Modeling Agents that Exhibit Variable Performance in a Collaborative Setting[*]

Ingrid Zukerman and Christian Guttmann

School of Computer Science and Software Engineering
Monash University
Clayton, VICTORIA 3800, Australia
`{ingrid,xtg}@csse.monash.edu.au`

Abstract. In a collaborative environment, knowledge about collaborators' skills is an important factor when determining which team members should perform a task. However, this knowledge may be incomplete or uncertain. In this paper, we extend our *ETAPP (Environment-Task-Agents-Policy-Protocol)* collaboration framework by modeling team members that exhibit non-deterministic performance, and comparing two alternative ways of using these models to assign agents to tasks. Our simulation-based evaluation shows that performance variability has a large impact on task performance, and that task performance is improved by consulting agent models built from a small number of observations of agents' recent performance.

1 Introduction

Collaboration plays a critical role when a team is striving for goals which are difficult to achieve by an individual. When a team is trying to perform a task, knowledge about collaborators' skills is necessary in order to determine which team members should perform which portions of the task. However, this knowledge may be incomplete, e.g., when collaborators are new to a team or face a new task, or uncertain, e.g., when the performance of collaborators is variable.

Our work focuses on how teams of agents make decisions when allocating tasks to team members. Group decisions are based on the opinions of team members, which in turn are based on their models of their collaborators. In previous work, we investigated joint decision making under the *ETAPP* framework [4], which expresses the collaboration of a team of agents in terms of five operating parameters (Environment, Task, Agents, Policy and Protocol). An important result from this research is that the main factor that influences task performance is the ability of the agents in a team to learn the models of team members from observations of their performance. However, this insight was obtained under a simplistic assumption whereby agents' performance is deterministic and invariant. This assumption implies that an agent's performance is the same every time a task is performed under the same conditions. Hence, an agent's level of performance under a particular set of conditions can be determined from a single observation of its actions.

---

[*] This research was supported in part by Linkage Grant LP0347470 from the Australian Research Council, and by an endowment from Hewlett Packard. The authors thank Yuval Marom for his assistance with the evaluation.

In this paper, we extend the ETAPP framework to build agent models under more realistic assumptions whereby the performance of agents is non-deterministic. That is, an agent's level of performance may change every time it performs a task due to the influence of factors that are not explicit. This extension requires a probabilistic representation of an agent's *task-related capabilities*, such as mean level of performance and stability; a procedure for building agent models from a sequence of observations; and a representation of an observer's *observation capacity*, i.e., how many observations can the observer remember. To illustrate these representations, a stable, high-performing agent exhibits a consistently high level of performance, while an unstable, medium-performing agent may sometimes perform well and other times poorly. An observer with a high observation capacity (it recalls many actions performed by team members) will derive an accurate model of the task-related capabilities of both types of agents, while an observer with a low observation capacity (it can recall only the last few observations) may still derive an accurate model of the stable agent, but its model of the unstable agent may be quite skewed.

In addition to our agent-modeling extensions, in this paper we compare two policies for assigning agents to tasks based on the proposals made by the agents in a team: an "optimistic" policy, which chooses the proposed agent with the most promising performance compared to all the other proposed agents [4], and a "majority" policy, which chooses the agent preferred by most team members.

We assessed the influence of these factors on task performance by means of a simulation where we varied the level of performance and stability of agents and their observation capacity, and applied the two policies for selecting agents for tasks.

Section 2 outlines the ETAPP framework and discusses the above mentioned extensions. Our evaluation is described in Section 3. In Section 4, we consider related research, followed by our conclusions.

2 The ETAPP Framework

The *ETAPP* [4] framework is designed for a decentralized setting, where agents act autonomously based on their knowledge of their team members. Our framework provides an explicit representation of five operating parameters of a collaboration: *Environment*, *Task*, *Agents*, *Policy* and *Protocol*. The *Task* given to the group is to be performed in the *Environment*, and the *Policy* and *Protocol* are procedures agreed upon by all the agents in the group, but performed autonomously (this is similar to abiding by certain rules in order to belong to a society). Central to the *ETAPP* framework is the idea that the real capabilities of the agents in a team are not known to the team members. Hence, individual agents employ models of collaborators' capabilities in order to estimate the contributions of team members to a task. The *Agents* component stores these models and the mechanisms to reason about them.

