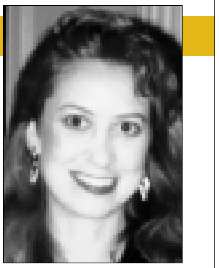


Mixed-initiative interaction

By Marti A. Hearst
University of California, Berkeley
hearst@sims.berkeley.edu

In the last few years, a series of well-publicized debates argued the merits of total automation of user needs (via intelligent agents) versus the importance of user control and decision making (via graphical user interfaces).[1]

Perhaps a more productive way to frame this discussion is to note that there is an interesting duality between AI and human-computer interaction. In AI, we try to model the way a human thinks in order to create a computer system that can perform intelligent actions. In HCI, we design computer interfaces that leverage off a human user to aid the user in the execution of intelligent actions.

What is the boundary between these two fields? An area that is becoming known as *mixed-initiative interaction* might turn out to be the missing link. Mixed-initiative interaction refers to a flexible interaction strategy in which each agent (human or computer) contributes what it is best suited at the most appropriate time. I became interested in this area when I read about AIDE, a system that helps a user explore a dataset using a statistics software package.[2] AIDE both makes suggestions to the user and responds to user guidance about what to do next.

In this installment of “Trends and Controversies,” we have three essays about the area of mixed-initiative interaction. James Allen of the University of Rochester introduces the area and creates a useful taxonomy of mixed-initiative dialog issues. He also summarizes several years’ worth of research on mixed-initiative planning systems. The second essay, by Eric Horvitz of Microsoft Research, describes the role of uncertainty in mixed-initiative interaction and describes two innovative systems for semiautomated assistance that make use of Bayesian reasoning. Finally, Curry Guinn of Duke University confronts the difficult task of evaluating such systems, including the creation of test sets and metrics for evaluating descriptive versus prescriptive dialog models. In earlier work, Guinn has developed extensive computer simulations of mixed-initiative dialogs.

—Marti Hearst

References

1. B. Shneiderman and P. Maes, “Direct Manipulation vs. Interface Agents,” *Interactions*, Vol. 4, No. 5, 1997, pp. 42–61.
2. R. St. Amant and P.R. Cohen, “Interaction with a Mixed-Initiative System for Exploratory Data Analysis,” to be published in *Knowledge-Based Systems,* Vol. 10, No. 5, 1999.

Mixed-initiative interaction

James F. Allen, University of Rochester

Mixed-initiative interaction is a key aspect of effective human-computer interaction and has great potential to affect work on multiagent systems. The term *mixed-initiative interaction* is sometimes conflated with human-computer interaction itself, but this is a mistake because almost all models of HCI so far are not mixed-initiative, and mixed-initiative systems need not involve a human. It is perhaps in HCI where we will see the greatest impact. The development of mixed-initiative intelligent systems will ultimately revolutionize the world of computing even more than the recent move to GUIs, in my view. This essay describes the goals of research in mixed-initiative interaction, suggests a general framework for thinking about work in the area based on the properties of human dialogue, and then briefly describes the key problems to overcome before mixed-initiative systems become a reality.

For simplicity, I will focus on a single scenario consisting of two agents: a human and an intelligent system. Mixed-initiative interaction can occur in many other scenarios as well, including between multiple machines cooperating to perform tasks (such as in distributed planning) or between multiple people and machines interacting to coordinate their activities (collaboration systems, for example). Most everything I say here generalizes to these other cases. In fact, many of the issues become even more crucial as the number of agents grows. In many examples, I will draw from our experience in building mixed-initiative planning systems over the last five years.[1–3.]

The goals of mixed-initiative interaction

Mixed-initiative interaction is an important aspect of effective multiagent collaboration to solve problems or perform tasks. In our minimal human-computer configuration, such tasks could include systems designed to interact with a user to design a kitchen, find the best airfare, coordinate an emergency relief mission, or teach the user how to use new equipment. *Mixed-initiative* refers to a flexible interaction strategy, where each agent can contribute to the task what it does best. Furthermore, in the most general cases, the agents’ roles are not determined in advance, but opportunistically negotiated between them as the problem is being solved. At any one time, one agent might have the initiative—controlling the interaction—while the other works to assist it, contributing to the interaction as required. At other times, the roles are reversed, and at other times again the agents might be working independently, assisting each other only when specifically asked. The agents dynamically adapt their interaction style to best address the problem at hand.

Mixed-initiative interaction lets agents work most effectively as a team—that’s the key. The secret is to let the agents who currently know best how to proceed coordinate the other agents. Involving a human in the interaction adds the complication that the system agents must use an interaction mode convenient to the human and support human-style problem solving. To do this, computer agents must be able to focus on different key subproblems, collaborate to find solutions—filling in details and identifying problem areas—and work with the person to resolve problems as they arise.

Current approaches to human-computer interaction with intelligent systems are typically not mixed-initiative. Rather, they fall into two approaches, illustrated by the following two examples:

- *human control: a scheduling workstation*—The system is a tool for scheduling vehicles for transporting freight. It provides a GUI for accessing a set of software tools for developing, manipulating, evaluating, and displaying plans, invoking simulators, and so on. The system responds to commands from the human, who specifies the plan using the tools available. If well-designed, such a system would be a useful planning tool, but it is not a mixed-initiative system, because the human always controls the interaction.
- *system control: automated call centers*—One of the fastest-growing applications of human-machine interfaces are automated call centers, in which the human uses a telephone keypad or speech to make a series of menu selections (for example, "if you want your account balance, press or say one…."). This is a prototypical example of system-controlled interaction, and our almost universal sense of frustration with such interfaces indicates how annoying they are to the human user. Such systems are becoming prevalent because they save companies money, not because they improve customer service.

Mixed-initiative interaction within a dialogue framework

The best way to view interaction between agents is as a *dialogue*, and thus mixed-initiative becomes a key property of effective dialogue. We all have intuitions about how human dialogue works, which can be exploited in developing new models of interaction. While natural-language interaction is the typical form of human dialogue, dialogue models can be characterized in terms of any communication protocol and are independent of natural language. People even engage in dialogues in other modalities, using gestures, drawing, and the like. A computer offers yet further modes of communication, graphics-based user interfaces, menu-base systems, and so on, used both for system output and point-and-click input. By dialogue, I refer to specific mechanisms such as contextual interpretation, turn taking, and grounding, that would be needed with any communication modality.

Turn-taking: When should or can I speak? When a particular agent communicates in a dialogue and the others listen, that agent has the turn. Turn-taking models address questions of when an agent is obliged to take the turn, when it cannot have the turn, and when it has an option of taking the turn or not. In fixed-initiative systems, turn taking is usually not an issue, because one agent initiates all interactions and waits for the appropriate response before moving on to the next interaction. Who has the turn is always well-defined, and the agent in control always initiates turns.

