A Hybrid Transfer of Control Model for Adjustable Autonomy Multiagent Systems

Michael Y.K. Cheng
School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
mycheng@cs.uwaterloo.ca

Robin Cohen
School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
rcohen@cs.uwaterloo.ca

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*intelligent agents, multiagent systems*

Keywords

adjustable autonomy, interaction

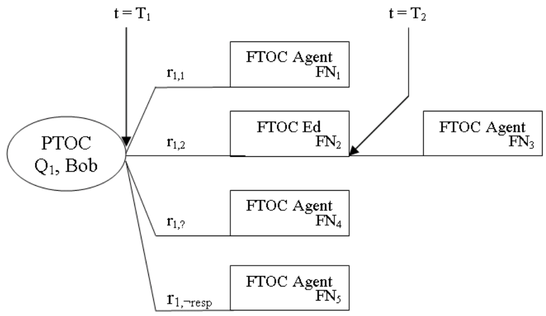
1. INTRODUCTION

Multiagent systems with agent-based adjustable autonomy are ones in which agents are provided with the ability to reason about adjusting their autonomy level depending on the situation [1]. One promising approach for the design of these systems is that of Electric Elves (E-Elves) [2]: a model for agents to reason about whether to retain autonomy or fully transfer decision-making control to another entity (user or agent).

In this paper, we present a model (based on E-Elves [2]) that allows agents to reason about adjusting their autonomy in multiagent systems, integrating both full transfers of decision-making control to other entities and information gathering interaction (referred to as partial transfers of control). By enabling agents to query for more information, agents can better determine the best entity to fully transfer control to, or improve their own decision-making ability.

2. HYBRID TOC MODEL

Our model produces an optimal *hybrid transfer-of-control (HTOC) strategy* that an agent should follow to maximize overall expected utility. A strategy specifies the transfers of control that the agent should perform, who to transfer to, how long to wait for a response, as well as what the agent should do next (which varies depending on the response received). We use the term 'hybrid' to emphasize the fact that our agents can employ strategies containing both full transfers of control (FTOC), that are present in E-Elves [2], and partial transfers of control (PTOC) that we are proposing. Visually, one can picture an HTOC strategy as a tree, with two types of nodes, *FTOC/Sequential nodes* and *PTOC/Branching nodes*.



Q_1 = "When rescheduling the meeting, which factor should be prioritized?"
$r_{1,1}$ = "Prioritize having the meeting earlier."
$r_{1,2}$ = "Prioritize having the meeting time be convenient for the presenter (Ed)."

Figure 1: Example Hybrid TOC Strategy

An FTOC node represents the agent fully transferring control to an entity at time point t_{i-1} and waiting until time point t_i for a response. If the entity does not respond within the allotted time, there is only one next action, i.e., the next node in the transfer-of-control strategy. For simplicity's sake, we will regard the case of the agent deciding autonomously as an FTOC to the agent itself.

A PTOC node represents the agent partially transferring control by asking an entity a query at time point t_{i-1} and waiting until time point t_i for a response. Each possible response to a query will be represented as a branch from the PTOC node to a branch strategy representing what the agent should do when it receives that particular response. In this paper, we use the following terminology. Q_j denotes a particular query, and $r_{j,1}, r_{j,2}, ...r_{j,n}$ denote its possible answer responses. We also include "Don't know" as a valid response, denoted as $r_{j,?}$, and also allow for the 'no response' case, $r_{j,\neg resp}$, which occurs when the entity does not respond within the allotted time.

Figure 1 illustrates an example HTOC strategy where the agent is responsible for rescheduling a presentation meeting time. Since it is uncertain about which factor it should prioritize when selecting a meeting time, it first queries Bob (the group leader) for that information, and then, depending on the response, either decides itself or first lets Ed (the presenter) try to make the decision.

The procedure for an agent to find the optimal HTOC strategy is a basic branch and bound search where the agent generates all possible strategies, of length up to a fixed number K, evaluates the generated strategies, and then selects the one with the highest expected utility value. Strategy generation is simply performed as follows. The agent first generates the simplest strategies (of length one), and then generates more elaborate strategies (of length up to K) by adding an FTOC or PTOC node to previously generated strategies.

2.1 Strategy Evaluation

In order to evaluate the expected utility (EU) of a strategy s, we must first find the optimal time instantiations of its transfers of control[1]. For example, for a simple strategy consisting of an FTOC to Bob, then an FTOC back to the agent, we need to determine the optimal time T that the agent should stop waiting for a response from Bob, and just decide by itself. The overall EU of strategy s is the sum of the EU of all the FTOC nodes in the strategy[2]. That is, $EU(s) = \sum_{fn \in s} EU(fn)$.

