Decisions about Turns in Multiparty Conversation: From Perception to Action

Dan Bohus
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
+(01) 425 706 5880
dbohus@microsoft.com

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
+(01) 425 706 2127
horvitz@microsoft.com

ABSTRACT
We present a decision-theoretic approach for guiding turn taking in a spoken dialog system operating in multiparty settings. The proposed methodology couples inferences about multiparty conversational dynamics with assessed costs of different outcomes, to guide turn-taking decisions. Beyond considering uncertainties about outcomes arising from evidential reasoning about the state of a conversation, we endow the system with awareness and methods for handling uncertainties stemming from computational delays in its own perception and production. We illustrate via sample cases how the proposed approach makes decisions, and we investigate the behaviors of the proposed methods via a retrospective analysis on logs collected in a multiparty interaction study.

Categories and Subject Descriptors
H.1.2 **[Models and Principles]**: User/Machine System – *Human Information Processing*; H.5.2 **[Information Interfaces and Presentation]** User Interfaces – *Natural Language*;

General Terms
Algorithms; Human Factors

Keywords
Multiparty turn taking; decision-theoretic approach; multiparty interaction; floor management; gaze; speech; spoken dialog; situated interaction; multimodal systems.

1. INTRODUCTION
The naturalness and usability of spoken dialog systems depends critically on the control of the *fine structure* of the timing of turns. Challenges in this realm include unexpected, long pauses after the completion of an utterance by a user, barge-ins by a system before a user has completed speaking, and floor conflicts resulting from confusion about turns. To date, most research on controlling turn taking has been undertaken in dyadic settings. Here, we consider the problem of turn taking in *multiparty settings*, where a spoken dialog system engages several people in a joint conversation.

ICMI'11, November 14–18, 2011, Alicante, Spain.

In multiparty interactions, beyond challenges with the detection of the end of the utterances of speakers, a system also must reason about the source and target of utterances, and, more generally, the state and dynamics of the floor. At any point, one participant may speak to another or to the system, or wait for a response from another participant or the system, or even openly reflect about one's own thoughts or plans before generating a contribution or while waiting for someone else to contribute. Given the multiparty and mixed-initiative nature of this process, a dialog system cannot assume that it can take the floor upon the detected end of another participant's turn, even if perfect end-of-turn predictions were available. Effective turn taking in multiparty dialog hinges on making accurate inferences about the multiparty conversational dynamics from streams of perceptual data, and on reasoning under uncertainty about various possible outcomes, their utilities, and tradeoffs between acting and waiting for additional information. In addition, given the time-critical nature of turn-taking decisions, a system should also take into account uncertainties arising from delays in its own perception and rendering pipelines.

We present a decision-theoretic approach for guiding turn taking in a spoken dialog system in multiparty settings. Our goal is to highlight opportunities and directions with moving from heuristics for turn taking to the use of principled decision policies guided by the computation of the expected utilities of different actions. We focus on decisions by a system to take the floor at different times after detecting the end of utterances of other participants in a multiparty dialog. We show how inferences about the conversational dynamics and the system's processing delays can be coupled with the assessed utilities of different outcomes to make floor control decisions under uncertainty. We explore the behaviors of the expected-utility policy and compare these with performance of heuristic procedures used in a previous version of the system. The comparative study is based on a retrospective analysis of logs captured by a spoken dialog system in multiparty settings.

We begin by reviewing related work. Then, we discuss a multiparty interaction user study, including lessons learned. Next, we formalize the turn-taking problem we shall focus on, and describe the approach. We showcase by example how the expected-utility methods guide actions, and we report results from a comparative analysis of the use of heuristics versus more principled inference and decision making.

2. RELATED WORK
Researchers in the psycholinguistics, sociolinguistics and conversational analysis communities have investigated various facets of

turn taking in human-human interaction. Sacks, Schegloff and Jefferson [11] proposed a basic model for the organization of turns in conversation, centered on the notions of *turn constructional units*, separated by *transition relevance places* which provide opportunities for speaker changes. Subsequent works have brought to the fore the important role played by gaze, gesture, and other non-verbal communication channels in regulating turn taking in interaction. For instance, Duncan [5] highlights the role of non-verbal signals in turn taking, and proposes that turn taking is mediated via a set of verbal and non-verbal cues. Wiemann and Knapp [22] survey a number of previous investigations and perform a quantitative analysis of turn-taking cues in dyadic conversations. Goodwin [7] also discusses various aspects of the relationship between turn taking and attention.

In contrast, most spoken dialog systems built to date operate in dyadic settings, and simply try to alternate turns with the user in a *"you speak then I speak"* manner. In this case, one of the key ingredients for making good turn-taking decisions is the ability to predict the end of the user's turns. A number of research efforts have addressed this problem. For instance, [1] describes a system that uses a semantic parser to classify incoming utterances as *closing* or *non-closing*. Machine learning techniques have also been used in conjunction with prosodic, syntactic, semantic, and/or higher-level dialog features to make predictions about end of turns [6, 12, 15, 18], or to optimize end-pointing thresholds [9].

Efforts with spoken dialog systems have also been directed towards developing, implementing and testing more comprehensive computational models and architectures for managing turn taking, in both dyadic [10, 19, 20] and multiparty settings [2, 4, 21]. With respect to turn-taking decisions, a number of more principled approaches have been proposed. As an example, [16] proposes a bidding approach to turn taking and investigates in simulations the use of reinforcement learning techniques for choosing appropriate turn bids, based on utterance importance. [8] proposes and evaluates via simulations a reinforcement learning approach whereby a system learns how to take turns online, by using a reward measure based on gaps and overlaps.

In more closely related work, [10] presents a turn-taking model for dyadic interactions based on a 6-state finite-state machine, and articulates a decision-theoretic approach for end-pointing user utterances and grabbing the floor. Uncertainties are considered about the turn-internal or turn-final nature of each detected pause. Under a specific set of cost functions motivated by turn-taking principles, the authors compute analytically the time at which the cost of waiting exceeds the cost of grabbing the floor and use it to perform end-pointing. Experiments in batch and with a live system indicate the approach improves the system's responsiveness.

