Harnessing Models of Users' Goals to Mediate Clarification Dialog in Spoken Language Systems

Eric Horvitz   **Tim Paek**

One Microsoft Way
Microsoft Research
Redmond, WA 98052-6399
{horvitz,timpaek}@microsoft.com

Abstract. Speaker-independent speech recognition systems are being used with increasing frequency for command and control applications. To date, users of such systems must contend with their fragility to subtle changes in language usage and environmental acoustics. We describe work on coupling speech recognition systems with temporal probabilistic user models that provide inferences about the intentions associated with utterances. The methods can be employed to enhance the robustness of speech recognition by endowing systems with an ability to reason about the costs and benefits of action in a setting and to make decisions about the best action to take given uncertainty about the meaning behind acoustic signals. The methods have been implemented in the form of a dialog clarification module that can be integrated with legacy spoken language systems. We describe representation and inference procedures and present details on the operation of an implemented spoken command and control development environment called DeepListener.

1 Introduction

Science-fiction writers have long assumed that computer systems will one day have the ability to understand spoken input and to interact with people in a fluid, natural manner via conversational dialog. To date, most automatic speech recognition (ASR) research has centered on the refinement of acoustic and language models for identifying words from acoustic signals. Progress has been steady but incremental with this approach, with recognition error rates dropping on average by a small percentage each year.

We have pursued opportunities for making a qualitative leap forward in the robustness of ASR through the coupling of traditional low-level speech models with higher-level models of a user's intentions. The user models take into consideration the costs and benefits of taking different actions based on utterances, and the expected utility of seeking clarification about a user's intentions before taking action.

We focus in this paper on work to enhance the robustness of speech recognition for *command-and-control* systems. Spoken command-and-control systems have been growing in popularity as a means for allowing people to control a variety of telephony and software services in a handsfree manner. A number of applications attempt to use speech recognition for guiding services in challenging acoustical contexts, such as interaction via a telephone or standard built-in laptop microphones. The perceived quality of such systems is limited by the

error rate in interpreting spoken commands. We believe that advances in the robustness, flexibility, and naturalness of spoken command and control systems could make such applications of ASR more valuable and ubiquitous.

We have pursued endowing speech recognition systems with the ability to refine their understanding of acoustical signals through the process of engaging in human-oriented clarification dialog with the user. Clarification dialog centers on gathering evidence about the meaning of an utterance or to discriminate utterances from other sources of signal, such as ambient noise or overheard conversation. We harness, at the heart of the approach, a dynamic Bayesian network to perform inference about a user's intentions, and to integrate, in a cohesive manner, evidence from multiple acoustical signals over time gathered via clarification dialog. As we shall see, the Bayesian fusion of evidence over time, coupled with utility-directed dialog, allows an ASR system to refine the probability of a user's intention, by overlaying the information from adjacent responses.

We shall present a model for representing key probabilistic relationships among intentions and utterances, describe utility-directed procedures for making dialog and domain-level actions, and present details on the operation of an implemented spoken command and control development environment that has been packaged as a module that can be integrated with legacy spoken command and control systems.
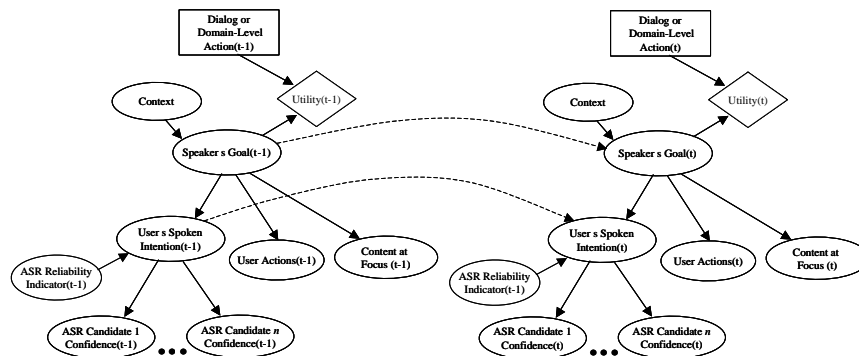
2   Intentions, Context, and Uncertainty

Fig. 1. A dynamic Bayesian network for reasoning about a user's goals given a sequence of utterances over time.

