extend ST2 with some of these techniques (e.g. event-based workflows).

Workload Modeling First, current practices show that workload is still a major bottleneck in human-robot interaction. Robot operators spend up to

[1] Proviado: http://www.uni-ulm.de/?id=9873

60% of their time attempting to establish and maintain situation awareness, and can cause "human" errors, such as driving mistakes and misses of victims [20], [21], [19]. We refined and implemented models of Cognitive Task Load (CTL) and Emotional State (ES) for adapting the individual (often mobile) interface to the workload and present the workload distribution to the team [11]. In this way, we could, real-time, analyse workload distributions in complex and realistic disaster response settings automatically. Depending on the Cognitive Task Load and the emotional state, the current task allocation of the team members (e.g., robot operators), can be adapted to optimize team performance.

Team Activity Reporting Team management in USAR teams depends to a high degree on team members reporting and documenting their activities and findings to establish and maintain shared knowledge about the progress of a mission. Robots in USAR teams typically are controlled by trained operators and do not participate directly in the information flow within the team. [27] propose verbalization of the robot internal action plans and goals to improve on the predictability of autonomous robot behavior and thereby its trustworthiness. [28] present an approach towards natural language generation of postmission debriefing reports from autonomous robot missions. Both approaches target mainly the information needs of the operators. But team members in USAR teams have various and differing information needs, depending on their role in the team. TRADR investigates how these different information needs and reporting requirements can be supported by the system. It must also take into account that in USAR teams each agent acts as information source and that fusion of information from the various channels and sources should be supported.

2 Annexes

2.1 Rozemuller, Chris, Hindriks, Koen and Neerincx, Mark (2015), "Coordination Requirements of Cooperative Teams"

Bibliography Chris Rozemuller, Koen V. Hindriks and Mark A. Neerincx (2015), "Coordination Requirements of Cooperative Teams". Submitted paper. Delft University of Technology, Delft, the Netherlands.

Abstract Coordination of cooperative multi-agent systems or teams has been recognized as important for improving task performance. A team of agents may reduce the time to complete a task, for example. It is less clear, however, which tasks require coordination to ensure task completion, and, if so, what kind of coordination mechanism is required. Our approach to identify team coordination requirements is based on a simple but systematic methodology to explore the need for a coordination mechanism. The idea is to first establish whether a single agent can ensure task completion and determine whether multiple of these agents would run into issues that require coordination. If so, conditions are identified that establish whether implicit coordination is sufficient or whether there is a need for explicit coordination. We introduce a formal task model and distinguish between no, implicit and explicit coordination mechanisms. This model is used to study which mechanisms guarantee task completion. It allows us to prove some intuitions expressed in the literature such as that a simple foraging task does not require coordination. We also show that explicit coordination is only required if performing an action depends on whether other actions have been performed or not.

Relation to WP This paper describes a theoretical framework for the modeling of coordination requirements of cooperative teams; as such it contributes directly to T5.1.

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

2.2 Bagosi, Timea (2015), "The TRADR Ontology"

Bibliography Timea Bagosi (2015), "The TRADR Ontology". Unpublished technical report. Delft University of Technology, Delft, the Netherlands.

Abstract In this report we describe the TRADR Ontology, its importance in the TRADR project, as well its three main parts: the domain, the team and the user model. After a detailed description of some most important concepts, the usage of ontological rules is introduced and explained. Lastly, the continuous development and future integration plans are presented.

Relation to WP This report describes the TRADR Ontology, which is a fundamental component of sharing knowledge within the TRADR system, us such contributing to shared controll.

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

2.3 Bagosi, Timea (2015), "Agent architecture"

Bibliography Timea Bagosi (2015), "'Agent architecture". Unpublished technical report. Delft University of Technology, Delft, the Netherlands.

Abstract In this report we describe the intelligent agents that orchestrate the high-level data, and control the TRADR visualization system in a smart, personalized manner. We present the distribution of information into one database shared between all agents, and several private databases for each agent. We illustrate the importance of agents in the project by examples of decisions taken by the agents to take over some tasks or to help humans.

Relation to WP This report describes the agent architecture, which directly contributes to T5.1.

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

2.4 Colin et al. (2014), "Real Time Modeling of the Cognitive Load of an Urban Search and Rescue Robot Operator"

Bibliography Colin, T.R., Mioch, T., Smets, N.J.J.M. and Neerincx, M.A.. "Real Time Modeling of the Cognitive Load of an Urban Search and Rescue Robot Operator" In *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN2014)*. 25-29th August Edinburgh, Scotland, UK.

Abstract Urban Search And Rescue (USAR) robots are used to find and save victims in the wake of disasters such as earthquakes or terrorist attacks. The operators of these robots are affected by high cognitive load; this hinders effective robot usage. This paper presents a cognitive task load model for real-time monitoring and, subsequently, balancing of workload on three factors that affect operator performance and mental effort: time occupied, level of information processing, and number of task switches. To test an implementation of the model, five participants drove a shape-shifting USAR robot, accumulating over 16 hours of driving time in the course of 485 USAR missions with varying objectives and difficulty. An accuracy of 69 % was obtained for discrimination between low and high cognitive load; higher accuracy was measured for discrimination between extreme cognitive loads. This demonstrates that such a model can contribute, in a non-invasive manner, to estimating an operators cognitive state. Several ways to further improve accuracy are discussed, based on additional experimental results.

Relation to WP The cognitive load modeling addressed in this work can be used for task allocation in a team and thus directly contributes to T5.1.

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

2.5 Giele et al. (2015), "Dynamic Task Allocation for Human-Robot Teams"

Bibliography Giele, Tinka R. A., Mioch, Tina, Neerincx, Mark A., and Meyer, John-Jules. "Dynamic Task Allocation for Human-Robot Teams." In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence (ICAART 2015).* Lisbon, Portugal, January 2015.

Abstract Artificial agents, such as robots, are increasingly deployed for teamwork in dynamic, high-demand environments. This paper presents a framework, which applies context information to establish task (re)allocations that improve human-robot team's performance. Based on the framework, a model for adaptive automation was designed that takes the cognitive task load (CTL) of a human team member and the coordination costs of switching to a new task allocation into account. Based on these two context factors, it tries to optimize the level of autonomy of a robot for each task. The model was instantiated for a single human agent cooperating with a single robot in the urban search and rescue domain. A first experiment provided encouraging results: the cognitive task load of participants mostly reacted to the model as intended. Recommendations for improving the model are provided, such as adding more context information.

Relation to WP By addressing dynamic task allocation this work directly contributes to T5.1.

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

2.6 Kaiser (2014), "The Prediction of an Emotional State through Physiological Measurements and its Influence on Performance"

Bibliography Mira Kaiser. "The Prediction of an Emotional State through Physiological Measurements and its Influence on Performance." Master's thesis, Computing Science, Utrecht University, August 2014.

Abstract In some branches of work, such as operating rescue robots during a disaster, a badly adjusted workload can result in errors with devastating consequences. If we could estimate the workload the operator is experiencing, we could better assign the amount of work among the operators, and so optimize performance. Existing approaches compute a cognitive taskload (CTL), which maps the cognitive resources required for a set of tasks, to a numeric value. To improve this approach, we wish to extend it with a component modeling the emotional state (ES) in terms of arousal and valence, from the operator's physiological measurements.

One experiment was set up with young children in a relaxed setting, where we measured heart rate, analyzed video footage with facial activity detecting software and manually annotated smiling and frowning as a type of ideal sensor. Another experiment was set up with adults performing cognitively demanding tasks, where we measured galvanic skin response, heart rate and activity of the corrugator (frowning) and orbicularis oculi (smiling) muscles.

None of the collected measurements were sufficiently accurate to detect a difference between the most and least cognitively demanding or exciting aspects of the sessions. However, when using the manual annotations of smiling, the fuzzy logic model computed valence with an average accuracy of 95%, and a minimal accuracy of 80%. We also discovered a Pierson's correlation between frowning and cognitive taskload of 0.9. Assuming that the future will bring an improvement in accuracy of physiological measuring devices, fuzzy logic offers a simple, transparent, fast way of modeling an emotional state.

Relation to WP This work contributes to T5.1 by investigating the possibility of improving cognitive load modeling by adding emotional state modeling.

Availablity Unrestricted. Available for download: `http://dspace.library.uu.nl/bitstream/handle/1874/298568/Eindverslag.pdf?sequence=2`

2.7 Kasper and Kruijff-Korbayová (2015), "Communication and Reporting in USAR-Teams"

Bibliography Walter Kasper and Ivana Kruijff-Korbayová, "Communication and Reporting in USAR-Teams". Unpublished Technical Report, DFKI GmbH, Saarbrücken, January 2015

Abstract In this document we analyze communication structures within USAR teams that are important for creating shared knowledge about situations and activities during USAR missions, thus enabling team work and supporting the decision making processes of individual team members and at team-level. Missions can involve multiple sorties. Therefore not only communication during an individual sortie but also communication and information exchange between sorties and before and after them have to be taken into account. We investigate especially the possible use and usefulness of natural language technologies to support especially human team members in their communication needs. We will derive concepts for supporting team communication in USAR missions especially by natural language technologies.

Relation to WP By investigating the forms and uses of synchronous and asynchronous reports in search and rescue teamwork this work directly contributes to T5.1.

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

References

[1] T. Bagosi. Agent architecture. Technical report, Delft University of Technology, 2015. Unpublished.

[2] T. Bagosi. The TRADR Ontology. Technical report, Delft University of Technology, 2015. Unpublished.

[3] N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, and W. Schwinger. Beaware! situation awareness, the ontology-driven way. *Data & Knowledge Engineering*, 69(11):1181–1193, 2010.

[4] N. Baumgartner and W. Retschitzegger. A survey of upper ontologies for situation awareness. In *In Proceedings of the 4th IASTED International Conference on Knowledge Sharing and Collaborative Engineering*, pages 1–9, 2006.

[5] J. Bermejo-Alonso, R. Sanz, and I. Lopez. A survey on ontologies for agents. *From Theory to Practice. ASLab-ICEA*, 2006.