3. BACKGROUND

We begin by outlining the computational model for multiparty turn taking that provides the basis for this work, and we review lessons learned from a multiparty interaction experiment with an initial, heuristic implementation of turn-taking policies [4].

3.1 Turn-Taking Model

In a multiparty turn-taking model that we described previously [2, 3], we consider turn taking as a mixed-initiative, collaborative process that emerges from coordinated floor actions produced by participants engaged in a conversation. The participant who is ratified to speak via this collaborative process is said to have the *conversational floor*. Participants continuously produce one of four *floor management actions*: the participant that currently has the floor produces either a *hold* action, indicating that they continue to hold the floor, or a *release* action, indicating that they are releasing the floor to someone else. The *release* action also specifies the set of participants the floor is being released to; an empty set allows for self-allocated next speaker selection [11]. Participants who currently do not have the floor produce either a *take* action, indicating that they want to take the floor, or a *null* action, indicating that they make no floor claims. Under this model, floor shifts emerge from the joint actions of the participants in the conversation. Specifically, the floor transitions from one participant to another when a *release* action is met with a corresponding *take* action.

The model subsumes components for sensing the conversational dynamics, for making floor control decisions, and for rendering these decisions into an appropriate set of coordinated behaviors. In short, the sensing component subsumes inference models that leverage audio-visual evidence to estimate the current speaker, the addressee roles for each participant with respect to the current utterance, as well as the floor state, actions and intentions for each engaged participant. The decision component uses the estimated state information as well as high-level dialog context to select the floor management actions to be performed by the system. The floor actions are communicated both to the dialog management layer, which generates the system's semantic contributions, and to a turn-taking behavioral control component that renders these actions into synchronized gaze, gesture and speech [2, 3].

3.2 Experiment and Lessons Learned

We implemented an initial version of the turn-taking model in a multimodal interactive dialog system that can play a *questions game* with groups of people [3, 4]. In the game, the system asks trivia questions (see Figure 1) and displays the list of possible answers on the screen. Users can discuss the question and provide answers. After a confirmation, the system provides an explanation if the answer is incorrect, and then moves on to the next question. We conducted a large-scale user study with this system [3, 4] and collected a total of 150 interactions: 90 with groups of two people interacting with the system and 60 with groups of three people interacting with the system. Sample interactions are available online [17].

The initial implementation of the turn-taking model was based on heuristics for sensing, decision making, and behavioral control, and is described and evaluated in detail in [3, 4]. On perception and inference, the current speaker and set of addressees for the current utterance were identified via handcrafted models that used sound source localization information from the microphone array in conjunction with information from a visual scene analysis (e.g. the location and estimated head-pose for each speaker), as well as some additional rules (e.g. non-understandings and utterances longer than three seconds were assumed to be addressed to others). At the end of each utterance, the system assumed that the floor was being released to the person or people that the utterance was addressed to. While this is not always true, we found it to be a good assumption for the questions game.

Turn-taking decisions were also based on a heuristic policy. When the system detected that the floor was being released to it, it generally took the floor immediately and generated a turn. In a limited number of cases, the system did not generate a verbal contribution, but instead tried to pass the floor to another participant via a non-verbal gesture: it turned the avatar's face towards the other participant and lifted the eyebrows [3]. When the system detected that the floor was not released to it at the end of an utterance, it waited for a predefined duration (in most cases 3.5 seconds), before trying to take the floor, giving someone else a

Figure 1. Questions game with two participants.

chance to take the floor. This duration was set to give participants a reasonable time for reflection about another participant's response or question. In a limited number of cases, where the system tried to quickly regain the floor after having been interrupted while posing a question, the waiting duration was set to 0.5 seconds. The system released the floor at the end of its own utterances, and also when interrupted while asking a question.

Following the experiment, the *addressees* of each utterance were tagged by a human annotator. This enabled us to assess the accuracy of the system's heuristic inference models. We found an overall error rate of 18% on detecting whether an utterance was addressed to the system or not, and therefore on identifying floor releases to the system, as described earlier.

For 9% of utterances, the system *incorrectly inferred that the floor was being released to it*. In the majority of these cases, the system took the floor and immediately issued a verbal contribution. As the floor was in fact not released to the system, this action often (42%) led to significant turn-taking problems, manifested as *floor transition battles* marked by *turn-initial overlaps*: a participant started talking at around the same time as the system did, and an undesirable overlap was created. The data shows that such turn-initial overlaps also occurred (albeit in lower proportion, 10%) when the system produced a verbal contribution after *correctly identifying* that the floor was being released to it. We believe these turn-initial overlaps are explained in part by the system's response delay. Due to technological limitations in the input (~700ms) and output (~150ms) processing pipelines the system's response arrives after a delay. This delay can be taken as a cue that the system does not intend to take the floor, provoking other participants to take initiative. Turn-initial overlaps are also caused by the mixed-initiative nature of turn taking, with participants vying for the floor and inserting their own contributions immediately after an answer directed to the system.

On another 9% of utterances, the system *failed to detect that the floor was being released to it*. In these cases the system waited for a specified duration (in most cases 3.5 seconds), after which, if no one generated another contribution, the system took the floor. In these cases the system can appear unresponsive and the participants often re-take the initiative prior to the system's timeout.

4. DECISION-THEORETIC APPROACH

The decision-theoretic approach we describe below provides a more principled solution for making turn-taking decisions and for dealing with some of the issues highlighted above. The approach allows the system to continuously deliberate about key uncertainties in the world and in its own processing delays, and resolve tradeoffs between waiting and taking the floor. It helps reduce the number of floor battles, and minimize gaps in the conversation.

We focus on the situation where the system does not have the floor and has to decide whether it should attempt to take the floor, or simply wait. We restrict the search to the subspace of turn-taking policies where: (1) the system may take the floor only if no one else is speaking, e.g. the system never interrupts a speaking participant, and (2) the system always takes the floor once a silence longer than a specified duration (δ_{MAX}) is observed. While we limit our focus to making take versus wait decisions, we believe similar methods can be applied to make decisions over richer sets of turn-taking strategies.