In mixed-initiative interaction, the situation can be more complex. Because different agents might take the initiative at different times, an agent must be able to tell when it should appropriately start an interaction by taking the turn. For example, in a plan-management system, the system might learn of a new problem that interferes with the current plan built so far. Assume that the user is currently asking about the weather. The system must decide whether to

- interrupt the user with a notification of the problem;
- wait for the user to finish but then ignore the question and notify the problem;
- answer the question and then state the problem; or
- wait until later (say, until the user asks if there are any problems).

The right decision here requires balancing the importance of the problem, the status of the weather, and other "social" constraints between the two agents.

Table 1. Different levels of mixed initiative

Mixed-initiative levels	Capabilities
Unsolicited reporting	Agent may notify others of critical information as it arises
Subdialogue initiation	Agent may initiate subdialogues to clarify, correct, and so on
Fixed subtask initiative	Agent takes initiative to solve predefined subtasks
Negotiated mixed initiative	Agents coordinate and negotiate with other agents to determine initiative

Different levels of mixed initiative. Single-initiative systems specify in advance which agent has the initiative in the interaction, while mixed-initiative systems offer a range of options and levels of complexity (see Table 1). Consider starting with an interactive-planning application in which the user has the initiative. The first step toward mixed-initiative is to allow *unsolicited reporting*. For example, say the system continually verifies whether the plan under development is likely to succeed in the current situation. As the situation or plan changes, the system might notice problems and then notify the user. At this basic level, however, the system does not then coordinate the subsequent interaction.

The next level involves *subdialogue initiation*. In this case, the system might initiate a subdialogue in certain situations, say, to ask a clarification. Because the user should then answer the question, and the clarification might take several interactions, the system has temporarily taken the initiative until the issue is clarified. Initiative then reverts to the user.

At the *fixed-subtask mixed-initiative* level, the system responsibility to perform certain operations. For instance, say the system is responsible for choosing particular vehicles, routes, and refueling stops for each planned transportation action. When the user suggests a transportation goal, the system takes over building the plan, asking the user to make decisions when it needs help. As long as the system is working on this aspect of the plan, it is maintaining the initiative. Once the subtask is completed, initiative reverts to the user.

At the final *negotiated mixed-initiative* level, there is no fixed assignment of responsibilities or initiative. Each agent constantly monitors the current task and evaluates whether it should take the initiative in the interaction, basing this decision on many factors, including

- the agent's capability to effective coordinate the current subtask (Can I coordinate the interaction to solve this problem?),
- the other demands on the agent at the present time (Do I have the time and

Table 2. What happens in a mixed-initiative collaborative planning between humans.

Action	Amount (%)
Evaluating and comparing options	25
Suggesting courses of action	23
Clarifying and establishing state	13.5
Clarifying or confirming the communication	13.5
Discussing problem-solving strategy	10
Summarizing courses of action	8
Identifying problems and alternatives	7

resources to do it?), and

- the other agents' evaluations of their own capability to coordinate the interaction at the present time (Am I the best qualified to coordinate given my current collaborators?).

Intention recognition. Another key technical problem that arises in mixed-initiative settings is intention recognition. When the system wholly controls the interaction, it can determine the allowable responses each time and decide how to interpret them. When the user has control, however, the system must identify the goals underlying the user's request to respond appropriately. In the simplest of tasks, simple techniques might suffice and be built into the interpretation strategy. With more complex tasks, however, the system might need to recognize some or all of the following from a user's contribution:

- What speech act the user is performing (for example, is this a request, a promise, or acceptance of a proposal?),
- What level the user is concerned with (are they talking about the interaction or about how to perform the task, or are they performing part of the task?), and
- What action they are trying to accomplish (what modification to the interaction, how is the task being modified, or what part of the task is the user performing?).

Mixed-initiative planning systems

At the University of Rochester, we have been working for more than five years on mixed-initiative planning systems. The goal is to develop systems that can enhance human performance in managing plans, for example, to maintain a transportation network or to coordinate emergency relief in response to natural disasters. At first, we had high hopes of harnessing the long history of work in AI on planning systems[4] to build effective planning tools. This hope disappeared rapidly, however, as we understood that there was a serious mismatch between the way planning systems solve problems and the way humans solve problems.

Automated planners require complete specifications of the goals and situation before starting to work, while people incrementally learn about the scenario and refine and modify their goals as the plan is being developed. Automated planners evaluate plans quantitatively and in black-and-white terms, while humans subjectively evaluate plans. Automated planners work on one solution at a time, whereas people compare options and alternatives before selecting a course of action.

Faced with this dilemma, we decided to try to design collaborative planning systems in which the user and machine collaborate to build plans, each providing the capabilities it does best. The human brought intuition, a notion of the goals and trade-offs between goals, and highly developed problem-solving strategies, while the machine brought an ability to manage detail, allocate and schedule resources, and perform quantitative analysis of proposed courses of action.

To determine how the system should interact with the human, we studied how humans interact with each other. We collected dialogues between two people interacting in a transportation-planning scenario, to see what types of interactions they used when they collaboratively planned. One person received information about the domain—the capabilities of trains, for example—while the other started with the goals to achieve and was ultimately responsible for the plan produced. The two could not see each other and did not know each other. The only information they shared at the start of the dialogue was an abstract map of the Trains world.

Table 2 summarizes the different classes of interaction and their frequency that we found in analyzing every utterance in one hour of a sample dialogue. The only type of interactions supported by a typical state-of-the-art planning system (namely, adding a new course of action) handles less than 25% of the interactions. Much of the interaction was concerned with maintaining the communication (summarizing and clarifying, for example) or managing the collaboration (discussing the problem solving strategy). Clearly, an effective collaborative planner required much more that traditional planning technology.

Faced with this analysis, we have focussed our research on several key areas. First, to provide the human with a convenient communication language, we have developed a spoken natural-language dialogue interface. While a complex task in itself, it did not seem feasible to develop a flexible enough communication language that would not have required too much training on the human's part. Second, we developed a more general model of plan reasoning and management, focusing on incremental development of plans, plan recognition, ways of managing and comparing different options, and ways of effectively communicating the structure and implications of proposed plans. (See our Web site at www.cs.rochester/research/trains for details.)

Summary

Research in mixed-initiative interaction is still in its infancy, and the research problems we will face are significant. The potential impact of such systems, however, cannot be overestimated. If we are ever to build computer systems that can seamlessly interact with humans as they perform complex tasks, these systems will need to support effective mixed-initiative interaction.

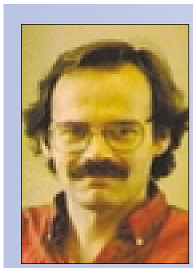
References

1. J. Allen et al., "A Robust System for Natural Spoken Dialog," *Proc. 31st Meeting ACL,* MIT Press, Cambridge, Mass., 1996, pp. 62–70.

