Towards Collaborative Adaptive Autonomous Agents

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

Keywords: Adaptive Autonomy, Autonomous Systems, Agent Architectures, Collaborative Agents, Multi-robot Systems.

Abstract: Adaptive autonomy enables agents operating in an environment to change, or adapt, their autonomy levels by relying on tasks executed by others. Moreover, tasks could be delegated between agents, and as a result decision-making concerning them could also be delegated. In this work, adaptive autonomy is modeled through the willingness of agents to cooperate in order to complete abstract tasks, the latter with varying levels of dependencies between them. Furthermore, it is sustained that adaptive autonomy should be considered at an agent's architectural level. Thus the aim of this paper is two-fold. Firstly, the initial concept of an agent architecture is proposed and discussed from an agent interaction perspective. Secondly, the relations between static values of willingness to help, dependencies between tasks and overall usefulness of the agents' population are analysed. The results show that a unselfish population will complete more tasks than a selfish one for low dependency degrees. However, as the latter increases more tasks are dropped, and consequently the utility of the population degrades. Utility is measured by the number of tasks that the population completes during run-time. Finally, it is shown that agents are able to finish more tasks by dynamically changing their willingness to cooperate.

1 INTRODUCTION

Adaptive autonomy (AA) refers to a specific type of an autonomous system, in which the level of autonomy is chosen by the system itself (Hardin and Goodrich, 2009). In general the changes of autonomy levels of a software agent are set either by (i) the software agent itself, (ii) other software agents that it is interacting with, or lastly by (iii) a human operator (in the remaining text agent is used instead of software agents for the sake of simplicity). Moreover, such decision could also be shared between human operators and agents. As a result, alongside adaptive autonomy, other common terminology includes the following: adjustable autonomy, mixed-initiative interaction, collaborative control, and sliding autonomy. Each of them addresses changes in autonomy from different perspectives. From one view, adjustable autonomy enables the human operator to change the agent's autonomy level (Hardin and Goodrich, 2009). The emphasis in this definition is on the party which has the authority to make such changes. On the other hand, the term is also employed to refer to all different ways in which decisions on autonomy are shared between human and agents (Johnson, et al., 2011). In mixed-initiative interactions (Hardin and Goodrich, 2009), both human and machine are able to trigger changes of the autonomy level. Specifically, the machine attempts to keep the highest level of autonomy, but lowers it in case the human intervenes. In collaborative control (Fong, et al., 2001) humans and agents solve their inconsistencies through dialogue. The human operator is responsible for defining the high-level goals and objectives to be fulfilled. The agents are not autonomous with respect to deciding on their own goals, but can still make autonomous decisions during execution. Another approach is sliding autonomy (Brookshire, et al., 2004). Two extreme modes are assumed, i.e. tele-operation and full autonomy and the level of autonomy could be switched between them on the task level. The human operator is able to take control of some tasks without taking control of the whole system.

Autonomy itself has been defined in connection to the notions of dependency and power relations (Castelfranchi, 2000). Moreover, in the aforementioned work, a distinction is made between autonomy as a function of interaction with the environment versus interaction with other agents. The

Frasheri M., CÃijrÃijklÃij B. and Ekstroem M.
Towards Collaborative Adaptive Autonomous Agents.
DOI: 10.5220/0006195500780087
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 78-87
ISBN: 978-989-758-219-6

HIGH	10. The computer decides everything, acts autonomously, ignoring the human.
	9. informs the human only if it, the computer, decides to
	8. informs the human only if asked, or
	7. executes automatically, then necessarily informs the human, and
	6. allows the human a restrcited time to veto before automatic execution, or
	5. executes the suggestion if the human approves, or
	4. suggests an alternative
	3. narrows the selection down to a few, or
	2. The computer offers a complete set of decision/action alternatives, or
LOW	1. The computer offers no assistance: human must take all the decisions and actions

Figure 1: 10 levels of autonomy (Parasuraman et al., 2000).

former indicates that an agent has some autonomy from the stimuli it gets from the environment, i.e. it is not merely a reactive entity. The latter refers to autonomy – or independence – from other agents. In case agent *A* has needs that could be fulfilled by an agent *B*, then *A* is dependent on *B* for those specific needs. The latter could refer to a need for information, a resource, or a goal. *B* could provide them either directly, e.g. by physically providing a resource, or by granting permission.

In this paper it is assumed that changes in autonomy stem from the dependency relations between agents. An agent facing some sort of dependency will ask another agent for assistance. The other agent will decide whether to engage itself or not based on its willingness to cooperate. The agents decide themselves when and if to ask or give assistance to one another, as a result it could be assumed that the decision to adapt autonomy is internal to the agents.