[6] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[7] R. Bobrik, T. Bauer, and M. Reichert. Proviado–personalized and configurable visualizations of business processes. In *E-Commerce and Web Technologies*, pages 61–71. Springer, 2006.

[8] M. E. Bratman. Shared cooperative activity. *The Philosophical Review*, 101(2):327–341, 1992.

[9] R. Brown and J. Jennings. A pusher/steerer model for strongly cooperative mobile robot manipulation. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 562–568 vol.3, Aug 1995.

[10] P. Chrobocinski, N. Zotos, E. Makri, C. Stergiopoulos, and G. Bogdos. Darius project: Deployable sar integrated chain with unmanned systems. In *Telecommunications and Multimedia (TEMU), 2012 International Conference on*, pages 220–226, July 2012.

[11] T. R. Colin, N. J. J. M. Smets, T. Mioch, and M. A. Neerincx. Real time modeling of the mental workload of an urban search and rescue robot operator. In *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, UK, August 2014.

[12] G. De Cubber, D. Doroftei, D. Serrano, K. Chintamani, R. Sabino, and S. Ourevitch. The eu-icarus project: Developing assistive robotic tools for search and rescue operations. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–4, Oct 2013.

[13] B. Dunin-Keplicz and R. Verbrugge. Collective intentions. *Fundam. Inf.*, 51(3):271–295, Mar. 2002.

[14] T. Fincannon, L. E. Barnes, R. R. Murphy, and D. L. Riddle. Evidence of the need for social intelligence in rescue robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1089–1095. IEEE, 2004.

[15] T. R. A. Giele, T. Mioch, M. A. Neerincx, and J.-J. Meyer. Dynamic task allocation for human-robot teams. In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence*, Lisbon, Portugal, January 2015.

[16] R. Girardi and C. Faria. An ontology-based technique for the specification of domain and user models in multi-agent domain engineering. *CLEI electronic journal*, 7(1):7, 2004.

[17] V. Goranko and W. Jamroga. Comparing semantics of logics for multi-agent systems. In *Information, Interaction and Agency*, pages 77–116. Springer Netherlands, 2005.

[18] J. Grant, S. Kraus, and D. Perlis. A logic-based model of intention formation and action for multi-agent subcontracting. *Artificial Intelligence*, 163(2):163 – 201, 2005.

[19] L. Gunawan. *Crowdsourced disaster response for effective mapping and wayfinding.* PhD-thesis Delft University of Technology, 2013.

[20] L. T. Gunawan, H. Alers, W. Brinkman, and M. A. Neerincx. Distributed collaborative situation-map making for disaster response. *Interacting with Computers*, 23(4):308–316, 2011.

[21] L. T. Gunawan, S. Fitrianie, Z. Yang, W. Brinkman, and M. A. Neerincx. Travelthrough: A participatory-based guidance system for traveling through disaster areas. In *Proceeding of Computer Human Interaction, CHI 2012*, pages 241–250, Austin, Texas, USA, May 2012.

[22] Y. Guo and L. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 2612–2619, 2002.

[23] K. Hindriks. Programming rational agents in GOAL. In A. El Fallah Seghrouchni, J. Dix, M. Dastani, and R. H. Bordini, editors, *Multi-Agent Programming: Languages, Tools and Applications*, pages 119–157. Springer, 2009.

[24] K. Hirayama and M. Yokoo. Distributed partial constraint satisfaction problem. In G. Smolka, editor, *Principles and Practice of Constraint Programming-CP97*, volume 1330 of *Lecture Notes in Computer Science*, pages 222–236. Springer Berlin Heidelberg, 1997.

[25] G. Hoffman and C. Breazeal. Collaboration in human-robot teams. In *Proc. of the AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, USA*, 2004.

[26] 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.

[27] 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.

[28] 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.

[29] C. M. Jonker, B. van Riemsdijk, I. van de Kieft, and M. L. Gini. Towards measuring sharedness of team mental models by compositional means. In *AAAI Fall Symposium: Robot-Human Teamwork in Dynamic Adverse Environment*, 2011.

[30] C. M. Jonker, M. B. Van Riemsdijk, I. C. Van De Kieft, and M. Gini. Compositionality of team mental models in relation to sharedness and team performance. In *Advanced Research in Applied Artificial Intelligence*, pages 242–251. Springer, 2012.

[31] M. Kaiser. The prediction of an emotional state through physiological measurements and its influence on performance. Master's thesis, Computing Science, Utrecht University, August 2014.

[32] W. Kasper and I. Kruijff-Korbayova. Communication and Reporting in USAR-Teams. Technical report, DFKI GmbH, 2015. Unpublished.

[33] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1):1–14, 2010.

[34] G. Klein, D. Woods, J. Bradshaw, R. Hoffman, and P. Feltovich. Ten challenges for making automation a "team player" in joint human-agent activity. *Intelligent Systems, IEEE*, 19(6):91–95, Nov 2004.

[35] M. M. Kokar, C. J. Matheus, and K. Baclawski. Ontology-based situation awareness. *Information fusion*, 10(1):83–98, 2009.

[36] M. D. Manner and M. Gini. Improving agent team performance through helper agents. In *Cognitive Agents for Virtual Environments*, pages 89–105. Springer, 2013.

[37] L. Marconi, S. Leutenegger, S. Lynen, M. Burri, R. Naldi, and C. Melchiorri. Ground and aerial robots as an aid to alpine search and rescue: Initial sherpa outcomes. In *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pages 1–2, Oct 2013.

[38] C. J. Matheus, M. M. Kokar, K. Baclawski, J. A. Letkowski, C. Call, M. L. Hinman, J. J. Salerno, and D. M. Boulware. Sawa: An assistant for higher-level fusion and situation awareness. In *Defense and Security*, pages 75–85. International Society for Optics and Photonics, 2005.

[39] R. R. Murphy. Human-robot interaction in rescue robotics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):138–153, 2004.

[40] M. A. Neerincx. Cognitive task load design: model, methods and examples. In *Handbook of Cognitive Task Design*, pages 283–305. Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[41] S. Ososky, D. Schuster, F. Jentsch, S. Fiore, R. Shumaker, C. Lebiere, U. Kurup, J. Oh, and A. Stentz. The importance of shared mental models and shared situation awareness for transforming robots from tools to teammates. In *SPIE Defense, Security, and Sensing*, pages 838710–838710. International Society for Optics and Photonics, 2012.

[42] S. H. Othman, G. Beydoun, and V. Sugumaran. Development and validation of a disaster management metamodel (dmm). *Information Processing & Management*, 50(2):235–271, 2014.

[43] 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.

[44] A. Rosenfeld, G. A. Kaminka, S. Kraus, and O. Shehory. A study of mechanisms for improving robotic group performance. *Artificial Intelligence*, 172(6-7):633–655, 2008.

[45] C. G. Rozemuller, K. V. Hindriks, and M. Neerincx. Coordination requirements of cooperative teams. In *Autonomous Agents and Multiagent Systems (AAMAS)*. 2015. Under review.

[46] C. Schlenoff and E. Messina. A robot ontology for urban search and rescue. In *Proceedings of the 2005 ACM workshop on Research in knowledge representation for autonomous systems*, pages 27–34. ACM, 2005.

[47] C. Schlenoff, R. Washington, T. Barbera, and C. Manteuffel. A standard intelligent system ontology. In *Defense and Security*, pages 46–56. International Society for Optics and Photonics, 2005.

[48] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. van der Aalst. Process flexibility: A survey of contemporary approaches. In *Advances in Enterprise Engineering I*, pages 16–30. Springer, 2008.

[49] F. Tang and L. E. Parker. A complete methodology for generating multi-robot task solutions using asymtre-d and market-based task allocation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3351–3358, April 2007.

[50] E. M. Tapia. Activity recognition in the home setting using simple and ubiquitous sensors. 2003.

[51] M. Uschold, R. Provine, S. Smith, C. Schlenoff, and S. Balikirsky. Ontologies for world modeling in autonomous vehicles. In *18Th International Joint Conference on Artificial Intelligence, IJCAI*, volume 3, 2003.

[52] J. van Diggelen, W. Post, M. Rakhorst, R. Plasmeijer, and W. van Staal. Using process-oriented interfaces for solving the automation paradox in highly automated navy vessels. In *Active Media Technology*, pages 442–452. Springer, 2014.

[53] M. Wooldridge and N. Jennings. The cooperative problem-solving process. *Journal of Logic and Computation*, 9(4):563–592, 1999.

[54] Y. Zhang and S. Kambhampati. A formal analysis of required cooperation in multi-agent planning. *CoRR*, abs/1404.5643, 2014.

Real Time Modeling of the Cognitive Load of an Urban Search and Rescue Robot Operator

Thomas R. Colin[1], Nanja J.J.M. Smets[2], Tina Mioch[2] and Mark A. Neerincx[2,3]

Abstract— **Urban Search And Rescue (USAR) robots are used to find and save victims in the wake of disasters such as earthquakes or terrorist attacks. The operators of these robots are affected by high cognitive load; this hinders effective robot usage. This paper presents a cognitive task load model for real-time monitoring and, subsequently, balancing of workload on three factors that affect operator performance and mental effort: time occupied, level of information processing, and number of task switches. To test an implementation of the model, five participants drove a shape-shifting USAR robot, accumulating over 16 hours of driving time in the course of 485 USAR missions with varying objectives and difficulty. An accuracy of 69% was obtained for discrimination between low and high cognitive load; higher accuracy was measured for discrimination between extreme cognitive loads. This demonstrates that such a model can contribute, in a non-invasive manner, to estimating an operator's cognitive state. Several ways to further improve accuracy are discussed, based on additional experimental results.**

I. INTRODUCTION

(Semi-)autonomous robots are becoming increasingly capable due to the combined progress of robotics and artificial intelligence. But in most operation fields, the achievement of full autonomy is not yet a realistic goal. In the meantime, humans and robots must cooperate harmoniously to accomplish complex tasks.

