Talking to Machines Today and Tomorrow: Designing for the User

Michael J. Cosky
Barry L. Lively
Linda A. Roberts
Blake L. Wattenbarger

Automatic speech recognition (ASR) is finding many new applications in both products and services. Designing good user interfaces with ASR poses a variety of challenges. This paper:

- Addresses basic principles in designing for the user,
- Describes the need for appropriate dialogues between the user and ASR-based systems,
- Discusses how these user-machine transactions are changing due to improved ASR technology, and
- Discusses the direction ASR should take to provide the kind of "natural" user-machine dialogue that is expected in Vision 2001.

Introduction

Hundred is a small town in West Virginia. About 45 years ago, it was a common occurrence for someone living in Hundred to go to the telephone, lift the receiver from the switch hook, turn the crank to signal the operator, and say:

"Good morning, Florence. Would you ring my mother, please?"

And Florence might reply:

"She's over at your Aunt Madge's. Do you want me to ring her there?"

The communication was natural and personalized. Then advances in technology provided a more economical telephone service by centralizing operators, installing advanced switching systems, and providing telephones with a wide range of features. The communication certainly was faster and more economical, but user actions were more indirect—less natural and less personalized.

Today, the caller dials a seven-digit number (roughly the maximum number of digits the average individual can remember) if "Mom" is not too far away, or an 11-digit number—or longer—if she is in some distant city. Technology, of course, continues to advance. As the year 2001 approaches—the beginning of the 21st century—it is possible that some of the naturalness that marked early electronic communication can return by using:

- Speech-based user interfaces that employ automatic speech recognition (ASR) for speech input, and
- Coded or synthesized speech for prompting and feedback.

Rather than dialing an arbitrary number, the user will be able to talk naturally to almost any telecommunications device:

"Computer, access December year-end report, display sales figures for Model 1341."

The device will respond with its own utterance, based on its findings, as well as any recent changes:

"Year-end report accessed. Accounts Receivable has hold on sales figures for Model 1341, pending inventory report."

There are many consumer products on the market now, particularly for turning on computers and dialing car phones, that allow a simple form of dialogue. The user might say to the cellular phone system:

"Mom."

and the system would respond:

"Dialing Mom."

This interchange clearly requires less effort than remembering and dialing

Panel 1. Abbreviations, Acronyms, and Terms

ASR—Automatic speech recognition
GUI—Graphical user interface
UI—User interface

1-813-555-1234. But notice that while the word recognizer can deal effectively with "Mom," it would find it difficult to handle what the user said to Florence—"Would you ring my mother, please?"

Technology is available that could come much closer to doing Florence's job; but, as other papers in this issue make clear, it has some distance to go to achieve Florence's level of sophisticated performance. The technology needed to support a more natural exchange between the user and the system is the kind envisioned in the year 2001. That world, it is assumed, will have seamless user-machine communication, no matter where the user is. In the office, the car, at home, or on vacation, it will be possible to stay in touch with both machines and other people—with simple and direct verbal communication.

Human-Factors Engineering

Human-factors engineering, also called *usability engineering* or *user-centered design*, refers to a set of methodologies that:

- Make products and services very easy to use, and
- Enhance productivity and efficiency of users.

Human-factors engineering is a formal discipline made up of trained professionals, typically with advanced degrees in behavioral sciences, such as experimental psychology, cognitive psychology, and human factors. These specialists have expertise in a number of areas, including statistical analysis and user interface design methodologies. Some of the common methodologies used by human-factors experts are described in Panel 2.

Also encompassed in human-factors engineering is the use of behavioral sciences and applied psychological principles and methods to design user interfaces for products and services—keeping in mind the user's goals, skills, and limitations throughout the design process.

Human-factors engineers incorporate user feedback at all stages of the design, including the earliest conceptual stages. Data from representative users must be included in the user-centered design process—from the early analyses of user tasks, to iterative user-interface design and usability testing, to final empirical usability testing of the completed application, including documentation and training.

The benefits of human-factors engineering include:

- Decreased development cycle time and costs;
- Increased productivity, reliability, and user satisfaction; and
- Marketplace differentiation on ease of use.