As delays with computational processing can influence time-critical turn-taking decisions, we need to build explicit machinery that takes them into account. We assume a stochastic input processing delay (ID) between the moment an utterance starts (or ends) and the moment that this event is detected by the system. Such processing delays are common with voice activity detectors, as they employ an audio buffer (and sometimes run a phone-loop or even leverage grammar) to segment spoken signals. Speech recognition may introduce additional delays. Similarly, we assume a stochastic output delay (OD) between the moment the system decides to speak and the moment speech is actually produced. We assume these delays can be observed retroactively by the system and can be probabilistically modeled from data.

4.1 Decision-Theoretic Representation

We consider that an instance of a decision problem is generated each time the endpoint of a participant's utterance is detected and the system does not already have the floor. We denote this moment by t'_e, the time of origin for our analysis, i.e. $t'_e = t_0 = 0$. Given delays in the input processing pipeline, the actual time at which the detected speech segment ended can be determined to be $t_e < t'_e$ (see Figure 2(a)) We use the regular notation t_v to denote the *actual time* a particular event v happens, and the prime notation t'_v to denote the time when the system *detected* that event.

Starting at t_0, the system faces a sequential decision-making problem: at each time step between t_0 and $t_{MAX} = t_0 + \delta_{MAX}$, it must decide whether to take the floor, i.e. speak, or perform a null action, i.e. wait. Note that once the system decides to take the floor and does so, the problem instance is completed. A new problem instance will be generated after the next detected segment of user speech. Similarly, if another participant takes the floor while the system is waiting and the system observes the take, the floor transitions to that participant and the problem instance is again completed. Due to this special structure of the decision problem, the number of possible action sequences (plans) that the system can follow is linear rather than exponential in the length of the $[t_0, t_{MAX}]$ time interval. If we discretize this time interval, the alternative action sequences to consider have the form: *null, null, … take*. Each action sequence can thus be described as $TakeAt(t'_s)$, meaning *wait until a time t'_s and take the floor at that point*. We also note that, given output processing delays, the system will actually start producing speech at a later time, $t_s > t'_s$.

We can compute the expected cost for each action sequence. Initially, at t_0, we compute the optimal time for the system to take the floor as:

$$t^* = \underset{t'_s \in [t_0, t_{MAX}]}{\operatorname{argmin}} \; E_{P(O|\psi_0)}[Cost(O, TakeAt(t'_s))]$$

where $P(O|\psi_0)$ is the *probability of the final outcome O* of the system's plan, conditioned on the *evidence we have collected so far* up to t_0, denoted by ψ_0; we shall discuss the set of possible outcomes O shortly. If the cost-minimizing time is $t^* = t_0$, the

system will perform a floor take right away, at t_0. If the cost-minimizing time is $t^* \neq t_0$, then the system waits at t_0. By the next time tick, t_1, additional evidence has accumulated: the new evidence set is ψ_1 and the system can re-compute the optimal point for taking the floor. Generally, if we have arrived by successive null actions, i.e. by waiting, to the current moment t_c (see Figure 2), we estimate the optimal point for taking the floor as:

$$t^* = \text{argmin}_{t'_s \in [t_c, t_{MAX}]} E_{P(O|\psi_c)}[Cost(O, TakeAt(t'_s))]$$

The proposed approach rests on two ingredients: a cost function, and a probabilistic model for tracking uncertainties in the world. The cost of the system's plan to $TakeAt(t'_s)$ depends ultimately on the final outcome of that plan, and on the conversational context. Depending on whether and when another participant starts speaking, three types of outcomes may stem from the system's decision to take the floor at some future time: the floor might transition to the system, the floor might transition to another participant, or a floor transition battle might arise. To define these outcome types more precisely, we denote by t_u the next moment at which a participant will start speaking again. Let t'_u ($t'_u > t_u$) denote the moment when the system detects this event. Finally, recall that the system starts speaking at time $t_s > t'_s$. The three potential outcome types can then be defined as follows:

1. *FloorTransitionToSystem:* $t_s + \epsilon < t_u$ (Fig. 2.b)
   The floor successfully transitions to the system if the system takes the floor and no one else claims the floor for up to some time ϵ after the system started speaking, i.e. $t_s + \epsilon < t_u$, i.e. no turn initial overlap occurs.
2. *FloorTransitionToOther*: $t'_u < t'_s$, (Fig. 2.c)
   The floor transitions to another participant if that participant starts speaking and the system detects that event before it starts its own attempt to take the floor, i.e. $t'_u < t'_s$. In this case, the system abandons its own plan to take the floor at t'_s.
3. *FloorTransitionBattle:*$(t'_u \geq t'_s) \wedge (t_u \leq t_s + \epsilon)$, (Fig. 2.d)
   A floor transition battle occurs if another participant starts talking around the same time as the system does, i.e. a turn-initial overlap occurs. First, the system must not have detected that the participant has started talking at the time it started its own take action, i.e. $t'_u \geq t'_s$ (otherwise the system would have let the floor transition to the participant). Second, the participant must have started talking no later than some ϵ after the system started talking, i.e. $t_u \leq t_s + \epsilon$ (otherwise the floor would have transitioned to the system).

The possible outcomes depend on the relationships among t_u, t_s, t'_u, t'_s. The cost of each outcome is also influenced by additional context. An important factor is the floor action produced by the floor holder at the end of the previous utterance: the cost of a system *take* will be different depending on whether the floor was released to the system or not. We model release actions via a binary variable fa. The magnitude of the timings involved in the floor transition also plays an important role. For instance, if the floor was released to the system, we expect that the cost increases with the amount of time the system waits until it takes the floor.