The rest of the paper is organized as follows. A short account on related work is provided in Section 2. Thereafter, an initial concept of the agent architecture is proposed in Section 3, which focuses on the agent interactions, and decision-making mechanisms based on the willingness to cooperate. The relations of the latter with the degree of dependencies between tasks and the utility of the agent population are depicted in Section 4. Moreover, it is shown that enabling agents to dynamically change their willingness to cooperate helps them to cope better in different situations. Finally, a discussion is provided on this work, and possible future ones.

2 RELATED WORK

Shared decision making on autonomy between agents and humans has been modeled in various ways. The classical concept (Parasuraman, et al., 2000), defines 10 levels of autonomy: from the lowest, in which the machine has no decision-making powers, to the highest, in which the machine is fully autonomous and potentially opaque to the user (Figure 1). On the other hand, more recent approaches are inspired from human collaboration in teams, e.g. Coactive Design (Johnson, et al., 2011). The focus is on soft interdependencies between agents which are working in a team towards some collective goal. Soft interdependencies are not crucial for success, but are considered to help the agents be more efficient while executing some task. On the other hand, hard interdependencies are crucial for the successful outcome of a task. From this perspective, earlier works are considered as being autonomy centred, i.e. the focus lies on self-sufficiency and self-directedness, and not on the interdependencies between the agents. Self-sufficiency refers to the agent's ability to take care of itself, whereas self-directedness refers to the agent's free will (Johnson, et al., 2011).

Several works investigate the performance of the different forms of shared decision-making between agents themselves and humans. Experiments by Barber *et al.* (Barber, et al., 2000) are conducted with different decision making frameworks, i.e. master-command driven, locally autonomous, and consensus, which are applied in scripted environmental conditions. The frameworks affect the agents at the task level. For instance, in the master-command case, an agent *A* (master) with authority over *B* can assign tasks to *B*, which the latter is required to perform. Agents become locally autonomous – they make decisions by themselves – when the communication is down. In the case of consensus, there is no leader, consequently agents have to reach an agreement. The authors' scenario involves agents which manage radio frequencies on military ships; no humans are involved. During the execution of the environmental scripts, the best decision making framework is applied in each case – the latter is chosen based on results from a previous study. It is shown that a system which dynamically switches between decision making frameworks performs better than the same system under one decision making framework.

AA, adjustable autonomy and mixed-initiative interaction are compared in search and rescue simulation environments by Hardin & Goodrich (Hardin and Goodrich, 2009). In their experiments, mixed-initiative interaction performs better than the other two, in terms of survivors found in the simulated environment.

Experiments in shared decision making between humans and a complex autonomous system – both are to coordinate teams of robots – are discussed by Barnes *et al.* (Barnes, et al., 2015). Three levels of autonomy are considered, either the human makes the decision with help, or the agent makes the decision alone, or the human makes the decision alone. They argue that shared autonomy between human and agent should be tailored according to the strengths and weaknesses of each. Also, the level of autonomy could be influenced by the workload of the operator at a given time.

Other work is directed toward developing policy systems that accommodate adaptive behaviour. The Kaa policy system (Bradshaw, et al., 2005) builds on top of the existing KaOS system – the latter implements policy services to regulate behaviour in a multi-agent system. Kaa adds support for adjustable autonomy by allowing the policies to be changed during runtime. A central coordinator takes the agents' requests for adjusting autonomy in given circumstances and decides whether to override the default policy for a given time. In case Kaa cannot make a decision it will ask for the human's feedback. Kaa was developed in the framework of the Naval Automation and Information Management Technology project, in an application concerning naval de-mining operations.

Adjustable autonomy is also considered in terms of meeting real-time requirements in a simulated environment where a human operator and 6 fire engines have to cooperate whilst sharing resources to extinguish fires (Schurr, et al., 2009). The RIAACT model (resolving inconsistencies in adjustable autonomy in continuous time) is proposed, which handles the resolution of inconsistencies between the operator and agents, allows the agents to plan in continuous time, and makes interruptible actions possible. They show that RIAACT can raise the performance of a human-multi-agent system.

3 THE AGENT MODEL

The adaptive autonomy approach presented in this work does not consider yet specific sensory/motor specifications, or concrete types of tasks. The focus is on the interaction between agents and the decision making mechanisms that would allow them to ask and give assistance, and the way they could do so without compromising their performance measures, e.g. utility. In principle, these measures could be subjective to each agent.

In the proposed model an agent could be in one of three states: *idle*, *execute*, and *interact* (Figure 2), and is associated with a willingness to assist others – expressed as a probability. Messages from other agents represent the input, and are handled in the message processing unit (Msg PU). The agent sends its broadcasts to others through the same unit.

