Comparison Between Static and Dynamic Willingness to Interact in Adaptive Autonomous Agents

Mirgita Frasheri, Baran Cürüklü and Mikael Ekström
Mälardalen University, Västerås, Sweden

Keywords: Adaptive Autonomy, Multi-agent Systems, Collaborative Agents.

Abstract: Adaptive autonomy (AA) is a behavior that allows agents to change their autonomy levels by reasoning on their circumstances. Previous work has modeled AA through the willingness to interact, composed of willingness to ask and give assistance. The aim of this paper is to investigate, through computer simulations, the behavior of agents given the proposed computational model with respect to different initial configurations, and level of dependencies between agents. Dependency refers to the need for help that one agent has. Such need can be fulfilled by deciding to depend on other agents. Results show that, firstly, agents whose willingness to interact changes during run-time perform better compared to those with static willingness parameters, i.e. willingness with fixed values. Secondly, two strategies for updating the willingness are compared, (i) the same fixed value is updated on each interaction, (ii) update is done on the previous calculated value. The maximum number of completed tasks which need assistance is achieved for (i), given specific initial configurations.

1 INTRODUCTION

Adaptive autonomy enables software agents to decide, on the fly, on their autonomy levels, i.e. on whether to be more or less autonomous in the context of a task or a goal, with respect to other entities, such as humans and other agents. There are several other theories on autonomy and the way it can change (Vernon et al., 2007). Castelfranchi (Castelfranchi, 2000) uses dependence theory to define autonomy. An agent A which is trying to complete a task/goal x but lacks any means to do so (e.g. knowledge, resources, or functional ability), will depend on another agent B for help, thus will become less-autonomous with respect to task/goal x. Note that, this represents the social aspect of autonomy which deals with agent-agent interaction, and is to be distinguished from issues related to an agent's autonomy from an environment. Johnson *et al.* (Johnson et al., 2011) consider the descriptive and prescriptive dimensions of autonomy, or also referred to as self-sufficiency, i.e. being able to carry out a task by oneself, and self-directedness, i.e. being able to choose one's own goals.

A 10 levels of autonomy model was proposed as a guideline for understanding the concept of several levels of autonomy which could be displayed by a system in the context of human-robot interaction (Parasuraman et al., 2000). This approach defines discrete changes in the level of autonomy from teleoperation to full autonomy. On the other hand, collaborative control (Fong et al., 2001) departed from the dominant view of human/master - agent/slave, and brought forward a perspective in which human and agent are peers and use dialogue to resolve inconsistencies. However, that is not to say that the agent is able to decide on its own goals, it will still operate under the goals defined by a human. Adjustable autonomy, in some cases, is used for a system in which the human decides on the autonomy levels (Hardin and Goodrich, 2009), and in others as an umbrella term that covers different ways in which autonomy is shared between humans and agents (Johnson et al., 2011). Mixed-initiative interaction allows for both agent and human operator to decide on autonomy, whilst adaptive autonomy places the decision-making on the agent (Hardin and Goodrich, 2009). Usually, the difference in these concepts lies on the party that is increasing or decreasing agent autonomy. In adaptive autonomy, this decision lies with the agent itself.

In this paper, it is assumed that autonomy changes when agents decide to become dependent on one another. These decisions are taken continuously, based on the circumstances. The willingness to interact models the two facets of these interactions, through the willingness to give, and ask for help (Frasheri et al., 2017a). The behavior of the agents with dy-

Frasheri, M., Cürüklü, B. and Ekström, M.
Comparison Between Static and Dynamic Willingness to Interact in Adaptive Autonomous Agents.
DOI: 10.5220/0006648002580267
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 1*, pages 258-267
ISBN: 978-989-758-275-2

namic and static willingness to interact is compared. The hypothesis is that agents with dynamic willingness complete more tasks than agents with static willingness. Moreover, two different strategies for updating the willingness are considered in the dynamic case, one in which the same initial value (base-line) is updated on each interaction, and another in which the update is done on the previous calculated value.

The rest of this paper is organized as follows. In Section 2 related work found in the field is discussed. Section 3 describes the agent model and in particular the willingness to interact. The simulation setup is described in Section 4, whilst the results are discussed in Section 5. Finally, Section 6 provides a discussion and future work directions.

2 RELATED WORK

Related work in the area is quite vast with respect to the different theories and approaches to autonomy. Johnson *et al.* (Johnson et al., 2011) assert that inter-dependencies in joint-activity should be at the heart of designing (software) systems with adjustable autonomy. In this model, agents are inter-dependent if they rely on each other during the execution of their goals/tasks. Moreover inter-dependencies could be soft, i.e. they are not necessary for the successful outcome of a task but can improve performance, and hard, i.e. they are in fact necessary for the successful outcome of a task. They further propose a design methodology (Johnson et al., 2014), which aims to provide concrete tools that can be used while implementing a system with adjustable autonomy. The method covers the following steps. In the beginning, possible inter-dependencies in the system (between tasks, and between humans and agents) are identified. Afterwards, different mechanisms are designed in order to support them. Lastly, it is analyzed how these mechanisms affect the existing inter-dependent relationships. In the paper presented here, a higher level of abstraction is considered, in which tasks are abstract. Their other work has focused on policy systems, such as Kaa (Bradshaw et al., 2005), which allows a central agent to change policies for some of the agents during run-time, depending on the circumstances. The human will be put in the loop if Kaa cannot reach a decision. In the paper presented here, there is no central solution. Each agent makes decisions on its own autonomy.

Frameworks such as STEAM (Tambe, 1997), and DEFACTO (Schurr et al., 2009), have been proposed to target support for teamwork and adjustable autonomy, respectively. STEAM extends the Soar (Laird, 2012) by adding support for teamwork through team operators. The agents also have their individual plans which do not require teamwork. As a result, they can reason for team and individual plans. The solution covers a synchronization protocol, based on whether communication is necessary (through the likelihood that others have it already), the cost for miss-coordination, and the threat that some event poses to the joint plan. In the paper presented here, the main assumption is that an activity (task) that may start as an individual one, could turn out to need support, thus it could become a team-task. The DEFACTO framework aims at providing support for transfers of control in continuous time, resolving human-agent inconsistencies, and making actions interrupt-able for real-time systems.