We shall focus on representations and inferential machinery for enhancing the robustness of speech recognition for controlling automated services. As an example service, the Lookout system [4] developed at Microsoft Research provides an automated calendaring and scheduling service. Lookout learns a user model that is used to guide if, when, and how the service will be executed, and uses these models to drive a mixed-initiative interaction with users. In one of the chief modes of operation, users interact with Lookout in a handsfree manner,

using voice commands to accept or reject offers of automated assistance, and to guide the calendaring activity.

We built our system on top of the command and control speech system provided with the Microsoft Agent (MS Agent) package. The MS Agent package is a freely available social agent development kit that provides a text-to-speech (TTS) generator and command and control ASR. MS Agent has been employed as the front end in a variety of software applications relying on speech recognition and TTS.

Our approach centers on taking the action with highest expected utility for the typical case where there is uncertainty about the intentions of the user. We include in the set of decisions under consideration, several dialog actions that seek clarification from the user by sharing the state of the system's confusion with the user in a natural manner.

We employ dynamic Bayesian networks to model the uncertainty in a user's goals and utterances over time. Dynamic Bayesian networks are probabilistic dependency models that include temporal dependencies among random variables at different times [8, 3, 10, 9]. Figure 1 displays two time slices of a larger iterative decision problem for inferring actions in response to output from a speech recognizer. Random variables are represented graphically as ovals with dependencies between variables as arcs. The dashed arcs indicate key temporal dependencies among variables in adjacent time slices.

Beyond reasoning about beliefs over time, we consider a decision problem at each time slice to identify dialog and domain-level actions in response to the user. As shown in the structure of the decision variable (square node) in Figure 1, the action associated with the largest expected utility is computed for each period, based on the inferred probability distribution over a user's goals.

To highlight the operation of the dynamic Bayesian network for increasing the robustness of speech recognition, let us consider the typical use of the ASR system for command and control. A listening session is initiated by a prompt from the TTS system inquiring if the user wants to activate a service in the current context. Following the initiation of a session, each recognition cycle, capturing the user's next turn in a dialog, is progressively represented by each time slice of a dynamic Bayesian network. At the end of each turn, the ASR system processes the acoustical signal and provides a set of recognized candidates.

At each step, a probability distribution over the goals of a user is inferred, based on observations about the user's activity, *e.g.*, events from the shell of a computer operating system or from a software application. Such computation of a user's goals based on activity and content has been a focus of work on probabilistic user modeling (*e.g.*, [5, 7, 1, 2, 6]).

In addition to considering evidence about a user's activity, we analyze a set of variables representing the output from a speaker-independent ASR system for command and control. For the case of the Lookout system—the first application domain of our work on clarification dialog—speech commands are used for a variety of tasks including accepting or rejecting offers of service. We shall focus on the simple case of clarification dialog for interpreting such acceptances and rejections of a service. We defined a set of candidate utterances, including a large set of responses capturing a range of ways people respond with to an offer of assistance with an acknowledgment (*e.g.*, "yes," "okay," "sure," "yeah," "go ahead," "right," "alright,""uh huh," etc.) or a rejection (*e.g.*, "no," "go away," "not now," "later," "no way," "get out of here," "nah," "nope," etc.). We also included candidates that represent utterances associated with a user's potential deliberation about the desirability of initiating a service (*e.g.*, "um," "hmmm," "let's see," "uh," etc.). Finally, we considered the cases where the

ASR system detects an unrecognized sound, or where nothing is heard. The set of variables represented in the model were abstracted as classes of response, including *affirmation*, *negation*, *reflection*, *no signal*, and *unrecognized signal*.