The result of human-factors engineering is high-quality user interfaces that are:

- Designed to match a user's capabilities and limitations;
- Easy to learn and to use;
- Consistent with the user's expectations, leading to decreased training costs; and
- Highly satisfying to customers, leading to increased sales.

Basic Principles in Designing for the User

Designing for the user begins with a set of basic principles for satisfying the user's needs. An understanding of these needs can help in the design of virtually any user interface, including those designed for speech-based applications. In one way or another, these principles have motivated the design of a variety of products, including airplane cockpits (from World War II to the present), as well as the way light switches work; traffic-light operation, accommodating the requirements of people with special needs, the layout of the touchtone dial pad, and the way users interact with computers. These issues will be introduced in this section in ways that will highlight some of the challenges facing designers and developers in achieving Vision 2001.

1. Understand the User and the Application. When a product concept is being considered for development, it is very useful to observe typical users in their own environment, using applications that are related to that particular product. Bottlenecks may be observed in the way users do their work, or more efficient ways of performing their tasks might be noted.

Understanding users and building a design around their needs entails a variety of different kinds of testing procedures that, together, help to answer many questions about what the new product should be and how it should work. Opaluch and Tsao[1] and Day and Boyce[2] described

Panel 2. Human Factors Methodologies

1) Literature Reviews. If the human-factors expert is working on a problem that has been around for a while, it may be appropriate to consider using the results of previous human factors studies by reviewing the literature, and checking the large body of scientific work reported in research journals and books in applied psychology, cognitive science, market research, management science, and other behavioral science disciplines.

2) Task Analysis/Customer Needs Analysis. Human-factors professionals observe users in their natural setting. They derive an understanding of the processes users employ while performing their jobs, and identify critical tasks and bottle-necks in their current processes, as well as causes of error. These analyses identify critical areas for planning and designing new features and services.

3) Focus Groups. Simulations, such as written descriptions, slide shows, prototypes of user interfaces of possible products, can be shown to customers in order to obtain feedback. These simulations and prototypes will elicit responses both on the usefulness of the feature or service being considered for the product, and on the manner in which the features are implemented in the user interface. Often a feature must be described concretely in the form of a specific user interface to get meaningful feedback from customers.

4) Participatory Design. If there is enough time in the process, it may be appropriate to have end users participate actively in the design process, particularly for simple features or applications. In such cases, representative customers are brought together to express their needs and preferences for each detail of the design. These representative users assist in producing user interface (UI) simulations and mock-ups in a small group setting moderated by a human-factors specialist. Marketing and development specialists often will participate, as well. The mock-ups are often made from index cards, post-it notes, and other low technology devices. The output of this activity is usually then translated into a simulation or prototype of the UI design.

5) Simulations/Prototyping. Because the product itself is unavailable early in the design process, simulations or prototypes must be created in order to facilitate an understanding of the feature or service. Simulations may take different forms, depending on their complexity and on the particular need. For example, if working prototypes are not available, the simulation might be a "Wizard-of-Oz" approach—an experimenter in another room simulating ASR performance.

6) Heuristic Evaluation. Human-factors specialists perform an evaluation of the user interface, based on professional training and experience with usability principles. This is also called an expert audit.

7). Laboratory Studies. When human-factors researchers are faced with decisions that require selecting among two or more alternative implementations of a feature, or are deciding whether a single implementation measures up to some criterion of performance or acceptance, traditional lab studies are most appropriate. Here, representative customers participate in an experiment in a human-factors laboratory in which they interact with the alternative implementations. By examining errors, response times, and preferences, usability engineers can easily determine which implementation best meets user needs.

8) Usability Studies. Usability testing early in the product design is particularly cost-effective because results can be used to ensure that usability objectives can be met and that an excellent user interface is designed before the product is built. Based on expert knowledge or on the findings from task analyses, simulations or prototypes are developed and end-users are brought into the lab to use the feature or service by performing tasks on the simulation. Using this method, human-factors experts can understand where potential problems or confusions occur, and where subsequent changes must be made. Data collection instruments include: observation, directed dialogue, videotape, software capture, electronic instruments, questionnaires, and post-experiment interviews.