We conjecture that USAR robots, if they are to cooperate efficiently with humans, should dynamically attune their behavior to their human partners; like team members, rather than like tools. Communications and the allocation of tasks between robots and their human operators should be adjusted depending on the level of workload of the humans operators, and on the abilities of the robots [1].

This is based on the observation of human teams. Indeed, human teammates adapt their communication patterns and the division of tasks to different levels of workload [2]. This is made possible by a shared understanding of the situation, including the status of the members of the team [3]; it results in adequate performance under increased levels of workload [3] [4].

A robot that "knows" the cognitive state of its human operator can inform the rest of the team; change the way it interacts with the operator (focusing on the most essential information); and, more ambitiously, dynamically change its level of autonomy to adapt to the workload of its operator [1].

This article presents a model of the cognitive load of an USAR robot operator, based on real-time observation of his tasks. A prototype of this model was implemented and a user study was set-up to determine the viability of the approach. During that user study, data was also collected to find directions for improving the model. Particular attention was paid to the characteristics of the tasks.

II. BACKGROUND

A. Cognitive load in USAR

The use of robots in search and rescue is heavily taxing on human operators. Indeed, operators spend up to 60% of their time attempting to establish and maintain situation awareness [5], leaving little time for the operation of the robot or the visual search for victims. This causes cognitive fatigue, and errors such as not seeing victims or crashing the robot [6].

There is considerable ongoing work to reduce the cognitive load of USAR robot operators, e.g. by improving the interface, devising novel control schemes, or providing help in the form of on-board AI[1]. Our focus is in estimating cognitive load, so that it can be managed using some of the techniques mentioned above.

B. Measuring cognitive load

Three approaches to measuring cognitive load are found in the literature: self-reported, physiological, and based on behavioral observation.

- *Physiological:* Efficient physiological assessment of cognitive load tends to be invasive (this is especially the case of EEG). The less invasive methods (such as heart rate or skin conductivity) tend to be fine-tailored to specific tasks (consider e.g. [8] or [9]), and are difficult to apply to real world activities, in which the biological signs of cognitive load are affected by noise from physical workload and emotions [10].
- *Self-reported:* Standardized tests are used, such as NASA-TLX [11] or RSME [12]. Self-reported measures can be impractical for real-time use in the field, as the induced overhead could reduce performance on the main task. However, such measures can be used for the experimental validation of models based on physiological data or behavior.

[1]Thomas R. Colin is with Utrecht University, Drift 10, 3512 BS Utrecht, The Netherlands thomas.r.colin@gmail.com

[2]Nanja J.J.M. Smets, Tina Mioch and Mark A. Neerincx are with TNO Soesterberg, Kampweg 5, 3769 DE Soesterberg, The Netherlands nanja.smets@tno.nl, tina.mioch@tno.nl, mark.neerincx@tno.nl

[3]Mark A. Neerincx is also with TU Delft, Stevinweg 1, 2628 CN Delft, The Netherlands

[1]For a review of the literature on mitigating strategies for high operator workload in USAR, see [7].

- *Behavior-based:* Examples include the Cognitive Task Load model (CTL-model) [13] and cognitive architectures such as ACT-R [14]. This approach has yet to be employed for USAR robots. However, it has been used in other domains, such as modeling car drivers and naval ship operators [1]. Some AI systems aim to detect the level of fatigue [15], the level of distraction [16], or even the short-term goals of a driver or operator [17].

The first two approaches, physiological and self-reported, do not seem well-suited for an USAR environment in the current state of the art [18]. This leads us to the choice of a behavior-based strategy. Among those, we prefer the CTL-model since it produces a high-level and possibly more robust result, compared to the fine cognitive modeling of ACT-R. Self-reported estimates were used for training and validating the model.

C. The CTL-model: overview

Neerincx [13] and Grootjen et al. [19] developed the CTL-model and applied it to the measurement of the cognitive load and performance of naval ship operators. They make use of three metrics affecting the cognitive state of the operator: level of information processing or LIP, time occupied or TO and task set switching or TSS. The relationship between these metrics and the cognitive state of the robot operator is shown in figure 1: for example, when all metrics are high, the operator is in "overload"; when all metrics are low, in "underload". When time occupied is high while task set switching and level of information processing are low, the operator faces vigilance issues: that is, he may fail to maintain the necessary level of focus on his task.

Neerincx et al. [20] showed that the cognitive states recognized by this model (optimal, underload, overload, cognitive lock-up and vigilance) affected performance, and that the model could be successfully used to estimate performance, with an accuracy of 86% in a laboratory setup, and 74% aboard a naval ship.

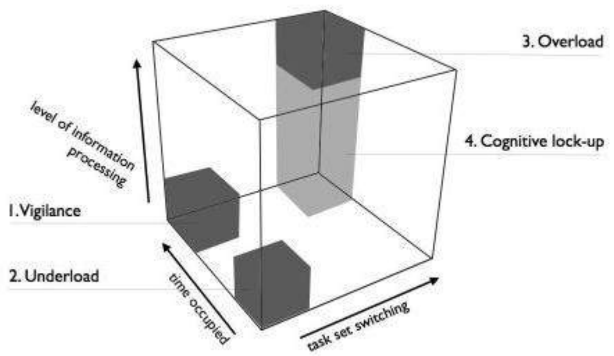


Fig. 1. Cognitive load space, with 4 problem regions (Neerincx, 2003)

Using only three metrics makes it easier for an expert to rate the tasks, and allows for automatic estimation of the cognitive state. Each metric is a measure of a specific characteristic of the user's tasks:

- LIP denotes the complexity of a task for the operator. The rating is given between 1 and 3, based on Rasmussen's distinction between skill- (perceptual-motor), rule- (procedural) and knowledge- (problem-solving) based tasks [21]. LIP is low for perceptual-motor tasks and high for problem-solving tasks.
- TO measures the proportion of time during which an operator is active (not resting).
- TSS measures the number of task switches or interruptions.

III. Cognitive load estimation module

This section describes how the CTL-model was refined and implemented to detect tasks, estimate the metrics, and produce an estimation of the cognitive state of the operator for a specific robot platform (cf. figure 2).

Below, we refer to the robot, GUI laptop and software as "the system", and to the specific module used to predict cognitive load as "the cognitive load module", or "the module" for short.

Fig. 2. The "NIFTi" robot platform

A. Refinement of the CTL model

To take into account changes in the application domain (USAR instead of naval ship operation), the following changes were made to the CTL model:

- "Time Occupied" (TO) is also used to address the difference between tasks that require occasional cognitive processing (e.g. every 0.5 seconds) and tasks that require constant cognitive processing.
- When piloting an USAR robot, some tasks make use of the same mental representations and of the same skills; whereas other tasks are radically different. To take this into account, the impact of each task set switch depended on the difference between the domains associated with each task.

B. Task model

A task analysis was conducted based on video footage of operators using the same USAR robot in an unrelated experiment [18], acquiring the following task knowledge:

- The set of tasks performed by the operators;

- The characteristics of each of these tasks: level of information processing (LIP), time occupied (TO), and domains of the task (used to estimate TSS);
- The relationship between observable events and the tasks.

Results from the task analysis were used to parameterize the cognitive load prediction module.

C. Implementation and integration

Figure 3 shows the architecture of the cognitive load module and its integration within a robotic system (a description of the experimental system is given in section IV). In short, the software modules of the robot (implemented as ROS nodes) transmit events to the cognitive load prediction module, which interprets them to produce, every second, an estimation of the cognitive load of the operator. The cognitive load estimation is then accessible by all software modules, e.g. for adaptation of the robot's behavior.

In the next paragraphs, we give a more detailed description of the internals of this module: task detection, metrics calculation and classification.

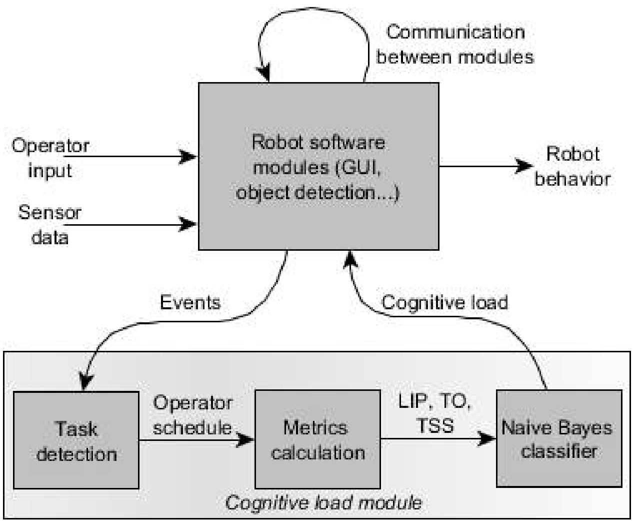


Fig. 3. Architecture and integration of the Cognitive load module

1) Task detection: Events are produced in real-time by the robot's other software modules; this allows us to defer most of the task detection to pre-existing modules on the robot. The events consist of a type, e.g. "driving start" or 'driving stop", and a timestamp. They are buffered by the cognitive load module; every second, new events are processed by a task detection sub-module, which translates events into a schedule. The task detection sub-module is implemented as a rule system: its rules can be parameterized based on the task model (e.g., a "driving" task begins with a "driving start" event and ends with a "driving stop" event).

The output of task detection is the schedule of the operator over the observed period.

2) Metrics calculation: The CTL metrics "LIP" and "TO" can then be estimated based on weighted averages over time, whereas TSS depends on task domains:

$$LIP = \sum_{i=1}^{n} \frac{l_i d_i}{f}$$

$$TO = \sum_{i=1}^{n} \frac{o_i d_i}{f}$$

$$TSS = \sum_{i=1}^{n-1} \frac{|D_i \ominus D_{i+1}|}{|D_i \cup D_{i+1}|}$$

Where the tasks $1, ...n$ are in chronological order, and for each task i, o_i is the time occupied, l_i the level of information processing, and D_i the set of information domains (as obtained from the operator schedule); d_i is the duration of the task in seconds, as obtained from task detection. f is the total duration of the operator's schedule, set to 120 seconds.

The three values for LIP, MO and TSS are then forwarded to a naive Bayes classifier.