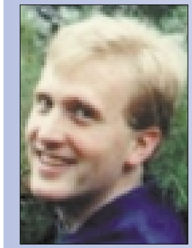
2. G. Ferguson, J. Allen, and B. Miller, "TRAINS-95: Towards a Mixed-Initiative Planning Assistant," *Proc. Third Conf. AI Planning Systems (AIPS-96),* AAAI Press, Menlo Park, Calif., 1996, pp. 70–77

3. G. Ferguson and J. Allen, "TRIPS: An Integrated Intelligent Problem-Solving Assistant," *Proc. Nat'l Conf. AI (AAAI-98),* AAAI Press, Menlo Park, Calif., 1998.

4. J. Allen, J. Hendler, and A. Tate, *Readings in Planning*, Morgan Kaufmann, San Francisco, 1990.



James F. Allen is a professor of computer science at the University of Rochester. His research interests include natural-language understanding, discourse, knowledge representation, common sense reasoning, and planning. He earned a PhD in computer science from the University of Toronto. He was one of the first winners of the Presidential Young Investigator Award, and is a Fellow of the American Association of Artificial Intelligence. Contact him at the Univ. of Rochester, Computer Science Dept., Rochester, NY 14627; james@ling.rochester.edu; http://www.cs.rochester.edu/u/james.



Curry I. Guinn is a research engineer at the Research Triangle Institute's Center for Digital Systems Engineering and an adjunct assistant professor at Duke University's Department of Computer Science. His research interests include human-computer collaboration, multimedia tools for education, text abstraction, argumentation theory, expert systems, and semantic networks. He earned a BS in computer science and philosophy from Virginia Polytechnic Institute, and an MS and PhD in computer science from Duke University. He is a member of the ACM, AAAI, and ACL. Contact him at the Research Triangle Inst., 3040 Cornwallis Rd., Research Triangle Park, NC 27709-2194; http://www.cs.duke.edu/~cig.

Eric Horvitz is a senior researcher and manager of the Adaptive Systems and Interaction Group at Microsoft Research. His research interests center on principles and applications of reasoning, learning, and action under uncertainty. He received a PhD and MD from Stanford University. He serves on the Executive Council of the American Association for Artificial Intelligence and the board of the Association for Uncertainty and Artificial Intelligence and is the editor of the Decisions, Uncertainty, and Computation area of the *Journal of the ACM*. Contact him at Microsoft Research, Redmond, WA 98052; horvitz@microsoft.com; http://research.microsoft.com/~horvitz.

Uncertainty, action, and interaction: in pursuit of mixed-initiative computing

Eric Horvitz, Microsoft Research

Recent debate has highlighted differing views on the most promising opportunities for user-interface innovation.[1] One group of investigators has expressed optimism about the potential for refining intelligent-interface agents, suggesting that research should focus on developing more powerful representations and inferential machinery for sensing a user's activity and taking automated actions.[2–4] Other researchers have voiced concerns that efforts focused on automation might be better expended on tools and metaphors that enhance the abilities of users to directly manipulate and inspect objects and information.[5] Rather than advocating one approach over the other, a creative integration of direct manipulation and automated services could provide fundamentally new kinds of user experiences, characterized by deeper, more natural collaborations between users and computers. In particular, there are rich opportunities for interweaving direct control and automation to create *mixed-initiative* systems and interfaces.

Computer scientists have used the term *mixed-initiative* in various ways. These include references to the automated control of turn taking in human-computer conversation[6] and the coordinated application of a set of problem-solving methodologies.[7] I shall use the phrase to refer broadly to methods that explicitly support an efficient, natural interleaving of contributions by users and automated services aimed at converging on solutions to problems.[8]

Taking a mixed-initiative approach promises to dramatically enhance human-computer interaction by allowing computers to behave more like associates, capable of working with users to develop a shared understanding of goals and of contributing to problem solving in the most appropriate way. Achieving such a dream of fluid collaboration between users and computers requires solving several difficult challenges. In particular, we need to develop machinery for gathering information and making inferences about the intentions, attention, and competencies of users—and for ultimately making decisions about the nature and timing of automated services. Computers will often be uncertain about the goals and needs of users. Thus, methods for reasoning under uncertainty play a critical role in mixed-initiative interaction.

Supporting joint activity under uncertainty

People appear to be well adapted to mixed-initiative problem solving. In daily life, we continue to engage one another in efficient, tightly woven collaborations. We assume and rely on a rich interleaving of efforts to achieve goals through a sharing of beliefs, needs, and context. A common arena for exploring mixed-initiative interaction is conversation, centering on a collaboration to achieve the goal of communicating needs and information. However, mixed-initiative interaction extends beyond conversation to encompass a wide variety of interactions that rely on a collaborative interleaving of contributions by participants, some of which might include conversational interaction.

Psychologists and computer scientists have referred to efficient collaborations converging on shared goals as *joint activity*.[9–11] Joint activity captures the behavior displayed in fast-paced, well-coordinated interactions among people who work together to solve a mutual goal. Examples of joint activity include the collaborative behaviors seen in conversation, dancing, and the familiar struggle of moving a large piece of furniture through cramped hallways. Participants in joint activity seek convergence on a shared set of beliefs about the setting, activity, goals, and the nature and timing of individual contributions. Uncertainties about goals and needs are resolved through a drive towards a mutual understanding or common ground in a process referred to by psychologists as *grounding*.[9,10,12]

Joint activity embodies an especially fluid and efficient form of mixed-initiative interaction. The pursuit of metaphors, designs, and reasoning machinery for supporting joint activity presents the most difficult challenges—and the greatest opportunities—for research on mixed-initiative interaction.

Beliefs, actions, and initiative

Mixed-initiative systems must consider a set of key decisions in their efforts to support joint activity and grounding. These include *when* to engage users with a service, *how* to best contribute to solving a problem, *when* to pass control of problem solving back to users for refinement or guidance, and *when* to query a user for additional information in pursuit of minimizing uncertainty about a task.

Systems that provide automated services rely on the ability to make good guesses

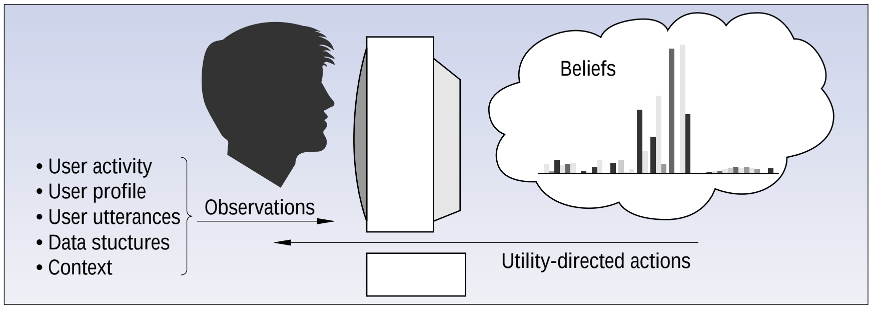


Figure 1. A Bayesian perspective on human-computer interaction. A probability distribution about a user's goals (bar graph) is computed from a set of observations and background information about the user. Actions are selected based on their expected utility. The probability distribution here, generated by the Lumière system, displays the likelihoods of a user's different goals while working with Microsoft Excel.

about a user's needs by considering evidence obtained through the narrow keyhole of user interface events. A system's ability to understand users can be enhanced by coupling richer systems for monitoring user activity with more expressive knowledge representations and sophisticated grounding skills. However, even given more complete knowledge about a user's activity, mixed-initiative systems must still grapple with significant uncertainties. Thus, building effective mixed-initiative systems requires the consideration of key uncertainties both at design time and in real time.