9) Iterative Design. If the feature is quite complex, iterative design is required. That is, following focus groups or usability testing, changes and improvements to the design of the feature or service are made and additional usability testing takes place to make sure that the changes were appropriate. This process continues until no problems remain or until usability objectives can be met.

10) Field Studies. It is critical to observe behaviors and collect data at customer sites. For example, in order to understand the needs of telephone attendants, it is essential to gather information in the user's actual environment. The goals of field studies are to understand user needs prior to developing a product, as well as to evaluate user satisfaction after the product's release in order to improve the user interface for the next release. Data from actual users can be collected in field settings with or without control of independent variables that are thought to impact user performance or satisfaction. Data can be collected through surveys, structured interviews, observation, video tape, software capture, electronic instruments, transaction logs, and user diaries. It may also be possible to solicit reactions to paper- or computer-based simulations of proposed user interfaces.

efficient techniques for gathering this information.

An issue that can be anticipated in most applications is the problem of accommodating both experienced and novice users. Sophisticated users want a quick transaction with minimal support in the way of prompting and feedback. New users want more support, either in being told which options are available or in understanding what the system will accept as valid input, such as whether the system will accept "I want to reverse the charges," or "I want them to pay for this," or just "Collect."

A procedure called *Question + Options* addresses the differences between experienced and novice users. With this procedure, the system asks a question and then, after a short pause, offers a menu. For example, the system asks:

"What type of call would you like to make?" (brief pause).

"Please say collect, credit card, third number, person-to-person, or operator—now."

In the simplest systems, the automatic speech recognizer can "listen" for the presence or absence of speech during the pause. In this case, less experienced users can say nothing during the pause and wait to hear the alternatives, while the more experienced users can speak during the pause, thus avoiding the time-consuming menu of choices.

More advanced systems offer a *barge-in* capability, which allows the user to speak at any time,[3] and *key word spotting*, which makes it possible to pick out key words when the user speaks some extraneous words, such as "I'd like a collect call, please," when only "collect" is required by the system, or "Would you ring my mother, please?", when only "mother" is needed.

It is likely that designs employing the Question + Options approach will be used into the next century because the approach yields good performance, is well liked by users, supports both novice and experienced users, and does not preclude the advancement of important forms of technology.[4]

A more powerful ASR system might simply accept a broader range of utterances, such as synonyms, and still offer a limited set of alternatives—although the specific alternatives offered may depend on the "meaning" extracted from what the user said.

2. Ensure Self-Evident Feature Operation. Part of the challenge of Vision 2001 is to make products and services so easy to use that aids that are external to the system, such as the customer instruction booklet, will not be needed. That possibility is part of the appeal in developing speech recognition as a means of user control. At some time in the future, the user will speak naturally and the device will "understand" variations in wording and syntax.

What exists now are systems that constrain the user to using only certain words or phrases. Novice users often become tense because of the differences between what they want to say and what the system will accept; the system must guide the user towards acceptable choices as simply and directly as possible.

If a visual display is available, the user can be given a rich array of prompts at any point in the exchange. But the user often has only an auditory channel available for prompting and feedback, such as when driving a car, using ASR across the room from a device, using a computer; or when accessing a service from a pay phone. In such cases, part of the solution is to choose speech prompts carefully to guide the user to make appropriate choices. For example, compare the following prompts for guiding the user's choice of utterances efficiently:

1) "Do you want to make a collect call?"
2) "How may I help you?"
3) "What kind of call do you want to make?"

In the first example, language conventions tend to constrain the user to "Yes" and "No" and their variants. The second example would require a system capable of understanding natural language. The third example is more open ended, and not suitable as it stands. Adding options to the question—the Question + Options method—would be more appropriate.

But even when a prompt seems clear in meaning to its designer, studies have shown that users sometimes don't "hear" what is said. Lively and Holinka[5] saw this when designing and testing a system that provided automated number reporting in directory assistance.

In an early trial, the number requested was inserted into the following sentence frame:

"The number is 555-1212" (followed by two-second pause).

"If you need a directory assistance operator, please remain on the line."

People remained on the line, even when they didn't need help. Apparently, they were not listening as they were copying the number and they didn't hear "If you need a directory assistance operator..." All they heard was "...please remain on the line."

