Why bother about bother: Is it worth it to ask the user?

Robin Cohen
School of Computer Science
University of Waterloo
rcohen@ai.uwaterloo.ca

Michael Y.K. Cheng
School of Computer Science
University of Waterloo
mycheng@cs.uwaterloo.ca

Michael W. Fleming
Faculty of Computer Science
University of New Brunswick
mwf@unb.ca

Abstract

In this paper, we discuss the importance of modeling the potential bother to the user, when reasoning about interaction in a mixed-initiative setting. We summarize our previous work on modeling bother as one of the costs of interaction, clarifying how to incorporate this estimated cost when reasoning about whether to initiate interaction. We then present a new method for modeling bother that appeals to classifications of users according to their willingness and attentional state and attaches a level of bother to the kind of task being asked of the user, when interacting. We demonstrate the usefulness of this model in the context of a multiagent system, where each agent faces the challenge of estimating the bother incurred by a user as a result of interactions from other agents. This is accomplished by the introduction of proxy agents for each user, to track the requests for interaction from multiple parties. We discuss the tradeoff between more accurate bother cost information and higher communication overhead, when reasoning about whether to initiate interactions with users, as part of the collaborative problem solving. Our conclusion is that bother is indeed important to model and that it is in fact possible to effectively integrate consideration of bother cost into methods for reasoning about interaction with users.

Introduction

In mixed-initiative settings, the system needs to reason about whether to interact with the user, in order to gather more information, when there is sufficient uncertainty about the path it is currently following and when the system feels the user will indeed be able to assist. In our previous work (Fleming & Cohen 2004)[1], we presented a domain-independent decision-theoretic model for reasoning about interaction that provides a principled method for a system to determine whether to interact, weighing the potential benefits against the costs of interaction. In that research, we outlined various factors about the user, the task and the dialogue that need to be modeled, as part of the overall reasoning about interaction. We also provided some guidance for system designers to select the appropriate reasoning model to use based on the nature of their application domain. One of the costs that we discussed as important is that of bother: essentially determining the degree to which the user will be interrupted or annoyed when asked by the system to provide additional information towards improving the problem solving. In this paper, we first summarize our proposed model for bother and discuss its flexibility in properly modeling the user's willingness to participate in interaction, over time. We then outline an alternative model for bother, allowing users to be labeled in certain classes and integrating other aspects of the user. Next, we move on to elaborate on how such a model of bother can be employed in a multiagent, multi-user context, when agents must determine just how open a user is to be bothered, based not only on its own experiences with that user in the past, but also with that user's interactions with other agents in the system. As such, our research explores the importance of modeling bother, mechanisms for incorporating a model of bother into a process for reasoning about interaction with users, and insights into how best to design not only mixed-initiative systems but also multiagent adjustable autonomy ones.

Fleming's Bother Cost Model

There are two main principles to Fleming's (Fleming 2003; Fleming & Cohen 2004) bother cost model. First is the idea that "recent interruptions and difficult questions should carry more weight than interruptions in the distant past and very straightforward questions." Second is the notion that a user's willingness to interact with the system is a critical factor in bother cost modeling. Some users are very willing and would prefer to actively help the system achieve a better result, while other users are not willing, and would prefer not to be bothered much. Fleming's model is as follows (Fleming 2003):

- First estimate how bothersome the dialogue has been so far. This *bother so far (BSF)* is given by $BSF = \sum_I c(I) \times \beta^{t(I)}$, where the system computes the sum over all the past interactions with the user. $c(I)$ is how bothersome the interaction was (e.g., cognitive effort required by the user to answer the question), $t(I)$ is the amount of time that has passed since that interaction, and β is a discount factor that diminishes the effect of past interactions

[1] An earlier version of this model appeared in the IJCAI'03 Workshop on Mixed-Initiative Intelligent Systems

as time passes.

- Let w represent the user willingness, with a range of 0 to 10, with higher w meaning more willingness.
- Let $\alpha = 1.26 - 0.05w$ and $Init = 10 - w$.
- Then, $BotherCost = Init + \frac{1-\alpha^{BSF}}{1-\alpha}$. From this formulation, a lower willingness w results in a higher $Init$ cost, and also a higher α value (which amplifies the effect of the bother so far BSF). As BSF increases, so too does $BotherCost$, but at different rates, depending on the α value. As shown in (Fleming 2003), for low w values, α will be greater than 1, and we will see an exponential-like increase due to BSF, while for high w values, α will be less than 1, and we see a log-like increase.

