A Decision Procedure for Autonomous Agents to Reason about Interaction with Humans

Michael Fleming
Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada
E3B 5A3
mwf@unb.ca

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

Abstract

In this paper, we present computational models for the design of autonomous agents that can make rational decisions about interaction with potentially helpful users. One challenge in designing these agents is to specify when the agent should take the initiative to interact with the user. We propose specific factors that must be modeled, as well as methods for how to combine these factors in order to make rational decisions about interaction, based on whether the perceived benefits of communication exceed the expected costs.

Introduction

Autonomous systems are designed to perform tasks on behalf of users, working with input provided initially by a human user, but operating independently afterwards. Under certain circumstances, however, these systems may reason that it is beneficial to acquire further information from the human, either because the environment has changed unexpectedly or simply because the system is sufficiently uncertain that its proposed problem solving will really meet the needs of the human. Our research aims to provide a decision procedure for systems to reason about interaction with humans, regardless of the area of the application.

Our approach is to model both the benefits and the costs of the possible interaction, and to have the system interact only if the benefits outweigh the costs. This is achieved by simultaneously modeling the system's perception of the human user in a user model, the system's perception of the current state of the dialogue and the system's perception of the state of the problem solving represented in a task model.

In this paper, we outline a specific proposal for what to model in order to reason about interaction, leading to exchanges that are efficient and meaningful. We also show how this decision procedure is designed to be sensitive to the current state of the system and, as such, is capable of responding to changing situations in the environment.

A Model for Reasoning about Interaction

Consider a scenario where an autonomous agent has been designed to perform a task on behalf of a human. In situations where both the human and the system can take the initiative to direct the problem solving or to engage in dialogue, to further advance the problem solving, this arrangement is typically referred to as a mixed-initiative system (see, for instance, (Kortenkamp *et al.* 1997; Cox & Veloso 1997)). Most designers, however, have simply come up with their own domain-specific mixed-initiative solutions to their particular problem of interest. What is needed is a principled method for designing such systems: one that takes into account information about the task, the user and the dialogue simultaneously.

To begin, consider the general situation in which an artificial intelligence system might find itself at any given time. It is common practice in artificial intelligence to speak of the system's current *state*. A state representation should capture everything that a system needs to know to distinguish its current situation from any other situation in which it might find itself while performing a task. It will be proposed here that the current state of any interactive system should depend on a number of factors, each of which can be classified as belonging to one of three general components of the system: the task model, user model, and dialogue model.

The various factors then need to be modelled in order to measure the relative benefits and costs of interacting with the user. These factors are summarized in Table 1 below.

Factor	Model
The user's knowledge	UM
The user's willingness to interact	UM
The user's preferences/utility function	UM, TM
Tolerance for suboptimality	UM, TM
Current context and expected understandability of system utterance	DM
Previous interactions	DM
The expected improvement of the system's task performance due to interaction	TM
Time and time criticality	TM
Resource costs and other task-specific costs	TM

Table 1: Summary of factors to be used in model

Single-Decision Problem

A certain class of problems can be viewed as "single-decision"; in other words, from an initial state, the system decides about interacting with the user, then makes a decision about what action to perform and then takes that action to complete the task.

Our general approach to reasoning about interaction in these environments is as follows.

Given a question that the system is considering asking the user, perform the following steps.

1. Determine the expected benefits of interacting with the user. More specifically, determine by how much the system's performance on the task is expected to improve (if at all) after asking the user the question.
2. Determine the expected costs of the interaction.
3. Proceed with the interaction only if the benefits exceed the costs.

Benefits The following formula is used for computing the *benefits* of interaction. Let EU_{ask} represent the expected utility of the outcome(s) that would result if the system *did* ask the user the question and then chose an action based in part on the user's response. Let $EU_{\neg ask}$ represent the expected utility of the outcome(s) that would result from the system making its decision without any further interaction. Then, the benefits are computed simply by taking the difference between these two values.

$$\text{Benefits} = EU_{ask} - EU_{\neg ask}$$

When we mention utility in this context, we are referring to the expected value of the problem solution that is reached in some final state. This does not take into account any costs that might be involved in reaching the solution. As will be seen soon, such costs are treated separately in our model.

EU_{ask} and $EU_{\neg ask}$ themselves are computed by summing over the possible outcomes in each case, weighted by the probability of each outcome.