Additional variables summarizing the status of a cycle of speech recognition can provide context-sensitive information about the overall proficiency of speech recognition. The variable labeled *ASR reliability indicator* in Figure 1 represents such a variable. We found it useful to consider the special case where the speech recognition analysis identifies a concatenation of two or more commands in a single turn (*e.g.*, "yes..uh..no, no"). In such cases, we analyze the last term recognized by the system. As indicated in the Bayesian network model, we condition the user's spoken intention on the status of such a reliability indicator.

In response to a user's utterances, the speech recognition system provides a list of recognized candidates and a measure of confidence for each candidate. This measure of confidence is based on a common measure of uncertainty that is computed in many low-level speech models, referred to as *perplexity*. As represented in Figure 1, the user's spoken intention influences the classes of response and their confidences. We allow dialog designers to assess in the development environment prior conditional probabilities representing the likelihood of hearing different classes of response with specific values of confidence, given the user's actual utterance, referred to as the user's *spoken intention*.

3   Utility-Directed Decisions about Dialog and Action

At run-time, the low-level speech recognition system is called, the utterance is evaluated, the reliability variable is set, and a list of candidates and confidences for the current time slice are communicated to the dynamic Bayesian network. A probability distribution is inferred over the classes of response represented by the processed acoustic signal and the local decision with maximum expected utility is identified. In the current system, base-level actions include:

– Execute the service being offered

– Ask the user to repeat the spoken intention

– Note the hearing of a noise and inquire

– Try to acquire the user's attention

– Confirm the user's intention to bypass the offer of service

We constructed a utility model by assessing a space of outcomes defined as the cross product of the situations under consideration and these actions. In our prototype for Lookout, we assessed utilities, $u(A_i, S_j)$, for the twenty-five outcomes, constructed from the five actions A and five situations S.

4   Global Analysis of Conversational Progress

Beyond considering the best next action at each time slice, we worked to overlay a more global consideration of *conversational progress*. Such consideration involves monitoring the overall nature of the efficiency of conversation by considering the number of turns and the convergence versus nonconvergence of the dialog system on a user's intentions over time. The function of this service is

to notice when things are not going well, and to consider the utility of troubleshooting the overall conversation. Conversational troubleshooting makes use of representations of the growing frustration of users with increasing numbers of steps of clarification dialog. Just as a person who is hard of hearing might stop and mention that things are just not going well in the conversation and suggest an alternate strategy, the system considers stepping outside of the stepwise dialog to engage in troubleshooting, including such behaviors as describing the problem, reflecting about what was heard so far, and then trying again.

We explored several representations for integrating such capabilities into DeepListener, including the embedding of distinctions about progress into each time slice and the overlaying of a monitoring system at a higher *conversational level* of analysis. The current implemented version of DeepListener bases its decisions to troubleshoot on a model of frustration that grows with the number of explicit steps of interaction. Explicit turns include all interactions except for steps where the user is likely pausing to reflect about an intention or where the user is likely not attending to the dialog. The development environment allows users or system developers to encode preferences about the value of engaging in troubleshooting at each step as a function of the number of steps and the dynamics of beliefs.

5 DeepListener Implementation

We built and tested several implementations of DeepListener. At the heart of DeepListener is a dynamic Bayesian model constructed with the MSBNX Bayesian modeling tool developed at Microsoft Research. Run-time components for dynamic Bayesian inference and decision making and MS Agent components were meshed to create a run-time system and a set of development tools. The tools include controls that enable developers or users to assess utilities of outcomes. The development environment also provides a programmatic interface and methods for defining the relationship between an external user model used by the application being extended with a spoken dialog system. The interface includes a graphical representation of the dialog system's belief about the user's goals as a function of beliefs inferred in an external user model.