This model of bother cost is incorporated into a formula for reasoning about interaction in mixed-initiative systems that weighs the benefits of interacting (the expected increase in the utility of the problem solving gained by asking a question) against the costs of interacting. These costs include a time cost (the cost associated with the additional time required for the interaction) and bother cost, and when the total of these costs exceeds the anticipated benefit, the system will decide not to initiate interaction. See (Fleming & Cohen 2004) for details.

Proposed Bother Cost Model

Below we present a richer model of bother cost, of use in settings where agents may be interacting with users either to query for information or to simply transfer decision making control. These are both referred to as transfer-of-control (TOC) actions[2]. It is important to note that we are not aiming to develop the definitive bother cost model. Instead, we are proposing a working model that we can use in this paper that incorporates the current bother cost research in the field. From previous works, we have extracted the following factors which are believed to influence bother cost:

- The difficulty of the interruption query, $TOC\_Base\_Bother\_Cost$. For example, usually, asking a user his/her preference is easier (i.e., cognitively less intense) than asking a user to decide on a plan of action.
- The attention state of the user, $Attention\_State\_Factor$[3]. For instance, a user is more interruptible when resting than when he/she is busy with important work.
- How willing a user is to interact with the system. $User\_Unwillingness\_Factor$[4] is a measure of how receptive the user is towards being TOC'ed, and how well they handle interruptions.
- The timings of the interruptions, $t(TOC)$, and the discount factor, β ($0 < \beta < 1$), which reduces the bother impact of past TOCs as time passes[5].

By logically adapting Fleming's (Fleming 2003) bother cost model to incorporate the findings of other researchers, we propose the following enhanced bother cost model:

- $Init = User\_Unwillingness\_Factor \times Attention\_State\_Factor \times$ *TOC_Base_Bother_Cost*
- BSF *(Bother So Far)* $= \sum_{toc \in PastTOC} TOC\_Base\_Bother\_Cost(toc) \times \beta^{t(toc)}$, where $PastTOC$ is the set of all the past TOCs experienced by the user, $TOC\_Base\_Bother\_Cost(toc)$ is just the $TOC\_Base\_Bother\_Cost$ of toc, and $t(toc)$ is the time point at which toc occurred.
- To determine the increase to the bother cost due to BSF, we have a function, $BC\_Inc\_Fn(BSF, User\_Unwillingness)$, that maps a BSF value to a bother cost increase, based on the user's unwillingness level.
- $BotherCost$ *(BC)* $= Init + BC\_Inc\_Fn(BSF, User\_Unwillingness)$.

Here are some suggestions for possible bother cost factor values[6]:

- [$TOC\_Base\_Bother\_Cost$] Easy=5, Medium=10, Hard=20
- [$Attention\_State\_Factor$] Relaxed=0.75, Neutral=1, Busy=1.25
- [$User\_Unwillingness\_Factor$] Willing=0.5, Neutral=1, Unwilling=2
- [β] 0.90
- [BC_Inc_Fn] For Willing, $BC\_Inc\_Fn(x) = x^{0.75}$, for Neutral, $BC\_Inc\_Fn(x) = x^1$, for Unwilling, $BC\_Inc\_Fn(x) = x^{1.25}$. This gives us roughly the same bother cost shape as used in (Fleming 2003)[7].

[2]Note that this extends the concept of a TOC strategy used in the Electric Elves (E-Elves) project (Scerri, Pynadath, & Tambe 2002), which is restricted to just transfers of decision making control

[3]This concept is drawn from the work of (Horvitz & Apacible 2003)

[4]In developing the proposed bother cost model, we found it more intuitive to think of unwillingness being a high value (compared to willingness being a high value)

[5]Note: The value of β depends on the size of the time step. If a time step is 'huge', then β should be low (to reflect that one time step means a lot of time has elapsed, and so we should discount more), while inversely, if the time step is 'small', then β should be high. Also, it is conceivable that the value of β will depend on the particular person. For this paper, we assume (for simplicity's sake) that it is the same value for all users.