Let us begin with $EU_{\neg ask}$, the expected utility of the action that the system would take without any further interaction. For each possible non-communicative action in the set $Actions$ that the system is able to perform, it computes the expected utility of that action by summing over the utilities of all possible outcomes of the action, weighted by the probabilities of each of those outcomes. The rational action for the system to choose is then the action with the highest expected utility; $EU_{\neg ask}$ is equal to the expected utility of this best possible action. In the equation below, $P(Result_i(A))$ replaces $P(Result_i(A)|E, Do(A))$, the probability that $Result_i(A)$ would occur given that the agent has gathered evidence E about the world and has performed action A. This simplification is made to improve the readability of this formula and others to follow in this section.

$$\begin{aligned} EU_{\neg ask} &= \max_{A \in Actions} EU(A) \\ &= \max_{A \in Actions} \left(\textstyle\sum_i P(Result_i(A)) \times U(Result_i(A))\right) \end{aligned}$$

To calculate EU_{ask}, the expected utility of the outcome that will result if we *do* interact with the user, we introduce a new variable. Let P_{UK} represent the probability that the user will actually have the knowledge required to answer the question. Also, let $Resp$ represent the set of possible responses to the question. Then EU_{ask} is computed as:

$$\begin{aligned} EU_{ask} &= P_{UK}[EU(\text{if user knows})] \\ &\quad +(1 - P_{UK})[EU(\text{if user does not know})] \\ &= P_{UK} \sum_{r \in Resp} P(\text{Resp} = r) EU(\text{S's action}|\text{Resp} = r) \\ &\quad +(1 - P_{UK}) EU_{\neg ask} \end{aligned}$$

In words, EU_{ask} is the sum of two terms. The second of these terms captures the case in which the user is unable to answer the question. In this case (occurring with probability $1 - P_{UK}$), the system would simply fall back on what it would have done without interacting (with expected utility $EU_{\neg ask}$). The first term considers the case in which the user *is* able to answer the question (probability P_{UK}). The expected value of the system's actions in this case is found by considering all the possible responses that the user might give and the expected utility of the action that the system would take in each of those cases.

In the formula above, $EU(\text{S's action}|\text{Resp} = r)$ is calculated by taking the maximum of the expected utilities of all possible actions.

$$\max_{A \in Actions} \left(\sum_i P(Result_i(A)) \times U(Result_i(A)) \right)$$

Costs The *costs* of interaction in our model are represented using a linear model: the total cost is a weighted sum of any individual cost measures C_i that have been identified for the application domain. Each of these factors is normalized so that the possible values range from 0 to 100, with a cost of 0 indicating no cost at all and a cost of 100 representing the maximum possible cost in this application domain.

$$\text{Costs} = \sum_i w_i C_i$$

Note that each of these cost measures C_i is actually a cost function that might depend on the current state, on the particular action or question being considered, and/or on certain components of the user model.

In the subsequent example, the cost of interaction will be measured by a weighted sum of only *two* cost factors: t, the cost associated with the estimated time required for the interaction, and b, the cost associated with bothering the user in the current situation.

A single-decision example We will now present a simple example to illustrate the application of our model to decision-making about interaction.

In this example, the system has been asked to plan a travel route for the user to get from City A to City B. It determines that it has the choice between two different paths. According to the system's knowledge, path 1 is shorter but is congested (due to heavy traffic) about 50% of the time. Path 2 is significantly longer, but it is never busy.

For the purpose of this example, we assume that the system has access to a utility function for this type of situation. The utility function assigns a value to each possible outcome in the domain, where an outcome consists of the decision that was made by the system (which route did it choose to take?) and the actual state of the world (was path 1 in fact congested?). For example, one outcome would be that the system opted to take path 1, but found that it turned out to be busy and, therefore, slow.

The values assigned by the utility function in this example are meant to capture the attitudes of the average user toward different possible outcomes in this domain, and are shown in Table 2.

The ideal outcome in this example would be if we were to choose path 1 and if it were to turn out to be clear. The worst outcome would involve choosing path 1 and then finding out that it is congested. In between these two extremes, choosing path 2 is a fairly safe decision, but we would be somewhat less pleased if we were to choose path 2 when, in fact, the shorter path had been available.

System's choice	**Actual state of path 1**	**Utility**
Path 1	Path 1 clear	100
Path 1	Path 1 busy	0
Path 2	Path 1 clear	50
Path 2	Path 1 busy	70

Table 2: Utility function for path example

Now, suppose that we believe that the user might have access to recent traffic information and could therefore help with the decision-making. We believe that there is a 60% chance that the user has accurate traffic information.