To complement the dialog decision making machinery, a set of appropriate utterances and gestures for the animated agent were designed for each DeepListener action. To enhance the naturalness of interaction, gestures and utterances are drawn randomly at runtime from a pool of candidate gestures for each action.

Figure 2 shows the user interface provided by DeepListener to developers for coupling the system with legacy applications, viewing inferences, and testing interactive behaviors. The sequence of utterances and their confidences are displayed in an utterance-trace window at the upper left of the display. In this case, the system heard a "yeah," followed by a "yes," followed by the name of the user, all heard with low confidence. The best action here is for DeepListener to share it's belief that it is overhearing sounds that were not intended for the system. This action involves relaying the system's reflection via a thought cloud over the animated agent. At the lower right-hand corner of the interface, feedback on the system's listening and reasoning is provided in the form of a colored lens. A green glow indicates that the system is listening for a user response, yellow indicates that the system is confused, and red indicates that the system is peforming recognition or inference and, thus, is temporarily unavailable for listening. Such feedback on listening and reasoning status provides information analogous to the communication backchannel of expressions and gestures about understanding that listeners may relay to speakers in natural conversations.
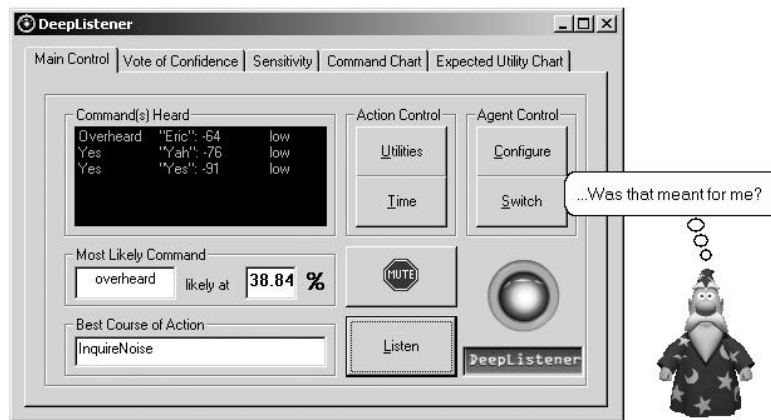
Fig. 2. Main screen of DeepListener system. The system provides tools for assessing utilities, displaying inferences over time, and interfaces for coupling DeepListener's beliefs to the beliefs of an external user modeling system.

Figure 3 shows DeepListener's preference assessment tools. The utility assessment tool on the right allows users to order their preferences about basic outcomes and to assess utilities using relative distance by moving sliders between the best and worst utilities. The figure displays outcomes for the case in which a user desires a service offered by the agent. The tool on the right demonstrates the facility for assessing a function that yields the utility of troubleshooting as a function of the number steps in a conversation without convergence on understanding.

6 Real-Time Interaction with DeepListener

At run time, DeepListener executes dialog and real-world actions at each turn in a manner consistent with its current uncertainty about the observed sequence of utterances and the expected consequences of alternative actions.