Imagine that an agent is in either *idle* or *execute* state. When it receives a request for assistance it will change its state (adapt) to the *interact* state. The outcome of the decision made in *interact* will send the agent either into *idle* or *execute* with the new task. In the latter case, after a task is finished, the agent will adapt to *idle* again – valid for both success and failure outcomes of the job.

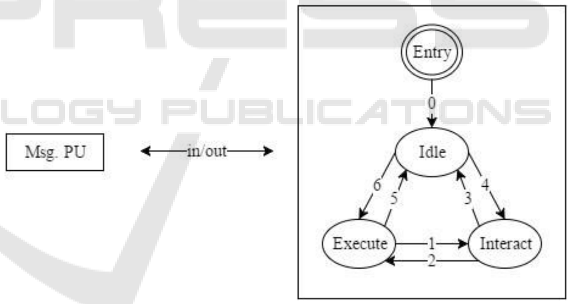

Figure 2: States of the agent, and possible transitions between them.

In *idle* (Figure 3) the agent is not engaged in any particular task, nonetheless it can decide whether to adopt and start the execution of a new one, e.g. it could generate a task to explore its surroundings. In principle, based on its perceptions from the exploration and set of its capabilities, the agent could possibly create another task for itself when it goes back to *idle*.

When the agent chooses to do a task, it will switch to the *execute* state (Figure 4). It is assumed that if the agent is not interrupted, then it will finish any task it starts. As a result, it is possible to focus only on the effects of agents assisting each other.

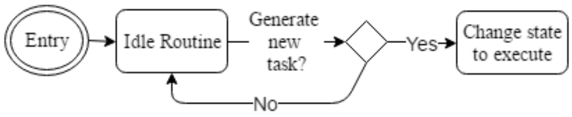

Figure 3: The agent starts its life-cycle in the idle state. It is possible for the agent to decide starting the execution of a task – in that case it will adapt to execute. Otherwise the agent will remain in idle until it decides to start performing a new task.

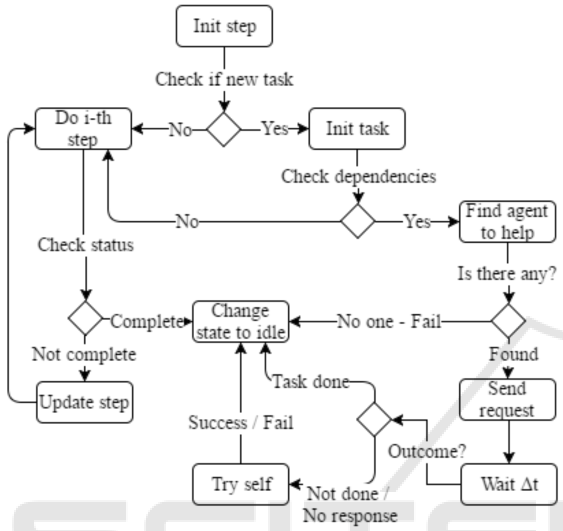

Figure 4: In the execute state the agent will perform all execution steps related to the task. If a task can be performed independently then the agent will execute the iteration steps until it finishes and succeeds. If the task cannot be done independently, the agent chooses whom to ask for assistance and sends a request. The agent will wait for a certain amount of time before giving up on getting help. In such case it will try to achieve the task by itself with a probability prob. Regardless of the outcome – success or failure, the agent will go to idle.

As aforementioned, it is possible for an agent to receive a request for assistance from another while either being in *idle* or *execute* state. In this case the agent will transition into the *interact* state (Figure 5), and other tasks will be left on-hold. Whilst in this state the agent cannot be interrupted – the process of making a decision is an atomic one. It follows that requests are processed one at a time. The agent returns from the *interact* state with a decision of what to do. It may either drop the past activitiy and pursue the new task, or it discards the request and continues where it left off. Such decision is made based on the willingness to cooperate.

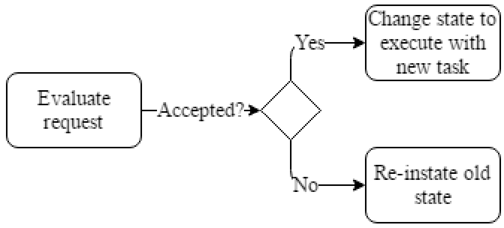

Figure 5: In the interact state the agent will evaluate the request and based on its willingness to cooperate will decide whether to accept it or not.