Changing the prompt to the following considerably reduced the user's holding time:

"If you need further assistance, an operator will return."

At a more general level, it is important to provide the user with a way of thinking about the product, or understanding what it is doing, to ensure its ease of operation. *Mental models* offer a powerful means of doing this. A mental model is a mental representation that sets up expectations about how a system works, and helps guide the right behavior. The user's expectation can be implied by design or it can come from user's experiences.

Technical understanding of the product is not required for mental models to be formulated. For example, Bennett[6] has shown that the conception of many people regarding how the telephone system works is two telephones joined by a wire—one step removed from two paper cups joined by a string. This mental model works well for talking on a simple telephone (although some "magic" is required to account for how the call was set up in the first place). The model runs into difficulty, however, when more advanced features are required, such as putting a call on hold or setting up a conference call.

A more advanced mental model is required to deal with this level of technical sophistication. It is not surprising, then, that when people first began connecting telephones to the network on their own, they ran into problems (the phone didn't ring or they couldn't make calls). They had a very difficult time learning to distinguish between problems that arose in the telephone set, versus those that occurred in the wiring or the network. Not having a good model is a likely contributor to those sorts of problems.

Similarly, the authors have observed that users sometimes treat ASR systems as live human beings who don't hear well and aren't at all intelligent—users talk slower and louder. The problem in speech processing applications is to create a model that will help guide the user through the task, often without the aid of specially labeled buttons or a visual display. The way the dialogue between the user and the system is structured can help them to formulate an appropriate model.

3. Employ Users' Knowledge Across Systems. The mental model for a new application is likely to be determined by the experience with a similar application. As speech-based control applications expand to include office, mobile, and residential products, users will be well served if they can carry over their experience from one product to the next. (In any case, they will try to do just that.) On the one hand, this encourages the use of common vocabularies and common operating procedures. On the other hand, it is a good strategy to accommodate synonyms, such as "dial" and "call," or "attendant" and "operator," that users might use during the application.

Another issue concerns consistent feature operation from one application to another. The order of entering a telephone number or a credit card number is an example of an operation that should be consistent from one device to another. Users will expect a particular sequence of these activities, and a well-designed system should meet that expectation. Understanding the importance of these issues, AT&T is promoting consistent user interfaces across ASR applications within the company. This is being done by means of a set of internal standards concerning the structure and wording of, for example, prompts, feedback, and standard command words.

4. Don't Slow the User Down. Experienced users are particularly concerned about the speed in which transactions are completed. The general rule is that simple transactions are better than complicated ones. For example, studies show that short prompts are generally preferred to long prompts, and that feedback messages should not delay the user.[7]

Suppose, for example, that the user has spoken a long series of digits, such as a long distance number or a credit card number. An important concern for the human-factors expert is whether the system should repeat the number so that the user can verify it. Experienced users would rather this didn't happen because it takes up too much time, whereas novice users would prefer feedback to confirm that they didn't make an error.

In work internal to AT&T, it was shown that some ASR systems can judge their own "confidence" in the correctness of *their own* performance in recognizing such a string of digits. The work found that the system could "tell" with a high level of accuracy whether or not an error would be likely to occur. The ASR system would then offer the string for verification only when the probability of an error was likely. Users preferred this differentiated form of verification. The Question + Options method is another way of speeding performance, since options are given only if the user waits for them.

A different approach to allowing users to decrease the time they spend in a transaction is to provide them with the ASR "barge-in" capability that, as noted previously, permits users to interrupt by talking over system announcements, or to enter touchtone responses while the announcements are being played. Both of these capabilities typically result in the same thing: immediate truncation of the announcement; system acceptance of user responses, whether uttered or keyed in; and a dramatic savings in overall transaction time.

5. Provide Simple Ways to Deal With User Errors. The issue here is no different from that in many complex applications. Users may find themselves in blind alleys, not knowing how to get out of an interactive process, or they may say one thing but mean another. The recognizer also will likely make errors.

The first approach is to design the application to deal with Issues 1-4 and avoid errors and confusion. At the next level, assume that errors and confusion will occur anyway and that they must be dealt with. Any solution will, of course, depend upon the application, but some general strategies are to:

- Tell the user what the current state of the system is;
- Provide a means for the user to get to a human operator;
- Provide a means to allow the user to back up a step, and
- Provide a way for the users to get back to the beginning of the application, that is, to start over.