Having identified fa, t_u, t'_u, t_s, t'_s as the key world state variables that condition the possible outcomes and cost, we model uncertainty over these variables and compute the expected cost for the system's plan. Given the evidence ψ_c accumulated up to t_c, the system's plan to $TakeAt(t'_s)$ will therefore lead to an outcome $O = \langle FA, T_u, T'_u, T_s, t'_s \rangle$ with probability P_O (capital letters denote random variables and lowercase letters denote known quantities):

$$P_O = P(FA, T_u, T'_u, T_s | t'_s, \psi_c)$$

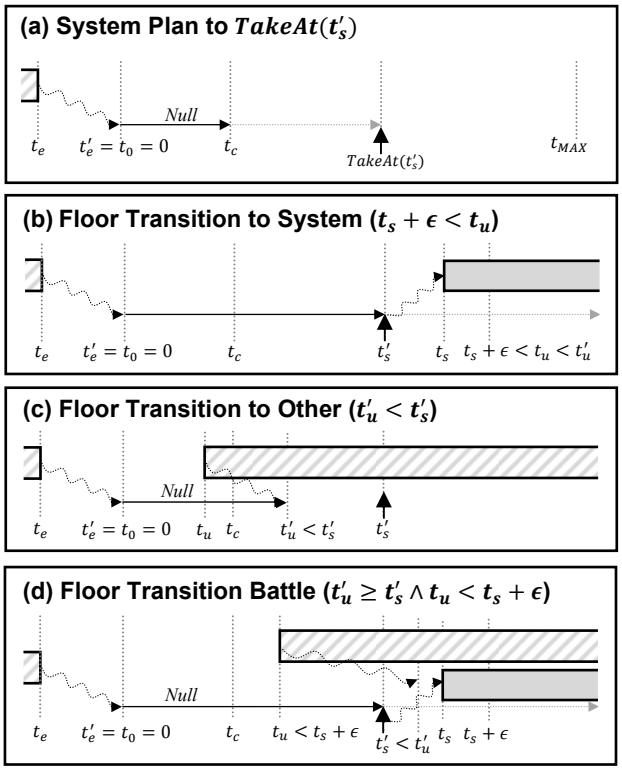

Figure 2. Illustrations of system plan to take the floor at time t'_s (a), and three possible outcomes (b,c,d).

Given a set of reasonable independence assumptions that we review below, we can factor P_O as follows:

$$P_O = P_{cd}(FA, T_u | \psi_c) P_{id}(T'_u | T_u, \psi_c) P_{od}(T_s | t'_s) \quad [1]$$

The first factor in the equation above, P_{cd}, represents uncertainty over the conversational dynamics in the world, including, in this case, the floor release action and the next moment a contribution will arrive from one of the participants. The second and third factors, P_{id} and P_{od}, further integrate the uncertainty about the system's input and output processing delays. We have made the assumptions that (1) given ψ_c, FA and T_u are independent of the time t'_s the system plans to take the floor, (2) T_u depends only on T'_u (and not FA), and (3) T_s depends only on t'_s. Below, we discuss each component factor from Equation 1 in more detail.

4.2 Modeling Conversational Dynamics

P_{cd} represents the joint distribution over the floor release action FA and the moment T_u when the next contribution will arrive from other participants, given the evidence ψ_c accumulated up to time t_c. We can factor P_{cd} as follows:

$$P_{cd}(FA, T_u | \psi_c) = P(FA | \psi_c) \cdot P(T_u | FA, \psi_c)$$

The first factor, $P(FA|\psi_c)$, represents the uncertainty over the floor release action. This model can be trained from labeled data, for instance using discriminative learning techniques. The second factor, $P(T_u|FA, \psi_c)$, models when the next contribution will arrive from one of the participants given the last floor action and additional evidence. We note that, since time t_c was reached without yet detecting that someone is speaking, the evidence ψ_c includes the time constraints that $T_u > t_e$, that $T'_u > t_c$, as well as additional evidence ϕ_c, e.g. audio-visual information, dialog context, etc. In this work we assume that given the floor release action FA, T_u only depends on the time constraints above, and is independent of the additional evidence ϕ_c, i.e.:

$$P(T_u|FA,\psi_c) = P(T_u|T_u > t_e, T'_u > t_c, FA)$$

If ID is a random variable that captures the input processing delay ($T'_u = T_u + ID$), this term can be computed by marginalizing over the delay, as follows:

$$P(T_u|FA,\psi_c) = \sum_d P(T_u, ID = d|T_u > t_e, T_u + ID > t_c, FA)$$

The term inside the sum can then be computed with Bayes' rule:

$$P(T_u, ID|T_u > t_e, T_u + ID > t_c, FA) \propto 1(T_u > t_e) \cdot 1(T_u + ID > t_c) \cdot P(T_u|FA) \cdot P(ID)$$

where $P(ID)$ models the input processing delay and $P(T_u|FA)$ models the prediction of when the next contribution will arrive from one of the participants, given the last floor release action. The other two terms integrate the time constraints.

The factorization described above is one approach to constructing a model of the conversational dynamics P_{cd}. Other alternatives can be envisioned. For instance, we are interested in directly learning the joint $P_{cd}(FA, T_u|\psi_c)$ from data, using structured discriminative learning techniques.

4.3 Modeling System Delays

To model the system's input processing delay $P_{id}(T'_u|T_u, \psi_c)$, we note that given T_u, T'_u depends on ψ_c only via the additional time constraint on $T'_u > t_c$:

$$P_{id}(T'_u|T_u, \psi_c) = P(T'_u|T_u, T'_u > t_c)$$

Given that $T'_u = T_u + ID$, we can write:

$$P_{id}(T'_u = t'_u|T_u = t_u, T'_u > t_c) = P(ID = t'_u - t_u|ID > t_c - t_u)$$

which yields via applying Bayesian reformulation:

$$P_{id}(T'_u = t'_u|T_u = t_u, T'_u > t_c) \propto 1(t'_u > t_c) \cdot P(ID = t'_u - t_u)$$

In effect, P_{id} can be directly computed from $P(ID)$.