Agents keep a profile of one another, based on the outcomes of past interactions (agents are not aware of how they are profiled by others). Such profile contains the following: the degree of perceived helpfulness, a set of capabilities and respective expertise. In this work, an agent *A* chooses to rely on an agent *B* based on the latter's perceived helpfulness. Thereafter, it will wait for a finite amount of time for *B* to respond. In case there is no response, *A* will do the following: give up on *B*, update the corresponding profile, and try to carry out the task itself with a low success rate (Figure 4). It would also be reasonable for *A* to first try by itself. It could also be that *A*, upon giving up on *B*, chooses some other agent *C* to ask for help. On the other hand, *B* keeps track of how good it is doing at the moment of the request. In this paper, if *B* concludes that it has dropped too many tasks – explained further in Section 4.2 – then it will lower its willingness to cooperate with *A* at that point. In the opposite case, *B* will raise its cooperation level, thus will become more inclined to help *A*.

3.1 Interactions Between Agents

Dependencies between agents can either arise with time, or they can be known in advance. In the former case, the agent might discover them either at the beginning of the task, or while the task is in progress. In order to increase their chance of a successful outcome, i.e. task completion, the agents will need to interact with each other. Agents can interact on several levels, as follows:

- **Non-committal interaction.** Agent *A* could broadcast pieces of information it deems important to other agents, i.e. its presence and capabilities, and messages of the form 'path x1 to x2 blocked'. Other agents could decide whether or not to accept this broadcast. When *A* sends such broadcasts it is not trying to establish a dialogue with others around it. Therefore, it does not expect any response or commitment to the message. The other agents could also evaluate how trustworthy agent *A* is, based on the validity of its broadcasts. Specifically, (i) is the information provided useful, and (ii) is it true?

- **One-to-one dialogue.** Agent *A* has knowledge gaps. Consequently, it asks agent *B* for specific information to address this issue. Also in this case, agent *A* could evaluate the validity of the responses of *B*, as in the non-committal broadcast. In addition, the overall helpfulness of *B* could also be estimated.
- **One-to-one delegation.** Agent *A* asks agent *B* to perform a task on which *A* depends on. It could also be that agent *A* is still able to perform its own task, however, with lower probability of success. Agent *B* will evaluate the request from agent *A* and decide whether it will adopt it as its own. As in the previous cases, *A* can also judge the behaviour of *B*, in terms of (i) the overall helpfulness of *B* and (ii) the quality of the outcome produced by *B*.
- **One-to-many dialogue/delegation.** In this case, a chain of one-to-one interaction emerges. There is another way to understand the one-to-many scenario. Agent *A* engages in interaction with several other agents, at the same time over the same task. This means that agent *A* can ask from each agent a different subtask to be performed, which will affect the success of its own task.

Each case discussed above could refer to hard or soft interdependencies as defined by Johnson *et al.* (Johnson, et al., 2011). For instance, if the non-committal broadcast contains an alarm message, then it is vital to the well-being of the other agents. On the other hand, if the message is the aforementioned 'path x1 to x2 blocked', then disregarding it might delay some mission without compromising its success. In the same way it could be argued for all the other cases.

Differently from Barber *et al.* (Barber, et al., 2000), in the present work an agent decides by itself if it will aid another agent at any point in time. Consequently, task delegation from an agent *A* to *B*, first has to be accepted by *B*.

3.2 Agent Organization and Autonomy

An agent population could either be organized in a hierarchy, or as peers. It might be possible for some structure to emerge in the latter case, e.g. the most successful agents go up in the ranks. Environmental conditions could also be used to predict the best hierarchy (Barber, et al., 2000). The type of organization will influence how an agent's autonomy is affected by the interaction with other agents.

Let us assume an agent *A* which is a superior of agent *B*, i.e. agent *A* has the power to delegate to *B* any task it sees fit, e.g. task x_i. In principle, *A* could be fully capable of performing x_i by itself. However, in order to conserve its resources, it chooses to delegate such task to *B*. There are two possibilities for *B*. It either has no choice at all but to execute task x_i, or it might have some degree of independence to refute doing x_i, in case the task could have catastrophic consequences that *A* has not foreseen. In general, *A* can and will interfere in the agenda of *B*, and *B* has to comply with *A* up to some degree. Overall, *B* depends on the will of *A*.

When agents *A* and *B* are peers, *A* does not have any authority over *B*. If during its lifetime agent *A* depends on *B* for some tasks, then *A* will make a request for assistance to *B*. Whether *B* decides to intervene or not will depend on its willingness to cooperate. Agent *A* will depend on the will of *B*. If *B* has perceived *A* to be helpful in the past, then it might be more difficult for *B* to reject the request from *A*. In general a more willing agent might be easier to interfere with. On the other hand, *B* might not be driven by unselfish motives. It can in fact decide to help *A* in order to make a better case for itself, should it need the help of *A* in the future.

