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

Robin Cohen
Michael Y.K. Cheng
Michael W. Fleming

AAAI-05 Symposium on Mixed-Initiative Problem-Solving Assistants
November 5, 2005

Overview

- Mixed-initiative settings: system reasoning about interacting with user

- Fleming and Cohen 2004 model: decision-theoretic, benefits vs. costs
  - factors to model about user, task and dialogue
  - costs included: bother to user

- This paper: modeling bother for extended interaction, multi-user, multiagent case
  - bother cost sensitive to bother generated by other agents
  - experimental results about how best to model bother

- Value for designing mixed-initiative and adjustable autonomy systems

Fleming's Bother Cost Model

- User's willingness to interact a critical factor in bother cost modeling
- Recent interruptions and difficult questions carry more weight
- Estimate the bother so far (BSF)
- User indicates willingness (w) in range of 0 to 10
- Formula combines BSF and willingness
- Log-like increases for high values of w; exponential increases for low w

Fleming's Formulae

- $BSF = \sum_I c(I) \times \beta^{t(I)}$
  where the system computes the sum over all the past interactions with the user

- $c(I)$: how bothersome the interaction was (cognitive effort required by the user to answer the question)

- $t(I)$: the amount of time that has passed since that interaction

- β : a discount factor that diminishes the effect of past interactions as time passes

- w represents the user willingness, with a range of 0 to 10, with higher w meaning more willingness

- $\alpha = 1.26 - 0.05w$ 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

- As BSF increases, so too does $BotherCost$, but at different rates according to the value of α

Using Bother Cost in Fleming's Model

- Benefits: expected increase in expected utility by asking user

- Costs of interaction: includes bother cost

- Interact if benefits exceeed costs

- User-specific decisions about interaction

Proposed New Bother Cost Model

- Setting: multiagent, multi-user systems
- Agent reasoning about transfering control to a user (TOC) including when to stop waiting
  - to transfer decision making control
  - to ask a query
- $TOC\_Base\_Bother\_Cost$: difficulty of interruption
- $Attention\_State\_Factor$: higher values if more busy
- $User\_Unwillingness\_Factor$: higher values if more inherently unwilling
- Timing of TOC $t(TOC)$ and discount factor for bother (as in Fleming and Cohen)
- Combine factors into equation reflecting willingness and bother so far

Enhanced Bother Model Formulae

- $Init\ =\ User\_Unwillingness\_Factor \times Attention\_State\_Factor \times$ *TOC_Base_Bother_Cost*

- BSF *(Bother So Far)* $= \sum_{toc \in PastTOC} TOC\_Base\_Bother\_Cost(toc) \times \beta^{t(toc)}$

  - $PastTOC$: the set of all the past TOCs experienced by the user

  - $t(toc)$ is the time since toc occurred

  - β as discount factor to diminish bother over time

- $BC\_Inc\_Fn(BSF, User\_Unwillingness)$ maps a BSF value to a bother cost increase, based on the user's unwillingness level.

- $BotherCost$ *(BC)* $= Init + BC\_Inc\_Fn(BSF, User\_Unwillingness)$.

Possible Bother Cost Factor Values

- [$TOC\_Base\_Bother\_Cost$] 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

- [BC_Inc_Fn] For Willing, $BC\_Inc\_Fn(x) = x^{0.75}$, for Neutral, $BC\_Inc\_Fn(x) = x^1$, for Unwilling, $BC\_Inc\_Fn(x) = x^{1.25}$.

- [β] 0.90

The Challenge of Reasoning about Bother in Multiagent Settings

- Ensuring up-to-date model of bother assumed by user

- BSF may change due to actions of other agents

- Stale bother estimates compromise selection of TOC strategies with highest utility

- Tradeoff: communication burden vs. accuracy of bother information

Approach for Reasoning about Bother

- Each user has own proxy agent

- Focus on sharing information about bother only

- Context: agents selecting optimal TOC strategies
  - key term for EU is EQ-BC
  - TOC: query to gather information or transfering decision making control
  - the latter incurs more bother

Different Possible Models for Reasoning About Bother

- Agent Type I: Broadcast

- Agent Type II: No Information Sharing

- Agent Type III: Verify Plan, Retry TOC

- Agent Type IV: Verify Plan Within Threshold and Retry

Agent Type I (Broadcast)