3) Naive Bayes classifier: Once trained (see section IV for an example), the Naive Bayes classifier can output the most likely cognitive load of the operator, between the classes"overload", "normal", "underload", "cognitive lock-up", and "vigilance". In the case of the experiment described below, a simplified classifier discriminates only between low and high cognitive load (corresponding roughly to extended overload and underload classes).

Note that the features (LIP, TO, TSS) are all real-valued. To take this into account in the classifier, an assumption is made that the distribution of each feature, with respect to each class, is Gaussian. This constrains learning but limits over-fitting, making the accuracy results more reliable.

IV. User Study: Method

A user study was conducted to evaluate the accuracy of the module. We also investigated cognitive load variations relative to the objectives of each mission, in order to find ways to improve the model.

A. Participants

Eight students from 21 to 28 years old participated in the user study. Each participant spent eight hours on the study. Because some events (used for task detection) were not satisfactorily logged, we could only estimate the accuracy of the cognitive load module for five participants; other results (sections V-A and V-C) include all eight participants.

B. Materials and set-up

A 65m^2 maze was set up, which served as a testing area for driving the robot. The maze contained 15 walls, whose positions were changed every three missions to prevent participants from learning the maze configuration. There were ten obstacles (pallets, metal bars and ramps) on which the robot could climb. Participants could not see the robot during the study, and relied instead on information from the robot's sensors (camera and laser). The cognitive load module was not run in real time during the user study; instead, events were collected for later processing, allowing for distributing the collected data into training and test sets (via cross-validation).

C. Objectives

Each mission lasted two minutes, and consisted of a combination of a driving objective (among three) and a distractor objective (among seven). These numerous objectives, and more numerous combinations of objectives (21 in total), together with the variations from mission to mission, created a wide range of situations for the cognitive load module to learn from, and on which to test it.

1) Driving objectives: Participants completed three types of driving objectives. Each objective corresponded to an aspect of an USAR mission:

- *Explore.* Signs were placed on the maze walls; participants were scored on the proportion of signs they could successfully find and identify.
- *Goto.* Participants were shown a map (not including the positions of the walls) indicating their current location and their target position. Participants were scored on the proportion of the distance they could cover.
- *Assess.* A dummy victim was marked in specific spots with different colors, to indicate whether it was hurt or not. Participants were scored based on how many of these markers they could identify, while driving the robot around the dummy victim.

2) Distractors: In addition, participants had to complete objectives simulating non-driving activities (e.g. talking with a teammate, remembering information about the mission, reasoning about other events around them). These activities constitute an important part of USAR operations [22]. They were simulated by arithmetic operations, presented in different manners:

- *Visual.* Additions were shown on a laptop screen. A countdown indicated the remaining time to solve them. There were three difficulty levels for this distractor, based on changes in frequency, number of digits, and the presence or absence of carries.
- *Audio.* Simple arithmetic operations were presented verbally at regular intervals. Participants also had to remember the result of the previous operation in order to answer correctly. As for the above, this distractor had three difficulty levels.
- *None.* Participants were awarded a perfect distractor score and could focus entirely on the driving task.

Distractors were scored based on the proportion of correct answers.

D. Procedure

Participants were trained on all aspects of the user study during a 100-minutes training session. They learnt how to operate the robot, and performed six missions with objectives of varying difficulty. Once trained, participants completed four additional sessions, each lasting 95 minutes and taking place on a different day. Each session consisted of 16 missions (± 2); each participant completed 60 missions (± 3) in total. After each mission, participants were informed of their total score, based on the multiplication of their driving and distractor scores.

E. Data analysis

1) Data collected: During the mission themselves, events were recorded - these include most interactions with the robot's interface (such as modifying the robot's shape, driving, moving the camera), as well as events related to the distractors (presentation of a question, answer, end of countdown). In total, about 31,000 events were recorded.

Participants rated the level of mental effort exerted on a scale from 0 to 150 (RSME [12]) immediately after each mission.

In addition, at the beginning and end of each session, they responded to a questionnaire about their state of mind (arousal and valence, using the SAM [23]); at the end of each session, they filled in a more in-depth questionnaire about how they experienced the session. These additional questionnaires were used for the manipulation check.

2) Model accuracy: The cognitive load per mission (normalized per participant) was categorized as either 'low' or 'high', based on the reported workload, thus creating a dataset combining, for each mission, the set of recorded events (used to extract the features LIP, TO and TSS) with the cognitive load ('low' or 'high'). Training and test sets were created by ten-fold cross-validation. Accuracy is the percentage of correctly classified test set missions.

It was especially important for the module to discriminate properly betwen extreme levels of cognitive load (i.e., not to rate its operator as being in underload when he suffers from overload), as this could lead to counterproductive reactions. Besides, higher accuracy was expected for these more sharply defined cases. To verify this, the accuracy of the model was also tested on the subset of missions corresponding to the top and bottom 33% reported mental effort, per participant.

3) Statistical analysis: The statistical analysis was conducted to discover and analyze cognitive load variations not captured by the module. Specifically, the variability associated with the type of objectives was measured (distractors and driving objectives), in order to verify whether other factors could influence cognitive load. Differences between participants were also measured, but the effects and number of participants were too small to provide significant correlations; the results are not reported here.

V. Results

A. Manipulation check

The complexity of the user study warranted conducting an extensive manipulation check.

1) Participant cognitive state:

- Fatigue: there was a significant but weak negative correlation ($c = 0.15$, $p = 0.003$) between the mission number within a session and cognitive load. No strong or significant effect was measured on performance ($c = 0.06$, $p = 0.3$). Participants did not report a loss of arousal (from 3.6/5 before to 3.8/5 after a session).
- Learning: between the first and last sessions, a limited increase in scores was measured (13% for driving

objectives, 4.4% for distractors), but reported mental effort remained stable (-1.6%).
- Motivation: motivation remained high throughout the study (participants rated *'I gave my best'* at 3.6/4 on average).

Furthermore, no extreme values were found for any of the participants, for any of these verifications.

2) Naive Bayes classification: Recall that the naive Bayes classifier operated over continuous data for LIP, TO, and TSS, fitting Gaussian distributions on training data in order to classify new values. To make sure that the Gaussian assumption was correct, it was verified that the three metrics respected a roughly Gaussian distribution.

B. Cognitive load estimation accuracy

TABLE I

MODULE ACCURACY BY PARTICIPANT

Participant	1	2	3	4	5	All*
Accuracy (%)	63	71	71	47	79	69

*: Using a normalized dataset

The accuracy of the module can be found in Table I; this represents the capacity of the model to match the participants' reported mental effort. In a follow up test, the model was run only on extreme values: the 33% lowest and 33% highest reported cognitive loads. Accuracy was slightly higher for participants 1 and 3 (65% and 72%), and much higher for participants 2, 4 and 5 (90%, 65%, and 89% respectively).

C. Statistical analysis

Recall that for each mission, participants had to complete both a driving objective (Explore, Goto or Assess) and a distractor objective. Figure 4 shows the average reported cognitive load for each objective. Note that distractor objectives correspond to strong variations of cognitive load, whereas driving objectives do not. This is despite the varying difficulty of driving objectives (measured based on the average score across participants per objective: 46, 62 and 71 for "Explore", "Go-to" and "Assess").

These results were confirmed when correlating participant cognitive load with the average difficulty of an objective (determined by the average score obtained across participants for that objective): indeed, a significant correlation was found for distractor objectives ($c = 0.43$, $p < 10^{-4}$), but no significant effects were observed for driving objectives. Furthermore, there is a significant correlation ($c = -0.20$, $p < 10^{-3}$) between distractor difficulty and driving score, but not between driving objective difficulty and distractor score.

VI. DISCUSSION

A. Cognitive load module validation

The module achieved an overall accuracy of 69% (cf. table I). It could make educated guesses, despite the multiple factors in play: task type and difficulty, participant personality and skill, and the variations caused by unpredictable participant mistakes. It could thus allow a robot or another intelligent system to make more informed decisions with regards to the cognitive state of its operator.

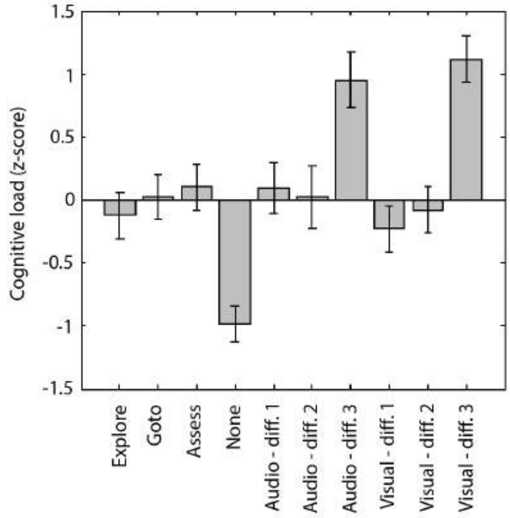


Fig. 4. Cognitive load by objective (z-score), ±95% confidence

However, the measured accuracy may be too low to directly share the information with human team members, who probably cannot handle noisy numerical input under the high-demand situations of USAR. Also, the user study did not reproduce all aspects of USAR missions which could contribute to an altered cognitive state, such as fatigue and intense emotions.

Below, we discuss potential ways to improve the cognitive load module, based on the statistical results of the user study.

B. Statistical analysis

There were important variations of cognitive load depending on the distractor. However, although driving was a difficult task occasioning high variations of cognitive load and score, the *type* of driving objective had very limited effects on cognitive load. Furthermore, the distractor type affected the driving objective score, but the reverse was not true. These were unexpected results.

A tentative explanation is that participants were affected by the urgency of the distractors (which had to be dealt with within a few seconds) compared to the driving objectives (which could be managed within the two minutes that a mission lasted). Participants seemingly always attempted to complete the distractors. Despite a scoring scheme (multiplication of the scores for each objective) that could not yield good scores when focusing on only one objective, participants did not decide to ignore a distractor for a period of time in order to focus on driving. This sometimes resulted in panicking participants and low scores.