The elements of the ETAPP framework are outlined below (for more details see [4], but note that the *Agents* component has been substantially modified since that publication). Our extensions are described in Section 2.1.

An ***Environment*** $\mathcal{E}$ is a state space described by predicates, which represent properties of objects and relations between objects. A state in the environment describes the values of these predicates at a particular step in a collaboration.

A ***Task*** $\mathcal{T}$ is represented by a tuple with two elements $< EC_T, EF_T >$.

- EC_T specifies the *Evaluation Criteria* relevant to task $\mathcal{T}$, e.g., speed, quality or profit. The value for each criterion ranges between 0 and 1, where 0 corresponds to the worst possible performance and 1 corresponds to the optimal performance.
- EF_T denotes the *Evaluation Function* for the task, which specifies the weights assigned to the Evaluation Criteria (i.e., their relative importance to the task), and the way in which the values for these criteria are combined. For instance, the Evaluation Function $EF_T = \max \sum_{i=1}^{n} ec_i w_i$ specifies that the task should maximize a linear combination of n Evaluation Criteria, where w_i for $i = 1, \ldots, n$ are the weights assigned to these criteria. These weights range between 0 and 1, where 0 indicates no impact of a criterion on task performance, and 1 indicates maximum impact.

A team of ***Agents*** $\boldsymbol{\mathcal{A}}$ comprises agents $\{A_1, \ldots, A_m\}$, where m is the number of agents in $\mathcal{A}$. Individual agents have *Internal Resources (IR)*, which represent the task-related capabilities of an agent, and *Modeling Resources (MR)*, which represent the ability of an agent to model agents and reason about them.

The *IR* of an agent represent how well it can perform an action in terms of the Evaluation Criteria of the task. The values for *IR* range between 0 and 1, with 0 indicating the worst performance and 1 the best. For instance, if the Evaluation Criteria of a task are time and quality, and one of the actions in the environment is *drive*, then $IR_{A_i}(\textit{drive})$ represent the driving performance of agent A_i in terms of time and quality, i.e., $IR_{A_i}(\textit{drive}) = \{\textit{Perf}_{A_i}^{\text{time}}(\textit{drive}), \textit{Perf}_{A_i}^{\text{qual}}(\textit{drive})\}$. These capabilities are *not* directly observable (only the resultant behaviour can be observed). Hence, they cannot be used to propose agents for tasks (but they are necessary to simulate agent performance, Section 3).

The *MR* of an agent comprise its *Models (M)* of the Internal Resources of agents, the *Resource Limits (RL)* of the agent in question, and its *Reasoning Apparatus (RA)*.

- M_{A_i} are the models maintained by agent A_i to estimate IR_{A_j} for $j = 1, \ldots, m$. A_i's estimation of the capabilities of the agents in the team (including its own capabilities) may differ from their actual performance, in particular if agent A_i has never observed the team in action. This estimation may be updated as agent A_i observes the real performance of the agents in the team.
- The *RL* of an agent pertain to the amount of memory available to store models of agents, the agent's ability to update these models and generate proposals, and its ability to send and receive proposals (an agent that has become disconnected cannot send proposals, even if it can generate them).
- The *RA* consists of the processes applied by protocol $\mathcal{P}$, which enable an agent to act in an environment and interact with collaborators. These processes are: (1) proposing agents for an action (selecting agents from a list of candidates); (2) communicating this proposal to other agents; (3) applying a policy $\mathcal{P}_A$ to select a proposal from the communicated proposals; and (4) updating *M* based on the observed performance of the selected agent(s).

A ***Policy*** $\boldsymbol{\mathcal{P}_A}$ is a joint policy (adopted by all the agents in the team) for making decentralized group decisions about assigning agents to activities. As stated above, each agent proposes one or more agents for an action (according to its models *M* and its *RA*). Upon receiving all the proposals, each agent uses $\mathcal{P}_A$ for selecting one proposal.

In the future, we plan to compare this decentralized decision-making process with a centralized process, where a leader assesses proposals and communicates the outcome to team members. It is expected that the centralized process would require less communication and computations than our current procedure, while our procedure would be more resistant to being subverted by corrupt or incompetent leaders.