In many situations, DeepListener hears well enough to simply perform the intended action within a single step. In such scenarios, the system performs just as a simple speech recognition would in response to an accurate recognition. However, the system's understanding of an intention may not be good enough to invoke an action in the world. The top portion of Figure 4 displays traces of the beliefs about a user's intentions, and the expected utility of alternate actions over time, for an interactive session where the user was in a relatively noisy environment (background noise was generated by HVAC and equipment fans) and sat approximately four feet from a standard built-in microphone embedded in a laptop computer. An automated service was offered per an inference about user goals, yielding prior probabilities on alternate intentions. The user uttered "ummm..." in the process of reflecting about the desire for the service, following by "okay." The system heard this response with a low confidence and requested a clarification, at which point the user said, "yes." Although this response was also heard with low confidence, the update to beliefs about the user's desire for the
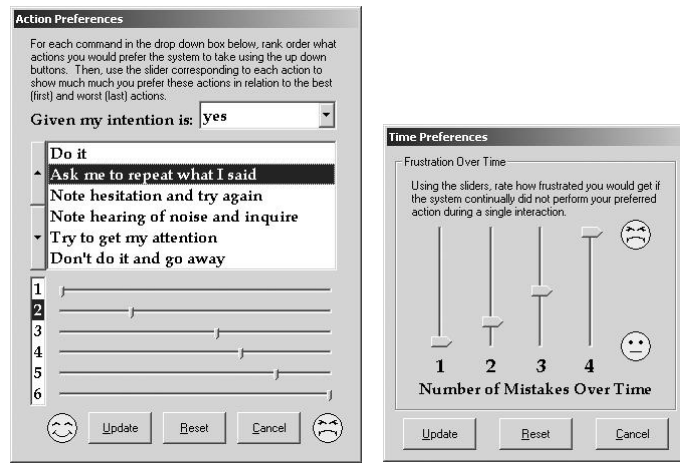
Fig. 3. Views of DeepListener's preference assessment tools. Left: Interface for assessing stepwise outcomes, showing outcomes for the case where the user desires a service. Right: Tool for assessing a troubleshooting function representing the utility of troubleshooting with increasing numbers of steps in the dialog.

service led to an invocation of the service, given a consideration of the expected utilities of all actions. This example highlights how utility-directed clarification dialog can fuse evidence across time to converge on an intended action. The lower portion of Figure 4 shows the behavior of DeepListener at steps 2 and 3 of this case.

Overall, the experience of communicating with a DeepListener-enabled application differs qualitatively from interacting with traditional speech command systems. In distinction to traditional ASR, DeepListener understands when it should gather additional information before taking (potentially erroneous) actions or simply expressing a failure to understand. For example, interacting with the system in noisy environments—or at a relatively long distance away from a microphone—invokes the impression of attempting to communicate with a person who is having difficulty hearing but desires to work with the speaker to clarify the utterance. Also, users can assume, as they do with people they are interacting with, that there is a shared memory of the recent communication experience. Thus, during a cycle of clarification, a user can utter different words to describe an intention and expect the system to have considered the history of interaction, rather than assuming that each clarification is starting anew.

Figures 5 and 6 display inference over sample sessions that led to troubleshooting. For the case presented in Figure 5, the utility of troubleshooting eventually dominated other actions, following a sequence of poor recognitions and detection of background noise. The system shared a summary of its analysis and recommended that the user adjust the microphone and speak with greater articulation. For the session presented in Figure 6, the system heard low confidence utterances (*e.g.*, a muffled "yeah") signaling an intention to go ahead with a service, interspersed with signs of reflection (*e.g.*, "hmmm, let's see,"). After three steps of clarification dialog, the system initiated conversational trou-
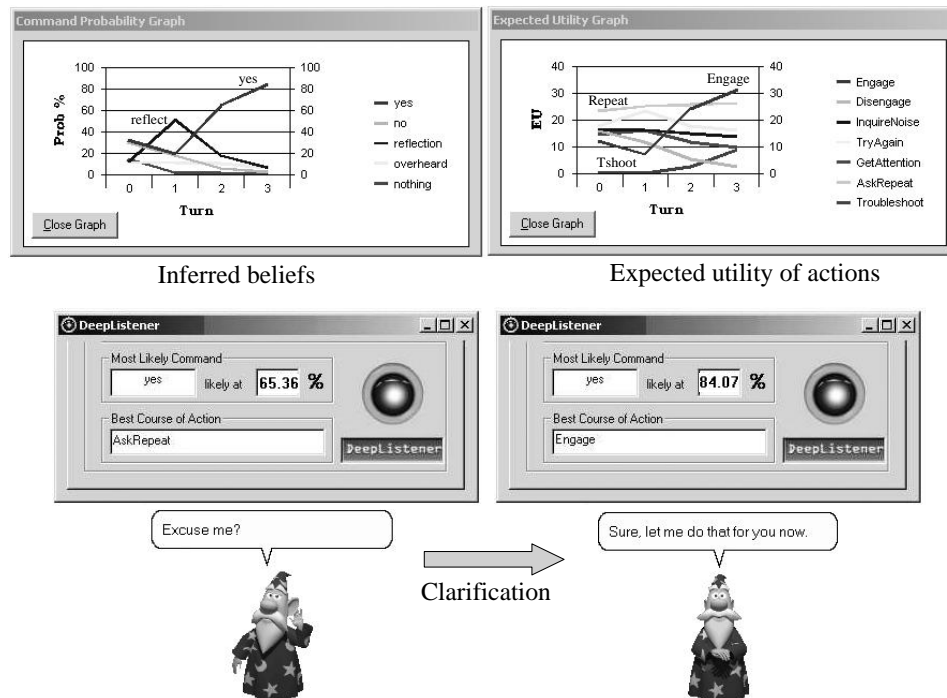
Fig. 4. Top Left: Beliefs about a user's intentions over a three-step interaction in a noisy environment. Top Right: Expected utilities inferred over the interaction, converging on an invocation of service. Bottom: Behavior of DeepListener at steps 2 and 3 of this session.