We conclude that urgency/time pressure likely had a major impact on cognitive load. Although the TSS metric is indirectly affected by task urgency, the effect should be modeled more directly. For further discussion of the impact of time pressure for cognitive state, see [24] or [25].

C. Other possible improvements

1) Cognitive load module output: The implementation described here outputs the most likely cognitive load, rather than the probability of each possible cognitive load state. This could lead to drastic decisions based on incomplete and uncertain information, for a domain in which high error rates are unacceptable. A probabilistic output ("the operator is in state S_1 with probability $P(S_1)$, S_2 with probability $P(S_2)$") is not difficult to generate from a naive Bayes classifier, and would fix this problem.

2) Task detection: This preliminary user study was conducted with a prototype, including rule-based task detection. If a cognitive load module is to be efficient in real world conditions, it has to rely on a robust task detection algorithm, most likely involving probabilistic estimation of the tasks based on pattern detection. This would be another step towards dealing with the complexity and unpredictability inherent to real-life USAR missions [6].

VII. Conclusion

The CTL-model makes use of a production rule system to determine an operator's schedule, and of the CTL-model metrics and a naive Bayes classifier to estimate the operator's cognitive state.

Applied to an USAR robot, in a simulated USAR environment, a prototype implementation was able to predict the cognitive load of participants with 69% accuracy; accuracy was higher when considering only extreme cognitive loads. This is encouraging; furthermore, a statistical analysis hinted that accuracy could be improved by taking into account the urgency of the tasks.

We believe cognitive modeling is essential to improve human-robot team-work; this user study suggests that real-time modeling is feasible, and proposed directions and insight for future implementations.

VIII. Acknowledgments

This research is supported by the EU-FP7 ICT Programmes, Project 247870 (NIFTi) and Project 609763 (TRADR).

References

[1] T. E. de Greef, H. F. R. Arciszewski, and M. A. Neerincx, "Adaptive automation based on an object-oriented task model: Implementation and evaluation in a realistic c2 environment," *Journal of Cognitive Engineering and Decision Making*, vol. 4, no. 2, pp. 152–173, 2010.

[2] J. Urban, J. Weaver, C. Bowers, and L. Rhodenizer, "Effects of workload and structure on team processes and performance: Implications for complex team decision making," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 38, no. 2, pp. 300–310, 1996.

[3] E. Entin and D. Serfaty, "Adaptive team coordination," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 41, no. 2, pp. 312–325, 1999.

[4] C. Porter, C. Gogus, and R. Yu, "When does teamwork translate into improved team performance? a resource allocation perspective," *Small Group Research*, vol. 41, no. 2, pp. 221–248, 2010.

[5] R. R. Murphy and J. L. Burke, "Up from the rubble: Lessons learned about hri from search and rescue," in *Proceedings of the 49th Annual Meetings of the Human Factors and Ergonomics Society. 2005*, pp. 437–2005, 2005.

[6] J. Casper and R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 33, no. 3, pp. 367–385, 2003.

[7] Y. Liu and G. Nejat, "Robotic urban search and rescue: A survey from the control perspective," *Journal of Intelligent and Robotic Systems*, pp. 1–19, 2013.

[8] B. Cinaz, R. La Marca, B. Arnrich, G. Tröster, and M. Macedo, "Monitoring of mental workload levels," in *Proceedings of IADIS eHealth Conference*, 2010.

[9] A. Henelius, K. Hirvonen, A. Holm, J. Korpela, and K. Muller, "Mental workload classification using heart rate metrics," in *Engineering in Medicine and Biology Society*, pp. 1836–1839, IEEE, 2009.

[10] M. Grootjen, M. A. Neerincx, J. C. M. van Weert, and K. P. Truong, "Measuring cognitive task load on a naval ship: implications of a real world environment," in *Foundations of Augmented Cognition*, pp. 147–156, Springer Berlin Heidelberg, 2007.

[11] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Human mental workload*, vol. 1, no. 3, pp. 139–183, 1988.

[12] F. Zijlstra, *Efficiency in work behaviour: A design approach for modern tools(Ph. D. Thesis).* PhD thesis, Delft University of Technology, 1993.

[13] M. A. Neerincx, "Cognitive task load design: model, methods and examples," *Handbook of Cognitive Task Design*, pp. 283–305, 2003.

[14] J. R. Anderson, M. Matessa, and C. Lebiere, "Act-r: A theory of higher level cognition and its relation to visual attention," *Human-Computer Interaction*, vol. 12, no. 4, pp. 439–462, 1997.

[15] R. Bittner, K. Hána, L. Poušek, P. Smrka, P. Schreib, and P. Vysokỳ, "Detecting of fatigue states of a car driver," *Medical Data Analysis*, pp. 123–126, 2000.

[16] J. Pohl, W. Birk, and L. Westervall, "A driver-distraction-based lane-keeping assistance system," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 4, pp. 541–552, 2007.

[17] D. Salvucci, E. Boer, and A. Liu, "Toward an integrated model of driver behavior in cognitive architecture," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1779, no. -1, pp. 9–16, 2001.

[18] T. Mioch, T. R. A. Giele, N. J. J. M. Smets, and M. A. Neerincx, "Measuring emotions of robot operators in urban search and rescue missions," in *Proceedings of the 31st European Conference on Cognitive Ergonomics*, EACE, 2013.

[19] M. Grootjen, M. A. Neerincx, and J. A. Veltman, "Cognitive task load in a naval ship control centre: from identification to prediction," *Ergonomics*, vol. 49, no. 12-13, pp. 1238–1264, 2006.

[20] M. A. Neerincx, S. Kennedie, M. Grootjen, and F. Grootjen, "Modeling the cognitive task load and performance of naval operators," in *5th International Conference on Augmented Cognition*, pp. 260–269, 2009.

[21] J. Rasmussen, "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-13, no. 3, pp. 257–266, 1983.

[22] J. Burke, R. Murphy, M. Coovert, and D. Riddle, "Moonlight in miami: Field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise," *Human–Computer Interaction*, vol. 19, no. 1-2, pp. 85–116, 2004.

[23] M. Bradley and P. Lang, "Measuring emotion: the self-assessment manikin and the semantic differential," *Journal of behavior therapy and experimental psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.

[24] N. Moray, M. I. Dessouky, B. A. Kijowski, and R. Adapathya, "Strategic behavior, workload, and performance in task scheduling," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 33, no. 6, pp. 607–629, 1991.

[25] Y. Kushleyeva, D. D. Salvucci, and F. J. Lee, "Deciding when to switch tasks in time-critical multitasking," *Cognitive Systems Research*, vol. 6, no. 1, pp. 41–49, 2005.

Dynamic Task Allocation for Human-Robot Teams

Tinka R.A. Giele[1,2], Tina Mioch[2], Mark A. Neerincx[2] and John-Jules C. Meyer[1]
[1] *Utrecht University, Drift 10, 3512 BS Utrecht, The Netherlands*
[2] *TNO, Kampweg 5, 3769 DE Soesterberg, The Netherlands*
tinka_g@hotmail.com, {tina.mioch, mark.neerincx}@tno.nl, j.j.c.meyer@uu.nl

Keywords: Human-robot interaction, Agent cooperation.

Abstract: Artificial agents, such as robots, are increasingly deployed for teamwork in dynamic, high-demand environments. This paper presents a framework, which applies context information to establish task (re)allocations that improve human-robot team's performance. Based on the framework, a model for adaptive automation was designed that takes the cognitive task load (CTL) of a human team member and the coordination costs of switching to a new task allocation into account. Based on these two context factors, it tries to optimize the level of autonomy of a robot for each task. The model was instantiated for a single human agent cooperating with a single robot in the urban search and rescue domain. A first experiment provided encouraging results: the cognitive task load of participants mostly reacted to the model as intended. Recommendations for improving the model are provided, such as adding more context information.

1 INTRODUCTION

Teams are groups consisting of two or more actors that set out to achieve a joint goal. A good task allocation is crucial for team performance, especially when teams have to cope with high-demand situations (e.g., at disaster responses). Task allocation should be flexible: when an environment is dynamic or states of team members change, reallocating tasks could be beneficial for team performance (Brannick et al., 1997). Making a (human) team member responsible for dynamically allocating tasks, causes extra workload (Barnes et al., 2008). To avoid this, tasks should be reallocated automatically. Such allocation is important for mixed human-robot teams (Burke et al., 2004), for example rescue teams including an robot to explore terrains unsafe for humans. The dynamic allocation of tasks to human or robot is called adaptive automation, distinguishing intermediate levels of autonomy for each task in a joint effort to complete the task. An example is way-point navigation, in which the operator sets the way-points and the robot drives along them. Recent research shows that dynamically adapting autonomy levels of robots could help optimizing team performance, when this process is automated (Calhoun et al., 2012).

An important challenge in adaptive automation is deciding when to change the level of autonomy of the robot, and to which level. This can be done based on the cognitive task load of the operator (Neerincx, 2003), as cognitive task load has an influence on performance (Neerincx et al., 2009). In addition, cognitive task load itself is influenced by changing levels of automation, as the level of autonomy and operator task load are inversely correlated if other factors remain stable (Steinfeld et al., 2006). This does not hold for the relation between autonomy levels and operator performance. Setting robot autonomy very high might cause human-out-of-the-loop problems, whereas setting autonomy very low might cause task overload for the operator; both decrease performance.

This study, first, aims at the design and formalization of a general dynamic task allocation framework that specifies concepts and their effect on team performance, which can be used to dynamically allocate tasks. Subsequently, this framework is used to design a practical model for adaptive automation, based on cognitive task load. Finally, the model is instantiated for an experimental setting in the urban search and rescue domain for a first validation of the model.