The system output processing delay is captured via the distribution $P_{od}(T_s|t'_s)$. If OD is a random variable that captures the output delay ($T_s = t'_s + OD$), we have:

$$P_{od}(T_s = t_s|t'_s) = P(OD = t_s - t'_s)$$

and therefore P_{od} can be directly computed from $P(OD)$.

4.4 Modeling Cost

We now turn our attention to utility of outcomes. As previously discussed, the cost for each outcome depends on several contextual factors including the last floor action, the timings involved, etc.

To further investigate the cost structure, we conducted a small-scale cost assessment experiment [4]. 9 human annotators were asked to review videos of 9 interactions from the user study. Each annotator reviewed 3 interactions and each interaction was reviewed by 3 annotators. The annotators were asked to identify turn-taking errors committed by the system in each interaction and to assess the cost of each of these errors on a scale from 0 ("no error") to 5 ("worst error"). The authors aligned each turn-taking error identified by the judges with one of the turn-taking decisions made by the system and its corresponding outcome.

Based on this data, we fit two cost models: one for the case where the floor was released to the system, and one for the case where it was not. The dependent variable in these models is the cost, as assessed by the human judges. The independent variables are the time elapsed until the transition occurs and the transition type. We assumed a sigmoid parameterization for extrapolating the assessments to time points of outcomes that were unavailable for assessment. The resulting fitted models, together with the points based on which they were constructed, are displayed in Figure 3.

In general, the fitted cost functions align with our intuitions about costs in turn taking. For the case when the floor was released to the system, cost increases with time, and a transition to other is more costly than a transition to the system (it was the system's responsibility to take the floor). There is no significant difference in the fitted functions between a transition to other and a floor battle; we found this result somewhat counterintuitive, as we had assumed floor conflicts would generally add to the cost. When the floor was not released to the system, the cost is zero if another participant takes the floor, regardless of the timing. If the system takes the floor right away with no turn-initial overlap, the cost is positive. If however no one else talks and the system takes the floor successfully after a delay the cost drops towards zero as the delay increases, which aligns with the intuition that it would be okay for the system to take the floor if no one else takes it after a while. Finally, floor transition battles are more severe in this case, especially if they happen right away; the floor was not released to the system and a system intervention that leads to an overlap is penalized significantly.

We note that this analysis suffers to some degree from biases and sparsity in the data. For instance, due to the system's runtime heuristic policy, floor transitions to the system and floor battles tend to happen either immediately after the detected end of an utterance or about 3.5 seconds later. Transitions at points in between, which would help to better estimate the influence of time on cost, are not available and therefore the fitted models general-

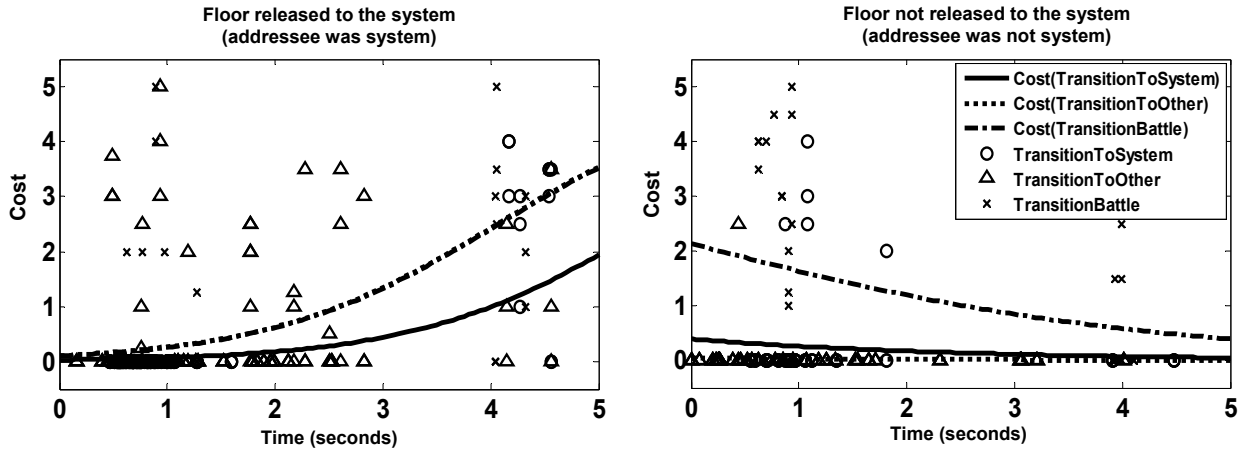

Figure 3. Estimating costs for actions over time from assessments.