Users at one time were inclined to blame themselves for errors, even when a system error occurred. To some extent, this is still true for errors in manual dialing, but people are beginning to understand that system design has a lot to do with whether errors will be made. They are becoming less tolerant of less-than-excellent designs. This tendency is fed, in part, by users becoming more educated through published evaluations of products' ease of use. *Consumer Reports* and personal computer magazines routinely provide this kind of information.

There are reasons to be particularly concerned about the ease-of-use issue in ASR applications. The ASR recognition process is very different from human recognition, and the kinds of errors the system makes will consequently be different from the confusion humans make. Users would probably understand how confusion could occur over the use of similar sounding words, such as "bun" and "done." But they would have difficulty understanding how "help" and "delete" could be confused, and probably wouldn't tolerate the problem. But to some recognizers, however, these words are close enough for technical reasons for designers to consider not having their functions active at the same time.

6. Customize for Market Segments and Users. A widely applied basic design for a telephone product or service is likely to encounter some conflicting requirements in meeting the needs of all of its users. People differ in their needs and in their preferences, so no single design will be best for all users. The ability to customize and personalize products and services enhances both their usability and attractiveness for users. Perlmuter et al.[8] demonstrated that the ability to select the material to be remembered enhanced performance in a memory task.

The issue of customization is the other side of the concern raised in Issue 3, allowing the user to carry over as much experience as possible from one system to another. A problem exists between the need to provide a standardized, predictable interface and the need to customize that interface to meet the demands of a specific business segment.

It also is important to allow users to personalize their products. Examples include the capability of users to record their own personalized message on an answering system, or to choose their own names for a voice dialing system. In ASR applications, there is sometimes the opportunity to do both.

Examining users' spoken vocabularies can easily demonstrate that providing only "Yes" and "No" as standard utterances for confirmation or denial is not adequate. Accommodating synonyms, such as "Yep," "Yeah," and "Nope," can sometimes be important for a large segment of users. Although most users could be accommodated by most of these choices, the user may want to use additional options that are more natural to them, such as "Oui" or "Non" for a bilingual user.

Internal AT&T design standards for ASR user interfaces include a core vocabulary of command words that should always be accepted by a system, where appropriate, but synonyms are permitted to accommodate the need for customization in particular situations.

7. Make Products and Services Attractive. User interface design has historically had a very functional orientation, focusing on a clean, crisp, and unassuming appearance. In World War II, for example, standardizing and optimizing the design of airplane cockpits had an immediate payoff—pilots and crews were more likely to survive and carry out their missions. This functional approach continues into the present. Putting a dot at the center of each hole on a rotary dial improved dialing speed and accuracy. Automating the reporting of the requested telephone number in directory assistance saved an average of five seconds per call. Giving an ASR service tasks that operators once performed saves money.[9,10]

An important consideration in incorporating each of these changes was whether the user found the change acceptable. Currently, the aesthetics of the user interface on a product or service is assuming a more strategic role. In addition to providing excellent functionality, the user interface is expected to distinguish the product from its competition and, thus, promote market leadership.

AT&T's True Voice® service is an example that distinguishes AT&T from its competitors. Here, sophisticated, real-time speech processing is used to enhance the sound quality of the received speech signal. This patented new technology is used not to enhance intelligibility but to produce a sound that users like better. People don't just find True Voice *acceptable*, they find it *preferable*. Not only has AT&T highlighted this enhancement in advertisements, but it is expected that this kind of strategy will add to market leadership.

The emphasis on aesthetics also puts a premium on the design of the user interface. There are many ways to make an auditory interface more attractive, such as choosing the right wording for prompts and feedback, selecting a pleasing voice and using it appropriately (soft or harsh, demanding, neutral, or inviting), or the effective use of tones and other auditory signals. When used across product lines, these common aspects of the user interface reflect a relatedness that gives a common look and feel, and suggests a common operation, even if technical implementations vary.

Visual aesthetics have been an important part of product design for a long time. The product's shape, colors, and the way visual accents are used have been the bread and butter for industrial design firms for decades. More recently, the design of graphic user interfaces (GUIs) for computer applications have carried this tradition into screen design.