2 BACKGROUND

Team Performance. Team performance is a measure of how well a common goal is achieved. Early frameworks describing team performance commonly follow the Input-Process-Output structure. For example, McGrath (McGrath, 1964) describes three input concepts: individual level factors (e.g. cognitive ability), group level factors (e.g. communication) and environmental factors (e.g. resource availability, task difficulty). These factors are input for the team's interaction processes; the output concept is team performance. This framework has some downsides. Feedback loops are excluded, e.g., team performance itself cannot serve as an input for interaction processes. Also, the Input-Process-Output structure suggests linear progression, but interactions between various inputs and processes or between different processes are also possible (Ilgen et al., 2005). Outside McGrath's framework, a vast amount of research has focused on the numerous factors that influence individual performance (Matthew et al., 2000), for example cognitive task load (Neerincx, 2003).

Dynamic Task Allocation. Dynamic task allocation benefits team performance (Brannick et al., 1997), it can be effectuated in numerous ways. First, responsibility can be distributed, or it can be centralized. Distributed responsibility for dynamic task allocation has the disadvantage that it causes extra workload for (human) team members (Barnes et al., 2008). Disadvantages of centralized coordination are that it might be unfeasible to implement for very large teams, and that task reallocations need to be clearly communicated to the team members. Second, Inagaki (Inagaki, 2003) argues that a dynamic form of comparison allocation is the best strategy for task allocation. Comparison allocation means tasks are allocated based on capabilities of actors.

Adaptive Automation. Traditionally, tasks in mixed human-robot teams are allocated either fully to a human or fully to a robot, e.g. based on a list of static human versus robot capabilities (Fitts et al., 1951). This way of allocating tasks has the problem that it is overly coarse. In addition, static task allocation is insufficient for dynamic environments, as capabilities needed for a task could change (Inagaki, 2003). Adaptive task allocation addresses these issues.

Numerous studies have shown the positive effects of dynamic task allocation via adaptive automation in single human-single robot teams, e.g., improved performance, enhanced situation awareness and reduced cognitive workload (Greef et al., 2010), (Bailey et al., 2006), (Calhoun et al., 2012). A few studies have looked at adaptive automation in the context of single human-multiple robot teams (Parasuraman et al., 2009), (Kidwell et al., 2012). In these studies however, only the level of autonomy of a single robot or of a separate system on a *single* task was adapted.

Different techniques for triggering reallocation are possible, for example techniques based on performance (Calhoun et al., 2012), psycho-physiological measures (Bailey et al., 2006), operator cognition (Hilburn et al., 1993), environment (Moray et al., 2000) or hybrid techniques (Greef et al., 2010). However, not all tasks allow for real-time performance measurement, psycho-physiological measures are not suitable for all settings, and environment-based techniques in isolation fail to capture changing states of team members. Hybrid techniques are more robust as multiple factors can be used (Greef et al., 2010). Only a limited amount of studies have used hybrid techniques (Greef et al., 2010).

Cognitive Task Load. An important factor for dynamic task allocation in teams, operating in high-demand situations, is cognitive task load (CTL) (Guzzo et al., 1995). A model of CTL was proposed by Neerincx (Neerincx, 2003). The model describes how task characteristics are of influence on individual performance and mental effort. CTL can be described as a function over three metrics. The *time occupied* is the amount of time a person spends performing a task, the number of *task-set switches* is the number of times that a person has to switch between different tasks. The *level of information processing* is the type of cognitive processes required by recent tasks. When the values for the three metrics fall into a certain range (corresponding to a certain region in CTL-space), the operator is diagnosed to be in a certain mental state, i.e., vigilance, underload, overload, and cognitive lock-up. Being in such a state has a negative influence on performance. The CTL model has been experimentally validated in the naval domain (Neerincx et al., 2009).

3 DYNAMIC TASK ALLOCATION FRAMEWORK

Dynamic task allocation can be seen as optimizing a utility (evaluation) function. Firstly, possible role assignments are generated from context information. Role assignments are a combination of a robot and a set of tasks this robot could execute. These role assignments are then evaluated using context information relevant to how well the robot is able to execute the set of tasks. Secondly, an optimization algorithm is applied, which finds the collection of options which has the highest utility and allocates every task to a robot. This collection of options is a task allocation (Gerkey and Matarić, 2004).

This approach has some limitations. The utility of a robot-task pair is assumed not to be influenced by other tasks the robot might be doing. Also, this analysis does not include mixed human-robot teams.

More importantly, multi-robot task allocation problems are reduced to optimization problems, but some important steps that are needed to realize this reduction are underspecified: generating the feasible role assignments and how to evaluate these. Our framework builds on Gerkey and Matarić's analysis, and improves it on these aspects. We specifically address the issues of option generation and utility calculation. Once we have dealt with these issues, we reduce the task allocation problem to the set-partitioning problem (SPP). Although the SPP is strongly NP-hard, it has been studied extensively and many heuristic algorithms that give good approximations have been developed (Gerkey and Matarić, 2004).

An overview of the proposed framework is shown in Figure 1. Three categories of factors that influence task allocation (individual, environmental, and task factors) are represented by the three input concepts in the top of the figure.

Task models represents task factors: $S_{\mathcal{T}}(T,t)$ where $S_{\mathcal{T}}$ is a name of a property or state, T is a task and t is a time point. Task models contains functions from a specific task to its properties (static) and states (dynamic) at a certain point in time. Examples include location and resource requirement.

Environment models represent environmental factors: $S_{\mathcal{E}}(E,t)$ where E is an environment. Environment models are functions that describe states and properties of the environment that are dependent on the location and possibly the time (e.g. resource availability and weather conditions).

Actor models represent individual factors. Actor models are functions that describe for each actor their relevant abilities and states, associated with a certain point in time: $S_{\mathcal{A}}(A,t)$ where A is an actor. Abilities are static, for example IQ, personality traits and skills. The dynamic counterpart of actor abilities are actor states, for example emotion, location and fatigue. An important influence on task allocation is the cost caused by the reallocation of tasks (Barnes et al., 2008); for that reason, our framework includes a feedback loop for the task allocation itself (denoted by the dashed arrow). The current task allocation itself thus is an actor state.

Some factors influencing task allocation can only be described by combining factors from the categories mentioned above. These factors are represented by the concept of *situation models* in our framework: $S_I(\langle A,\mathcal{T}\rangle,t)$ where $\mathcal{T}$ is a set of tasks. Situation model functions are always described using functions from actor models, environment models and/or task factors. An example is the distance between an actor and a task, a function that is described using both actor location and task location.

To come to an optimal task allocation, three processes are identified, namely *option generation and pruning*, *utility calculation*, and *determining the optimal task allocation* (see colored boxes in Figure 1).

The first process is *option generation and pruning.* An option is a actor-task set combination, $O = \langle A,\mathcal{T}\rangle$. Options are generated from the *set of actors* (input) and the *set of tasks* (input). Then, *restrictive factors* are used to prune the set of possibilities. For example, an actor might lack the proper sensors to execute a task.

The second process is *utility calculation.* For this process, *preference factors* are used. Preference factors give an indication of how well the task set can be executed by the actor. For example, if an actor has been assigned a single, but difficult task, he might do better on this task than if he has also been assigned to do several other tasks. All actor-task set combinations are mapped to a utility value using some function that combines the outcomes of all the preference factors.

The final process is *determining the optimal task allocation*. With the utility function and the set of possible actor-task set pairs, we can use a SPP solving algorithm (Gerkey and Matarić, 2004) to arrive at the best task allocation for a specific time.

Solving the task allocation problem by using the SPP introduces the assumption that all tasks need to be allocated to an actor. This excludes scenarios where it might not be possible or preferable to allocate all tasks. We relieve this assumption by introducing a placeholder for tasks that are not executed, a dummy actor. Tasks allocated to the dummy actor are not executed. We can now model mandatory tasks by defining a restrictive factor that prunes role assignments that assign the dummy actor to mandatory tasks. Also, the costs of not executing certain tasks can be easily modeled using a preference factor, since the set of tasks that are not executed is the set of tasks assigned to the dummy actor.

4 MODEL FOR ADAPTIVE AUTOMATION

In adaptive automation, tasks are dynamically allocated at a specific level of autonomy. Based on the framework, we build the model by defining the factors to be included as influence on adaptive automation. As argued in Section 2, cognitive task load is a good candidate as it affects performance and is influenced by the tasks an actor has. Specifically, it is likely to be influenced by at which level of autonomy an allocated task is. We will include the predicted cognitive task

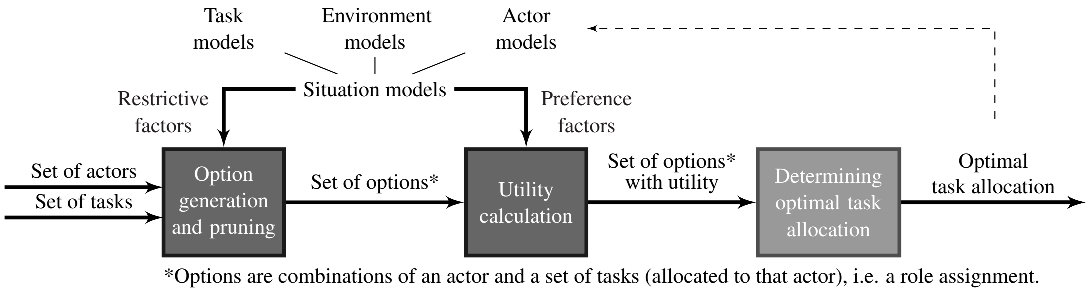


Figure 1: Overview of the proposed framework. Boxes denote processes, arrows represent flow of information. Opposite to Gerkey and Matarić's (Gerkey and Matarić, 2004) focus, we focus on the process of pruning generated options and calculating utility of options (darker boxes) and less on the process of optimization (lighter box).

load of an actor on a set of tasks as a preference factor in our model. Cognitive task load encompasses the metric *task switching*. We define this metric to only cover task switches that are *not* caused by task reallocations, but only by an actor switching between tasks he is both assigned to (for example switching between driving and looking around while exploring an area). We define costs that *are* caused by task reallocations as coordination costs and include this as a separate preference factor. Team performance could benefit from an actor switching between different (levels of autonomy of) tasks if it reduces the negative effect on performance of the cognitive state he is in, but only if the coordination costs do not outweigh the cost of the negative effect on performance of the cognitive state (Inagaki, 2003).