A ***Protocol*** $\mathcal{P}$ is a process that is followed by all the agents in the group to coordinate their interaction. According to this protocol, all agents generate a proposal and communicate it to the other agents. Next, each agent applies $\mathcal{P}_A$ to select a proposal, observes the performance of the selected agent(s), and updates its model(s) accordingly.

2.1 Extensions of ETAPP

In this paper, we extend the ETAPP framework along three agent-modeling dimensions – Internal Resources, Resource Limits and Reasoning Apparatus, and consider a new Policy for agent selection.

Internal Resources. As mentioned in Section 1, in the original framework we assumed that agents' performance is deterministic and invariant. Thus, $IR_{A_i}(\textit{action})$ comprise a set of numbers between 0 and 1. However, in realistic settings, agents exhibit variable performance (e.g., they could be having a bad day). We represent such a performance by means of a truncated normal distribution, where the mean represents the ability of an agent, and the standard deviation represents its stability (truncation is required so that we don't exceed the [0,1] thresholds). As stated above, these values are not observable, but they are the basis from which the observed performance of an agent is obtained during simulations.

Resource Limits. Originally, due to the deterministic performance of agents, a single observation of an agent's performance yielded an accurate model of its ability. However, this is clearly not the case if the performance is non-deterministic. In order to cope with this situation, we include *Observation Capacity (OC)* in our model of the Resource Limits of agents. This parameter, which is similar to attention span [8], specifies how many observations of the performance of each agent can be stored by an agent in its memory. When this limit is exceeded, the observer agent retains a window of the last K observations (forgetting the initial ones).

Reasoning Apparatus. The variable performance of agents also demands the implementation of a new model-updating procedure. As for Resource Limits, our previous single-update method is unlikely to yield accurate results. We therefore propose a simple procedure whereby an agent re-calculates the mean and standard deviation of the observed performance of an agent every time it performs an action. Notice, however, that the results obtained by this procedure are moderated by the observation capacity of the observing agent. That is, if the observing agent can remember only the last K observations of an agent's performance, then the mean and standard deviation are calculated from these observations.

Policy. In previous work, we implemented an *optimistic* policy, where the agent with the most promising performance was chosen for an action. We now consider the *majority* policy, where the agent that receives the most votes is chosen.

2.2 Example – Surf Rescue Scenario

In this section, we present an example that illustrates the ETAPP framework in the context of the Surf Rescue (SR) scenario used in our simulation-based evaluation (Section 3). In this scenario, the environment $\mathcal{E}$ consists of the *beach* and the *ocean*, and the task $\mathcal{T}$ is to rescue a distressed person (DP) in the shortest time possible. This means that the set of evaluation criteria is $EC_T = \{ec_{time}\}$, and the evaluation function is $EF_T = \max\{ec_{time}\}$ (recall that the best performance has value 1, i.e., a short time has a high score).

In this example, we have three lifesavers $\mathcal{A} = \{A_1, A_2, A_3\}$ at the beach. The task consists of performing one action – to rescue the distressed person. The values for the *IR* of A_1, A_2 and A_3 for this action are $IR_{A_1}(\textit{rescue}) = 0.5$ (STDV=0.4), $IR_{A_2}(\textit{rescue}) = 0.8$ (STDV=0.3), and $IR_{A_3}(\textit{rescue}) = 0.3$ (STDV=0.2). That is, agent A_1 has a medium performance and is unstable, agent A_2 has a high performance and is a bit more stable, and agent A_3 has a low performance and high stability.

For clarity of exposition, we assume that only agents A_1 and A_2 can select agents for a rescue. These two agents (which are both observers and lifesavers) maintain models of lifesaver agents A_1, A_2 and A_3 ($M_{A_1}(A_1), M_{A_1}(A_2)$ and $M_{A_1}(A_3)$, and $M_{A_2}(A_1), M_{A_2}(A_2)$ and $M_{A_2}(A_3)$), and generate proposals involving the lifesaver agents. The models are initialized randomly (i.e., each agent has an *a priori*, random opinion of the other agents). Both A_1 and A_2 store the last three observations made of the performance of the lifesavers (*OC*=3), and apply the majority policy for selecting a lifesaver for a rescue. This policy chooses the lifesaver that most agents voted for (in the event of a tie, the top agent in an ordered list of agents is selected).