To make the problem slightly more interesting, we will assume that there is an additional 10% chance that the user has traffic information that turns out to be *incorrect*.

The remaining 30% is assigned to the case in which the user states that he has no additional traffic information. In this case, the system should fall back on what it would have done without asking the user.

Let us first consider the decision that the system would make in the absence of any further information from the user. In other words, we want to compute the value of $EU_{\neg ask}$, the expected utility of the best action the system could take if it did not ask the user the question.

There are two possible actions in this simple example: choose path 1 or choose path 2. According to the problem description above, there is a 50% chance that path 1 will be congested. The expected utility of choosing path 1 is computed as follows:

$$\begin{aligned}
&EU\ (A = \text{path } 1) \\
&\quad = \textstyle\sum_i P(Result_i(A)) \times U(Result_i(A)) \\
&\quad = P(\text{path 1 clear}) \times U(A = \text{path 1} \wedge \text{path 1 clear}) \\
&\qquad + P(\text{path 1 busy}) \times U(A = \text{path 1} \wedge \text{path 1 busy}) \\
&\quad = 0.5\ (100) + 0.5\ (0) \\
&\quad = 50
\end{aligned}$$

Similarly, the expected utility of path 2 depends on the system's current beliefs about the state of path 1.

$$\begin{aligned}
&EU(A = \text{path } 2) \\
&\quad = P(\text{path 1 clear}) \times U(A = \text{path 2} \wedge \text{path 1 clear}) \\
&\qquad + P(\text{path 1 busy}) \times U(A = \text{path 2} \wedge \text{path 1 busy}) \\
&\quad = 0.5\ (50) + 0.5\ (70) \\
&\quad = 60
\end{aligned}$$

$$\begin{aligned}
EU_{\neg ask} &= \max_{A \in Actions} EU(A) \\
&= \max(50, 60) \\
&= 60
\end{aligned}$$

Therefore, with no additional information, it appears that the best solution for the system is to play it safe and choose path 2, since the expected utility of path 2 is higher than the expected utility of path 1.

Now that we know the expected utility of what the system could do *on its own*, let us consider the possible outcomes if the system *were* to ask the user for further information. Table 3 summarizes the possible scenarios that might arise.

User resp.	**Actual state of path 1**	**Prob.**	**S's choice**	**Util.**
Path 1 clear	Path 1 clear	0.30	Path 1	100
Path 1 clear	Path 1 busy	0.05	Path 1	0
Path 1 busy	Path 1 clear	0.05	Path 2	50
Path 1 busy	Path 1 busy	0.30	Path 2	70
No answer	Path 1 clear	0.15	Path 2	50
No answer	Path 1 busy	0.15	Path 2	70
Overall expected utility of system choice after asking				71.5

Table 3: Possible scenarios in path-choosing example

The probabilities in Table 3 are computed by considering both the system's beliefs about whether or not the user will have the knowledge and the system's prior beliefs about the actual state of the path. For example, the system believes that the user will know the correct answer with a probability of 0.6. Since the system initially believed that path 1 was equally likely to be clear or congested, there is a probability of 0.3 of the user correctly saying that the path is clear (Row 1) and a probability of 0.3 of the user correctly saying that it is busy (Row 4).

The overall expected utility (71.5) shown in the final row of Table 3 is computed by summing over all the possible outcomes, weighted by their probabilities. This tells us that, in the average case, interacting with the user will lead the system to a choice with an expected utility of 71.5. Recall that, without asking the user, the expected utility of the system's best action – simply choosing path 2 as a safe route – was 60.

Our conclusion is that, despite the fact that the user might not know the answer or might even mislead the system, there is a clear expected benefit to requesting the additional information from the user. We represent this benefit by looking at the expected gain in performance on the task if we were to interact – in other words, by taking the difference between the expected utility after interaction and the expected utility of the system's default action in the absence of further interaction.

In this example,

$$\text{Benefits} = EU_{ask} - EU_{\neg ask} = 71.5 - 60 = 11.5$$

Costs of interaction It is important now to consider the fact that there are also *costs* involved in interacting with the user.

As presented earlier, costs are calculated with a weighted sum over all cost measures C_i that have been identified for the domain.

$$\text{Costs} = \sum_i w_i C_i$$

In this example, the cost of interaction will be measured by a weighted sum of *two* cost factors: t, the cost associated with the extra time required for the interaction, and b, the cost associated with bothering the user in the current situation. To keep this example simple, we will simply state that the time cost is 10 on a scale of 0 to 100 (the interaction will not take long at all), the bother cost is 10 (the communication will not be perceived by the user as being very bothersome), and the weights associated with time and bother are 0.2 and 0.3, respectively.