A Bayesian approach to human-computer interaction provides a valuable perspective for the design of mixed-initiative systems. Bayesian agents maintain beliefs about such critical variables as a user's intentions and attention. Bayesian agents also update their beliefs continuously with probabilistic procedures that consider both passively observed and actively gathered evidence.

Recent work on the use of real-time Bayesian inference suggests that dynamic reasoning under uncertainty can be a valuable component of mixed-initiative interaction. Both hand-built and automatically learned probabilistic user models, including Bayesian networks, have been embedded as key components of user-interface prototypes. For example, in the Lumière system,[4] a Bayesian network model analyzes a stream of events generated by the user's interaction with Microsoft Excel. It continuously infers probability distributions over the user's goals and user's interest in receiving active assistance. When the user makes an explicit query for assistance, information about this query is added to the analysis. The bar graph in Figure 1 represents a snapshot of a probability distribution inferred by Lumière over a user's goals. My colleagues and I have developed prototypes that not only reason about a user's goals and needs, but additionally harness Bayesian networks to infer a probability distribution over the attentional focus of users.[13]

Guiding mixed-initiative action with expected utility

A system endowed with the ability to infer beliefs about the states of a user's intention and attention can make ideal decisions about how and when an automated service should step in to assist a user. More specifically, access to beliefs about a user's goals give a mixed-initiative system the ability to take information-gathering and problem-solving actions that have the highest *expected utility*, taking into consideration the expected benefits and costs of attempting to participate in problem solving. Expected benefits represent the gains in efficiency associated with offering a contribution under uncertainty. Expected costs capture the frustration and inefficiency associated with the distraction of presenting an otherwise valuable contribution—or of executing an inappropriate contribution. Beyond reasoning about goals, inferences about the attention of users are critical in making decisions about the best time to provide assistance. Significant costs may be associated with querying a user or providing a partial solution when the user is not ready to accept the intervention.

The policy of taking actions associated with maximum expected utility has a long tradition, founded on the axioms of utility, formulated originally by John von Neumann and Oskar Morgenstern over 50 years ago. Although expected utility has enjoyed a rich history of application in such fields as economics and decision analysis, it has only recently been applied in human-computer interaction.

Designing for a mix of initiatives

Harnessing probabilistic inference to provide an awareness of users and expected utility to guide actions offers an overall perspective that can guide the development of mixed-initiative architectures. However, the basic principles do not provide detailed blueprints for creating specific, valuable interleavings of direct manipulation and automation. Designs for mixed-initiative systems benefit greatly from careful consideration—from the earliest phases of the design process—of the detailed interactions between potential automated services and options for user manipulation and display.

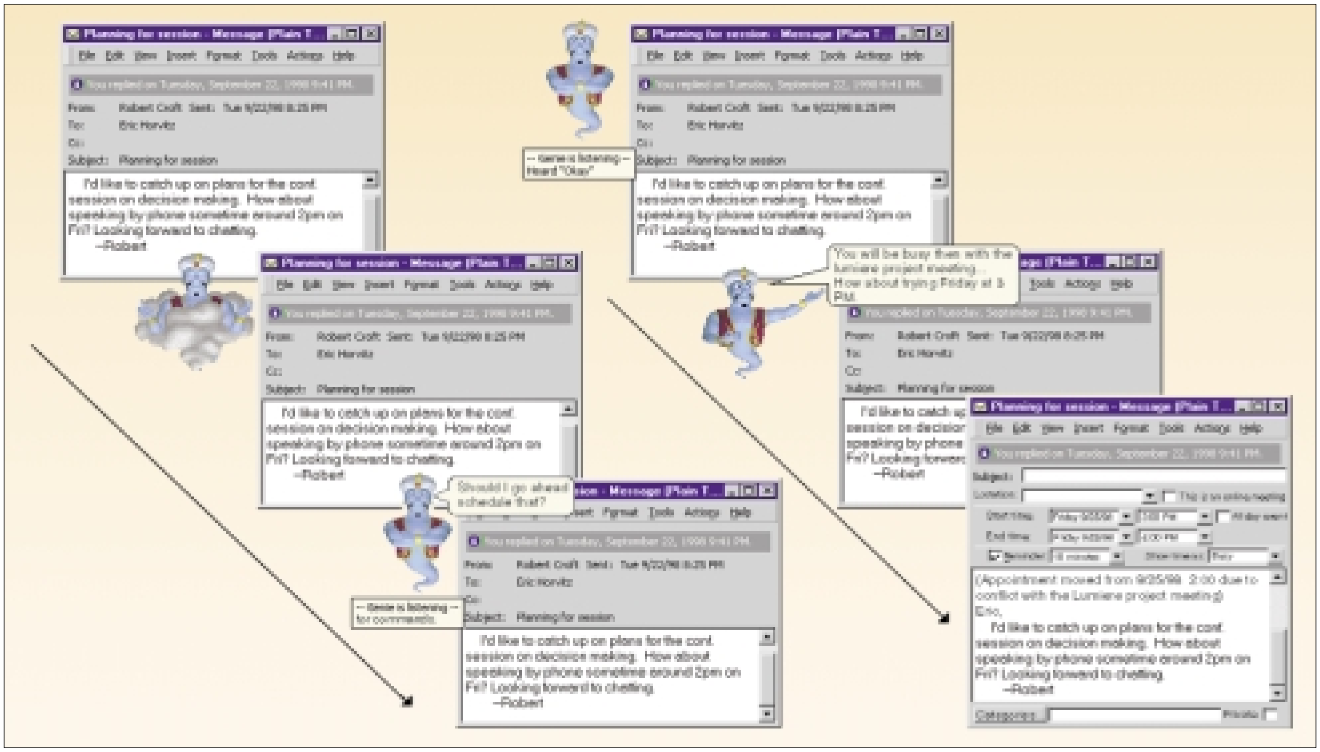
The large space of design opportunities for mixed-initiative interaction includes

- developing automated services that are performed *in line* with a user's activity, allowing users to take advantage of contributions provided by a system while they work in a natural manner,
- identifying elegant metaphors that promote efficient grounding by providing efficient means for users and computers to communicate information *about* intended or ongoing contributions to a solution, and
- developing automated services that can provide solutions at varying levels of precision or completeness, giving mixed-initiative systems the flexibility to scope the precision of contributions in accordance with the uncertainty about a user's goals or the competency of an analysis.