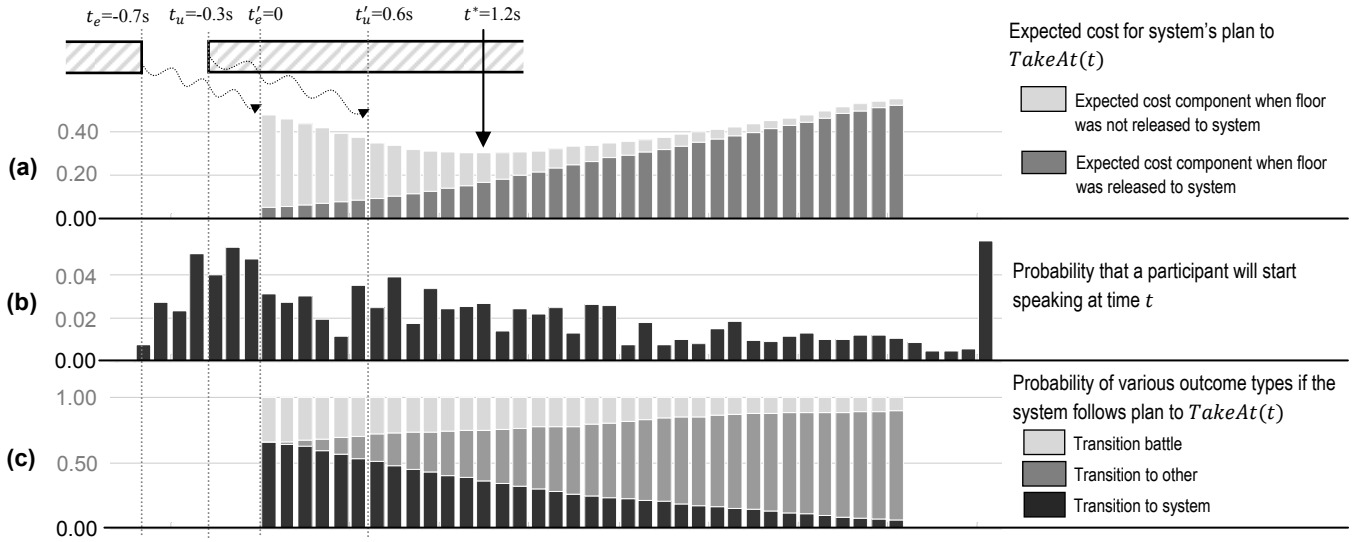

Figure 4. Decision-theoretic model avoiding floor conflict. (a) Expected cost for plan to $TakeAt(t)$, (b) probability distribution over time that the user will speak, and (c) inferred outcome-type probabilities for plan to $TakeAt(t)$.

ize from the available points and the sigmoid assumption. While we believe that this assumption is a reasonable first approximation (the costs are bound in the 0-5 interval and we assume monotonicity in the effect of time elapsed), data collected under a more exploratory policy would allow for fitting more accurate nonparametric models that make less assumptions about the overall structure of the cost function and effect of various outcomes.

We use the cost functions assessed above in the preliminary model assessment described in the next section. We note that the decision-theoretic approach can use any cost functions, either assessed through experimentation, or asserted by system engineers.

5. MODEL ASSESSMENT

We now illustrate with specific examples how the proposed model makes turn-taking decisions, and we investigate its behaviors based on a retrospective analysis of logs.

In the assessments described below, we assume t_{MAX}=3.5s. In addition, we used ϵ=0.3s, based on an analysis reported in [4]. The dataset was split into a train set containing 76 interactions and a test set containing 67 interactions. The distributions described below were learned or estimated from the training set, and the assessments were conducted on the test set.

We discretized the time axis into 100 millisecond intervals and modeled time as a multinomial variable over these intervals, with one last bin capturing times greater than t_{MAX}. $P(ID)$, $P(OD)$, and $P(T_u|FA)$ were therefore modeled as multinomial distributions and estimated from data. The latter distribution was estimated from cases where the system inferred that the floor was not released to it and decided to wait for 3.5 seconds or longer.

The model for inferring floor actions, $P(FA|\psi_c)$, was approximated by $P(FA|\psi_e)$, i.e. no incremental inference was performed for floor actions. This model was learned as a maximum entropy model from labeled data, based on a large set of multimodal features, including: *acoustic* features, e.g. average energy in the signal; *sound source localization* features, e.g. direction of the microphone array beam and its relative position to the actors; *spatial* features, e.g. location of actors; *visual focus of attention* features, e.g. for each actor a face detector reports whether a face is frontal, slightly turned left or right, or completely turned left or right; *understanding* features, e.g. whether the last utterance was a non-understanding, the confidence score, number of decoding alternates, etc.; *turn-taking* features, e.g. duration of current utterance, time since previous utterance, overlap with the system, etc.; *dialog* features, e.g. descriptors of the current dialog state. To account for structure in the temporally streaming signals (e.g. energy, beam location, visual focus-of-attention, etc.), we constructed features by computing relevant statistics of these streams in various windows relative to the current utterance. The trained model improves the accuracy of the floor action inferences significantly, attaining on test non-overlap utterances a classification error rate of 13% versus 20% error for the heuristic previously used by the system (described in Section 3.2.)

The first example we discuss, displayed in Figure 4, illustrates an instance from our collected data where the system made a poor decision to take the floor and entered into a floor battle. The system detected the end of a participant's utterance at time t'_e=0s (the actual end of the utterance was at t_e=-0.7s), and incorrectly inferred that the floor was being released to it. Based on its heuristic policy, the system took the floor and started speaking. However the participant's utterance was immediately followed by another utterance, which in fact begins at t_u=-0.3s, but is actually not detected by the system until t'_u=0.6s. The system therefore overlaps with the participant, inadvertently creating a floor conflict. The estimated cost for this outcome is 1.78.

The decision-theoretic policy would have avoided this conflict. The new floor action inference model is less confident that the floor was actually released to the system, i.e. $P(FA|\psi_0)$=0.6. At time t_0, the system computes the expected cost for the plans to take the floor at different time-points in the future, shown in Figure 4(a). Figure 4 also shows the inferred probability that a participant starts speaking at a time t, i.e. $P(T_u = t|\psi_0)$, and the probability of different outcome types if the system follows the plan to take the floor at a given time t. Since the time that minimizes the expected cost is t^*=1.2s, the system chooses to wait. After 100ms, at the next time tick, the system re-assesses the expected cost, but the minimizing cost is again found at 1.2s. On subsequent reassessments the optimal time shifts to t^*=1.5s, and the system keeps waiting, up until that time. The decision-theoretic system would have been able to detect the next utterance at t'_u=0.6, and would have avoided the floor conflict. The estimated cost would have been 0.04 (versus 1.78 incurred by the runtime model).