bleshooting, relaying to the user its sequence of beliefs about the user's intentions over time, and giving the user a tip about how to best communicate the most likely intention.

We are currently pursuing methodical tests of DeepListener in different kinds of environments to see how much better the system performs over a naive, spoken command and control system. We are also investigating the use of machine learning to learn key probabilistic dependencies from data and the extension of the system to consider additional sensory information about a user's context and attention. Some of this work is being pursued in the related Quartet project, focused on developing a multilevel conversational architecture [11].

7 Summary

We presented methods for extending traditional speech recognition systems with representations and inference machinery that consider the intentions associated with one or more utterances, and that continue to weigh the costs and benefits of alternate behaviors. The approach centers on the use of temporal probabilistic inference and expected utility to guide clarification dialog and domain-
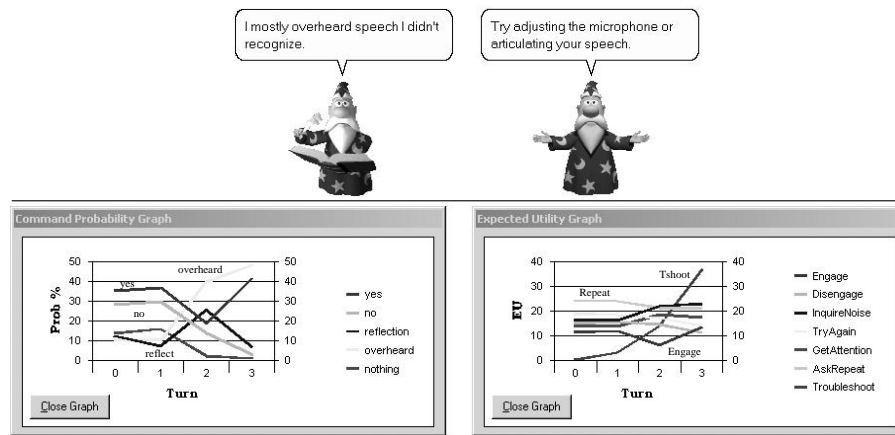
Fig. 5. Troubleshooting of the overall conversation following a sequence of poor recognitions and detection of background noise.

level actions. We described DeepListener, a clarification-dialog development environment that can be used to construct run-time spoken command and control systems. DeepListener employs utility-directed clarification dialog to acquire a growing sequence of utterances that is subjected to Bayesian analysis at each turn in a dialog. We reviewed the use of DeepListener to extend the robustness of spoken command and control for guiding the provision of automated services. We believe that the methods embodied in DeepListener show promise for enhancing the performance of spoken language systems in a variety of real-world situations and environments. We believe that the use of utility-directed clarification dialog transforms in a qualitative manner the overall experience of interacting with a spoken language system. Our ongoing work on DeepListener centers on quantitatively assessing the value of the approach in real-world settings, and on extensions to both the overall architecture and the methodology for constructing user models for clarification dialog.