The relation of dependence is present in both situations. Moreover, choosing to depend and delegate always constitutes a risk (Castelfranchi and Falcone, 1998). Even if *A* is the superior of *B*, by delegating it depends on *B*. Even if *A* could perform the task by itself, the failure of *B* will delay its own success, i.e. if the outcome is expected at a certain time, then the failure of *B* might entail the failure of *A*. Also, if *A* is not able to do the task by itself, then it will be even more dependent on *B*. As a result, the changes of autonomy may become blurred. In this paper, the agents are considered to be peers. Consequently, when *A* asks *B* for assistance with respect to a task x_i, it is deciding to depend on *B*, and thus it is lowering its autonomy over x_i.

4 EXPERIMENT

4.1 Setup

In this paper the simulation model is tested against values of dependency degrees and willingness to cooperate (Δ), in order to investigate the utility of the agent population. Utility is measured in terms of the number of dependent tasks completed as a whole (completion degree *CD*), and the total number of unfinished tasks (dropout degree *DD*). The degree of dependencies represents the percentage of tasks which are dependent on other tasks in order to have a higher chance of being completed – also referred to as dependent tasks. The parameter Δ represents the probability that an agent *A* will accept to help an agent *B*, upon receiving a request from *B*.

In the simulation a task is defined by the following characteristics: energy levels, reward, and dependencies on other tasks, i.e. task x_i depends on task x_j. This list is not exhaustive. An agent is assumed to have a list of tasks it can perform, with value mappings between each task and the characteristics described. This abstraction could be useful even if tasks are concretely defined. On every run, each agent has the same set of tasks that it provides. In every set, there are tasks that depend on other tasks and tasks that the agent can perform alone. In total there are 10 different tasks. More than one agent can do each task. This is to ensure the diversity of individuals with which an agent could interact.

In this experiment only two types of the interactions discussed above are used: the non-committal broadcast and the one-to-one delegation. Agents make themselves and their list of tasks known to each other through the non-committal broadcast. One the other hand, they make requests to each other through the one-to-one delegation. The Robot Operating System (ROS) (Quigley, et al., 2009) is used to simulate agents and their interaction through services and publish/subscribe mechanisms. The one-to-one delegation is implemented through ROS services.

It is important to note that agents in the population are not working to achieve the same set of goals, in other words no global objective/goal is assumed. Each agent has its own agenda; nevertheless, its capabilities could be of use for other agents too.

Three sets of trials were conducted. A set of trials is composed of 3 independent simulation runs for the same population size (*popsize)*, degree of dependencies, and Δ. In the first set, simulations are run for *popsize* = 10, alternatively *popsize* = 30, and static Δ. The percentages of tasks that depend on other tasks are in the segment [10%, 25%, 50%, 75%, 100%]. The parameter Δ is taken from the segment [0.0, 0.25, 0.5, 0.75, 1.0].

These values capture different degrees of dependencies and selfishness in the agent population. The experiments were conducted for each combination of Δ with each dependency degree.

The second set of trials is conducted with a *popsize* = 10, and a finer resolution of the Δ segment: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]. The segment for dependencies is the same as in the first trial.

The third set of trials is conducted again with *popsize* = 10, with a dynamic Δ that changes during runtime on each interaction. Simulations are run for several initial values of Δ, in the segment [0.0, 0.3 0.5, 0.7, 1.0]. Two cases are studied, only one agent has a dynamic Δ, and all agents have dynamic Δ. The segment for dependencies is the same as in the other trials.

During any simulation, at a point in time t, agent *A* might decide to do a task x_i, or receive a request for such task. In the case x_i depends on some task x_j, agent *A* chooses whom to ask for assistance by consulting its list of known agents. In the first steps of the simulation the agent will make the selection randomly. Consequently, it will either select the one which it has perceived as more helpful in the past, or randomly with a probability equal to 0.3. This value is chosen arbitrarily in order to help the agent explore its options. Agent *A* computes the perceived helpfulness (*PH*) of some agent *B*, by comparing the number of times it has gotten a response over the total number of requests made to *B* (Equation 1):

$$\mathrm{PH} = \frac{\text{Handled Requests}}{\text{Total Requests}} \tag{1}$$

This is relevant because agent *B*, upon receiving and adopting some other task, i.e. from *C*, will drop the request of *A* and continue. After a time out, *A* assumes that its request has been dropped. If *B* does indeed perform x_j, then *A* is considered to have succeeded. Otherwise *A* will succeed by itself with *prob* = 0.3.