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 I Overview

- A Push-type approach: whenever user is bothered, his agent will broadcast this news to all agents in the society

- Value of push vs. pull: agent can start planning right away vs. waiting to receive information

Agent Type II (No Information Sharing)

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 II Overview

- Agents look only at their own past actions

- Optimistic: hoping no other TOCs to users they want to transfer control to

- The BC value used to find optimal strategy is less than or equal to the actual BC value

- Agent may select less than optimal TOC strategy (actual EU is lower)

Agent Type IV (Verify Plan Within Threshold)

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, *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.

Agent Type IV Overview

- Agent verifies expected BC with proxy agent before executing strategy

- Proxy provides agent with actual BC value, if difference greater than threshold

- Agent will recalculate optimal TOC strategy and execute it

  - May involve not TOCing and making decision itself

  - May involve retrying a TOC (to same user or another user)

- Reduces communication cost (vs. Type I) but has retry cost

  - Retry delays executing of TOC and incurs two extra messages (request rejection and TOC verify request)

Experiments

- Each trial involves a set of agents, a set of users and a set of need-decision events

- Default settings for each trial established

- One simulation run: results

- Vary number of agents and users: results

- Vary threshold of acceptable error: results

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

- 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

- 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].

Experiment Set-up continued

- For TypeIV agents, $Threshold_{acceptable\_error} = 10$.

- Bother cost model parameters:
  $\beta = 0.90$, $Attention\_State\_Factor = 1$,
  $TOC\_Base\_Bother\_Cost = 10$

- Each of the users is uniformly randomly assigned a user willingness type from the set, {Willing, Neutral, Unwilling}.

- Run 20 simulation trials and average the results

  – offsets possible odd results from oddly generated random numbers

Results from One Simulation Trial Run

	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

- Type I agents: highest Average Utility
- Type II agents: high Standard Deviation (sometimes unbothered, sometimes bothered users)
- Type I agents: large number of broadcasted messages
- Type II agents: worst Average Utility, but no extra communication
- Type IV agents: seems like a nice compromise
- Retry chain: number of TOC retries before TOC request accepted

Varying the Number of Agents and Users

- Three separate experiments: 5, 50 and 500 agents and users

	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

Varying the Number of Agents Continued

- Not much difference between agent types when low numbers of agents
  - not much chance of overuse of same user
- With large numbers of agents, Type II approach suffers greatly
  - all agents trying to access the same high utility users
- Type I requires huge number of broadcasted messages as number of agents increases
- Type IV increases in overhead costs, but not as dramatically
- Low number of users: less bother but lower average utility
  - less likely to get user with high EQ, from randomly generated EQ values

Varying threshold values for Agent IV

- Tradeoff between utility and number of retries

- As threshold decreases, Type IV agents exhibit higher average utility and higher number of retries

- As threshold increases, Type IV agents shift towards Type II agents with lower average utility but almost no retries

- System designer selects compromise threshold: 10 suffers little average utility loss but reduces number of retries significantly

Conclusions

- It is worth it to model bother
  - better decision making for whether to initiate interaction and which user to ask

- One possible representation for bother shown here; user-specific

- Tradeoffs in how to manage sharing of BC information to improve decision-making

- Ultimately designing better mixed-initiative, adjustable autonomy systems
  - systematic foundation
  - in multiagent setting: something that enables coordination of decisions of individual agents is necessary and possible
  - value of user modeling, categories of users for estimating bother costs

Related Work and Future Work

- Proxy agents in Schreckenghost et al.
  - not just for completion of tasks, but to coordinate bother
  - addresses acknowledged need for interruptions
- User initiative, as in Myers and Morley
  - starting point for addressing multiple users directing agents
- Global strategies, as in Barber et al.
  - value of modeling bother, to prefer not bothering certain parties

Related Work and Future Work Continued

- Other models of bother: Horvitz, Bauer, etc.
  - addressing multiple agents and multiple users now

- Current model has coordination strategy (not presented here)
  - algorithms for adjusting TOC strategies after learning of bother costs
  - future work: more with sophisticated multiagent systems