Assuming these values, the total cost associated with the interaction is:

$$\begin{aligned}\text{Costs} &= w_t t + w_b b \\ &= 0.2(10) + 0.3(10) \\ &= 5\end{aligned}$$

Since the benefits of interaction were computed earlier to be 11.5, the benefits outweigh the costs and our system's optimal decision would be to proceed with the interaction with the user.

Table 4 shows some variations on the above example, demonstrating how the system's decisions about interaction are affected by modifying the values of the relevant factors. For example, while the first row summarizes the earlier example, the second row shows that if the system had believed that the user was not very likely at all to be able to answer the question (with all other factors remaining unchanged), the ultimate decision would have been instead to forgo interaction. The third and fourth rows show that if the scenario had been changed so that the system is initially almost certain that Path 1 is clear or almost certain that it is congested, then interacting would not be beneficial at all.[1] The fifth row shows a case where, despite the fact that the system was initially quite sure about the state of path 1, it still decides to interact because it is certain that the user will know the answer. The final two rows demonstrate the effects of modifying the values of the two cost factors.

[1]The *negative* value for benefits in fact demonstrates that the system would likely *decrease* its performance on the task if it were to ask the user, since the likelihood of obtaining new information is quite low and since there is a chance of actually being misled by the user's response.

The utilities of the possible outcomes are assumed to be the same in all cases.

P_{UK}	**Prob. P1 clear**	t	b	EU_{ask}, $EU_{\neg ask}$	**Ben.**	**Costs**	**Ask?**
0.6	0.5	10	10	71.5, 60	11.5	5	Yes
0.2	0.5	10	10	61.5, 60	1.5	5	No
0.6	**0.9**	10	10	89.7, 90	-0.3	5	No
0.6	**0.1**	10	10	64.7, 68	-3.3	5	No
1.0	**0.9**	10	10	97, 90	7	5	Yes
0.6	0.5	**50**	10	71.5, 60	11.5	13	No
0.6	0.5	10	**40**	71.5, 60	11.5	14	No

Table 4: Variations on path-choosing example

Extended Modeling of Costs

In cases where the system may be interacting with the user multiple times within a given system-user dialogue, we can build on the proposed solution, continuing to model user model, task model and dialogue model factors and to weigh benefits against costs in order to determine whether to interact.

Some additional issues arise, however. First of all, it is important to develop more detailed models of the inherent costs of interaction.

The actual cost associated with bothering the user with a particular question should depend on a user-defined willingness factor w, measured on a scale from 0 to 10. In the single-decision example, this willingness factor is the only concern when it comes to estimating the cost of bothering the user. However, in a sequential decision problem, the bother cost should also incorporate a measure of how much the user has been bothered in the dialogue so far. In essence, recent interruptions and difficult questions should carry more weight than interruptions in the distant past and very straightforward questions. Further research effort must be devoted to determining an appropriate function for the cost of bothering the user. However, according to the desired behaviour just described, we propose the following function for the bother cost.

For every past interaction I with the user, let $t(I)$ be the amount of time that has elapsed since that interaction. The specific implementation assumed in this paper involves dividing time into discrete time steps and using the number of steps as the value for $t(I)$.[2] Let $c(I)$ be an estimate of how bothersome the interaction actually was, in terms of the cognitive effort required of the user to answer the questions. Then, we use the formula

$$\text{(Bother so far)} BSF = \sum_I c(I) \beta^{t(I)}$$

[2]In domains where time is better treated as a continuous variable, the system designer can adjust the formula so that $t(I)$ is the actual time elapsed divided by some pre-determined constant.

to give us an idea of how bothersome the dialogue has been so far. The term β is a discount factor, $0 < \beta \leq 1$, that is used to accomplish the goal of diminishing the impact of interactions that took place a long time ago.

Suppose we bothered the user 2 time steps ago, 7 time steps ago and 13 time steps ago. Assuming that each interaction had a cost[3] of $c(I) = 1$ and that $\beta = 0.95$, our estimate of "bother so far" BSF is $0.95^2 + 0.95^7 + 0.95^{13} = 2.11$.