4.2 Results

The simulation results, visualized as heat maps, show how the utility measures relate to the dependency degree and willingness to cooperate (Figures 6a-6h). The x-axis represents the degree of dependency expressed in percentage, whereas the y-axis represents the willingness to cooperate. The colour represents the degree of completed dependent tasks averaged over 3 trials. The completion degree (*CD*) for each agent is calculated as seen in Equation 2:

$$\mathrm{CD} = \frac{\text{Depend Tasks Completed}}{\text{Depend Tasks Attempted}} \tag{2}$$

On the other hand, the dropout degree (*DD*) for each agent is calculated in Equation 3:

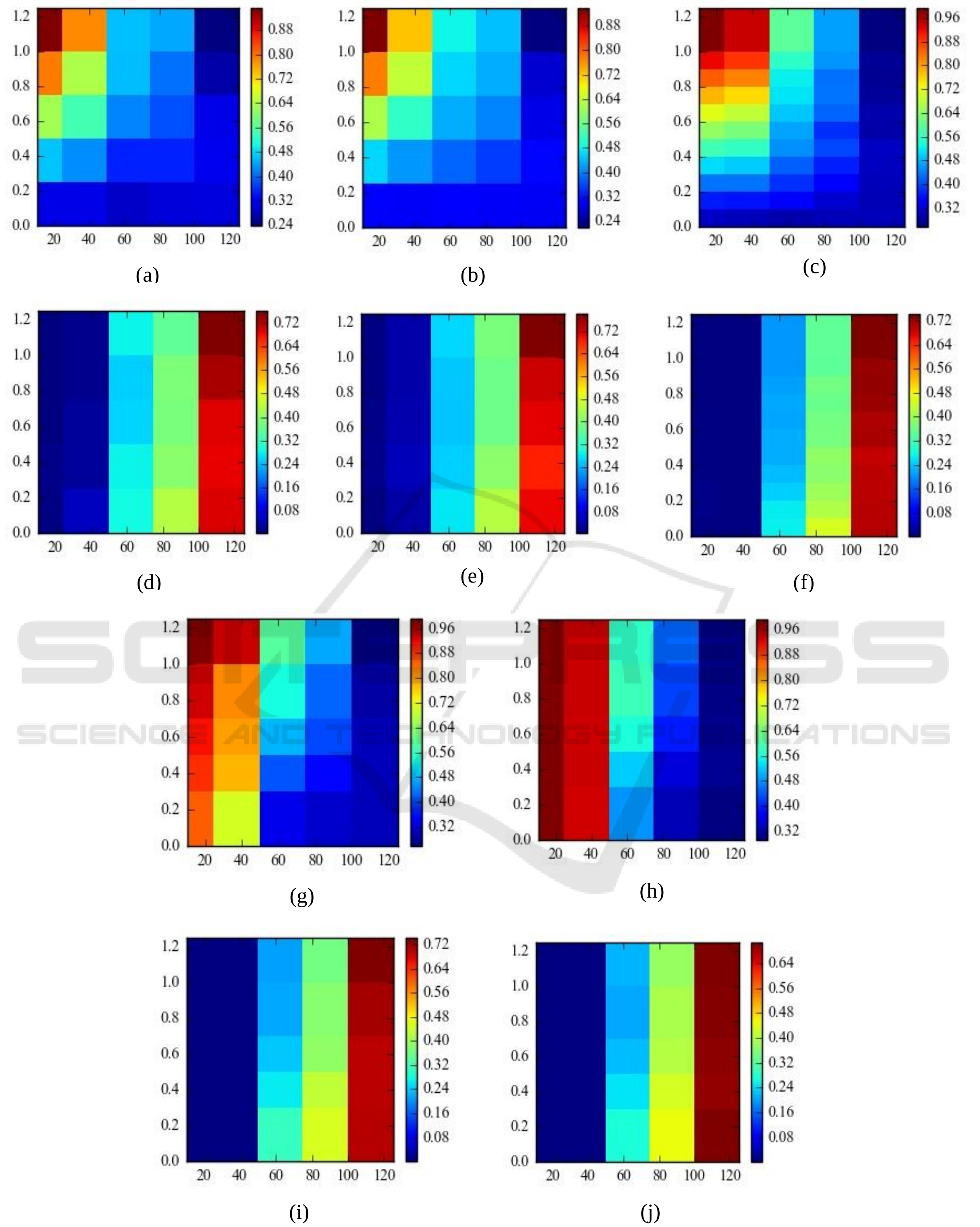

Figure 6: Heat maps of CD and DD utility measures, for simulations with static Δ and dynamic Δ, and different popsize. (colors on the blue side of the spectrum represent low values, whilst the ones on the red side represent high values) (a) CD for popsize = 10 with static Δ. (b) CD for popsize = 30 with static Δ. (c) CD for popsize = 10 with finer resolution of static Δ. (d) DD for popsize = 10 with static Δ. (e) DD for popsize = 30 with static Δ. (f) DD for popsize = 10 with finer resolution of static Δ. (g) CD for popsize = 10, one agent with dynamic Δ. (h) CD for popsize = 10, all agents with dynamic Δ. (i) DD for popsize = 10, one agent with dynamic Δ. (j) DD for popsize = 10, all agents with dynamic Δ.