Levels of Autonomy. Tasks that have multiple possible levels of automation are replaced in the task model by a separate version of the task for each different level of autonomy, T becomes $\{T^1, T^2, ..., T^k\}$. Tasks at intermediate levels of autonomy (for example way-point driving) are divided into two subtasks, one for an operator (setting way-points) and one for a robot (driving along the way-points). The separate versions all need to be described in terms of task state concepts. The same task at several different levels of autonomy can be modeled as several mutually exclusive subtasks. All but one of the mutually exclusive tasks (which could consist of two subtasks) should be forcibly allocated to the dummy actor, ensuring a task is only allocated at a single level of autonomy to a real actor.

Cognitive Task Load. We use the predicted CTL level of an actor on a task set to help decide how well this task set is suited to be executed by the actor (relative to other tasks sets). All three metrics of CTL are situation state concepts, they are some function over an option (actor-task set), using the properties of the tasks in the task set. Using the three metrics, we can estimate whether the CTL level of an actor will be in a problem region given a set of tasks. Task allocations that keep actors out of CTL problem regions should be preferred. Timing is also an important aspect in CTL. The longer a person's CTL is in a problem region, the more negative the effect on performance will be. Typically, vigilance and underload problems occur only after some time (900 seconds), while overload and cognitive lock-up problems can occur even if the CTL has only been in the problem region for a short time (300 seconds) (Neerincx, 2003). Cognitive task load as proposed by Neerincx only makes sense in the context of humans, not for robots. For example, robots cannot suffer from vigilance problems if they are bored, because generally robots cannot be bored.

The formal description of the preference concept CTL can be seen in Equation 1. Preference based on CTL ranges from 1 (most preferred) to 0 (least preferred). The 'isHuman' function describes whether an actor is a human, the 'cognitiveState' functions describe whether an actor is in a certain cognitive state and the 'cognitiveStatePast' functions describe for how long (seconds) an actor has been in a certain cognitive state.

The first line of the equation describes that preference of a actor-task set pair based on CTL is 1 if the actor is not human or the actor's CTL is not is a problem region. The second to fifth line describe the preference to be in between 0.7/0.5 and 0.2/0, depending on how long an actor has been in the corresponding problem region (preference decreasing faster for overload and cognitive lock-up as they can occur faster than other problem states). As cognitive lockup is slightly less problematic than the other states the person can be in, the preference associated therewith is set somewhat higher.

Coordination Costs. The coordination costs have to take into account two aspects of switching between tasks, namely how much attention is needed to switch to a new task set, and how often task reallocations take place. The first aspect covers how much atten-

$$\mathrm{ctl}_I(\langle A,\mathcal{T}\rangle,t) = \begin{cases} 1 & \text{if } \neg\mathrm{isHuman}_{\mathcal{A}}(A,t) \text{ or } \mathrm{neutral}_I(\langle A,\mathcal{T}\rangle,t) \\ 0.7 - 0.5*(\min(300,\mathrm{cognitiveLockUpPast}_{\mathcal{A}}(A,t))/300) & \text{if } \mathrm{cognitiveLockUp}_I(\langle A,\mathcal{T}\rangle,t) \\ 0.5 - 0.5*(\min(300,\mathrm{overloadPast}_{\mathcal{A}}(A,t))/300) & \text{if } \mathrm{overload}_I(\langle A,\mathcal{T}\rangle,t) \\ 0.5 - 0.5*(\min(900,\mathrm{vigilancePast}_{\mathcal{A}}(A,t))/900) & \text{if } \mathrm{vigilance}_I(\langle A,\mathcal{T}\rangle,t) \\ 0.5 - 0.5*(\min(900,\mathrm{underloadPast}_{\mathcal{A}}(A,t))/900) & \text{otherwise (if } \mathrm{underload}_I(\langle A,\mathcal{T}\rangle,t)) \end{cases}$$

Eq. 1: Formal description of the preference concept CTL. The function $\min(x,y)$ returns the lesser of its two arguments. All parameters used here and in other formulas are based on relevant literature and were tweaked using data from pilot studies.

tion is needed to switch to a new task set. The formal description of this aspect is seen in Equation 2. If the task set of an actor does not change, there are no coordination costs, which is preferable (fourth line of Eq. 2). If a task gets assigned to an actor that was not previously assigned to this actor at all, this has a relatively high cost (first line). If a task gets assigned to an actor that *was* previously assigned to this actor, but at a different level of autonomy, there are two scenarios. The level of autonomy of a robot could increase, in this case the coordination costs for the human actor are small (third line). If the level of autonomy of a robot decreases, the cost is a bit higher as the human actor has increased responsibilities (second line).

The second aspect that coordination costs have to take into account is how often task reallocations take place. Changing the level of autonomy too often could cause extra workload (Inagaki, 2003). The formal description of this aspect is seen in Equation 3. The first line describes that there is no effect if the last task reallocation is more than 300 seconds ago or if the task was already assigned to the actor at the same autonomy level. The second line describes that a task reallocation in the last 300 seconds gives a penalty to the preference (the longer ago, the smaller the penalty).

The full preference function for coordination costs is seen in Equation 4. It defines preference based on coordination costs of a actor-task set pair to be the average preference based on coordination costs for all separate tasks in the task set.

Utility Function. The utility function maps role assignments at a certain point in time to their utility. The utility of a role assignment is some combination of all preference concepts, in this case the preference based on CTL and the preference based on coordination costs (CC). Team performance benefits from an actor switching between different (levels of autonomy of) tasks if the the negative effect on performance of the cognitive state he is in outweighs the costs of switching. The utility of a role assignment thus is the preference of the role assignment based on CTL minus the coordination costs. The preference concept CC is high if the coordination costs are low (because this is preferred) and vice versa. Therefore the utility of a role assignment is the addition of the two preference concepts CTL and CC. We define that the lowest utility equals 0 and the highest utility equals 1. To fit this range, we scale the sum of the preference concepts CTL and CC (which also both range from 0 to 1) by dividing it by two. More formally, the utility of a role assignment (an option) $O = \langle A, \mathcal{T}\rangle$ at time t is: $\mathrm{utility}(O,t) = (\mathrm{ctl}_I(O,t) + \mathrm{cc}_I(O,t))/2$

5 EXPERIMENT

An experiment was set-up to test if the model reallocates tasks at the right moment and if it chooses the appropriate reallocations. We instantiated the model to be used for a single operator-single robot team in the urban search and rescue domain. This involved specifying tasks, possible levels of autonomy of these tasks and task properties. Furthermore, we used an existing model that calculates CTL specifically for the urban search and rescue domain (Colin et al., 2014).

Experimental Method. Twelve participants (aged 21 to 38) completed three fifteen minute sessions and one participant performed a single session. Participants were given the role of robot operator and asked to execute a typical urban search and rescue task. The task was to explore a virtual office building with a virtual robot after an earthquake, and to map the situation in the building. This was done by navigating the robot through the building and adding findings (large obstacles and victims) to a tactical map, a screen shot of the interface is seen in Figure 2. Sometimes information appeared on the map (e.g., “We think there are two people in this room.”). As there might be victims in the building that are in need of medical attention, participants were told to hurry. The tasks were allocated to the participants by the task allocation model: the optimal level of autonomy for the robot, as calculated by the model, was chosen. Four tasks were specified: navigation, obstacle recognition & avoidance, victim recognition and information processing. The level of autonomy of the robot could change separately for each of these four tasks. During task execution, the CTL of the participant was calculated. When the CTL was in a problem region, the task allocation model was run. If the task allocation model deter-

$$\mathrm{cc}_{\mathrm{attention}}(\langle A, T^v\rangle, t) = \begin{cases} 0 & \text{if } \neg\exists w : T^w \in \mathrm{currentTasks}_{\mathcal{A}}(A,t) \\ 0.2 & \text{if } \exists w : T^w \in \mathrm{currentTasks}_{\mathcal{A}}(A,t) \wedge v < w \\ 0.5 & \text{if } \exists w : T^w \in \mathrm{currentTasks}_{\mathcal{A}}(A,t) \wedge v > w \\ 1 & \text{otherwise (if } \exists w : T^w \in \mathrm{currentTasks}_{\mathcal{A}}(A,t) \wedge v = w \text{)} \end{cases}$$

Eq. 2: The function describing preference based on how much attention is needed for switching between tasks. The 'current-Tasks' function describes the set of tasks currently allocated to an actor.

$$\mathrm{cc}_{\mathrm{time}}(\langle A, T^v\rangle, t) =$$

$$\begin{cases} \mathrm{cc}_{\mathrm{attention}}(\langle A, T^v\rangle, t) & \text{if reallocation}(\langle A, T^v\rangle, t) \geq 300 \text{ or } \mathrm{cc}_{\mathrm{attention}}(\langle A, T^v\rangle, t) = 1 \\ \max(0, \mathrm{cc}_{\mathrm{attention}}(\langle A, T^v\rangle, t) - \mathit{penalty}) & \text{otherwise} \end{cases}$$

where $\mathit{penalty} = ((300 - \mathrm{reallocation}(\langle A, T^v\rangle, t))/300) * 0.25$

Eq. 3: The preference function also taking into account how often task reallocations take place. The 'reallocation' function describes how long ago the last reallocation of a task was (in seconds).

mined that a task reallocation was needed, this new task allocation was communicated to the robot and its operator.

Results. In the experiment, we evaluated whether the participants thought that the task reallocations of the model were done at the right time, whether the task reallocations were thought to be appropriate, and whether, after a task reallocation, the CTL of the participants changed as predicted by the model.

Six statements about *timing of reallocations* were given to participants after the experiment. Cronbach's alpha was used to check the internal consistency of these six statements, which yielded 0,607. This is quite low, but expected as the concept of timing is rather broad and we use only six statements. The average response over all six statements describes if participants think the model reallocated tasks at the right moment ranging from 1 (strongly disagree) to 5 (strongly agree). The average value over all participants is 2,65 (standard deviation 0,68). Participants are thus quite neutral about the timing of the model. We cannot say, based on this data, that the model reallocates tasks at the right moment. Conversely, we also cannot say the timing of the model was fully off.