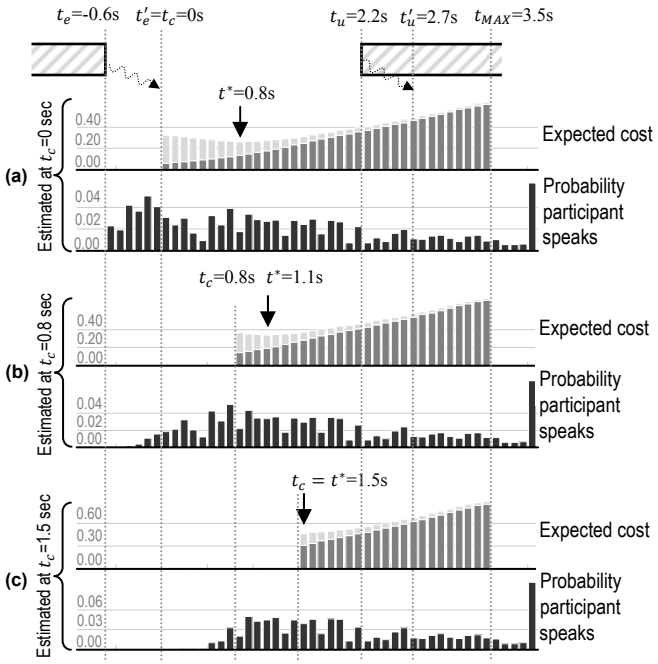

Figure 5. Decision-theoretic model avoiding undesirable gap.

The second example, shown in Figure 5, corresponds to a case where the floor was released to the system. At runtime the system inferred incorrectly that the floor was not released to it, and, according to its heuristic policy, decided to wait for 3.5 seconds prior to taking the floor. In this case, a participant intervened at t_u=2.2s, and the beginning of that utterance was detected by the system at t'_u=2.7s. The floor transitioned to the participant and the estimated cost for this outcome is 1.09.

With the new model, this situation could have also been averted. The learned predictive model infers that the probability that the floor was released to the system is $P(FA|\psi_e)$=0.76. The expected cost computation at time t_0 indicates that the optimal time to take the floor would be at t^*=0.8s. Hence the system would wait initially. As time goes by, the system re-computes the expected cost at each tick. For instance, consider the inferences at time t_c=0.8s as illustrated in Figure 5(b). At this point, the optimal time to take the floor has shifted in light of new evidence, and the expected cost computation indicates that the system should wait even longer, up until t^*=1.1s. Note that, because of time constraints (e.g. we haven't yet detected any participants' speech up to t_c=0.8s, i.e. T'_u>0.8s), the probability distribution over when a participant is expected to start speaking $P(T_u = t|\psi_8)$, has shifted accordingly. Once time t_c=1.5s is reached, the expected cost computation indicates that this is now the optimal time to take the floor, i.e. $t_c = t^*$ =1.5s, and the system does so. Under the new model the outcome would have been a transition to the system, with an estimated cost of 0.22 (versus 1.09 incurred by the runtime model).

The two illustrated cases highlight the abilities of the decision-theoretic model to integrate over inferred uncertainties to identify actions that minimize turn-initial overlaps and undesirable gaps in the conversation. Moving from our focus on methods to a comprehensive evaluation of the decision-theoretic variant of the system requires a second user study. While such an experiment remains for future work, we present a preliminary assessment for the proposed approach by investigating how some of the system's turn-taking decisions would have changed if the new model was used. As in the examples discussed above, we can revisit portions of the data observed at runtime and explore how the revised decisions compare to actions taken with the prior heuristic policy.

Table 1. Cost assessment for various decision models.

Model		Cost
Floor Release Inference	**Policy**	
Heuristic	Heuristic	0.43
Learned	Heuristic	0.29
Learned	Decision-theoretic	0.21

We cannot re-run in batch all situations observed in the original study as turn taking is an interactive process and outcomes hinge on user responses. However, sets of states are available for revisitation in a comparative analysis. Recall that three possible outcomes resulted from the system's floor decisions in the user study: transitions to others, transition battles, and transitions to the system. Because we know when the user intervened, in the first two situations we can actually revisit the data and assess *what would have happened had the system been using the new policy*. If the new policy leads the system to wait until after the point where the participant spoke at runtime, the new outcome would be a *transition to other*, and we can assess its cost. Similarly if the new policy decides to take the floor before the user did, we can also assess the new outcome—which will be *either a transition to the system or a floor transition battle*. In contrast, when the floor transitioned to the system during the user study (see Figure 2(b)), we can re-evaluate the situation only if the decision-theoretic model decides to take the floor at an earlier time than the original system did. If the new model decides to wait longer, we cannot know whether or not another participant would have intervened. Thus, in this comparative analysis, we focus only on the portion of the test set where we can fully assess the new policy, i.e. cases where in the collected data we had a transition to other, or a transition battle. We additionally remove cases where the run-time system employed special turn-taking strategies (one for rapidly re-acquiring the floor when the system was interrupted during the posing of a question, and the other where a non-verbal gesture was used to prompt for a response [3, 4]). The cases used in the assessment below account for 33% of all of the study data.

Running the heuristic policy that had been originally used on these cases instructs the system to take the floor immediately when the floor is released to it, or otherwise wait 3.5 seconds before trying to take the floor, giving someone else a chance to take it. As shown in Table 1, the mean cost associated with this heuristic policy is 0.43. In contrast, the decision-theoretic approach leads to outcomes with a mean cost of 0.21. An investigation of outcome details revealed that the largest reductions in cost were attained by *avoiding turn-initial overlaps* (i.e. avoiding floor conflicts), and *reducing response time* following floor releases to the system. The situations are represented by the cases discussed earlier and displayed in Figures 4 and 5. Although these findings are encouraging, we note that they address performance only on a specific subset of data as drawn from the prior user study. A comprehensive evaluation requires analysis of the performance of the revised system in a new user study.

As previously discussed, the decision-theoretic approach uses a learned model for predicting floor release actions $P(FA|\psi_e)$ that performs better than the original heuristic that had been used for inference. We also assessed the gains that can be attained by solely switching to this more accurate inference model, while keeping the same heuristic policy for actions: take the floor immediately if the floor is released to system or otherwise wait 3.5 seconds giv-

ing someone else a chance to take the floor. The resulting cost, shown in the third line of Table 1, is 0.29. This represents a significant reduction from the baseline, confirming the importance of accurate inferences. At the same time, the result is worse than when using the learned model *together with* the decision-theoretic policy for identifying ideal actions, which highlights the importance of reasoning about uncertainty, timings, and taking actions that minimize the expected costs of outcomes.