The latter class of design opportunities is motivated by the notion that, as uncertainty grows about a user's intentions or about the quality of the result, a system should gracefully degrade its contribution so as to "do less, but do it well." That is, we prefer that a system provide users with a clear, valuable advance towards a solution—an advance that minimizes the need for the user to perform costly undoing or backtracking. We can enhance the ability of systems to make decisions about the most appropriate contribution by endowing those systems with the ability to decompose prototypical tasks into sets of subtasks that span a spectrum of precision or completeness.

Lookout

A prototype system named Lookout provides concrete instantiations of several key concepts that highlight the role of decision making under uncertainty in mixed-initiative interaction.[8] Lookout assists with the



Planning for session - Message (Plain T...

From: Robert Croft Sent: Tue 9/22/98 8:25 PM
To: Eric Horvitz
Cc:
Subject: Planning for session

I'd like to catch up on plans for the conf. session on decision making. How about speaking by phone sometime around 2pm on Fri? Looking forward to chatting.
—Robert

(Appointment moved from 9/25/98 2:00 due to conflict with the Lumiere project meeting)
Eric,
I'd like to catch up on plans for the conf. session on decision making. How about speaking by phone sometime around 2pm on Fri? Looking forward to chatting.
—Robert

File Edit View Insert Format Tools Actions Help
You replied on Tuesday, September 22, 1998 9:41 PM.
-- Genie is listening -- Heard "Okay"
You will be busy then with the Lumiere project meeting... How about trying Friday at 3 PM.
Should I go ahead and schedule that?
-- Genie is listening -- for commands.
Subject:
Location:
This is an online meeting
Start time:
End time:
All day event
Reminder
Show time as:
Categories...
Private

Figure 2. Lookout in action. Procedures that harness probability and utility guide Lookout's actions to assist users with accessing their calendar and composing appointments, based on a background analysis of e-mail being reviewed.[8]

tasks of calendar review and appointment creation. A group of interested people scattered throughout Microsoft have employed the system since it was released for internal testing in early 1998.

Lookout monitors a user's interactions with the Microsoft Outlook messaging and calendar system. The system recognizes when users open and attend to new e-mail messages. Lookout decides whether, when, and how to best assist users with the tasks of accessing the appropriate view of their calendar and scheduling appointments associated with the messages.

For each message being reviewed, Lookout infers the probability that a user has the goal of invoking Outlook's calendar and scheduling subsystem. This is done by considering information in the header and patterns of text in the body of e-mail messages. Given this probability, and the costs and benefits of providing automation, the system performs an expected-utility analysis and decides whether to simply do nothing (letting users continue to perform direct manipulation), or to interact with the user. The system considers the expected utility of pausing to ask the user if he or she might like assistance and of simply going ahead and performing the most appropriate calendar view or scheduling action without requesting the user's input. When the system decides that automated calendar access or appointment generation would be a valuable contribution, it displays results in a manner that makes it easy for the user to further refine or undo the analysis.

In providing its service, Lookout uses knowledge about the typical ways people describe meetings and times. It understands the temporal implications of such phrases appearing in e-mail as "Fri afternoon," "tomorrow at 3," "next week," "in December," "get breakfast," "grab lunch," and so forth. In preparing its analysis, Lookout considers a spectrum of contributions at differing levels of precision. The system seeks to provide a user with a valuable step forward by displaying an automatically generated appointment or a calendar view with the most appropriate scope.

If the system can identify a single date and time with confidence, it will construct a proposed appointment by filling out the day, time, and subject fields of an Outlook appointment form and present it to the user for confirmation or modification. If the target appointment conflicts with another meeting on the calendar, the system will search to find an alternative time for the event before composing and presenting the appointment. If the system cannot identify a specific day and time with confidence, it will opt to introduce a less precise contribution. For example, the system will open the calendar to the most likely day, or the most likely week, and pass control back to the user for refinement.

Lookout relies on reasoning, learning, and communication to provide services in line with the flow of a user's work. Lookout employs a model for gauging the status of a user's attention in making decisions about when to jump in and query the user or to perform its service. The system infers the amount of time a user wishes to dwell on an e-mail message at hand by considering attributes of the message and the user's activity. Specifically, Lookout considers the length of the message and the time since the last paging or scroll event to decide on the ideal time to step in. Early versions of Lookout that did not employ such a model of attention had a very different feel; the appropriate timing of services dramatically improves the experience and relays a remarkable sense that an intuitive assistant is attempting to work with the user.

Lookout can be instructed to run in a hands-free, social-agent modality, employing an explicit animated assistant coupled with speech synthesis and recognition. When operating in the social-agent mode, Lookout establishes a separate audio channel for communicating with users about contributions, minimizing the potential conflict with ongoing keyboard and mouse activity. Figure 2 shows a sample interaction with Lookout as an embodied agent.

Integrating an explicit social presence has let us explore the use of gestures and utterances that might be expected in natural mixed-initiative interaction among people. For example, the agent selects a behavior from a set of gestures and utterances that communicates its confidence about taking an action. Also, the agent displays signs of confusion when the speech-recognition subsystem has difficulty interpreting the audio signals. If the system does not receive a response when offering the user assistance, it uses gestures to communicate, in a noninvasive manner, the notion that it understands that the user is too busy to respond before disappearing (for example, the agent will shrug, relaying with visual cues that "I was just trying to be of service—no problem..."). At such times, the agent will wait patiently on the sidelines for a period of time that is computed dynamically as a function of the inferred belief that the system could have provided a valuable service.

Lookout continually attempts to improve its ability to provide valuable contributions by performing background learning. The system's models for inferring the goals and attention of users are updated over time through implicit observation of a user's behavior, using a learning process that collects evidence about such variables as the content of messages associated with a user's scheduling activity and the period of time between a message being opened and a user's direct execution of calendar and scheduling tasks.

Beyond implicit learning, Lookout allows users to directly indicate their preferences about the system's decision-making behavior. Preferences input during configuration are used in Lookout's cost-benefit analyses and timing decisions. Additionally, users can take the initiative to invoke Lookout's services at any time by simply clicking on the Lookout icon that is always available on the System Tray of the Windows shell.

We have explored designs for more deeply integrating Lookout's automated services with direct manipulation and display. Lookout's current version was designed to work with a legacy software application, rather than built as part of a more global design process taking a more comprehensive approach to interweaving direct manipulation and automation. As such, the behavior and value of the Lookout prototype hinges on the design of the direct-manipulation capabilities provided by Outlook. Without Lookout, users typically navigate to the appropriate graphical button or menu item to access their calendar, search for the appropriate day, input the appropriate times, and fill in the subject of the meeting. Changes in the details of how Outlook operates would likely entail modifications of the actions and cost-benefit considerations employed by Lookout.