The expected utility of an FTOC node is $EU(fn) = P_{trans} \times \int_{t_s}^{t_e} PR^d_{e_i}(t) \times (EQ^d_{e_i}(t) - W(t) - BC)dt$. We retained the fundamental components of the E-Elves [2] EU calculation, namely the concepts of $PR^d_{e_i}(t)$, $EQ^d_{e_i}(t)$, and $W(t)$ and how they are used. $PR^d_{e_i}(t)$ now denotes the probability that entity e_i will respond to the delegation of decision d at time t, *given that the transfer occurred at time* t_s. $EQ^d_{e_i}(t)$ denotes the expected decision quality of entity e_i at time t. $W(t)$ denotes the cost of waiting until time t to make a decision. t_s denotes the time point that the FTOC is issued, and t_e denotes the time point that the agent stops waiting for a response. We introduce the term BC (motivated by [3]) to denote the accumulated bother cost incurred to entities due to interruptions from all the transfers of control that the agent has issued so far. P_{trans} is the probability that the agent will actually reach/execute this particular transfer node and is computed as follows, $P_{trans} = \prod_{fn}(1 - \int_{t_s}^{t_e} PR^d_{e^d_{prev}}(t)dt) \times \prod_{pn} P^{Q_j}_{e^q_{prev}}(resp = r)$, where the first product is iterated over all the previous FTOCs, and represents the probability that the decision was not made in an earlier FTOC node. The second product is iterated over all the previous PTOCs, and represents the probability that for all the previously asked queries, the agent received the responses such that the node currently under consideration will be executed. $P^{Q_j}_{e^q_{prev}}(resp = r)$ denotes the probability that asking entity e^q_{prev} question Q_j will result in a particular response r. The computation of this term will be described in a section further below.

It is important to note that the model parameters used in the above EU calculation depend on the branch path that the FTOC node is on (i.e., the model parameters will be adjusted to reflect the information gathered from earlier PTOC responses, and different responses lead to different parameter values). For instance, after a PTOC node where an agent asks a preference elicitation query, some of the answer branches will have a higher EQ^d_{agent} value than if the agent did not ask the query, since agents can make a better decision with the user's preference. This is one benefit of PTOCs, namely improving the agent's own decision-making ability. Another critical benefit of PTOCs is that the agent can now perform different branch strategies for different responses received (in essence, taking the action that best suits the current situation).

Here we will elaborate on the computation of $P^{Q_j}_{e_i}(resp = r_{j,k})$, the probability of getting a particular response $r_{j,k}$ when asking entity e_i query Q_j. The relevant entity characteristics are the $PEK^{Q_j}_{e_i}$ value, denoting the probability that entity e_i knows the answer to query Q_j, and the $PR^{Q_j}_{e_i}(t)$ value, denoting the probability that e_i responds to Q_j at time t. The idea is that the probability of getting an answer response is contingent on e_i responding, and e_i knowing the answer. The three possible cases for how to compute the value of $P^{Q_j}_{e_i}(resp = r_{j,k})$, are as follows: (No response) $P^{Q_j}_{e_i}(resp = r_{j,\neg resp}) = (1 - \int_{t_s}^{t_e} PR^{Q_j}_{e_i}(t)dt)$, ("I don't know") $P^{Q_j}_{e_i}(resp = r_{j,?}) = \int_{t_s}^{t_e} PR^{Q_j}_{e_i}(t)dt \times (1 - PEK^{Q_j}_{e_i})$, and (Answer response) $P^{Q_j}_{e_i}(resp = r_{j,a}) = \int_{t_s}^{t_e} PR^{Q_j}_{e_i}(t)dt \times PEK^{Q_j}_{e_i} \times PA(r_{j,a})$, where $PA(r_{j,a})$ denotes the probability that the answer to query Q_j is $r_{j,a}$. Note that $\int_{t_s}^{t_e} PR^{Q_j}_{e_i}(t)dt$ gives the probability of e_i responding to Q_j during time $[t_s, t_e]$.

3. CONCLUSIONS AND FUTURE WORK

In this paper, we present a domain-independent decision-theoretic adjustable autonomy model that enables an agent to reason about the trade-offs between three different levels of autonomy: (i) full autonomy, where the agent just decides by itself, (ii) no autonomy, where the agent transfers decision-making control to another entity, and (iii) partial autonomy, where the agent queries another entity for information that determines how the decision should be made. In contrast to earlier works, rather than restricting to only full transfers-of-control (as in [2]) or to interaction without any transfers of decision-making control (as in [3]), we allow agents to initiate interaction in order to determine the best transfers of decision-making control.

For future work, we are looking into heuristics to reduce the number of strategies generated and searched. Also, our model currently computes a conservative estimate of the expected utility of a strategy containing PTOCs, since a user can respond earlier than the time the agent allotted for the interaction. Therefore, a future work would be to derive a more precise calculation of the strategy's expected utility.

4. REFERENCES

[1] R.T., Maheswaran, M. Tambe, P. Varakantham, and K. Myers. Adjustable Autonomy challenges in in Personal Assistant Agents: A Position Paper. In *Proc. of Autonomy'03*, 2003.

[2] P. Scerri, D.V. Pynadath, and M. Tambe. Why the elf acted autonomously: Towards a theory of adjustable autonomy. In *Proc. of AAMAS'02*, 2002.

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

[1] One way is to use numerical methods.

[2] Note that decisions are only made in FTOC nodes. The benefit of PTOCs are *indirect*, and is reflected in the overall EU calculation.