6. CONCLUSION

We presented a decision-theoretic approach for managing turn-taking actions in multiparty settings. The proposed methodology endows a dialog system with the ability to reason continuously about uncertainties in multiparty conversational dynamics, and about its own processing delays for perception and rendering. The model couples these uncertainties with assessments of potential outcomes and takes actions that minimize expected cost. While we focused on floor take versus wait decisions, similar machinery for inference and action can be harnessed for making decisions about when to release the floor.

Our main goal was to lay out key concepts with the time-dependent guidance of turn-taking actions in multiparty settings via computations of expected utility. As part of this effort, we have investigated the behavior of the proposed approach via a retrospective analysis of logs collected in a prior study. While new user studies are required for a comprehensive evaluation of the value of the approach, our preliminary comparative analysis highlights how the decision-theoretic methods can minimize turn-initial overlaps and reduce unnecessary gaps in the conversation.

We are interested in several directions for refining the performance of turn taking in multiparty settings. We believe that turn-taking decisions will benefit from further enhancements in the accuracy of inferences. As one direction in this realm, we are exploring the discriminative training of models for jointly and incrementally inferring the last floor action and start time of the next utterance as new evidence streams in. Inferential competencies can also be enhanced by developing means for better identifying the intentions and semantics associated with utterances. We are also interested in opportunities for extending the set of floor control actions available to a system. For example, we can add graded floor management strategies, such as a softer version of a floor take, where the system starts a contribution with a filler, e.g. *"So…* [pause] *What do you think?"*, constructing an opportunity to more gracefully back out of a potential floor conflict right after *"So…"*. We believe that taking a principled approach to turn taking in multiparty dialog systems will lead to more natural and flexible systems, with the ability to respond effectively to a wide range of dialog situations.

7. ACKNOWLEDGMENTS

We thank Isabelle Bouanna, Qin Cai, Ece Kamar, Zicheng Liu, Anne Loomis Thompson, and Cha Zhang for their contributions to this project.

8. REFERENCES

[1] Bell, L., Boye, J., and Gustafson, J., 2001. Real-time handling of fragmented utterances, In *Proceedings of NAACL-2001 workshop on Adaptation in Dialog Systems*.

[2] Bohus, D., and Horvitz, E., 2010. *Computational Models for Multiparty Turn Taking*, Microsoft Technical Report, MSR-TR-2010-115

[3] Bohus, D., and Horvitz, E., 2010. Facilitating Multiparty Dialog with Gaze, Gesture and Speech, In *Proceedings of ICMI'2010,* Beijing, China

[4] Bohus, D., and Horvitz, E., 2011. Multiparty Turn Taking in Situated Dialog: Study, Lessons, and Directions, In *Proceedings of SIGdial'2011*, Portland, OR.

[5] Duncan, S. 1972. *Some Signals and Rules for Taking Speaking Turns in Conversation*, Journal of Personality and Social Psychology 23, 283-292.

[6] Ferrer, L., Shriberg, E., and Stolcke, A. 2003. A Prosody-Based Approach to End-Of-Utterance Detection That Does Not Require Speech Recognition, In *Proceedings of ICASSP-2003*, 608-611, Hong Kong.

[7] Goodwin, C. 1980. *Restarts, pauses and the achievement of mutual gaze at turn-beginning*, Sociological Inquiry, 50(3-4), 272-302.

[8] Jonsdottir, G.R, Thorisson, K.R. and Nivel, E. 2008. Learning Smooth, Human-Like Turntaking in Realtime Dialogue, In *Proceedings of IVA-2008*, Tokyo, Japan.

[9] Raux, A. and Eskenazi, M., 2008. Optimizing end-pointing thresholds using dialogue features in a spoken dialogue system, In *Proceedings of SIGdial-2008*, Columbus, OH.

[10] Raux, A. and Eskenazi, M., 2009. A Finite-State Turn-Taking Model for Spoken Dialog Systems, In *Proceedings of HLT'09*, Boulder, CO.

[11] Sacks, H., Schegloff. E., and Jefferson, G. 1974. *A simplest systematics for the organization of turn-taking in conversation*, Language, 50, 696-735.

[12] Sato, R., and Higashinaka, R., 2002. Learning decision trees to determine turn-taking by spoken dialogue systems, In *Proceedings of ICSLP-2002*, 861-864, Denver, CO.

[13] Schegloff, E. 2000a. *Accounts of Conduct in Interaction: Interruption, Overlap and Turn-Taking,* The handbook of sociological theory, 287-321, New York: Plenum.

[14] Schegloff, E. 2000b. *Overlapping talk and the organization of turn-taking in conversation*, Language in Society, 29, 1-63.

[15] Schlangen, D., 2006. From reaction to prediction: Experiments with computational models of turn-taking, In *Proceedings of Interspeech 2006*, Panel on Prosody of Dialogue Acts and Turn-Taking, Pittsburgh, PA.

[16] Selfridge, E.O., and Heeman, P.A. 2010. Importance-Driven Turn-Bidding for Spoken Dialogue Systems, In *Proceedings of ACL-2010*, Uppsala, Sweden.

[17] Situated Interaction project web-page, 2011. http://research.microsoft.com/~dbohus/si.html

[18] Takeuchi, M., Kitaoka, N., Nakagawa, S. 2004. Timing detection for realtime dialog systems using prosodic and linguistic information, In *Proceedings of International Conference: Speech Prosody 2004*, 529-532.

[19] Thorisson, K.R. 2002. Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perceptions to Action, *Multimodality in Language and Speech Systems*, Kluwer Academic Publishers, 173-207.

[20] Traum, D., 1994. *A Computational Theory of Grounding in Natural Language Conversation*, TR-545, U. of Rochester.

[21] Traum, D., and Rickel, J., 2002. Embodied Agents for Multiparty Dialogue in Immersive Virtual World, In *Proceedings of AAMAS'02*, 766-773.

[22] Wiemann, J., and Knapp, M., 1975. *Turn-taking in conversation*, Journal of Communication, 25, 75-92.