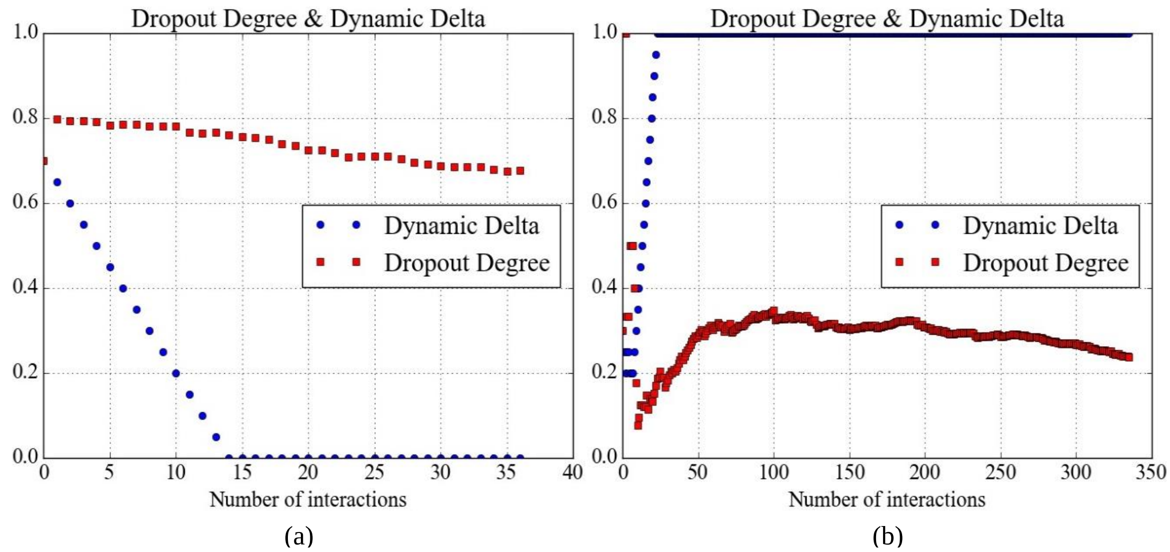

Figure 7: Simulations under different conditions of dependency degree and different Δ_{init} show that (a) for $\Delta_{init} = 0.7$ and dependency degree = 75% the agent becomes more selfish, (b) whereas for $\Delta_{init} = 0.3$ and dependency degree = 50% the agent becomes less selfish.

$$\mathrm{DD} = \frac{\text{Tasks not Completed}}{\text{Tasks Attempted}} \tag{3}$$

The heat maps show the values for *CD* and *DD*, each summed over all the agents. The outcomes of the first set of trials are depicted in Figures 6a, 6b, 6d and 6e. In the case of low dependency degrees, agents with low Δ complete circa 0.3 of the dependent tasks, whereas those with higher Δ complete noticeably more with no relevant impact on DD. Results from the initial tests seem not dependent on *popsize* with respect to *CD* (Figures 6a and 6b) and *DD* (Figures 6c and 6d), thus *popsize* = 10 was used in the succeeding simulations. The utility measures are calculated through Equations 2 and 3.

The outcomes of the second set of trials are given in Figures 6c and 6f. The results using a finer resolution for Δ are consistent with the first set of trials.

In the case of dynamic Δ (third set of trials), in the y-axis its initial values are shown, Δ_{init} (Figures 6g-6j). It is observable how the agent population accomplishes more tasks – *CD* increases – for lower dependency degrees due to dynamic Δ. There is a noticeable difference between the results for static Δ and results for both cases with dynamic Δ: only one agent with dynamic Δ (Figures 6g and 6i) and all agents with dynamic Δ (Figures 6h and 6j). Moreover, the benefit of having all agents with dynamic Δ is observable. On the other hand, the value of *DD* increases in all cases with static and dynamic Δ, due to the increase of dependency degree. In the case the latter is 100%, all tasks depend on each other. Consequently, the value of *CD* is approximately equal to *prob*.

Changes of Δ for an agent with respect to the *DD* shows that adaptation of behaviour takes place (Figures 7a and 7b). In this specific experiment, two thresholds are considered, $\theta_{low} = 0.3$, and $\theta_{high} = 0.7$. If the value of *DD* is higher than θ_{high}, then the agent will decrease its Δ with a $\Delta_{step} = 0.05$. If it is lower than θ_{low}, the agent will increase its Δ with the same $\Delta_{step} = 0.05$. If the value of *DD* is between θ_{low} and θ_{high}, the agent will compare the current value with the one before last. In case the difference in absolute value is bigger than 0.01 the agent will update Δ. The value of Δ will increase if the value of *DD* has gone down, and decrease otherwise.