Table 1 illustrates the assignment of agents to a sequence of rescues under the majority selection policy (the values obtained after each rescue are boldfaced). The first column shows the time of the rescue; the second column lists the observer agents; the third and fourth columns show the agent proposed by each observer agent and the agent selected by the majority selection policy, respectively. Columns 5-7 contain the observed performance of the lifesaver agents; and columns 8-10 contain the models resulting from these observations (we have listed only the mean of the observed performance).

The first two rows in Table 1 (for time T_0) contain the initial conditions of the collaboration. Columns 8-10 contain the initial values of the models maintained by A_1 and A_2 for the Internal Resources (rescue performance) of A_1, A_2 and A_3. These initial values, which are *not* consistent with the real performance of the agents in question, are also recorded as the first "observed" performance of A_1, A_2 and A_3. This is done to model a behaviour whereby an agent's initial "opinion" of the members of its team can be influenced, but not immediately replaced, by observations of their performance.

According to the models maintained by A_1 and A_2, A_3 has the best performance. Hence, A_3 is selected by both A_1 and A_2 when a rescue is announced at time T_1. However, as expected from the *IR* of A_3, the agent's actual performance (0.4 at time T_1, Column 7) is poorer than that anticipated by the observer agents. Both agents observe this performance, and update their models accordingly (Column 10).

Now, when a new rescue must be performed (at time T_2), agent A_1 proposes A_3, as it is still the best according to its models, but agent A_2 proposes A_1. As indicated above, according to our tie-breaking rule, the first agent in the ordered list of agents is chosen. This is A_1, as it appears in the list before A_3. However, A_1 does not perform

Time	**Observer agent**	**Proposed agent**	**Selected agent**	**Observed performance of**			**Models**		
				A_1	A_2	A_3	$M(A_1)$	$M(A_2)$	$M(A_3)$
T_0	A_1			0.3	0.4	0.5	0.3	0.4	0.5
	A_2			0.6	0.5	0.7	0.6	0.5	0.7
T_1	A_1	A_3	A_3	0.3	0.4	0.5 **0.4**	0.3	0.4	**0.45**
	A_2	A_3		0.6	0.5	0.7 **0.4**	0.6	0.5	**0.55**
T_2	A_1	A_3	A_1	0.3 **0.3**	0.4	0.5 0.4	**0.3**	0.4	0.45
	A_2	A_1		0.6 **0.3**	0.5	0.7 0.4	**0.45**	0.5	0.55
T_3	A_1	A_3	A_3	0.3 0.3	0.4	0.5 0.4 **0.2**	0.3	0.4	**0.37**
	A_2	A_3		0.6 0.3	0.5	0.7 0.4 **0.2**	0.45	0.5	**0.43**
T_4	A_1	A_2	A_2	0.3 0.3	0.4 **0.8**	0.5 0.4 0.2	0.3	**0.6**	0.37
	A_2	A_2		0.6 0.3	0.5 **0.8**	0.7 0.4 0.2	0.45	**0.65**	0.43
T_5	A_1	A_2	A_2	0.3 0.3	0.4 0.8 **0.7**	0.5 0.4 0.2	0.3	**0.63**	0.37
	A_2	A_2		0.6 0.3	0.5 0.8 **0.7**	0.7 0.4 0.2	0.45	**0.67**	0.43

Table 1. Sample agent assignment to a sequence of rescues

well in the rescue (0.3 at time T_2, Column 5), which significantly lowers $M_{A_2}(A_1)$ to 0.45 (Column 8). As a result, A_3 is once more the top choice of both observer agents for the next rescue (at time T_3). But A_3 performs quite badly (0.2 at time T_3, Column 7), thereby further lowering its expected performance according to the models maintained by the observers (Column 10).

At this stage, the bad performance of both A_1 and A_3 has yielded models with low mean values for these agents. Hence, for the next rescue, A_2 is chosen by both observer agents (at time T_4). This is a high-performing agent that has been under-estimated by both observers. Its good performance (0.8 at time T_4, Column 6) raises the expected value in the models maintained by both observers (Column 9). As a result, A_2, who is now clearly preferred by both observers, is chosen for the rescue at time T_5, rendering once more a good performance (0.7 at time T_5, Column 6).