Frontiers of mixed-initiative interaction

The Lookout system has provided a testbed for utility-directed mixed-initiative interaction on relatively short-run sequences of interaction. Work is underway on leveraging Bayesian inference and expected-utility decision making in richer mixed-initiative systems that work with users on longer, more sophisticated communication and problem-solving sessions. For example, work on the Bayesian Receptionist focuses on methods for supporting joint activity and grounding in conversation about goals that are typically handled by receptionists at the Microsoft corporate campus.[14,15] The Bayesian Receptionist decomposes goals into a hierarchical set of subgoals and employs sets of Bayesian networks and expected-utility decision making to navigate through a subgoal hierarchy in pursuit of common ground.

Lookout and the Bayesian Receptionist have highlighted the necessity and promise of endowing agents with beliefs and of employing probability and expected utility to mesh automated services with direct manipulation. Opportunities abound for harnessing probabilistic methods to weave automation more tightly together with methods that enable users to control, inspect, and guide computing. Although great challenges lie ahead, we believe these early prototypes, and others being developed by colleagues pursuing principles and machinery for mixed-initiative interaction, provide glimmers of the future of human-computer interaction.

References

1. P. Maes and B. Shneiderman, "Direct Manipulation vs. Interface Agents: A Debate," *Interactions*, Vol. 4, No. 6, ACM Press, New York, 1997.
2. P. Maes, "Agents that Reduce Work and Information Overload," *Comm. ACM*, Vol. 37, No. 7, July, pp. 31–40.
3. L. Birnbaum et al., "Compelling Intelligent User Interfaces: How Much AI?," *Proc. 1997 ACM Int'l Conf. Intelligent Interfaces, ACM Press, New York*, 1996; http://www.merl.com/reports/TR96-28/index.html.
4. E. Horvitz et al., "The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users," *Proc. 14th Conf. Uncertainty in AI*, Morgan Kaufmann, San Francisco, 1998, pp. 256–265; http://research.microsoft.com/~horvitz/lumiere.htm.
5. B. Schneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, ACM Press, New York, 1992.
6. M.A. Walker and S. Whittaker, "Mixed Initiative in Dialogue: An Investigation into Discourse," *Proc. 28th Ann. Meeting Assoc. Computational Linguistics*, 1990.
7. G. Ferguson, J. Allen, and B. Miller, "TRAINS-95: Towards a Mixed-Initiative Planning Assistant," *Proc. Third Conf. AI Planning Systems*, AAAI Press, Menlo Park, Calif., 1996, pp. 70–77.
8. E. Horvitz, "Principles of Mixed-Initiative User Interfaces," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems*, ACM Press, 1999, pp. 159–166; http://research.microsoft.com/~horvitz/uiact.htm.
9. H.H. Clark, *Using Language*, Cambridge Univ. Press, New York, 1996.
10. P.R. Cohen and H.J. Levesque, "Preliminaries to a Collaborative Model of Dialogue," *Speech Communication*, Vol. 15, 1994, pp. 265–274.
11. B.J. Grosz and C.L. Sidner, "Plans for Discourse," *Intentions in Communication*, MIT Press, Cambridge, Mass., 1990, pp. 417–444.
12. H.H. Clark and S.A. Brennan, "Grounding in Communication," *Perspectives on Socially Shared Cognition*, APA Books, Washington D.C., 1991, pp. 127–149.
13. E. Horvitz, A. Jacobs, and D. Hovel, "Attention Sensitive Alerting," *Proc. Conf. Uncertainty and Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1998, pp. 305–313; http://research.microsoft.com/~horvitz/attend.htm.
14. E. Horvitz and T. Paek, "A Computational Architecture for Conversation," *Proc. Seventh Int'l Conf. User Modeling*, Springer Verlag, New York, 1999, pp. 201–210; http://research.microsoft.com/~horvitz/converse.htm.
15. T. Paek and E. Horvitz, "Uncertainty, Utility, and Misunderstanding: A Decision-Theoretic Perspective on Grounding in Conversational Systems," *AAAI Fall Symp. Psychological Models of Communication in Collaborative Systems*, AAAI Press, Menlo Park, Calif., 1999; http://research.microsoft.com/~horvitz/grounding.htm.

Evaluating mixed-initiative dialog

Curry I. Guinn, Research Triangle Institute

Researchers in mixed-initiative interaction are trying to make computers be collaborators with their human users. In this two-way information exchange, the computer can do some tasks better alone, some tasks require joint work, and some tasks are better done by the human user. The challenge is to define computation models of how initiative is or should be controlled in a dialog.

In current user-interface design, the predominate initiative structure has the human user initiating almost every interaction. Only in some very fixed, a priori-designed instances might the computer initiate interactions ("All files in directory will be deleted! Are you sure (Y/N)?"). In early human-computer interfaces, these designer-selected interfaces were often faulted for being too rigid. So, today, almost all complex software comes with a multitude of user-preference selections.

In moving away from these rigid paradigms of interaction, we strive toward the more loose, open, and dynamic interaction patterns seen in human-human conversations. Flow control there is often very fast-paced, with many dialog turns and shifts in initiative.

In this essay, I will focus on how to evaluate and compare computational models of mixed-initiative dialog. I will focus on two aspects of this evaluation:

- What are the metrics used for evaluating dialog systems?
- What is the nature of the data set being used for evaluation?

Metrics for evaluating models of mixed-initiative dialog

How would we describe a good model of dialog? One approach to goodness—the descriptive model of dialog—is how well the model fits existing data. A second approach is prescriptive: whether dialogs that result from the model's use have desirable qualities.

Evaluating descriptive models of dialog. Does the theory fit the data? That is the essential question in evaluating descriptive models. The problem for researchers in mixed-initiative dialog is that existing data is extremely limited and the challenge of gathering appropriate data is formidable.

Once gathered, the key issue is appropriately transcribing and then tagging the data. In tagging the data, there is no current standard of what should be tagged. Different research agendas—speech recognition, prosody, parsing, co-reference, and discourse structure—have generated different formalisms for tagging spoken conversations. Recently, efforts such as the IRCS Workshop on Discourse Tagging (http:// ftp.cis. upenn.edu/ ~ircs/discoursetagging/ toplevel. html), the International Workshop on Discourse Tagging, and the ACL 99 Workshop towards Standards and Tools for Discourse Tagging (http://www. research.att.com/~walker/dtagwrk/ ac199-dtag.html) have attempted to bring together researchers from these different areas to reach consensus.