Acknowledgments

References

1. D.W. Albrecht, I. Zukerman, A.E. Nicholson, and A. Bud. Towards a bayesian model for keyhole plan recognition in large domains. In *Proceedings of the Sixth International Conference on User Modeling, Sardinia, Italy*, pages 365–376. User Modeling, Springer-Verlag, June 1997.
2. C. Conati, A.S. Gertner, K. VanLehn, and M.J. Druzdzel. Online student modeling for coached problem solving using bayesian networks. In *Proceedings of the Sixth*
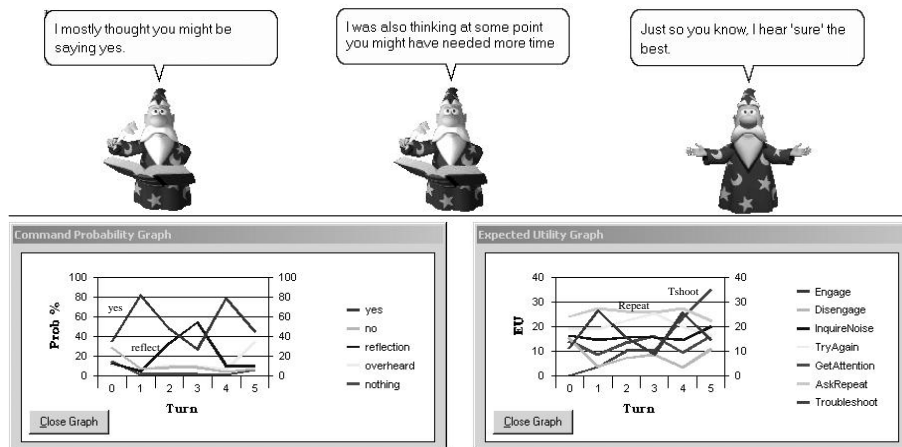
Fig. 6. Troubleshooting of conversation following clarification dialog demonstrating communication of a summary of DeepListener's time-varying beliefs about a user's intentions.

    International Conference on UserModeling, Sardinia, Italy, pages 231–242. User Modeling, Springer-Verlag, June 1997.

3. P. Dagum, A. Galper, and E. Horvitz. Dynamic network models for forecasting. In *Proceedings of the Eighth Workshop on Uncertainty in Artificial Intelligence*, pages 41–48, Stanford, CA, July 1992. Association for Uncertainty in Artificial Intelligence.

4. E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*, pages 159–166. ACM Press, May 1999.

5. E. Horvitz and M. Barry. Display of information for time-critical decision making. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 296–305, Montreal, Canada, August 1995. Morgan Kaufmann, San Francisco.

6. E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265. Morgan Kaufmann, San Francisco, July 1998.

7. A. Jameson. Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction*, 5:193–251, 1996.

8. K. Kanazawa and T. Dean. A model for projection and action. In *Proceedings of the Eleventh IJCAI*. AAAI/International Joint Conferences on Artificial Intelligence, August 1989.

9. K. Kanazawa, D. Koller, and S. Russell. Stochastic simulation algorithm for dynamic probabilistic networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, pages 346–351, Montreal, Quebec, Canada, 1995.

10. A.E. Nicholson and J.M. Brady. Dynamic belief networks for discrete monitoring. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(11):1593–1610, 1994.

11. T. Paek and E. Horvitz. Conversation as action under uncertainty. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 445–464. AUAI, Morgan Kaufmann, August 2000.

This article was processed using the LaTeX macro package with LLNCS style