Scenarios with static and dynamic autonomy are compared (Martin and Barber, 2006). The authors use different decision frameworks, such as master-slave, peer to peer and locally autonomous, which are dynamically switched between each other based on the specific conditions. The mapping between the right decision framework and conditions is set in advance. The authors aim to show the usefulness of dynamic autonomy. Whereas, other work compares different implementations (Hardin and Goodrich, 2009), such as adaptive autonomy (agents change their own autonomy), mixed-initiative interaction (both human and agent are able to change autonomy), and adjustable autonomy (the human is able to change the autonomy). The implementation of mixed-initiative interaction performs better than the other two, measured by the number of victims found in a search and rescue simulation environment. In their case, AA agents would always try to maintain the highest-level autonomy. In the paper presented here, two different scenarios are considered. One in which, the agents go back to a fixed level after each interaction, and another in which agents adapt continuously.

Specific reasoning mechanisms have been proposed that weigh the influence of outside stimuli (e.g. order of a superior) on the agent, by considering task urgency and dedication level to the organization(van der Vecht et al., 2009). This paper extends on the factors that shape agent reasoning as is detailed in the following sections. Others allow the change in autonomy to happen at the individual task level (Brookshire et al., 2004), i.e. the agent can execute a task autonomously, whilst being tele-operated for another one. The scenario in the current paper assumes that agents perform tasks sequentially, and that they can delegate when assistance is needed. Another approach is to categorize tasks in two groups, i.e. tasks which the robot is able to perform by itself, and tasks

that need human supervision/assistance (Kim et al., 2016). Algorithms are designed based on this classification. The current paper does not assume such categorization done before hand, and any task can turn out to need assistance. Moreover, the presence of the human operator is not modeled. Silva *et al.* (da Silva et al., 2017) propose an agent framework for when to ask for and give advice to others, in a multi-agent learning environment. These two behaviors are shaped from the confidence level an agent has in its current state, i.e. an agent will ask for advise if not confident in its current policies, or give advice if it is confident in them. The concepts discussed in the present paper are complementary to these ideas, but the type of interactions assumed represent task delegation from one agent to another.

3 THE AGENT MODEL

The agent model described in this section stems from (Frasheri et al., 2017b), (Frasheri et al., 2017a). In this model, the problem of embedding adaptive autonomous behavior in an agent is addressed from two perspectives. Firstly, a generic high-level agent model models the internal operation of the agent. Secondly, the willingness to interact is introduced, based on which the agent can modify its autonomy as needed. Willingness to interact is composed of two separate relationships, to help and to ask for help. In order to calculate the willingness, an initial computational model has been proposed and is incorporated in the general agent framework.

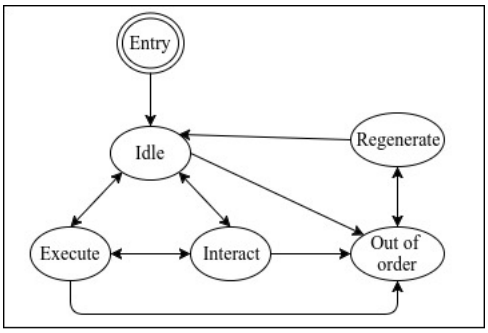

Figure 1: The proposed agent model composed of five states.

The agent model is composed of five states, *idle*, *execute*, *interact*, *regenerative* and *out_of_order* (Figure 1). It is also assumed to have the following characteristics:

1. b: battery
2. e: equipment (sensors, motors, actuators)
3. a: abilities (2D vision, reasoning, planning, moving, object manipulation and so on)
4. t: tools
5. k: knowledge (with respect to a task, its environment, other agents)

These characteristics are implemented in an abstract way, detailed in Section 4. Furthermore, the agent is assumed to be able to estimate the following:

1. e_R: environment risk factor
2. a_R: potential collaborator risk
3. μ: its own performance
4. t_P: progress of the task for which it is requiring help (partially implemented)
5. t_T: trade-off of adopting a new task and postponing/dropping the current task (not yet implemented)

The general operation of the agent runs as follows. The agent starts its operation in the *idle* state. In *idle*, the agent has not committed to any goal. The implication of this is that the agent might be doing nothing (simply waiting for a request), or it might decide to adopt a goal. When an agent commits to a goal, it is added into its FIFO queue, switches to the *execute* state and performs the task related to that goal. During the execution, the agent will evaluate if it needs to ask another agent for assistance (based on the willingness to ask for help). When the task is complete, the agent will return to *idle*. If the agent is in *idle* or *execute*, when a request is received, it will switch to the *interact* state. The latter is not possible to interrupt, and by the end of this state the agent will decide (based on the willingness to give help) whether to accept the received request. If accepted, the corresponding task is put into the agent's FIFO queue, otherwise the request will be discarded. Multiple requests are handled sequentially, depending on the incoming order. If accepted they are put into the queue, otherwise they are discarded. If an agent's battery level goes below the threshold, it will switch to *out_of_order*. Thresholds are taken the same for each agent for simplicity, but it is not necessary that they are so. As a result, the agent will immediately switch to *regenerate*, in which the recharge takes place. Afterwards the agent returns to *idle*. The switch from *regenerate* to *idle* happens immediately. Once more, this choice was for simplicity. A time delay could be introduced to make the process more realistic.

3.1 Willingness to Interact

In this paper an agent's willingness to interact shapes its autonomy and is composed of two elements which

are willingness to give help δ and willingness to ask for help γ. The reasoning of the agent on its willingness is explained in the following scenario.