[6]Note: These are only example suggestions. In the real world, the system designer would want to tailor the values to the domain. For instance, the domain might require finer granularity in terms of the number of attention states, or, perhaps the differences between willing and unwilling users are greater, necessitating greater differences in $BC\_Inc\_Fn(BSF, User\_Unwillingness)$.

[7]Note: For BSF less than 1, we might want to use different functions, else we get the somewhat odd result that there is more bother increase for Willing users than Unwilling users. A work-around is to do something like $BC\_Inc\_Fn(x) = (x + 1)^{willingness\_rate} - 1$. However, with BSF less than 1, the increase in bother cost is negligible and so this slight difference is virtually irrelevant.

Reasoning about Bother in Multiagent Settings

Interesting challenges arise when an agent is trying to model bother cost in a cooperative multiagent system. While an agent has up-to-date records regarding the bother cost of various users in the single agent case, this is no longer true in the multiagent systems case. From an individual agent's perspective, the bother so far (BSF) of users may change for reasons other than as a result of its own actions. In particular, BSF of users may change due to the actions of other agents in the system. Unless mechanisms are in place to address this, agents will likely have 'stale' local data about users' bother so far. The problem with stale bother data is that it is possible for an agent to think that it is optimal to interact with a user, when in actuality, it is not, since that user has already been bothered too much by other agents in the system. As such, we would like the agents to propagate bother cost information amongst themselves in order to keep each other updated. However, at the same time, we do not want to overburden the communication channels. So, there is a trade-off between keeping agents updated (which affects the quality of agent operation), and minimizing the amount of agent communication required. We now outline three different designs of agents with respect to bother cost information sharing, namely TypeI, TypeII, and TypeIV agents[8]. We first make the following assumptions/simplifications in this paper:

- We focus on sharing users' bother cost information amongst the agents, not other types of information such as domain specific information.
- Each user ($User_i$) has his/her own proxy agent (*Proxy*$_i$). This is a reasonable assumption and is similar in fact to what occurs in the E-Elves project (Scerri, Pynadath, & Tambe 2002) where each user has a personal assistant agent[9]. In order for an agent to interact with a user, it has to go through that user's proxy agent first. To differentiate between the proxy agent, and the agent which initiates the interaction, we shall denote the latter as 'requesting agent' or sometimes simply as 'agent', while the former will always be noted as 'proxy agent'.
- Representing bother cost is employed in conjunction with a model for reasoning about interaction that allows agents to select the optimal TOC strategy, determining who to transfer control to (either for information gathering or decision making) and how long to wait for a response before attempting a different transfer. The particular model that we have developed is not explained in detail here (see (Cheng & Cohen 2005) for further details). It is important to note that selecting a TOC strategy requires modeling the expected quality of decision obtained as a result of the transfer (EQ, as in (Scerri, Pynadath, & Tambe 2002)) and the probability of response, balanced against the costs of waiting for a response and bothering the user. As a simplification, we will assume that users who are asked are able to provide responses and that the length of TOC strategies is at most 1. As such, the key term to consider for the expected utility (EU) of a strategy is $EQ - BC$.

[8]In our research, we have also developed another type of information sharing agent named TypeIII. TypeIII agents are in fact subsumed by TypeIV agents, and as such, we only describe TypeIV agents in this paper.

[9]Note that in E-Elves, the personal assistant agent does not perform the coordination function that we envision our proxy agents doing.

Agent Type I (Broadcast)

This type of agent emphasizes up-to-date information, with the benefit of high utility ($EQ-BC$) achieved by the agents, at the cost of high communication overhead. There are two ways to do this: *[Push]* Whenever a user is bothered (i.e., TOC'ed), his/her proxy agent will broadcast this news to all agents in the society, and *[Pull]* Whenever an agent is about to enter its optimal TOC strategy reasoning process, it will first broadcast poll all the proxy agents for the current bother cost data of their users. For this research, we go with the push approach, since there is no time lag involved (i.e., agent can start planning right away, instead of waiting to receive all the poll information). When a TypeI agent needs a decision made, the process is as follows:

1. Using its up-to-date bother cost information, the agent determines an optimal TOC strategy, which specifies transferring control to a particular user, $User_i$.
2. The agent sends a TOC request (which includes the TOC question to ask) to proxy agent *Proxy*$_i$ who will in turn, relay the TOC question to $User_i$.
3. *Proxy*$_i$ broadcasts an update/notification message to all agents in the system, to alert them of the TOC event.
4. When an agent receives a notification message, it will update the bother so far (BSF) value for $User_i$, so that future TOC strategy planning will be accurate.