Current and Future ASR Applications

Although many of today's user interfaces that are supported by speech processing technologies are becoming more "natural," they fall far short of what is expected in 2001.

Previous ASR-based applications imposed narrow restrictions on *what* users could and couldn't say, and *when* they could and couldn't say it. The user, often an unsuspecting caller, was required to:

- Speak only the target word, with no filled pauses, noises, or extraneous words permitted.
- Speak the target word *after* the prompt or announcement was completed, and not before it had run its course.

Early user interface design efforts for ASR devices were focused primarily on the design of the announcements. Considerations of wording and word choice, intonation, announcement length, and pauses all contributed to maximizing the likelihood that the user would, in fact, utter only the right word at just the right time. Even small changes in an announcement's design could decrease error rates or call abandons significantly. For example, internal trial results showed that call abandon rates varied from 2 to 30 percent as a function of changes in announcements.

Data collected during early trials of applications with these constrained ASR user interfaces showed that users were unwilling to be so "well-behaved." Often utterances involved extra words, including expressions of amazement, and began well before the prompt ended, since these users were accustomed to dialing through prompts with touchtone keys.

Issue 1 (understand the user and the application), Issue 3 (allow the user to carry over as much experience as possible from one system to another), and Issue 4 (don't slow the user down) were all brought to bear when ASR technology was first introduced to users. Word-spotting and barge-in technology improvements to ASR, implemented as a result of these early trials, increased the usability of ASR-based applications—as measured by call abandon rates and transaction completion rates. They also significantly relaxed the constraints imposed on the user.

However, talking to a machine today is still not at all like talking to a person. Most ASR-based user interfaces employ a command-and-control, menu-driven approach. Users are essentially asked to provide verbal equivalents of touchtone responses to navigate across or

Panel 3. Order entry: isolated word versus advanced ASR

Isolated word		Advanced ASR	
Order system	Caller	Order System	Caller
Thank you for calling. What type bread would you like for your sandwich?	Rye	Thank you for calling What type of sandwich would you like?	Boiled ham on rye, with Swiss cheese, lettuce and mustard.
Light, dark or marble?	Dark	Light, dark, or marble rye bread?	Dark
What type of meat would you like?	Ham	Would you like tomato?	No
Boiled or baked?	Boiled		
What type of dressing–mustard, mayonnaise, commercial whip, butter or none?	Mustard		
Would you like lettuce?	Yes		
Would you like tomato?	No		
Would you like cheese?	Yes		
Swiss, muenster, cheddar, American or pepper?	Swiss		
You have ordered boiled ham on dark rye with Swiss cheese, lettuce, and mustard. Is this correct?	Yes	You have ordered boiled ham on dark rye with Swiss cheese, lettuce, and mustard. Is this correct?	Yes
Thank you.		Thank you.	

down menus. These responses are both functional and acceptable to users, but they are still not natural.

Because full, unrestricted natural language processing is still a few years off (see Rabiner[11] in this issue), current efforts to improve the naturalness of ASR-based user interfaces are beginning to focus on limited applications, such as product ordering, information retrieval, and reservations. This focus includes a detailed characterization of what a user typically does and says in the particular application—Issue 1, again.

By characterizing the naturally occurring linguistic behavior of users, and by specifying the user's intent and task, grammars can be produced that give the ASR device a much better chance at recognition because it knows what to expect from the user. Users, on the other hand, are apparently unconstrained and, thus, can speak naturally. They are also given the added sense of controlling the machine—instead of being controlled by it.

There are several ways that user-machine dialogues can be made to sound more like natural language, and two types will be illustrates here. Consider the order entry example shown in Panel 3. It portrays the difference between an isolated word interaction and an advanced ASR interaction in a simple task—ordering sandwiches from a deli. The natural language approach is a dramatic improvement over the isolated word approach and provides for an interaction that is more natural and a lot less tedious for the caller. And it clearly is sensitive to users' needs as expressed in Issue 1 (understand the user and the application), Issue 2 (ensure self-evident feature operation), and Issue 4 (don't slow down the user).