Assume an agent *A* which decides to do a task *t*. At first, *A* will need to consider if it has enough energy before continuing with the pursuit. Then, it needs to make sure that it has all the required abilities in its repertoire. Moreover, agent *A* might have different levels of expertise for its abilities. Afterwards, the agent needs to make sure it has all the required resources, in the necessary quantity and quality. Resources are composed of internal and external resources. Internal resources include energy/battery, sensors, motors, actuators, knowledge that are part of the agent. External resources include physical objects in the environment and other agents (individually and as groups). Whether the agent has the needed levels of energy, abilities, and resources will depend upon the specific requirements of the task. Assume that *A* has an ability a_i with level of expertise *q*. It might be that even though the value of *q* is above a specific threshold for a task t_1, the same value may not be acceptable for another task t_2. If agent *A* complies well with what is needed for the task, it could in principle conduct the activity autonomously on its own.

However, other factors might be taken into consideration. Agent *A* has a performance value determined by its past successes and failures. The lower the performance, the higher the inclination to ask for help might be. Furthermore, the agent might be more inclined to ask for help if it has not progressed by itself with respect to the task, or if the environment it is operating within is highly threatening. However, if the agent knows no one, or if it lives among unhelpful agents, it might be inclined otherwise. When agent *A* decides to require assistance, it will forward its request to some agent *B*. It is reasonable to assume that *B* is not chosen at random, but based on past history of collaboration between *A* and *B* and the corresponding successes. Naturally, when there is not any of the latter, the agent might choose at random between the agents it knows. Agent *B* will then need to process the request of agent *A*. The reasoning of *B* will be similar, with the added consideration of the trade-off between dropping or postponing its current activities in order to help *A*. The influence of each of the factors on the components of the willingness is captured in Figures 2 and 3. In Figure 2 it is shown that if the agent misses any of the required internal resources, then it cannot perform the task, thus it will ask for help. This is deterministic. On the other hand, the other factors model the probabilistic nature of the willingness to ask for help. If the perceived environmental risk increases, then the willingness to ask increases, otherwise it decreases. If the perceived risk from another agent increases, then the willingness to ask decreases, otherwise it increases. If the agent's own performance increases, then the willingness increases, otherwise it decreases. Finally, if the task progress increases, the willingness to ask decreases (if the agent progresses, the need for assistance decreases), otherwise it increases. In Figure 3 it is shown that missing the internal resources will decrease the willingness to give help, otherwise the willingness increases. It is clear in this case, that if the agent still accepts to help, it will in turn ask for help another agent, thus creating a chain reaction (Frasheri et al., 2017b). Another approach could be to have the agent directly reject any request if it misses any of the internal resources, and let the agent in need figure out what to do next. If the environmental risk decreases, then the willingness to give increases, otherwise it decreases. If the agent risk increases, then the willingness decreases, otherwise it increases. If the agent's own performance increases, then the willingness increases, otherwise it decreases.

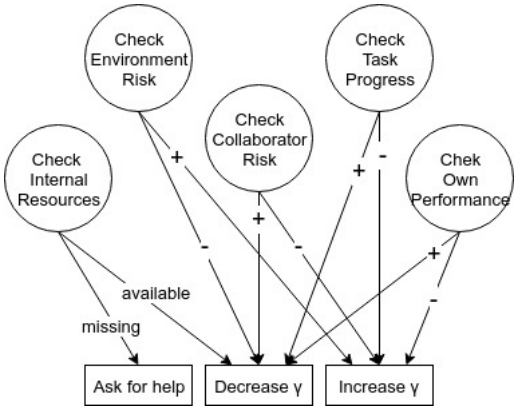

Figure 2: Influence diagram for the willingness to ask for help, γ.

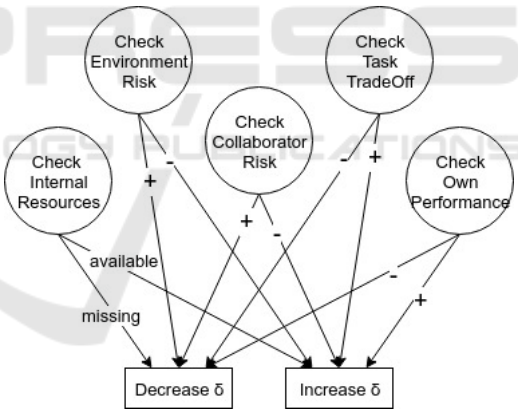

Figure 3: Influence diagram for the willingness to give help, δ.

Finally, if the task trade-off is in favor of the new task, then the willingness will increase, otherwise it will decrease.

Agents are assumed to start their operation with predefined values $\langle \delta_0, \gamma_0 \rangle$, and corresponding changing steps $\langle \Delta\delta, \Delta\gamma \rangle$. Moreover, they are equipped with a set of abstract sensors, actuators, motors, knowledge and abilities. Also, there are no restrictions assumed as to whether they change in time, e.g. an agent could have an initial minimal amount of knowledge which expands during the operation through learning, or the agent might be updated to support new abilities and so on. Agents are assumed to be able to compute a risk factor for the environment they operate in and for each other agent in that environment. In addition, they are able to compute a performance measure for themselves. An agent can generate a task t_i to do, or can receive a request from another agent for another task t_j. The task comes with specific execution requirements such as: estimated amount of energy required, equipment, knowledge, abilities, tools. It is worth noting that the functions used to determine each of the variables that affect willingness may vary. For the purposes of this paper, they are computed in simple terms (Section 4). Currently the following limitations hold. (i) The degree to which an agent has an ability, or a resource is not considered, i.e. these properties take binary values. (ii) An agent checks if it needs assistance at the beginning of each task. (iii) A task is simulated as an atomic step, as such the trade-off between tasks is not simulated in the current work.