Agent Type II (No Information Sharing)

This type of agent stresses low communication overhead (actually no overhead) by not communicating bother cost information at all. Agents only look at their own past actions (TOCs) when determining the BC values used in the optimal TOC strategy reasoning. In a way, they can be viewed as extreme optimists, hoping that no other agents have done any TOCs recently to the users that they want to transfer control to. The consequence of this is that the BC value used to find the optimal TOC strategy will always be less than or equal to the actual BC value. This is easy to see from the bother cost model, because TOC events can only increase BSF, which increases BC. When an agent is not aware of TOC events, its estimated BC value will be lower than the actual BC value. Subsequently, the perceived EU of a strategy may be higher than the actual EU of the strategy, leading the agent to possibly select a TOC strategy that is less than optimal (in the sense that there is another strategy that has a lower perceived EU, but a higher actual EU value). When a TypeII agent needs a decision made, the process is as follows:

1. Using its possibly stale bother cost information, the agent determines an optimal TOC strategy, which specifies transferring control to a particular user, $User_i$.

2. The agent sends a TOC request (which includes the TOC question to ask) to proxy agent *Proxy*$_i$ who will in turn, relay the TOC question to $User_i$.
3. Only the requesting agent and *Proxy*$_i$ are aware of the TOC event, and so only they update their bother so far (BSF) value for $User_i$.

Agent Type IV (Verify Plan within Threshold)

This type of agent aims to achieve a compromise between the two extreme types of agents described above. It will plan its TOC strategy based on possibly stale information, but will verify that its plan is accurate enough (in terms of BC) with the proxy agent, before executing the strategy. If the verification succeeds (i.e., anticipated BC value is reasonably correct), then the plan execution proceeds. Otherwise, the proxy agent will notify the requesting agent of the actual BC value. After receiving the update, the agent will recalculate its optimal TOC strategy, and execute it. This may involve the agent not TOC'ing and making the decision itself, or retrying a TOC (possibly to the same user, or another user). Note that a proxy agent will always have up-to-date bother cost information about its own user, since all TOC requests go through it.

While this cuts down on the communication cost (as compared to Type I agents), and it is sensitive to bother cost by other agents (which is not the case with Type II agents), there is however, a new type of cost introduced in terms of *retry*, where a Type IV agent may have its plan rejected, and have to recompute a new optimal strategy and assuming the new strategy involves a TOC, verify the estimated bother cost again. This introduces a slight time lag between when an agent first determines a TOC strategy, and when it actually executes the strategy, not to mention two extra messages being sent per retry (one message for request rejection, which contains the updated BSF value, and one extra message for TOC request/verify).

It is for this reason that we allow for some possible lack of optimality in EU in order to reduce the number of retries encountered. Basically, due to the nature of the bother cost model, the impact of a TOC event diminishes with the passage of time. So, an unknown TOC event that occurred in the distant past would cause very little increase in BC (due to the discounting factor). In these cases, it makes sense for the agent's plan to go ahead, and not get rejected. For instance, if an agent's estimate of BC is off by 0.1, then we should just go ahead with the strategy, instead of doing a retry. So, with the TypeIV approach, upon receiving the estimated BC value from the requesting agent, the proxy agent will see by how much the estimated BC differs from the actual BC. If the difference (error) is below some threshold, $Threshold_{acceptable\_error}$, then the proxy agent will accept. Due to this, a TypeIV agent is no longer guaranteed to have the EU of TypeI agents, but it can still be reasonably assured of a high EU because its selected strategy EU can never be off by more than $Threshold_{acceptable\_error}$ per proxy agent involved. By being more 'lenient', we can reduce the number of retries. When a TypeIV agent needs a decision made, the process is as follows:

1. Using its possibly stale bother information, the agent determines an optimal TOC strategy, which specifies transferring control to a particular user, $User_i$
2. The agent sends a TOC request (which includes the TOC question to ask *and* the estimated BC of $User_i$) to proxy agent *Proxy*$_i$.
   (a) [If $(ActualBC - EstimatedBC \leq Threshold_{acceptable\_error})$] Then *Proxy*$_i$ relays the TOC question to its user $User_i$, and sends a reply to the requesting agent that the plan was accepted. Both the requesting agent and *Proxy*$_i$ update their information about $User_i$'s bother so far value.
   (b) [Otherwise $(ActualBC$ - $EstimatedBC > Threshold_{acceptable\_error})$] Then *Proxy*$_i$ replies to the requesting agent with a rejection message, which contains information about the actual BSF value. The requesting agent updates its information, and goes back to step 1.

Experiments

To see how the different agent types would fare in different situations, we ran simulations where we varied the model parameters. Each simulation mimics a real world situation, where there are a set of agents and a set of users, and as time progresses, an agent may find that it needs a decision made (we refer to this as an 'event'). When an agent encounters an event, it will proceed to obtain a decision in a manner specified by its agent type. In this section, we first describe the experiment set-up, and then the results of the experiments.

Experiment Set-up

A simulation trial involves three main components: (i) a set of agents, (ii) a set of users, and (iii) a set of 'need decision' events. Here are the *default* settings of each trial:

- There are 50 users and 50 agents in the system.
- There are 5 decision classes, and for each decision class, a uniformly randomly generated number from the range [50,100] is assigned to each user, to serve as that user's EQ value for that decision class. This is done to address the fact that different users have different capabilities in handling different types of decisions.
- There are 100 timesteps taken per trial, and for each timestep t_i and agent $agent_j$, there is a 0.05 chance that a 'need decision' event will occur for $agent_j$ at timestep t_i. The particular class of decision needed will be uniformly randomly assigned from the range [1,5].
- For TypeIV agents, $Threshold_{acceptable\_error} = 10$.
- The bother cost model parameter values used in the simulation are the ones suggested in the Bother Cost Model section. In particular, $\beta = 0.90$, $Attention\_State\_Factor = 1$, $TOC\_Base\_Bother\_Cost = 10$, and for each of the users, they are uniformly randomly assigned a user willingness type from the set, {Willing, Neutral, Unwilling}.

A simulation trial starts by generating all the random values it needs (as described above). Then, it proceeds to run the simulation for each agent type, and records the performance measurements (e.g., total utility and communication overhead). An experiment runs 20 simulation trials, and averages the results. This is to offset possible ‘odd’ results from ‘oddly’ generated random numbers. As a side note, this is why in the results section, we see decimals for certain performance measurements that are ordinarily integers (e.g., the number of retries may be a decimal number).

Experiment Results

Results of One Simulation Trial Run The results of doing one simulation trial run, using the default trial parameters are as follows:

	TypeI	TypeII	TypeIV
AverageUtility	82.23	73.33	80.97
STDev (Utility)	4.47	12.01	4.73
# Broadcasted Msgs	11564	N/A	N/A
# Retries	N/A	N/A	223
Average Retry Chain	N/A	N/A	0.94

As we can see, TypeI agents achieved the highest AverageUtility, with AverageUtility being the average utility achieved for a decision event[10,11]. However, TypeI agents sent an enormous number of broadcasted messages (essentially every TOC results in $\#Agents - 1$ messages). TypeII agents fared the worst in terms of AverageUtility, but had absolutely no extra communication messages. TypeIV seems to be a nice compromise, with just slightly lower AverageUtility than TypeI, and a reasonably low number of retries (and so a low average retry chain[12]).

