Dynamic Task Allocation for Human-Robot Teams

CONFERENCE PAPER · JANUARY 2015

DOI: 10.5220/0005178001170124

4 AUTHORS, INCLUDING:

Tina Mioch

TNO

Mark Neerincx

TNO

John-jules Meyer

Utrecht University

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*

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 exam-

R. A. Giele T., Mioch T., A. Neerincx M. and C. Meyer J..
Dynamic Task Allocation for Human-robot Teams.
DOI: 10.5220/0005178001170124
In *Proceedings of the International Conference on Agents and Artificial Intelligence* (ICAART-2015), pages 117-124
ISBN: 978-989-758-073-4

ple, 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

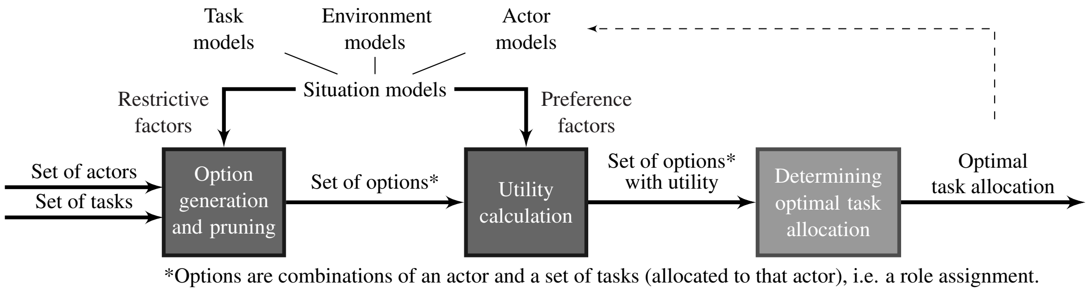

Figure 1: Overview of the proposed framework. Boxes denote processes, arrows represent flow of information. Opposite to Gerkey and Matarić's (Gerkey and Matarić, 2004) focus, we focus on the process of pruning generated options and calculating utility of options (darker boxes) and less on the process 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 attention is needed to switch to a new task set. The formal description of this aspect is seen in Equation 2. If the

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

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.

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: $\text{utility}(O,t)=(\text{ctl}_I(O,t)+\text{cc}_I(O,t))/2$

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 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 determined 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

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

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.

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

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

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).

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

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

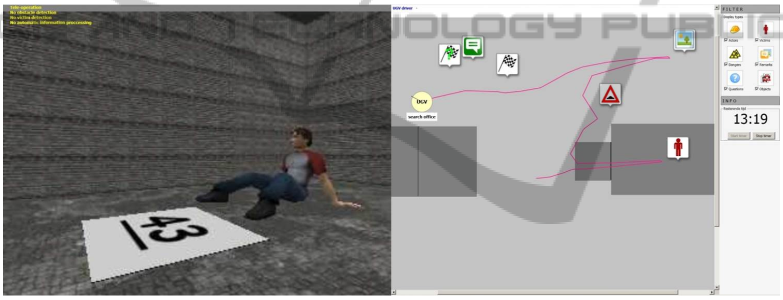

Figure 2: A screen shot of the practice level. The left screen shows the building through the camera mounted on the robot. A victim can be seen, accompanied by a number that could be used to look up information about the victim. The right screen shows the tactical map. The circle on the left corresponds 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.

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 one. 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.

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 do this 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 1st Conference on Human-Robot Interaction*, pages 33–40. ACM.