4 SIMULATIONS

The behavior of the agents was investigated through computer simulations. One trial was conducted, and is composed of two separate computer simulations, referred to as phases. In the first phase, the values of δ and γ are static throughout the whole time. Simulations are repeated for the following combinations of $\langle \delta, \gamma \rangle$: $\langle 0.0, 0.0 \rangle$ $\langle 0.5, 0.0 \rangle$, $\langle 1.0, 0.0 \rangle$, $\langle 0.0, 0.5 \rangle$, $\langle 0.5, 0.5 \rangle$, $\langle 1.0, 0.5 \rangle$, $\langle 0.0, 1.0 \rangle$, $\langle 0.5, 1.0 \rangle$, $\langle 1.0, 1.0 \rangle$, $\langle 0.2, 0.2 \rangle$ $\langle 0.5, 0.2 \rangle$, $\langle 0.8, 0.2 \rangle$, $\langle 0.2, 0.5 \rangle$, $\langle 0.8, 0.5 \rangle$, $\langle 0.2, 1.0 \rangle$, $\langle 0.5, 0.8 \rangle$, and $\langle 0.8, 0.8 \rangle$. These values are chosen as representatives of extreme and average selfish/unselfish agent behavior. In the second phase, their values will change during run-time according to the scheme shown in Figures 2 and 3 with $\Delta\delta = \Delta\gamma = 0.05$. Simulations are repeated for all combinations of $\langle \delta_0, \gamma_0 \rangle$: $\langle 0.0, 0.0 \rangle$ $\langle 0.5, 0.0 \rangle$, $\langle 1.0, 0.0 \rangle$, $\langle 0.0, 0.5 \rangle$, $\langle 0.5, 0.5 \rangle$, $\langle 1.0, 0.5 \rangle$, $\langle 0.0, 1.0 \rangle$, $\langle 0.5, 1.0 \rangle$, $\langle 1.0, 1.0 \rangle$, $\langle 0.2, 0.2 \rangle$ $\langle 0.5, 0.2 \rangle$, $\langle 0.8, 0.2 \rangle$, $\langle 0.2, 0.5 \rangle$, $\langle 0.8, 0.5 \rangle$, $\langle 0.2, 1.0 \rangle$, $\langle 0.5, 0.8 \rangle$, and $\langle 0.8, 0.8 \rangle$, in which δ_0 and γ_0 are the initial values for δ and γ. Moreover, for the second phase of simulations, two update strategies for δ and γ are investigated. In one strategy, the values for δ and γ will always be calculated from δ_0 and γ_0. In the other strategy, the values for δ and γ will be calculated based on values calculated on the previous interaction.

All experiments are repeated for three levels of difficulty. Difficulty refers to the probability (P_D) that an agent lacks any of the abilities, knowledge, equipment, or tools. The higher this probability, the higher the chances that an agent will ask for help. The values taken for P_D are $0.2, 0.5, 0.8$, which means that for $P_D = 0.2$, the probability that the agent will ask for help is 0.2, and the difficulty of the simulation is low. The size of the population is 30, and remains fixed across phases.

Each simulation runs for circa 20 minutes. Within this amount of time, the agents are able to attempt and complete a considerable amount of tasks, in the range of hundreds. Consequently, it is possible to identify a trend for the number of tasks completed over those attempted. This choice was considered adequate for the purposes of this paper. Naturally, it is possible to let the agents run for a longer time.

Each agent is composed of a set of ROS (Quigley et al., 2009) nodes. The communication between agents happens by broadcast (when agents make themselves known to each other), and by a tailored action-server mechanism (when agents make one to one help requests to each other). All agents in the simulation start in their *idle* state. Each time they are in this state, a task could be generated with a probability $P = 0.6$.

In the current state of the implementation an agent reasons at the beginning of each task on whether it needs assistance. If it does not, then it is supposed to always succeed by itself. Moreover, the execution of each task is simulated by having the system pause for a specific amount of time Δt which corresponds to a predefined completion time of a task. This is a simplified scenario that was deemed adequate for the purposes of the simulations described here. The success criterion for the static case is defined as follows. An agent fails when it attempts a task while lacking any of the following internal resources: abilities, knowledge, battery, equipment, and tools. Otherwise it will succeed. The presence of any of the abilities, knowledge, equipment, and tools, is decided by the difficulty of the simulation.

All agents start with the same level of battery $b = 4000$ of energy units, which decreases after every finished task by the amount of energy required by that task. If b becomes lower than a threshold $b_{low} = 300$,

then the agent goes into the *out_of_order* state. Parameters b and b_{low} are not bound to these specific values. These were chosen for running the simulation. The energy level required by tasks is generated at random at task generation as follows. First, a task difficulty is selected randomly between low, medium, and high. If the task is of low difficulty, then the energy required for it, is chosen randomly in the range $1-30$. Whereas, if the difficulty is medium, then the energy is taken in the range $31-60$. Finally, for the high difficulty, energy is chosen in the range $61-90$. When giving help, the agent always checks whether its current battery level minus the energy required by the task is above the threshold b_{low}.

The agent to ask for help is the one with the highest perceived helpfulness with probability $p = 0.6$ and randomly otherwise. Perceived helpfulness in this case is interpreted as the least amount of risk, calculated as in Equation 1.

$$\beta = max(\{ph_1, ...ph_i, ...ph_n\}) \quad (1)$$

where ph_i - perceived helpfulness of agent i, n - number of agents.

Agents keep track of how many tasks they are attempting, completing, and also whether they needed help for a task or not. Tasks for which help is either needed or asked for, are called dependent tasks. A task is by default dependent if the agent lacks any of the following: battery, abilities, resources, knowledge, external resources. The performance for each agent is expressed in terms of completed tasks over attempted tasks, as follows.

$$\mu = \frac{Tasks\ Completed}{Tasks\ Attempted} \quad (2)$$

Factors such as environmental risk and task progress are kept constant throughout the simulation, therefore they affect the willingness to interact in the same way on each interaction. The environmental risk is fixed to a low value equal to 0.2, and affects γ by $-\Delta\gamma$, and δ by $+\Delta\delta$. This means that, when environmental risk is perceived as low, the willingness to ask for help will get lower, whilst the willingness to give help will get higher. Furthermore, since the reasoning for asking for help is done at the beginning of the task and the agent cannot yet have a sense of progress, the task-progress affect is by $-\Delta\gamma$ over γ.