At this point, the models maintained by the observer agents are closer to the *IR* of the lifesavers than the initial (random) models. Since both observer agents have an observation capacity of three observations, the next time a rescue is performed, the initial value will be dropped, which will further increase the accuracy of the models.

3 Simulation-Based Evaluation

We evaluated our extensions of the ETAPP framework by means of simulation experiments which assess the impact of the following parameters on task performance: (1) Internal Resources, (2) Observation Capacity, and (3) Agent-Selection Policy. The same model-updating procedure was used in all our experiments (when *OC*=1, this procedure reverts to that used in our original framework). Our simulation is based on the Surf Rescue (SR) scenario introduced in Section 2.2, where the task is to rescue a person in distress. However, in our simulation the team of lifesavers is composed of five agents.

3.1 Simulation parameters

The parameters corresponding to our extensions were varied as follows.

- **Internal Resources –** We defined teams of agents with different degrees of stability: *Invariant, Stable, Medium, Unstable* and *Mixed*. The agents in Invariant teams exhibit the same performance in all the rescues. Agents in Stable teams exhibit low performance variability – the standard deviation of their performance distribution ranges between 0 and 0.2. The standard deviation for the performance of agents in Medium teams ranges between 0.2 and 0.8, and for agents in Unstable teams between 0.8 and 1. The Mixed team includes a mixture of stable, medium and unstable agents. The mean of the performance distribution is randomly initialized for the agents in all types of teams. In the future, we propose to conduct experiments with high-performing, medium-performing and low-performing teams.

- **Observation capacity –** We varied the *OC* of the agents between 1 and 8. When *OC*=i, agents retain the last i observations made, and when *OC*=1, their observation capacity is as for the original ETAPP framework.
- **Agent selection policy –** We experimented with the two policies mentioned in Section 2.1: *optimistic* and *majority*.

In addition, we constructed two benchmark collaboration settings: RAND and OMNI.

- The RAND (or random) setting defines a lower bound, where a rescue is conducted by an agent that has been chosen randomly from the team. In this setting, agents do not maintain or update models of their collaborators' resources, and do not communicate proposals.
- The OMNI (or omniscient) setting defines an upper bound, where the best-performing agent in the team is always assigned to a rescue. This setting is consistent with the traditional assumption of multi-agent systems whereby agents have accurate knowledge about the performance of team members prior to the collaboration (i.e., $M_{A_i}(A_j) = IR_{A_j}$ for $i, j = 1, \ldots, m$). In this setting, all agents have the same accurate models, and hence do not update their models or communicate proposals.

3.2 Methodology

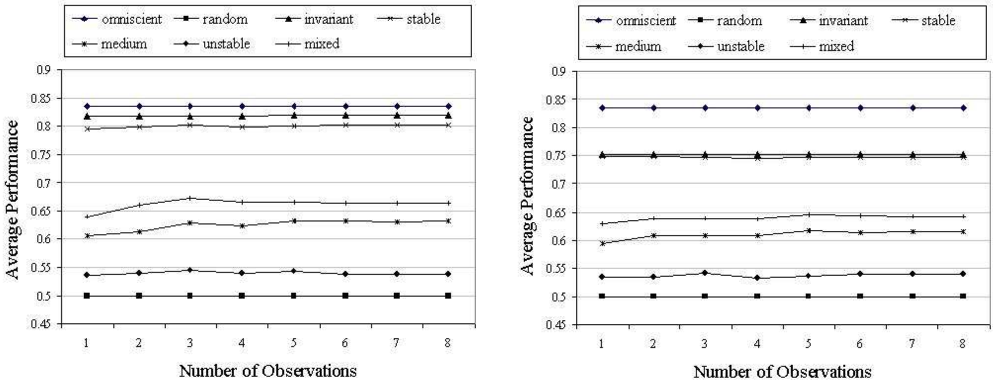
We ran one simulation for each combination of the simulation parameters ($IR \times OC \times \mathcal{P}_A = 5 \times 8 \times 2 = 80$), plus one simulation for each of the benchmark settings, RAND and OMNI. Each simulation consists of ten trials, each divided into 1000 runs (we selected this number of trials and runs because it yields stable and continuous behaviour patterns). Each run consists of a rescue task that is repeated until convergence is reached.