The willingness of the user to interact (the variable *w*, on a scale of 0 to 10, introduced earlier) is then incorporated as follows. We define two new terms $\alpha = 1.26 - 0.05w$ and $Init = 10 - w$. [4] The bother cost is then computed as

$$bother = Init + \frac{1 - \alpha^{BSF}}{1 - \alpha}$$

Suppose, for example, that the user is a very willing one, with $w = 9$. Then α would be $1.26 - 0.05w = 0.81$. If the bother so far is computed to be 2.11 as shown above, then the cost of bothering would be

$bother = Init + \frac{1-\alpha^{BSF}}{1-\alpha} = 1 + \frac{1-0.81^{2.11}}{1-0.81} = 2.89$

If we had bothered the user more frequently and more recently, say at 1,4,6,9 and 15 time steps ago, then the bother so far would be 3.59 and the cost of bothering would be 3.79.

For a less willing user (one with $w = 1$), the bother costs for the same two situations described above would work out to be 11.36 and 13.68, respectively.

Another important aspect to the consideration of costs of interaction is the consumption of system resources. For example, in some domains, the system (1) might have to perform several database queries (which might come at some cost), (2) might have to communicate with *other* agents if information cannot be obtained from a user (bother cost analogous to the user bother, *plus* cost of communication channel, etc.), (3) might have to use CPU time, memory, disk space and other computational resources.

These costs will be domain-specific, but any that are deemed to be relevant should be incorporated as a factor C_i in the costs formula introduced earlier:

$$\text{Costs} = \sum_i w_i C_i$$

This is especially important when comparing the value of asking the user a question and the value of performing other actions to try to obtain this information. Interaction with the user might take some time and might inconvenience the user. However, if the information involved is essential for the system, and if the only alternative to asking the user is an extremely expensive query to a remote database, then the interaction with the user should turn out to be the better choice.

Multiple Decision Problems

Most autonomous agents will in fact have to reason about interacting with the human at several points in their processing. In (Fleming 2003), we discuss various options for going beyond the single decision case. We summarize these briefly, in this section.

One possibility is to use a Markov Decision Process (MDP), using strategies such as value iteration to determine the optimal policy of an agent. In this scenario, it will still be important to model various user-specific factors and various costs, to determine whether it is beneficial to initiate interaction. The expected utility of actions will be affected by the probability that the user will know the information being requested, the cost of bothering the user at this point in the dialogue and the cost of losing time due to the proposed interaction, for example.

There are some difficulties in using MDPs, however. The agent's state representation must include information from the user model and from the dialogue history. This results in possibly explosive state spaces.

An alternative approach is to make use of current information available to the system, without projecting into the future. This so-called information-theoretic approach sets the value of interaction to be:

$$\text{Value of question} = \kappa \times P_{UK} \times \text{Importance} \times \text{Uncertainty}$$

Importance captures how critical it is to answer this particular question. As importance increases, the system becomes more likely to interact with the user. The same is true for uncertainty (if the question to be asked has n possible answers and one is very likely to be true, then there is less value to interaction). Again, all of this calculation is tempered by a model of whether the user will indeed have the knowledge being requested. κ is a constant intended to place benefits and costs on the same scale; more details can be found in (Fleming 2003).

We have then formulated a design procedure for those designing systems of autonomous agents, to select the most appropriate decision process for reasoning about interaction, based on whether the single decision case is appropriate or whether there is a potential for difficulties in employing a MDP model for multiple decisions. Once more, the details are omitted here but can be seen in (Fleming 2003).

To gain a deeper understanding of the alternative information-theoretic model, consider again the simple path-choosing example. The question that was being considered in that example was a simple question with only two possible answers and with the system believing that those two answers were equally likely. Therefore, we can use the information-theoretic approach to measuring uncer-

[3]In this example, we assume that all interactions are equally costly; for some domains, system designers might choose to make certain types of questions more or less costly.

[4]These are simply suggested values; the system designer might consider doing some empirical research to determine the most appropriate values for a given domain. The proposed formula for α is intended to give a nearly linear bother curve for users with moderate willingness values and bother curves with more exponential and logarithmic appearances, respectively, for more unwilling and willing users. The value of $Init$ is intended to reflect the cost of bothering a user for the first time. Our choice for $Init$ assumes that this cost will be negligible for a very willing user ($w = 10$) and quite high for an unwilling user ($w = 0$).

tainty. The information content[5] of the question is:

$$I\left(\frac{1}{2},\frac{1}{2}\right) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1 \text{ bit}$$

.

As specified in the original description of the example, the probability of the user actually knowing the correct answer to the question is 0.6.