Note that the underlying grammar of the interaction in this application resembles a check-list or "fill-in-the-missing-data" routine. Once items for inclusion on the sandwich are recognized and populated in the checklist—regardless of when or how they are spoken—the system merely needs to re-prompt the user for any missing information, recognize that information, and then confirm that all recognized items constitute the intended sandwich order.

As one might expect, applications rarely have grammars that are quite so simple, and few applications are so well-bounded and predictable as this one. However, data collection and analytic methodologies of behavioral scientists are well suited to dealing with the apparent unpredictability found in natural language. Well-designed and systematic observational techniques produce data from which inductions can be made about the underlying patterns or rules governing the user's linguistic behavior.

A more demanding application, such as a pay-per-view movie selection system, would enable users to query it with questions like:

"What is the title of the movie that starred Al Pacino? Uh, he played a lawyer, I think. It was made several years ago."

If there was a match in its database, the system could respond to the key worlds "Al," "Pacino," "movie," "title," and "lawyer," with:

"And Justice for All"

Or it would list several titles that matched the attributes of the movie that the user had provided, and then prompt the user to make a selection or provide a less ambiguous attribute.

The underlying grammar of this kind of interaction would be considerably more complex than the other grammars considered so far. The ways in which people can, and would, ask about movies are surely to be much more variable than the ways in which people would list the contents of a sandwich that they order. Similar to being in a "tip-of-the-tongue" state,[12] people would bring any number of clues to the task, such as lead actors and actresses, director's name, co-stars, memorable characters, unforgettable scenes and lines (the legendary but incorrect "Play it again, Sam"), themes, messages, and settings.

To address this more complex application, an iterative design-by-use methodology typically would be used to specify and refine the underlying grammar and to test it against additional data that would be gathered in each iteration. But even such a human-factors approach would be hard pressed to fully characterize this kind of application. Yet this is the very challenge that Vision 2001 presents. As the 21st century moves closer, AT&T's customers will come to expect and demand user interfaces that permit unconstrained and "intelligent" interactions.

Future Design Challenges

The preceding sections have discussed user-machine communications in the context of traditional telephony-based applications, and the discussions have consequently focused mostly on speech technology. It is clear, however, that a revolution is under way in telecommunications that will go well beyond the fairly simple telephone call that has been evolving for over 100 years. Telecommunications in the year 2001 will include much more than simple speech communications; it will entail receiving information in a variety of forms, including some as yet unimagined, and interacting with communications systems in entirely new ways.

Our vision of multimedia communications in the year 2001 contains a rich mixture of data inputs, such as speech, music, still and motion video, and graphics, combined with simple-to-use ways for people to manipulate their otherwise complex and rich communications environment. Speech recognition, just one of many different ways for people to control the services they use, will need to fit into an environment that includes graphical interfaces that can be used with the click of the familiar mouse or the touch of a finger.

This richer environment brings with it an increase in the number and complexity of choices that must be made by the user interface designer. There will be a need to develop an understanding of what mixtures of media and user control works best for each application. There will also be a need to understand, for example, when a spoken command is superior to pressing a button, and the practical limits of a "language" of gestures made with a stylus.[13] When using a text editor, it is necessary to understand the optimal roles of the keyboard, mouse, speech, or other device, as control mechanisms. There are an almost unlimited number of such questions that must be answered through experimentation and experience.

The basic user issues discussed earlier in this paper are offered as guidelines for user interface design. It still will be essential to completely understand both the users and the applications to which they will put AT&T's products and services. Feature operations should be self-evident to the user, and consistency across systems will continue to be very important. AT&T's customers will still desire speed of operation, as well as easy ways to deal with and recover from errors. Aesthetics will assume increased importance, since there will be more opportunities to provide an interface that is not just functional and attractive, but compelling. The interface should also be fun. It should be possible to do all of these things far better than they can be done today, but the design choices will be tougher because of the increased number of options from which users will be able to choose.

The future holds many possibilities and challenges for user interface designers trying to make human conversations with machines easier and more productive. It is not enough to just ask potential users what they want. It is necessary to be more active than that, by making informed predictions that are also based on laboratory tests and field experience.