Five statements about the *appropriateness of reallocations* were given to participants after the experiment. Cronbach's alpha yielded 0,694. The average response over the five statements describes if participants think the model chose appropriate task reallocations, ranging from 1 (strongly disagree) to 5 (strongly agree). Averaged out over all participants, this value is 2,10 (standard deviation 0,39). Participants are thus quite negative about the appropriateness of the reallocations. We cannot say, based on this data, that the model chooses appropriate reallocations. Conversely, we *can* say participants think the model *does not* choose appropriate reallocations.

The real shift in CTL was compared to the predicted shift in CTL for each task reallocation. This comparison was done separately for the three metrics. We checked whether the difference between the predicted CTL for the old and new task allocation is the same as the difference between the average real CTL in the two minutes before and after the reallocation. This difference is calculated by subtracting the value for the new task allocation from the value for the old task allocation. The correlation coefficients are 0,32 ($p < 0,05$) for LIP, 0,43 ($p < 0,01$) for TO, and 0,29 ($p = 0,06$) for TSS. The correlations for LIP and TO are significant ($p < 0,05$), the correlation for TSS is not. Based on this data, we can validate that the LIP and TO respond to the task reallocations as the model predicts.

6 DISCUSSION

Trust in Model. During the experiment, participants found it hard to trust the model and to have no control over the task allocation. Making work agreements could help improve trust as they give an operator room to restrict which tasks can be done by the robot(s) and when. Work agreements can also give insight into what tasks actors can expect to be reallocated and when reallocations occur. To further give actors insight and even some influence, we could adapt the level of automation of the task reallocation model itself. A hybrid approach might be most suitable. The model could decide for high workloads and suggest for low workloads (operator decides). Furthermore, it benefits trust if the actor has insight into how the model chooses a task reallocation, e.g., through showing how options are rated. It needs to be further specified and evaluated how the internal processes of the model can be made visually available to the user to improve his understanding and trust of the model. In addition, future research on work agreements and hybrid models is needed to investigate how trust affects the effectiveness of the model.

$$
\mathrm{cc}_I(\langle A, \mathcal{T}\rangle, t) = \begin{cases} \left(\sum\limits_{\forall T^v \in \mathcal{T}} \mathrm{cc}_{\mathrm{time}}(\langle A, T^v \rangle, t) \right) / |\mathcal{T}| & \text{if isHuman}_{\mathcal{A}}(A, t) \\ 1 & \text{otherwise } (A \text{ is a robotic or dummy actor}) \end{cases}
$$

Eq. 4: The full preference function describing preference based on the cost of switching between tasks.

Figure 2: A screen shot of the practice level. The left screen shows the building through the camera mounted on the robot. A victim can be seen, accompanied by a number that could be used to look up information about the victim. The right screen shows the tactical map. The circle on the left corresponds to the location of the robot, the trail to the driven route. Other items shown on the map are (from left to right) a point of interest, a remark, a waypoint, an obstacle, a victim and a picture.

Factors in Choosing a Task Allocation. CTL is a very important factor in choosing a task allocation, but two possible additional factors were identified during the experiment. The first factor is the capability of an actor to do a task. A second factor is the preference for particular tasks of the actor. Taking this into account could greatly benefit actor trust towards the model and reduce reluctance to accept its decisions. Also, the actor is probably more likely to execute a task well that he likes. Future research is needed to explore the effects of including additional factors such as capability and preference, both on the trust and on the performance of the tasks.

Configuration. The exact moment of a task allocation relies on the configuration of the CTL model. Participants' opinion about the timing of the task allocation model will likely benefit from personalizing configuration of CTL problem region boundaries, which was not done in the current experiment. Future research should be executed to determine these boundaries and to explore the effects of personal configuration. Configuration poses additional challenges: Results of experiments using task allocation models with different configurations are hard to generalize and configuration takes a lot of time and effort. Ideally, models will need to become self-learning, adapting themselves to novel tasks and actors when needed.

Representation and Notification. This study did not address how to communicate this task allocation to the actors using the model. More research is needed to investigate how to keep all actors aware of which tasks are allocated to themand how to communicate the task allocation in the most intuitive and understandable way.

7 CONCLUSION

A high-level framework for dynamic task allocation, aimed at improving team performance in mixed human-robot teams, was presented. The framework describes important concepts that influence team performance and can be used to dynamically allocate tasks. The framework applies to a wide array of problems, including heterogeneous teams that might include multiple human actors and multiple robots or agents, a variety of tasks that might change over time and complex and dynamic environments.

We used the framework as a basis for designing a model for adaptive automation triggered by cognitive task load. The framework was general and flexible enough to cover all aspects needed to formalize the model, mainly cognitive task load (as a preference factor) and adaptive automation (as dynamic task allocation). We noticed that although cognitive task load is an important factor, some other factors are also important, such as capability, preference and trust or perceived capability. As the adaptive automation model is based on the framework, it can be quite easily extended to include other factors, which will be done in future work. The model addresses a wider range of problems than most current adaptive automation research, as it focuses on multiple tasks each with their own variable level of autonomy.

We designed an experiment using the model, to explore the effects of the resulting adaptive automation. The model was instantiated for a single human agent cooperating with a single robot in the urban search and rescue domain. An experiment was conducted aimed at testing the model. The experiment did not result in conclusive evidence that the model worked as it should, but encouraging results were found. Two of the three cognitive task load metrics (both the level of information processing and the time occupied) of participants could be managed using the model. Furthermore, important focus points for improving the model and furthering research on adaptive automation in general were identified.

ACKNOWLEDGEMENTS

This research is supported by the EU-FP7 ICT Programmes project 247870 (NIFTi) and project 609763 (TRADR).

REFERENCES

Bailey, N. R., Scerbo, M. W., Freeman, F. G., Mikulka, P. J., and Scott, L. A. (2006). Comparison of a brain-based adaptive system and a manual adaptable system for invoking automation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(4):693–709.

Barnes, C. M., Hollenbeck, J. R., Wagner, D. T., DeRue, D. S., Nahrgang, J. D., and Schwind, K. M. (2008). Harmful help: The costs of backing-up behavior in teams. *Journal of Applied Psychology*, 93(3):529.

Brannick, M. T., Salas, E., and Prince, C. (1997). *Team Performance Assessment and Measurement: Theory, Methods, and Applications*. Series in Applied Psychology. Lawrence Erlbaum Associates.

Burke, J. L., Murphy, R. R., Rogers, E., Lumelsky, V. J., and Scholtz, J. (2004). Final report for the DARPA/NSF interdisciplinary study on human-robot interaction. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):103–112.

Calhoun, G. L., Ruff, H. A., Spriggs, S., and Murray, C. (2012). Tailored performance-based adaptive levels of automation. *Proceedings of the Human Factors and Ergonomics Society*, 56(1):413–417.

Colin, T., Mioch, T., Smets, N., and Neerincx, M. (2014). Real time modeling of the cognitive load of an urban search and rescue robot operator. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN2014)*.

Fitts, P. M., Viteles, M. S., Barr, N. L., Brimhall, D. R., Finch, G., Gardner, E., Grether, W. F., Kellum, W. E., and Stevens, S. S. (1951). Human engineering for an effective air-navigation and traffic-control system. Technical report, Ohio State University Research Foundation Columbus.

Gerkey, B. P. and Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954.

Greef, T. E. d., Arciszewski, H. F. R., and Neerincx, M. A. (2010). Adaptive automation based on an object-oriented task model: Implementation and evaluation in a realistic C2 environment. *Journal of Cognitive Engineering and Decision Making*, 4(2):152–182.

Guzzo, R. A., Salas, E., and Goldstein, I. L. (1995). *Team Effectiveness and Decision Making in Organizations*. Jossey-Bass San Francisco.

Hilburn, B., Molloy, R., Wong, D., and Parasuraman, R. (1993). Operator versus computer control of adaptive automation. In *The Adaptive Function Allocation for Intelligent Cockpits (AFAIC) Program: Interim Research Guidelines for the Application of Adaptive Automation*, pages 31–36. DTIC Document.

Ilgen, D. R., Hollenbeck, J. R., Johnson, M., and Jundt, D. (2005). Teams in organizations: From input-process-output models to IMOI models. *Annual Review of Psychology*, 56:517–543.

Inagaki, T. (2003). Adaptive automation: Sharing and trading of control. In *Handbook of Cognitive Task Design*, pages 147–169. Mahwah, NJ: Lawrence Erlbaum Associates.

Kidwell, B., Calhoun, G. L., Ruff, H. A., and Parasuraman, R. (2012). Adaptable and adaptive automation for supervisory control of multiple autonomous vehicles. In *Proceedings of the Human Factors and Ergonomics Society*, pages 428–432.

Matthew, G., Davies, D. R., Westerman, S. J., and Stammers, R. B. (2000). *Human Performance: Cognition, Stress, and Individual Differences*. Psychology Press.

McGrath, J. E. (1964). *Social Psychology: A Brief Introduction*. Holt, Rinehart and Winston.

Moray, N., Inagaki, T., and Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44.

Neerincx, M. A. (2003). Cognitive task load design: Model, methods and examples. In *Handbook of Cognitive Task Design*, pages 283–305. Mahwah, NJ: Lawrence Erlbaum Associates.

Neerincx, M. A., Kennedie, S., Grootjen, M., and Grootjen, F. (2009). Modeling the cognitive task load and performance of naval operators. In Schmorrow, D., Estabrooke, I., and Grootjen, M., editors, *Foundations of Augmented Cognition*, volume 5638 of *LNCS*, pages 260–269. Springer Berlin / Heidelberg.

Parasuraman, R., Cosenzo, K. A., and Visser, E. d. (2009). Adaptive automation for human supervision of multiple uninhabited vehicles: Effects on change detection, situation awareness, and mental workload. *Military Psychology*, 21(2):270–297.

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., and Goodrich, M. (2006). Common metrics for human-robot interaction. In *Proceedings of the 1^{st} Conference on Human-Robot Interaction*, pages 33–40. ACM.