Results of Varying $Threshold_{acceptable\_error}$ The main results of varying $Threshold_{acceptable\_error}$ are shown in Figure 1. As can be seen from both graphs, varying the threshold level is basically making a tradeoff between utility and the number of retries. As the threshold decreases, TypeIV agents exhibit higher average utility (like TypeI agents) and higher number of retries. The opposite is true; as threshold increases, TypeIV agents shifts towards TypeII agents, with lower average utility, but almost no retries. A system designer should set the threshold level that achieves a beneficial medium between the two extremes. For instance, for the default settings in our experiment, a threshold set at 10 may be a good compromise, as it suffers little average utility loss, but manages to reduce the number of retries significantly. By assigning a utility cost to retry, a system designer can examine a plot such as Figure 1 and select the

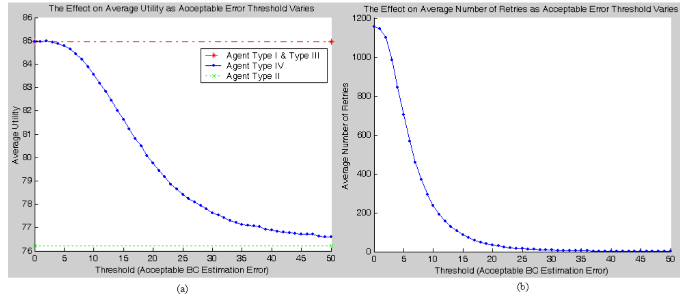

Figure 1: (a) Graph showing how average utility varies as acceptable BC estimate error varies. (b) Graph showing how the number of retries varies as acceptable BC estimate error varies.

optimal threshold level.

Results of Varying the Number of Agents and Users We ran three separate experiments, varying the number of agents and users in the system, as a test of how well each of the agent types scale. The agent/user numbers are 5, 50, 500. The results of the experiments are follows:

	5 Agents and 5 Users		
	TypeI	TypeII	TypeIV
AverageUtility	78.18	77.74	77.98
# Broadcasted Msgs	108.2	N/A	N/A
# Times Retry	N/A	N/A	1.7
	50 Agents and 50 Users		
	TypeI	TypeII	TypeIV
AverageUtility	85.12	77.10	83.67
# Broadcasted Msgs	12218.15	N/A	N/A
# Times Retry	N/A	N/A	218.2
	500 Agents and 500 Users		
	TypeI	TypeII	TypeIV
AverageUtility	85.56	31.88	83.34
# Broadcasted Msgs	1252664.65	N/A	N/A
# Times Retry	N/A	N/A	23260

The main things to note from the results are the following: When the number of agents in the system is low (and assuming ceteris paribus), then there is not much difference between the different agent types, since there is not much chance of ‘overuse’ of the same user. On the other hand, from the 500 agents experiment, we see that the TypeII (no coordination) approach suffers greatly in terms of average utility, as all the agents are trying to access the same users. Not surprisingly, the coordinated approaches (TypeI, and TypeIV) all manage to do fairly well in terms of average utility, but note that they don’t all scale equally well in terms of overhead cost. For instance, TypeI requires a huge number of broadcasted messages as the number of agents in the system increase. While TypeIV also requires more overhead (i.e., retries), it is not as dramatic. As well, even though there was undoubtedly less bother cost being incurred in the experiment with only 5 agents and users, the average utility was lower due to the low number of users. With a low number of users, and randomly generated user EQ values, it is less likely that there will be a user that will

[10]Note: For the utility value, we only count $EQ - BC$. The coordination overhead cost is counted separately.

[11]In the results table, we also show STDev. This is the standard deviation of the utility values. Note that TypeII agents have really high STDev, since sometimes agents TOC unbothered users (so higher utility) and sometimes agents TOC very bothered users (so much lower utility).

[12]A retry chain denotes the number of TOC retries that an agent has to perform before getting a TOC request accepted. This is equivalent to the number of times an agent gets rejected before having a TOC request accepted.

have a very high EQ value for a particular decision class.

Discussion and Conclusion

This research can be seen as contributing to the development of mixed-initiative systems by proposing methods, rooted in user modeling, for managing dialogues between system and user, towards effective overall problem solving. In particular, we believe that a principled method for designing mixed-initiative systems should reason about whether to interact, and that this should be sensitive to an appropriate model of bother to be incurred by the user. We also describe how this method of reasoning can be applied in contexts where there are multiple users and multiple agents, outlining the relative tradeoffs between improving the accuracy of the bother cost model and reducing communication overhead, to direct the problem solving.