5 RESULTS

The simulations were run in order to investigate the following hypothesis. Agents which change their willingness to interact during run-time are able to complete more tasks (as a population) compared to the agent population with static willingness. This problem was partially targeted in (Frasheri et al., 2017b) by considering only the willingness to give help. Moreover, two different methods of updating the willingness are compared, (i) agents update their willingness based on previous calculated values, (ii) agent update their willingness from fixed baseline values. Each separate simulation was run 10 times, thereafter the mean and standard deviation values were calculated.

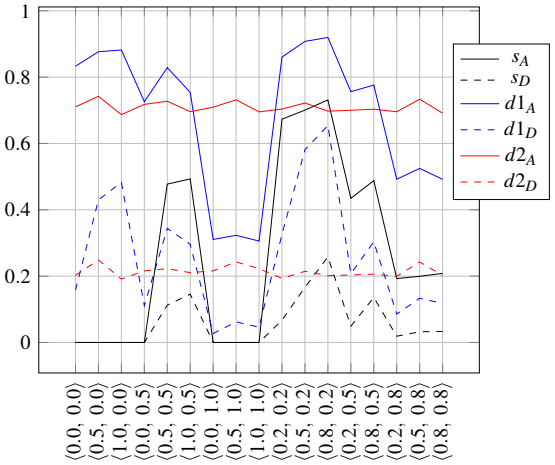

Figure 4: Task completion percentages for $P_D = 0.2$. The x-axis contains different configurations for $\langle\delta, \gamma\rangle$. s_A refers to the completion degree for all tasks in the static phase, whereas s_D corresponds to completion degree of dependent tasks. The interpretation of the subscript is the same for $d1$ and $d2$. Both refer to the dynamic case, $d1$ to the base-line update, and $d2$ to the continuous update respectively.

The results are shown in graphic, Figures 4-6, and tabular form, Tables 1-3, (where the standard deviation values are shown as well in round brackets). In each graph, the different combinations $\langle\delta_0, \gamma_0\rangle$ are displayed along the x-axis, whereas the percentages of completed tasks are along the y-axis (results correspond to the population as a whole). Two completion degrees are considered: on one hand the one with respect to all tasks computed as in Equation 2, on the other hand the one with respect to dependent tasks (tasks that cannot be achieved without help) computed as in Equation 3.

$$\mu_d = \frac{Dependent\ Tasks\ Completed}{Dependent\ Tasks\ Attempted} \quad (3)$$

5.1 Static vs. Dynamic Willingness

It is possible to observe from Figures 4-6 that agents with dynamic willingness perform better in most

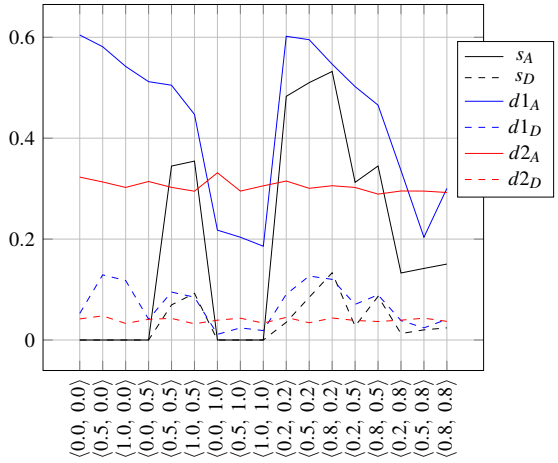

Figure 5: Task completion percentages for $P_D = 0.5$. Notation interpretation as in Figure 4.

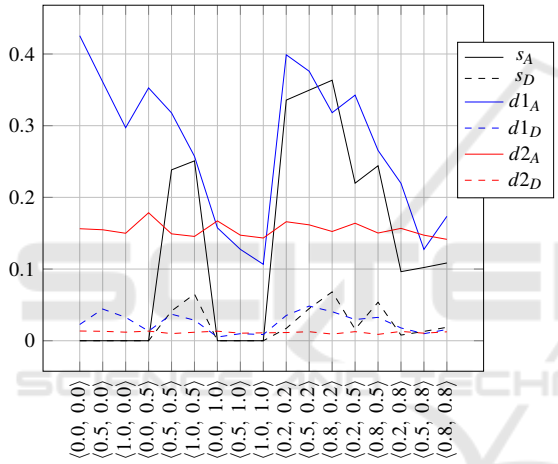

Figure 6: Task completion percentages for $P_D = 0.8$. Notation interpretation as in Figure 4.

cases than their static counterpart. Thus the hypothesis holds across the three difficulty levels for most combinations of $\langle \delta, \gamma \rangle$, considering the completion of both dependent and all tasks. Obviously, as the probability of failure (P_D) increases, the performance of the agents degrades independently of whether the willingness is static or dynamic. Moreover, in Figure 6, it is also possible to observe that static combinations of willingness such as $\langle 1.0, 0.5 \rangle$, $\langle 0.8, 0.2 \rangle$, and $\langle 0.8, 0.5 \rangle$ perform slightly better than the dynamic counterparts.

In the static scenario, it is clear that for $\delta = 0.0$, the agents will never help one another, thus no dependent tasks will be achieved. Also, for $\gamma = 1.0$, all agents will always ask one another, and will not accomplish any tasks. When $\gamma = 0.0$, agents will never ask, so the outcome will be the same as in the previous cases. For other combinations, some dependent tasks are accomplished. As mentioned previously, if the agent does not ask for help when a task is necessarily dependent (i.e. either battery, ability, knowledge, equipment, or external resources lack), then it will fail. Asking for help is determined with fixed probabilities, which are the initial values for γ. In the dynamic case, this is not probabilistic. If an agent determines a task to be dependent (based on the difficulty level of the simulation P_D), then it will for sure ask for help, thus increasing the chances of the task being completed.