For researchers interested in mixed-initiative systems, strategies for tagging dialogs can range from simply identifying utterance boundaries to complex taxonomy of labels applied to each dialog segment. While we have not yet established a standard of tagging, certain features seem common across a variety of tagging schemes. The corpus is broken up into discourse segments. Labels applied to each segment indicate the function of that segment. For example, the coding scheme used by Sherri Condon and Claude Cech identifies utterance boundaries and labels each utterance with a simple function `MOVE`, `RESPONSE`, or `OTHER`.[1] In contrast, the VerbMobil coding scheme applied to appointment-scheduling dialogs used over 50 domain-specific dialog acts as labels such as `SUGGEST_EXCLUDE_TIME`.[2]

Mixed-initiative researchers must determine whether there should be explicit tags for initiative in a corpus. An early version of the Penn Multiparty Standard Coding Scheme has explicit labels for initiative:

- `INITIATE`—where the speaker maintains or takes control of what is being discussed,
- `RESPONSE-TO-ACCEPT`—where the utterance is a response to a previous utterance and is accepting,
- `RESPONSE-TO-REJECT`—where the utterance is a response to a previous utterance but rejects its proposal or content, and
- `RESPONSE-TO-OTHER-RESPONSE`—where the utterance is a response to a previous utterance that neither confirms nor rejects its content.[3]

These labels, however, are heavily loaded with meaning. The very definition of what initiative is (does it refer to the dialog or the domain task or both?) is still controversial. Thus, coding schemes that require taggers to make decision on initiative are likely to have low reliability across coders and across projects. In a tagging scheme Jennifer Chu-Carroll and Michael Brown devised, taggers had to decide at each dialog turn which participant had task initiative and which participant had dialog initiative.[4] The kappa statistic for dialog initiative was 0.69, while the kappa for task initiative was 0.57. (For an overview of the kappa statistic as a means of evaluating a coding scheme's reliability, see Carletta et al.[5]) As a general rule, $K > 0.8$ represents high reliability, $0.67 < K < 0.\ 8$ moderate reliability, and $K < 0.67$ low reliability. If other studies show similar results, this suggests using initiative-neutral labeling schemes.

One initiative-neutral tagging system is the Coconut scheme, which is a derivative of the DRI tagging scheme, which uses the concept of forward-looking and backward-looking functions to label discourse segments.[6]

- Forward-looking functions include *Statements* (assertions and reassertions), *Influence-on-Hearer* (information request, action directives, laying out options), and *Influence-on-Speaker* (offers and commitments).
- Backward-looking functions include

Answers (to previous requests) and *Agreement* (acceptance or rejection of a belief or proposal embodied in a previous utterance).

In a study of inter-coder reliability using this system, the kappa coefficient for Statement and Answer were quite high (0.83 and 0.79, respectively), making these labels highly reliable.[7] The kappas for Influence-on-Hearer and Influence-on-Speaker were reasonable (0.72 for both). However, the cross-coder reliability of Agreement was only 0.54. These results indicate that many of the features believed to be important in setting and changing initiative (most forward-looking functions) are highly codeable. However, there might be important classes of utterances that are challenges or acceptance of initiative (the Agreement function) that require subjective evaluation.

An important area of study is determining how and why agents challenge the initiative of their collaborator. These backward-looking functions apparently have very low cross-coder reliability, indicating that evaluation of these utterances is much more subjective. This reliability result seems to present a serious challenge to mixed-initiative researchers. It is not surprising that tags related to initiative have low reliability—a precise all-acceptable definition of what initiative is has not been specified. Until an algorithm is devised that identifies which agent has what level of initiative, it is unlikely that high cross-coder reliability will be achievable within a study, much less across studies.

Evaluating prescriptive dialog models. Here, we are less concerned with the theory fitting the data. Rather, we want computational agents carrying out our dialog theory to produce conversations with desirable qualities. What qualities might we look for? Past evaluations of human-computer systems have focused on a variety of performance factors summarized nicely by Marilyn Walker and her colleagues.[8] The performance factors divide into objective metrics such as mean response time, which require no human judgment, and subjective metrics such as appropriateness of responses, which require human judgment. Table 3 lists objective and subjective metrics. While not meant to be exhaustive, this list gives a sense of the potential complexity of the analysis. (Walker gives a subset of this list. [8]) As this table shows, many of the features that are of interest to mixed-initiative researchers are subjective.

Walker's Paradise evaluation system uses decision theory to combine various metrics into a single performance metric.[8] Using a linear-regression analysis, it determines which factors are most important to maximizing task success while minimizing cost. Their analysis of Elvis, a voice-interactive email service, compared two initiative strategies:

- System initiative, where the system prompts the user by giving a list of possible responses ("Select messages by subject, by sender, or in order of arrival.") and
- Mixed initiative, where the system does not prompt with possible responses ("I've got your mail."). The user is expected to know the system capabilities and what is appropriate given the interaction's context.

Their analysis revealed that the three significant user-satisfaction factors in this spoken interactive system were the user assessment of task completion, the mean recognition score (performance of the speech recognizer), and elapsed time. The overall performance with mixed initiative was not as good as system initiative. Subjects had a more difficult time learning how to use the mixed-initiative system, and the mean recognition score tended to be worse.

Data sets for evaluation

Three types of data sets have been used to evaluate models of dialog: human-human dialogs, human-computer dialogs, and computer-computer dialogs.

Human-human dialogs. Gathering the appropriate data to model is a significant challenge for dialog researchers. But in evaluating dialog systems, we have very limited data sets with which to work. These data sets are expensive to obtain, and the quality of the data is affected by many confounding factors:

- How do we gather naturally occurring conversations between human participants?
- How does the intrusion of recording affect the data?
- If the communication between humans is partly linguistic, partly gestural, and partly intonational, how is this data captured and transcribed? (Tools for automating (or semiautomating) the transcription process are an important research area.[9])
- How can an external observer ever be confident of understanding the nuances of conversation between human participants? The underlying knowledge that makes interpretation possible might be inaccessible to the transcriber.

Also, the data collected is often far more complex than limited dialog theories can model. To simplify the data collected, researchers often will limit the contextual elements of the dialogs in the laboratory setting. Preventing face-to-face interaction and having typed interaction are common techniques in eliminating variables (essentially gesture and intonation) that computers currently are not good at interpreting or generating. Another (perhaps obvious) technique is to provide a rigid task for the conversants to work on. This predefined task makes modeling the domain's semantics much easier and can limit the amount of hidden knowledge.

For a variety of reasons, the corpus of human-human conversations is quite limited. They are usually gathered with a specific research agenda in mind, which might bias their use in particular directions. For instance, in the Map Corpus domain, one participant is usually the expert while the other is a novice.[10] The resulting conversation, then, might have very few changes in initiative. Specifically trying to study the effect of dialog initiative, Terry Moody had a human play the role of an expert system that had to follow certain predefined initiative rules (in what is often called a Wizard of Oz experiment).[11]

Human-computer dialogs. Here, researchers actually implement some or all of their dialog theory on a computer and have human subjects interact with the system. By actually implementing a system, they can directly test aspects of a theory. A typical experiment might involve running sets of subjects with the system using various initiative strategies. Recent examples of spoken-language systems used for testing initiative strategies include Elvis, Toot, Trains, the CSELT Italian railway timetable system, and the Circuit Fix-it Shoppe.[12–15]

There are a number of problems with running human-computer dialog experi-

Table 3. Objective and subjective metrics.