The *IR* and *M* for each agent are initialized at the beginning of each run. *IR* are initialized as specified by the type of the team (e.g., Stable or Unstable), and *M* are initialized with random values.[1] The *IR* of each agent remain constant throughout a run (the agent's performance is drawn from the distribution specified in the *IR*), while *M* are updated from the observations made for each rescue in the run.

The process for reaching convergence works as follows. At the beginning of a run, different lifesavers may be proposed for a rescue task due to the discrepancy between

---

[1] We also conducted experiments where all the models are initialized with a value of 0.5 (medium expected performance), and with a value of 1.0 (high expected performance). The overall results are similar to those obtained with the randomly initialized models, except for the Invariant and Stable group of agents and the 0.5 initialization, which yield a worse average performance. This is because a run terminates when the chosen agent's performance is repeatedly better than 0.5, and so other agents who may be better are not given a chance, thereby converging to a local maximum (Section 3.3).



(a) Optimistic policy

(b) Majority policy

Fig. 1. Average task performance obtained with the optimistic and the majority agent-selection policy plotted against observation capacity for several types of teams

the models maintained by the different agents. After each rescue, the agents update their models based on the performance of the chosen agent. Hence, when a rescue task is announced in the next turn, more agents are likely to propose the same lifesaver (but not necessarily the lifesaver chosen for the previous task). A run is terminated when the same lifesaver is chosen in N consecutive turns (we have experimented with $N=2$ and $N=5$; the results presented in Section 3.3 are for $N=5$).

3.3 Results

The results of our experiments are shown in Figure 1, which depicts the average task performance obtained with our two selection policies as a function of *OC* for our seven types of teams – RAND, OMNI, Invariant, Stable, Medium, Unstable and Mixed. Figure 1(a) shows the results obtained with the optimistic policy, and Figure 1(b) shows the results for the majority policy.

Our measure of task performance for a run is the mean of the *IR* distribution for the agent on which the observers eventually converged. For instance, in the example in Table 1, this agent is A_2, whose $IR_{A_2}(\textit{rescue})$ has mean 0.8 (STDV=0.3). This measure reflects the final outcome of the combination of the parameters of the simulation for the run in question.

As expected, the results for the RAND and OMNI settings correspond to the worst and best performance respectively, and are used as a benchmark for comparison with the other settings. The performance for the Invariant team is slightly lower than that for the OMNI setting. This is due to the fact that the Invariant team sometimes converges to a local maximum, which is reached when the agents in the team repeatedly select an agent that is not the best. This happens when the agents under-estimate the performance of the best agent to the extent that it will never be proposed by any agent in the group, and hence will never perform the task. These results are consistent with the results obtained for the RAND, OMNI and default scenarios in our previous work [4].

As seen in Figure 1, the average performance obtained for the other types of teams is generally worse than that obtained for the Invariant team. This is due to the higher variability in agent performance. In fact, the more unstable the agents in the team are, the worse the performance becomes. We posit that the main reason for this outcome is that the observing agents are unable to build reliable models when team members exhibit unstable performance.

The optimistic policy yields a substantially better performance for the Invariant and Stable teams than the majority policy, and it yields a slightly better performance for the Medium and Mixed teams than the majority policy (these results are significant with p=0.01). This is because if we assume that agents are honest and helpful (i.e., they always make the best proposal according to their models, and select the best of the proposals communicated by team members), the optimistic policy is similar to a global optimization, where the agent that appears to be the best overall is selected. In contrast, the majority policy yields the most popular agent, which may not be the best overall.

Task performance improves for Medium and Mixed teams when agents are able to remember observations of the performance of team members. This improvement is larger for the optimistic selection policy than for the majority policy (these results are significant with p=0.01). Further, this improvement is achieved with only 3 observations for the optimistic policy, and with 5 observations for the majority policy. This discrepancy may be caused by the need for additional "evidence" in order to get several agents to prefer the same agent, as required by the majority policy.

Finally, the performance of Unstable teams is not affected by the agent-selection policy or the agents' observation capacity, as the agents in these teams exhibit too much performance variation for the observer agents to reach reliable conclusions.