5.2 Base-line Update vs. Continuous

The comparison between the cases with base-line update and continuous update can be made from several perspectives. For low difficulty of the simulation (Figure 4), it is possible to observe that the best performance (dependent tasks) is achieved in the base-line update for the configuration $\langle \delta_0 = 0.8, \gamma_0 = 0.2 \rangle$, followed by $\langle \delta_0 = 1.0, \gamma_0 = 0.0 \rangle$.

In Figure 5, best performance in base-line update is achieved for the configurations $\langle \delta_0 = 0.5, \gamma_0 = 0.0 \rangle$, $\langle \delta_0 = 0.5, \gamma_0 = 0.2 \rangle$, and $\langle \delta_0 = 0.8, \gamma_0 = 0.2 \rangle$. Whereas, in the static update, the performance peaks are for $\langle \delta_0 = 0.8, \gamma_0 = 0.2 \rangle$, and $\langle \delta_0 = 1.0, \gamma_0 = 0.5 \rangle$. The maximum performance, with respect to dependent tasks, is achieved for static update with $\langle \delta_0 = 0.8, \gamma_0 = 0.2 \rangle$. Nevertheless, with respect to all configurations the base-line update does slightly better.

In Figure 6, there are three peaks for static update, $\langle \delta_0 = 1.0, \gamma_0 = 0.5 \rangle$, $\langle \delta_0 = 0.8, \gamma_0 = 0.2 \rangle$, and $\langle \delta_0 = 0.8, \gamma_0 = 0.5 \rangle$. The difference with the base-line update is higher than in Figure 5. The base-line update does better for other configurations.

The continuous case does always worse, and is overall better than the static case only for $P_D = 0.2$, (Figure 4), with maximums achieved for $\langle \delta_0 = 0.5, \gamma_0 = 0.0 \rangle$, $\langle \delta_0 = 0.5, \gamma_0 = 1.0 \rangle$, and $\langle \delta_0 = 0.5, \gamma_0 = 0.8 \rangle$. For the other values of P_D, the continuous update has overall the worst performance outcomes, with no clear peaks. It is also noticeable that for the continuous update, all the agents, independently of their start seem to reach a stable state, i.e. the range of values for the performance is small, around 0.2% for low difficulty level of the simulation. This phenomenon is true for both dependent and total tasks completed rates (Figure 4).

Finally, it is possible to observe that the same patterns for static, base-line, and continuous update, hold across difficulty levels. Furthermore, the base-line and static updates follow similar patterns across different combinations of $\langle \delta, \gamma \rangle$.

Table 1: Task completion percentages for $P_D = 0.2$. s_A refers to the completion degree for all tasks in the static phase, whereas s_D corresponds to completion degree of dependent tasks. The interpretation of the subscript is the same for $d1$ and $d2$. Both refer to the dynamic case, $d1$ to the base-line update, and $d2$ to the continuous update respectively.

	Static		Base-line		Continuous	
$\langle\delta_0,\gamma_0\rangle$	s_A	s_D	$d1_A$	$d1_D$	$d2_A$	$d2_D$
<0.0,0.0>	na	0	0.833 (0.005)	0.158 (0.009)	0.711 (0.023)	0.203 (0.050)
<0.5,0.0>	na	0	0.877 (0.016)	0.430 (0.062)	0.742 (0.024)	0.249 (0.049)
<1.0,0.0>	na	0	0.882 (0.017)	0.483 (0.068)	0.687 (0.031)	0.191 (0.044)
<0.0,0.5>	na	0	0.726 (0.009)	0.111 (0.017)	0.718 (0.020)	0.216 (0.050)
<0.5,0.5>	0.477 (0.015)	0.113 (0.010)	0.829 (0.009)	0.344 (0.022)	0.727 (0.017)	0.223 (0.024)
<1.0,0.5>	0.493 (0.012)	0.146 (0.017)	0.754 (0.013)	0.296 (0.032)	0.696 (0.031)	0.211 (0.049)
<0.0,1.0>	na	0	0.311 (0.016)	0.027 (0.005)	0.709 (0.021)	0.216 (0.046)
<0.5,1.0>	na	0	0.323 (0.019)	0.062 (0.010)	0.732 (0.021)	0.243 (0.041)
<1.0,1.0>	na	0	0.306 (0.029)	0.046 (0.009)	0.695 (0.023)	0.223 (0.040)
<0.2,0.2>	0.673 (0.008)	0.067 (0.008)	0.861 (0.006)	0.326 (0.019)	0.704 (0.022)	0.193 (0.028)
<0.5,0.2>	0.701 (0.006)	0.166 (0.008)	0.908 (0.008)	0.581 (0.033)	0.722 (0.022)	0.215 (0.032)
<0.8,0.2>	0.731 (0.006)	0.258 (0.014)	0.920 (0.013)	0.654 (0.051)	0.697 (0.025)	0.201 (0.034)
<0.2,0.5>	0.435 (0.014)	0.049 (0.008)	0.756 (0.009)	0.207 (0.015)	0.700 (0.014)	0.203 (0.017)
<0.8,0.5>	0.488 (0.008)	0.136 (0.008)	0.776 (0.019)	0.303 (0.024)	0.703 (0.019)	0.206 (0.032)
<0.2,1.0>	0.193 (0.017)	0.019 (0.005)	0.492 (0.015)	0.086 (0.011)	0.696 (0.022)	0.200 (0.027)
<0.5,0.8>	0.199 (0.017)	0.032 (0.006)	0.525 (0.015)	0.133 (0.012)	0.734 (0.026)	0.243 (0.064)
<0.8,0.8>	0.208 (0.015)	0.033 (0.006)	0.492 (0.015)	0.118 (0.018)	0.691 (0.028)	0.202 (0.052)

Table 2: Task completion percentages for $P_D = 0.5$. Notation interpretation as in Table 1.