Objective metrics	Subjective metrics
Percentage of correct answers	Percentage of implicit recover utterances
Percentage of successful transactions	Percentage of explicit recover utterances
Number of dialog turns	Percentage of appropriate system utterances
Dialog time	Cooperativity
User response time	Percentage of correct and partially correct utterances
System response time	User satisfaction
Percentage of error messages	Number of initiative changes
Percentage of "non-trivial" utterances	Number of explicit initiative changing events
Mean length of utterances	Number of implicit initiative changing events
Task completion	Level of initiative
	Number of discourse segments
	Knowledge density of utterances
	Co-reference patterns

ments. Some of these difficulties encountered by researchers often have little to do with the actual dialog theory:

- Creating an actual dialog system involves a very intensive programming effort that introduces variables of its own.[14]
- Because typed input systems drastically affect user perception and performance, a speech interface is preferred. Despite tremendous advances in speech recognition in the last decade, the errors that occur because of speech recognition often dominate the development cycle and performance.[17] In the analysis carried out by Walker, the mean recognition score was one of the three dominant variables affecting system performance.[7]
- Once the words are recognized, the system requires excellent natural-language parsing.
- The inevitable variability in human subjects requires a large test set. Ronnie Smith and Richard Hipp, for instance, found that subject outliers made model evaluation substantially more difficult.[17]
- Finally, the time to run and transcribe experiments inevitably limits the number of possible subjects.

Computer-computer dialogs. If the computational model of dialog developed is intended to model both participants in a dialog, it is theoretically possible to conduct computer-computer conversations. Two computational agents could each use the same dialog model (presumably with different knowledge sets) and have a conversation. The advantages of running computer-computer experiments mirror the disadvantages of running human-computer experiments. While a complex software system might have to be built, it does not require a user interface, speech recognition, or a sophisticated parser. Most importantly, computer-computer simulations can generate large test sets. Parameters that affect the dialog can be easily changed and tested. A number of researchers have performed such ablation studies.[18–21]

As an example, my initiative model has each computational agent attempt to assess each agent's ability to solve a goal based on a probabilistic analysis of its own knowledge and its model of its collaborator's knowledge. By varying how and when the agent carried out such analysis, we can gather empirical data on the effectiveness of various initiative-setting strategies.

What do computer-computer simulations say about our models? Simulations provide one tool for analyzing a complex model. These simulations give us detailed information about the underlying model. By observing the resulting dialogues, we can ascertain whether the underlying model generates interactions that have the target behaviors observed in human-human or human-computer dialogs.

References

1. S.L. Condon and C.G Cech, "Functional Comparison of Face-to-Face and Computer-Mediated Decision-Making Interactions," *Computer-Mediated Communication: Linguistic, Social, and Cross-Cultural Perspectives*, John Benjamins, Philadelphia, 1996, pp. 65–80.
2. J. Alexandersson et al., *Dialogue Acts in VERBMOBIL-2*, Verbmobil Report 204, DFKI Saarbrücken, Univ. of Stuttgart, Stuttgart, Germany, 1997.
3. M.A. Walker et al., *Penn Multiparty Standard Coding Scheme Draft Annotation Manual*, 1998; http://www.cis.upenn.edu/~ircs/discourse-tagging/newcoding.html.
4. J. Chu-Carroll and M. Brown, "An Evidential Model for Tracking Initiative in Collaborative Dialogue Interactions," *User Modeling and User-Adapted Interaction*, Vol, 8, Nos. 3–4, 1998, pp. 215–253.
5. J. Carletta, "Assessing Agreement on Classification Tasks: The Kappa Statistic," *Computational Linguistics*, Vol. 22, No. 2, 1996, pp. 249–254.
6. B. Di Eugenio, P.W. Jordan, and L.Pylkkanen, *The Coconut Project: Dialogue Annotation Manual,* ISP Tech. Report 98–1, Intelligent Systems Program, Univ. of Pittsburgh, Pittsburgh, 1998.
7. B. Di Eugenio et al., "An Empirical Investigation of Proposals in Collaborative Dialogues," *Proc. 17th Int'l Conf. Computational Linguistics and 36th Ann. Meeting Assoc. Computational Linguistics (COLING-ACL'98)*, Assoc. for Computational Linguistics, Nantes, France, 1998.
8. M.A. Walker et al., "Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies," *Computer Speech and Language*, Vol. 12, No. 3, 1998, pp. 317–347.
9. J. Passonneau and D. Litman, "Discourse Segmentation by Human and Automated Means," *Computational Linguistics*, Vol. 23, No. 1, 1997, pp. 103–141.
10. J. Carletta et al., "The Reliability of a Dialogue Structure Coding Scheme," *Computational Linguistics*, Vol. 23, No. 1, 1997, pp. 13–33.
11. T.S. Moody, *The Effects of Restricted Vocabulary Size on Voice Interactive Discourse Structure*, PhD dissertation, North Carolina State Univ., Raleigh, N.C., 1988.
12. J.F. Allen et al., "The TRAINS Project: A Case Study in Building a Conversational Planning Agent," *J. Experimental and Theoretical AI,* Vol. 7, Jan. 1995, pp. 7–48.
13. J.F. Allen, G. Ferguson, and L.K. Schubert, "Planning in Complex Worlds via Mixed-Initiative Interaction," *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative*, AAAI Press, Menlo Park, Calif., 1996, pp. 53–60.
14. M. Danieli and E. Gerbino, "Metrics for Evaluating Dialogue Strategies in a Spoken Language System," *Proc. 1995 AAAI Spring Symp. Empirical Methods in Discourse Interpretation and Generation*, AAAI Press, Menlo Park, Calif., 1995, pp. 24–39.
15. R.W. Smith and D.R. Hipp, *Spoken Natural Language Dialog Systems: A Practical Approach*, Oxford Univ. Press, New York, 1994.
16. C. Guinn, "An Analysis of Initiative Selection in Collaborative Task-Oriented Discourse," *User Modeling and User-Adapted Interaction*, Vol. 8, Nos. 3–4, 1998, pp. 255–314.
17. M.A. Walker, "The Effect of Resource Limits and Task Complexity on Collaborative Planning in Dialogue," *AI Journal*, Vol. 85, Nos. 1–2, 1996, pp. 181–243.
18. R. Power, "The Organization of Purposeful Dialogues," *Linguistics*, Vol. 17, 1979, pp. 107–152.
19. J. Carletta et al., "The Coding of Dialogue Structure in a Corpus," *Proc. Twente Workshop on Language Technology: Corpus-Based Approaches to Dialogue Modeling*, Univ. of Twente, Enschede, The Netherlands, 1995, pp. 25–34.
20. J. Carletta, "Planning to Fail, Not Failing to Plan: Risk-Taking and Recovery in Task-Oriented Dialogue," *Proc. 14th Int'l Conf. Computation Linguistics*, Assoc. for Computational Linguistics, Nantes, France, 1992, pp. 896–900.