4 Related Research

Several research projects have demonstrated that maintaining models of features of collaborators can benefit different aspects of task performance [6, 3, 7].[2]

Suryadi and Gmytrasiewicz [6] and Gmytrasiewicz and Durfee [3] investigated agents that apply a decision-theoretic procedure to make decisions that maximize their own individual payoffs. This procedure takes into account the "payoff matrix" of collaborators, which in turn is learned from observations of their behaviour. Our system also learns the behaviour of other agents from observations (although we learn only the mean and standard deviation of their performance). However, whereas Suryadi and Gmytrasiewicz's agents make individual decisions and do not communicate with each other, our agents communicate proposals in order to make a joint decision. Vassileva *et al.* developed I-Help [7], which is a large scale multi-agent system that provides students with distributed help resources. Personal agents represent students' personal preferences. Matchmaker agents collect this information from personal agents, and match students that require help in a certain topic with students that are able to provide help. The incorporation of our model-update mechanism into the models maintained by the matchmaker agents would increase the accuracy of these models, and hence improve their usefulness for help-seeking students.

Our *OC* parameter is similar to the attentional limitations considered by Walker [8], and is related to the memory boundedness investigated by Rubinstein [5]. However,

---

[2] Garrido *et al.* [2] and Suryadi and Gmytrasiewicz [6] provide an overview of research on modeling other agents.

both Walker and Rubinstein also considered inferential limitations, while we consider agent-modeling limitations.

Finally, our agents' ability to build models of agents from observations resembles the work of Davison and Hirsh [1]. Their model gave greater weight to more recent events than to earlier events, while we achieve a similar behaviour through our *OC* parameter, which specifies that only the last K observations should be considered.

5 Conclusion

We have extended our ETAPP collaboration framework to model team members that exhibit variable performance. This requires a probabilistic representation of agent performance, the specification of the number of observations retained by observer agents, and a procedure for building agent models from these observations. In addition, we have offered the majority policy for assigning agents to tasks, and compared its impact on task performance with the impact of the optimistic policy.

We evaluated our extensions by means of a simulated rescue scenario, where we varied the performance stability of teams of agents, the number of observations retained by observer agents, and the policy used to allocate agents to tasks. Our results show that performance variability has a large impact on task performance, that a small number of observations of agent behaviour is sufficient to improve task performance, and that the task performance obtained by applying the optimistic selection policy is at least as good as that obtained with the majority policy.

References

1. Brian Davison and Haym Hirsh. Predicting sequences of user actions. In *Notes of the AAAI/ICML 1998 Workshop on Predicting the Future: AI Approaches to Time-Series Analysis*, Madison, Wisconsin, 1998.
2. Leonardo Garrido, Katia Sycara, and R. Brena. Quantifying the utility of building agents models: An experimental study. In *Proceedings of the Agents-00/ECML-00 Workshop on Learning Agents*, Barcelona, Spain, 2000.
3. Piotr J. Gmytrasiewicz and Edmund H. Durfee. Rational communication in multi-agent environments. *Autonomous Agents and Multi-Agent Systems*, 4(3):233–272, 2001.
4. Christian Guttmann and Ingrid Zukerman. Towards models of incomplete and uncertain knowledge of collaborators' internal resources. In Jörg Denzinger, Gabriela Lindemann, Ingo J. Timm, and Rainer Unland, editors, *Second German Conference on MultiAgent system TEchnologieS (MATES) 2004*, LNAI 3187, 58–72, Erfurt, Germany, 2004. Springer.
5. Ariel Rubinstein. *Modeling Bounded Rationality*. Zeuthen lecture book series. MIT Press, Cambridge, Massachusetts, 1998.
6. Dicky Suryadi and Piotr J. Gmytrasiewicz. Learning models of other agents using influence diagrams. In *Proceedings of the Seventh International Conference on User Modeling*, 223–232, Banff, Canada, 1999.
7. Julita Vassileva, Gordon McCalla, and Jim Greer. Multi-agent multi-user modeling in I-Help. *User Modeling and User-Adapted Interaction*, 13(1-2):179–210, 2003.
8. Marilyn A. Walker. The effect of resource limits and task complexity on collaborative planning in dialogue. *Artificial Intelligence*, 1-2(85):181–243, 1996.