5 DISCUSSION

In this paper, the willingness to cooperate is used to model adaptive autonomy. An agent that asks for assistance is attempting to establish a dependency relation. The agent that accepts to give assistance establishes such a relation. The results show how the willingness to cooperate influences the utility of a population of agents. It is clear that selfish agents, as defined here, will only be as succesful as their individual potential allows them (Figures 6a-6c). On the other hand, unselfish agents can improve group utility up to a certain point. For low dependency degrees, they achieve more dependent tasks without compromising the dropout degree. When the dependencies become quite complex, due to the increase of tasks that require assistance, their utility degrades. In the latter case it seems quite reasonable to act more selfishly and rely more on oneself (Figure 7a). On the other hand, if one agent can afford to assist then it can adapt its behavior to that end (Figure 7b). A dynamic willingness to cooperate captures these shifts in behavior. As shown by the results in Section 4.2 (Figure 6g), even one agent with dynamic degree of willingness to help is able to positively impact the whole population.

In the simulations, the dropout degree served as a regulator. Agents were continuously keeping track of how many tasks they were concluding (each agent for itself) and based on that value their behavior adapted. Consequently, dependency relations are established with agents in need, based on current circumstances.

In other research areas, this kind of parameter is used to model risk tolerance (Cardoso and Oliveira, 2009). Agents which are representatives of business entities, are spawned with different willingness to sign contracts with other entities – the latter might be subject to fines. Fines are considered punishment for undesired behavior. The higher the fines, the higher the risk is of signing a contract with an agent.

On a different note, the dependency degree was kept fixed during a single run of the simulations. Therefore, it can be assumed that the dependencies are known in advance. However, this might not always be the case, because dependencies could also arise during the agent's lifespan. In principle, the model presented in this work does not make any restrictions for how dependencies should be.

Future research will be concerned with the further development of the agent model, and the establishment of an agent framework.

Firstly, the model will be expanded to include a willingness to ask for assistance which changes depending on the agent's chance of success if it would attempt the task by itself. As a result, autonomy will be shaped by both the willingness to cooperate and willingness to ask for assistance.

Secondly, the factors which should influence these parameters such as: health, reward, hierarchy, and trust, need to be taken into account. A general definition considers trust in terms of how much an agent will want to depend on another (Jøsang, et al., 2007). Integration of this dimension with the current model will aid the agents to make better choices about whom to give assistance, and whom to ask for it. The presence of a hierarchy, also creates interesting scenarios. As an example, in which cases should an agent obey its superior? The case in which the superior sends wrong information continuously is tackled by Vecht *et al.* (Vecht et al., 2009), which results in the agent taking more initiative. Additional scenarios could include a superior which is in conflict with agents of a higher rank than itself, or a superior which asks the agent to do tasks associated with low reward, thus not exploiting the agent's full capacity.

Lastly, the model will also be expanded to include two more auxiliary states, which are *regenerative* and *out_of_order*. The agent can go to *out_of_order* from any other state. If the agent attempts by itself to recover it will change its state to *regenerative*. In the case it does indeed recover it will go to *idle* and continue normal operation, otherwise it will return to *out_of_order*.

ACKNOWLEDGEMENTS

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 the third authors). In part it is also funded by the Erasmus Mundus scheme EUROWEB+ (the first author).

REFERENCES

Barber, S. K., Goel, A., Martin, C. E. (2000). Dynamic adaptive autonomy in multi-agent systems. *Journal of Experimental \& Theoretical Artificial Intelligence, 12*(2), 129-147.

Barnes, M. J., Chen, J. Y., Jentsch, F. (2015). Designing for Mixed-Initiative Interactions between Human and Autonomous Systems in Complex Environments. *IEEE International Conference on Systems, Man, and Cybernetics (SMC).*

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

Brookshire, J., Singh, S., Simmons, R. (2004). Preliminary results in sliding autonomy for coordinated teams. *Proceedings of The 2004 Spring Symposium Series.*

Cardoso, H. L., Oliveira, E. (2009). Adaptive deterrence sanctions in a normative framework. *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02.*

Castelfranchi, C. (2000). Founding agents' "autonomy" on dependence theory. *ECAI.*

Castelfranchi, C., Falcone, R. (1998). Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. *Multi Agent Systems, 1998. Proceedings. International Conference on. IEEE.*

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

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

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

Jøsang, A., Ismail, R., Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision support systems 43.2*, (pp. 618-644).

Parasuraman, R., Sheridan, T. B., 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.*

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., . . . Ng, A. Y. (2009). ROS: an open-source Robot Operating System. *ICRA workshop on open source software.*

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

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