	Static		Base-line		Continuous	
$\langle\delta_0,\gamma_0\rangle$	s_A	s_D	$d1_A$	$d1_D$	$d2_A$	$d2_D$
<0.0,0.0>	na	0	0.605 (0.016)	0.053 (0.009)	0.323 (0.021)	0.042 (0.008)
<0.5,0.0>	na	0	0.581 (0.015)	0.129 (0.017)	0.313 (0.033)	0.048 (0.012)
<1.0,0.0>	na	0	0.542 (0.016)	0.118 (0.014)	0.302 (0.019)	0.033 (0.015)
<0.0,0.5>	na	0	0.512 (0.019)	0.041 (0.007)	0.314 (0.012)	0.041 (0.016)
<0.5,0.5>	0.345 (0.019)	0.070 (0.013)	0.505 (0.021)	0.095 (0.010)	0.302 (0.025)	0.043 (0.013)
<1.0,0.5>	0.355 (0.017)	0.092 (0.012)	0.447 (0.024)	0.085 (0.012)	0.295 (0.054)	0.032 (0.008)
<0.0,1.0>	na	0	0.218 (0.016)	0.011 (0.003)	0.331 (0.051)	0.039 (0.007)
<0.5,1.0>	na	0	0.204 (0.023)	0.024 (0.004)	0.295 (0.022)	0.043 (0.012)
<1.0,1.0>	na	0	0.186 (0.018)	0.019 (0.004)	0.306 (0.034)	0.033 (0.010)
<0.2,0.2>	0.483 (0.009)	0.035 (0.004)	0.602 (0.019)	0.091 (0.011)	0.315 (0.030)	0.045 (0.016)
<0.5,0.2>	0.510 (0.008)	0.085 (0.007)	0.595 (0.023)	0.127 (0.020)	0.301 (0.030)	0.034 (0.011)
<0.8,0.2>	0.532 (0.005)	0.133 (0.009)	0.547 (0.023)	0.120 (0.022)	0.306 (0.017)	0.044 (0.015)
<0.2,0.5>	0.312 (0.013)	0.027 (0.003)	0.502 (0.023)	0.071 (0.009)	0.302 (0.018)	0.039 (0.012)
<0.8,0.5>	0.345 (0.011)	0.086 (0.009)	0.466 (0.033)	0.089 (0.017)	0.289 (0.024)	0.037 (0.011)
<0.2,1.0>	0.133 (0.019)	0.013 (0.004)	0.336 (0.021)	0.039 (0.007)	0.295 (0.024)	0.039 (0.011)
<0.5,0.8>	0.142 (0.014)	0.020 (0.006)	0.204 (0.023)	0.024 (0.004)	0.295 (0.022)	0.043 (0.012)
<0.8,0.8>	0.151 (0.023)	0.024 (0.008)	0.300 (0.037)	0.041 (0.005)	0.292 (0.037)	0.037 (0.008)

6 DISCUSSION

In this paper, the feature defined as willingness to interact has been investigated through multi-agent simulations. The evaluation is defined as the percentage of completed tasks. The results show the benefit of dynamic willingness to interact, using a base-line update strategy, as compared to its static counterpart in the implemented scenario. Thus, the simulations show that the performance of the agents improves, i.e. percentage of completed tasks, when willingness becomes a dynamic function, in all difficulty levels. In this context, difficulty level refers to the fraction of tasks that require assistance from other agents to be finished. Moreover, the effects of the update strategy on the dynamic willingness to interact were investigated. The results show that, in the case of base-line update, it is possible to reach a maximum performance under specific initial conditions. Whilst in the case of continuous update, the performance is con-

Table 3: Task completion percentages for $P_D = 0.8$. Notation interpretation as in Table 1.

	Static		Base-line		Continuous	
$\langle\delta_0,\gamma_0\rangle$	s_A	s_D	$d1_A$	$d1_D$	$d2_A$	$d2_D$
<0.0,0.0>	na	0	0.425 (0.019)	0.023 (0.005)	0.156 (0.017)	0.014 (0.006)
<0.5,0.0>	na	0	0.361 (0.019)	0.044 (0.008)	0.155 (0.021)	0.013 (0.004)
<1.0,0.0>	na	0	0.297 (0.025)	0.033 (0.002)	0.150 (0.022)	0.012 (0.006)
<0.0,0.5>	na	0	0.353 (0.015)	0.013 (0.004)	0.179 (0.038)	0.013 (0.007)
<0.5,0.5>	0.238 (0.017)	0.041 (0.008)	0.318 (0.022)	0.037 (0.006)	0.149 (0.033)	0.010 (0.004)
<1.0,0.5>	0.251 (0.016)	0.065 (0.011)	0.257 (0.021)	0.029 (0.008)	0.146 (0.024)	0.012 (0.004)
<0.0,1.0>	na	0	0.157 (0.024)	0.005 (0.003)	0.167 (0.039)	0.013 (0.004)
<0.5,1.0>	na	0	0.127 (0.017)	0.010 (0.002)	0.147 (0.035)	0.011 (0.004)
<1.0,1.0>	na	0	0.107 (0.012)	0.009 (0.003)	0.143 (0.022)	0.011 (0.004)
<0.2,0.2>	0.336 (0.005)	0.017 (0.002)	0.399 (0.018)	0.035 (0.005)	0.166 (0.030)	0.011 (0.006)
<0.5,0.2>	0.350 (0.010)	0.045 (0.005)	0.376 (0.013)	0.048 (0.007)	0.162 (0.033)	0.013 (0.003)
<0.8,0.2>	0.363 (0.013)	0.068 (0.005)	0.318 (0.018)	0.041 (0.008)	0.152 (0.027)	0.009 (0.004)
<0.2,0.5>	0.220 (0.009)	0.016 (0.004)	0.343 (0.016)	0.030 (0.005)	0.164 (0.029)	0.013 (0.006)
<0.8,0.5>	0.244 (0.014)	0.054 (0.009)	0.266 (0.024)	0.033 (0.006)	0.150 (0.026)	0.009 (0.005)
<0.2,1.0>	0.096 (0.008)	0.008 (0.002)	0.220 (0.013)	0.018 (0.006)	0.157 (0.024)	0.013 (0.004)
<0.5,0.8>	0.102 (0.015)	0.013 (0.005)	0.127 (0.017)	0.010 (0.002)	0.147 (0.035)	0.011 (0.004)
<0.8,0.8>	0.108 (0.011)	0.019 (0.003)	0.174 (0.016)	0.016 (0.005)	0.141 (0.019)	0.012 (0.004)

sistently worse. The simulation results seem to point to the conclusion that a high willingness to ask for help hinders the performance of the agents. Thus, the best configuration for the agents is the one in which $\langle\delta_0 = 0.8,\gamma_0 = 0.2\rangle$, and $\langle\delta_0 = 1.0,\gamma_0 = 0.0\rangle$ (for low difficulty level), which change on run-time with a base-line update strategy. If all agents start with these configurations, it is possible to predict results similar to the ones presented here.