The final components of the formula are the importance of the question and the constant value κ. In this case, the answer to the question is fairly important, but it is definitely possible for the system to proceed without the user's help. Suppose that the importance of this question has been set at 40, and suppose that κ has been learned to be 0.4 Then,

Value of question
= $\kappa \times P_{UK} \times$ Importance $\times$ Uncertainty
= $0.4 \times 0.6 \times 40 \times 1$
= 9.6

Table 5 shows the same variations on the path-choosing example that were included in Table 4. Costs are computed in exactly the same way as before and so the cost values are simply copied from the earlier table. However, the benefits are now computed by using the above formula for the value of asking the question, rather than by considering the expected utilities of the course of action that would be taken if the system did or did not interact. If the two tables are compared, it can be seen that the decisions made by the two agents are identical. However, the agent described in this section accomplishes this without expensive reasoning about future sequences of events.

P_{UK}	**Prob. P1 clear**	t	b	**Info. cont.**	**Ben.**	**Costs**	**Ask?**
0.6	0.5	10	10	1	9.6	5	Yes
0.2	0.5	10	10	1	3.2	5	No
0.6	**0.9**	10	10	0.47	4.5	5	No
0.6	**0.1**	10	10	0.47	4.5	5	No
1.0	**0.9**	10	10	0.47	7.5	5	Yes
0.6	0.5	**50**	10	1	12	13	No
0.6	0.5	10	**40**	1	12	14	No

Table 5: Variations on path-choosing example revisited

Discussion

When to initiate interaction between autonomous agents and their human users is an important topic of research. In this paper, we have proposed a design strategy that operates, independent of the domain of application of the system. The proposal, to model and compare the relative benefits and costs of any possible interaction, is enhanced with details of specific factors that are especially useful to include, as part of the modeling, such as: the probability that the human knows, the willingness of the human to interact, the criticality of the underlying task and the criticality of the processing time. In contrast with approaches such as (Scerri, Pynadath, & Tambe 2001) that focus on how best to use an MDP approach, we discuss other models of reasoning that may be applicable, depending on the domain of application. Through this research, we have emphasized the need to reason about the specific humans who are directing the processing of the agents and their needs and limitations, as well as specific information about the task at hand and about the dialogue that will be generated, as interactions continue to occur. As such, we provide a principled framework for making these important decisions about interaction.

We view the agents' decision to interact as a way of relinquishing autonomy to the human, allowing the human to set the direction of the problem solving. This perspective on adjustable autonomy is discussed further in (Cohen & Fleming 2003), and contrasts with the approach discussed in (Brainov & Hexmoor 2003) where an agent is considered autonomous if it is beyond the control and power of another agent. In our case, the agent is sufficiently autonomous in order to give up its control to the human. (Hexmoor, Falcone, & Castelfranchi 2003) includes a more detailed discussion on the concept of agent autonomy.

[5] See (Russell & Norvig 1995), for example, for the definition of information content.

References

Brainov, S., and Hexmoor, H. 2003. Quantifying autonomy. In Hexmoor, H.; Falcone, R.; and Castelfranchi, C., eds., *Agent Autonomy*. Kluwer Publishers. chapter 4.

Cohen, R., and Fleming, M. 2003. Adjusting the autonomy in mixed-initiative systems by reasoning about interaction. In Hexmoor, H.; Falcone, R.; and Castelfranchi, C., eds., *Agent Autonomy*. Kluwer Publishers. chapter 7.

Cox, M., and Veloso, M. 1997. Controlling for unexpected goals when planning in a mixed-initiative setting. In *Proceedings of the 8th Portuguese AI Conference, Coimbra, Portugal*, 309–318.

Fleming, M. 2003. Reasoning about interaction in mixed-initiative artificial intelligence systems. Technical report, University of Waterloo Ph. D. thesis.

Hexmoor, H.; Falcone, R.; and Castelfranchi, C., eds. 2003. *Agent Autonomy*. Kluwer Publishers.

Kortenkamp, D.; Bonasso, R.; Ryan, D.; and Schreckenghost, D. 1997. Traded control with autonomous robots as mixed-initiative interaction. In *Papers from the 1997 AAAI Symposium on Computational Models for Mixed Initiative Interaction*, 89–94. AAAI Press.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc.

Scerri, P.; Pynadath, D.; and Tambe, M. 2001. Adjustable autonomy in real-world multi-agent environments. In *Proceedings of Agents 2001 Conference*.