Our research discusses the value of proxy agents in facilitating the exchange of information about bother incurred by users, towards the selection of more effective interaction strategies by agents. This provides a more productive mechanism for agents to solicit user feedback as part of their processing. The work of Schreckenghost et al. (Schreckenghost *et al.* 2002) is relevant, as it also advocates the use of proxy agents in the context of multi-user, multiagent systems with adjustable autonomy. That effort is focused, however, on the use of proxies to coordinate the completion of tasks among multiple agents. In addition, we address a problem acknowledged as important by the authors, that of allowing interruptions to users and agents, during the processing.

Even in contexts where there is some preferable global strategy for balancing the level of autonomy between the system and user, for example allowing for strong collaboration between parties, as outlined in (Barber, Gamba, & Martin 2003), it is useful to allow some specific modeling of the users in the community, in order to make decisions about interaction. Our research outlines how bother can be an important factor in making these interaction decisions; for instance, users may be currently occupied with other tasks and even if they are committed to being helpful, it may be preferable not to initiate interaction with them.

For future work, it would be interesting to explore the usefulness of our approach in contexts where the user takes the initiative to direct the problem solving. For instance, Myers and Morley (Myers & Morley 2003) propose a framework for a user to specify when an agent should adjust its autonomy and elicit further input. In environments where there may be multiple agents directed by a single user, it will be critical for those agents to coordinate their efforts, even if governed by rules initially set by the user. This coordination could be achieved if the agents were, at the same time, corresponding with the proxies of the users, to confirm whether it was indeed desirable for them to interact with the user, at this point in time.

Others have explored the concept of bother as a factor in designing mixed-initiative systems, including Horvitz (Horvitz & Apacible 2003) and Bauer (Bauer, Dengler, & Paul 2000). Our research aims to deepen the modeling of bother to incorporate various aspects of the user and of the task, and to appreciate the challenges of properly modeling that bother in environments where there are multiple users and multiple agents. Our conclusion is that it is not sufficient to simply equip each individual agent with a similar process for reasoning about interaction but that there must also be a way of coordinating their modeling. We have also shown some value to modeling the bother to be incurred by users in terms of distinct factors, the values of which are determined on the basis of classifying the user into one of a number of distinct categories.

Acknowledgements

We thank the Natural Sciences and Engineering Research Council of Canada for the financial support.

References

Barber, K.; Gamba, I.; and Martin, C. 2003. *Agent Autonomy (H. Hexmoor, C. Castelfranchi and R. Falcone, eds.)*. chapter 3. Representing and analyzing adapative decision making frameworks.

Bauer, M.; Dengler, D.; and Paul, G. 2000. Instructible agents for web mining. In *Proceedings of International Conference on Intelligent User Interfaces*, 21–28.

Cheng, M., and Cohen, R. 2005. A hybrid transfer of control model for adjustable autonomy multiagent systems. In *Proceedings of AAMAS'05*.

Fleming, M., and Cohen, R. 2004. A decision procedure for autonomous agents to reason about interaction with humans. In *Proceedings of the AAAI 2004 Spring Symposium on Interaction between Humans and Autonomous Systems over Extended Operation*, 81–86.

Fleming, M. 2003. *Reasoning about Interaction in Mixed-Initiative Artificial Intelligence Systems*. Ph.D. Dissertation, University of Waterloo.

Horvitz, E., and Apacible, J. 2003. Learning and reasoning about interruption. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI'03)*, 20–27.

Myers, K., and Morley, D. 2003. *Agent Autonomy (H. Hexmoor, C. Castelfranchi and R. Falcone, eds.)*. chapter 9. Policy-based Agent Directability.

Scerri, P.; Pynadath, D.; and Tambe, M. 2002. Why the elf acted autonomously: Towards a theory of adjustable autonomy. In *Proceedings of AAMAS'02*.

Schreckenghost, D.; Martin, C.; Bonasso, P.; Kortenkamp, D.; Miliam, T.; and Thronesbery, C. 2002. Supporting group interaction among humans and autonomous agents. In *Proceedings of AAAI02 workshop on Autonomy, Delegation and Control: From Inter-Agent to Groups*, 72–77.