The agent model discussed in this paper is relevant for different application domains, both as software (distributed software services, social/behavioral population models), and hardware solutions (search and rescue, precision agriculture, collaborative industrial robots). The latter explains why battery is taken as a factor in the reasoning of the agent, which could be a robot. Thus, the intention is to keep the model as general as possible. In these scenarios heterogeneous agents/robots need to interact to exchange information or assist each other. It might not always be the case that human operators are available to take control or give advice. When agents/robots are dispatched far from the operators, communication links can become unreliable. In such cases, the agents will need to solve the problems by relying and helping one another in order to complete their tasks. Adaptive autonomy, realized through the willingness to interact, can be a mean to approach the issues of when and with whom to interact, given specific circumstances.

Future work will address the following issues. First, the computational model of the willingness to interact needs to be refined in order to include different weights for the different factors. It might be that different application domains, require different weights for these factors. This paper considered such weights to be equal to each other for the sake of simplicity. Secondly, as mentioned in Section 3, an agent could map an ability to a value between 0 *and* 1, which could reflect the accuracy for such ability (similar reasoning could be applied to resources as well). This means that, either the agent has an ability which it can use with accuracy *x%*, or it does not have the ability at all. Whether the *x%* value is acceptable depends on the particular task, and it should have an appropriate influence on the willingness to interact. Therefore, this granularity needs to be taken into account. Also, the impact of factors such as: environmental risk, task trade-off, and task progress need to be investigated further with dedicated simulations, while keeping the other factors fixed.

ACKNOWLEDGMENTS

The research leading to the presented results has been undertaken within the research profile DPAC – Dependable Platforms for Autonomous Systems and Control project, funded by the Swedish Knowledge Foundation (the second and third authors). In part it is also funded by the Erasmus Mundus scheme EUROWEB+ (the first author).

APPENDIX

The source code to reproduce the simulations described in this paper is publicly available on github: *https://github.com/gitting-around/gitagent_18.git*. The simulations were conducted using the Microsoft Azure Cloud services, with machines with the following characteristics: Ubuntu 14.04 LTS, Standard B4ms (4 vcpus, 16 GB memory).

REFERENCES

Bradshaw, J. M., Jung, H., Kulkarni, S., Johnson, M., Feltovich, P., Allen, J., Bunch, L., Chambers, N., Galescu, L., Jeffers, R., et al. (2005). Kaa: policy-based explorations of a richer model for adjustable autonomy. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 214–221. ACM.

Brookshire, J., Singh, S., and Simmons, R. (2004). Preliminary results in sliding autonomy for assembly by coordinated teams. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 706–711. IEEE.

Castelfranchi, C. (2000). Founding agent's 'autonomy' on dependence theory. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pages 353–357. IOS Press.

da Silva, F. L., Glatt, R., and Costa, A. H. R. (2017). Simultaneously learning and advising in multiagent reinforcement learning. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1100–1108. International Foundation for Autonomous Agents and Multiagent Systems.

Fong, T., Thorpe, C., and Baur, C. (2001). *Collaborative control: A robot-centric model for vehicle teleoperation*, volume 1. Carnegie Mellon University, The Robotics Institute.

Frasheri, M., Çürüklü, B., and Ekström, M. (2017a). Analysis of perceived helpfulness in adaptive autonomous agent populations. *LNCS Transactions on Computational Collective Intelligence*.

Frasheri, M., Çürüklü, B., and Ekström, M. (2017b). Towards collaborative adaptive autonomous agents. In *9th International Conference on Agents and Artificial Intelligence 2017 ICAART, 24 Feb 2017, Porto, Portugal*.

Hardin, B. and Goodrich, M. A. (2009). On using mixed-initiative control: A perspective for managing large-scale robotic teams. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 165–172. ACM.

Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., Van Riemsdijk, B., and Sierhuis, M. (2011). The fundamental principle of coactive design: Interdependence must shape autonomy. In *Coordination, organizations, institutions, and norms in agent systems VI*, pages 172–191. Springer.

Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., Van Riemsdijk, M. B., and Sierhuis, M. (2014). Coactive design: Designing support for interdependence in joint activity. *Journal of Human-Robot Interaction, 3 (1), 2014*.

Kim, S., Kim, M., Lee, J., Hwang, S., Chae, J., Park, B., Cho, H., Sim, J., Jung, J., Lee, H., et al. (2016). Team snu's control strategies for enhancing a robot's capability: Lessons from the 2015 darpa robotics challenge finals. *Journal of Field Robotics*.

Laird, J. E. (2012). *The Soar cognitive architecture*. MIT Press.

Martin, C. and Barber, K. S. (2006). Adaptive decision-making frameworks for dynamic multi-agent organizational change. *Autonomous Agents and Multi-Agent Systems*, 13(3):391–428.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 30(3):286–297.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe.

Schurr, N., Marecki, J., and Tambe, M. (2009). Improving adjustable autonomy strategies for time-critical domains. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 353–360. International Foundation for Autonomous Agents and Multiagent Systems.

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

van der Vecht, B., Dignum, F., and Meyer, J. C. (2009). Autonomy and coordination: Controlling external influences on decision making. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 92–95. IEEE.

Vernon, D., Metta, G., and Sandini, G. (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE transactions on evolutionary computation*, 11(2):151–180.