As the telephone operators of the past applied their own very sophisticated information processing capabilities to serve customers, the use of intelligent agents is, perhaps, the best way to economically return to personalized, intelligent service in the future. We must try to provide automated systems that do as well as Florence did. It won't be easy!

Acknowledgments

The authors are grateful to Harry Blanchard, Jim Dooling, Jim Farber, and Kelly Morgan for their thoughtful review of the paper.

(Manuscript approved January, 1995)

References

1. R. E. Opaluch and Y-C Tsao, "Ten Ways to Improve Usability Engineering–Designing User Interfaces for Ease of Use," *AT&T Technical Journal*, May/June 1993, Vol. 72, No. 3, pp. 75-88.
2. M. C. Day and S. J. Boyce, "Human factors in human-computer system design," In M. C. Yovits (Editor), *Advances in Computers*, Vol. 36, 1993, San Diego, Calif., Academic Press, pp. 333-430.
3. B. H. Juang, R. J. Perdue, and D. L. Thompson, "Deployable Automatic Speech Recognition Systems–Advances and Challenges," *AT&T Technical Journal*, March/April 1995, Vol. 74, No. 2, pp. 45-56.
4. B. L. Wattenbarger, R. B. Garberg, E. S. Halpern and B. L. Lively, "Serving Customers With Automatic Speech Recognition–Human Factors Issues," *AT&T Technical Journal*, May/June 1993, Vol. 72, No. 3, pp. 75-88.
5. B. L. Lively and H. Holinka, "Automating Number Report in Directory Assistance," 9th International Symposium on Human Factors in Telecommunications, 1980, Red Bank, New Jersey.
6. R. W. Bennett, "Telephony Interfaces Derived from Users' Conceptual Models." In *Proceedings of the Human Factors in Telecommunications Conference*, Torino, Italy, September, 1990.
7. D. J. Brems and B. L. Wattenbarger, "Dialog Design for Automatic Speech Recognition of Telephone Numbers and Account Numbers," 1994 International Workshop on Interactive Voice Technology for Telecommunications Applications, September 26-27, 1994; Kyoto, Japan.
8. L. C. Perlmuter, K. Scharf, K. Karsh, and R. A. Monty, "Perceived Control: A Generalized State of Motivation," *Motivation and Emotion*, Vol. 4, 1980, pp. 35-45.
9. D. Karis and K. M. Dobroth, "Automating Services with Speech Recognition over the Public Switched Telephone Network: Human Factors Considerations," *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 4, May, 1991.
10. M. Lennig, "Putting Speech Recognition to Work in the Telephone Network," *Computer*, Vol. 23, No. 8, 1990, pp. 35-42.
11. L. A. Rabiner, "Towards Vision 2001: Voice and Audio Processing Considerations," *AT&T Technical Journal*, March-April 1995, Vol. 74, No. 1, pp. 4-13.
12. R. Brown and D. McNeill, "The 'Tip-of-the-Tongue' Phenomenon," *Journal of Verbal Learning and Verbal Behavior*, 1966, Vol. 5, pp. 325-337.
13. N. A. Aykin, J. P. Cunningham, and J. P. Rotella, "Designing Operations Interfaces That Are Easy to Use," *AT&T Technical Journal*, July-August 1994, Vol. 73, No. 4, pp. 14-21

Michael J. Cosky *is technical manager of the Human Factors and User Interface Group in the Services and Speech Technology Department at AT&T Bell Laboratories in Naperville, Illinois. His group designs user interfaces that employ speech processing technologies, assesses the usability and usefulness of other emerging technologies, contributes human factors expertise to the user interface design of a variety of Network Systems–Global Public Networks' products, and provides general human factors consulting. He joined AT&T in 1981. He has a B.A. degree from the Unversity of Detroit, and an M.A. degree and Ph.D. from the University of Texas at Austin, all in psychology.*

Barry L. Lively *is technical manager of the User Interface Design Group in AT&T Consumer Products in Indianapolis, Indiana. His group is responsible for platform and product user interface design for various segments of Consumer Products. He joined AT&T in 1976. He has a B.S. degree from Pennsylvania State University in University Park, Pennsylvania, an M.S. degree from Kent State University, Kent, Ohio, and a Ph.D. from the University of Michigan, Ann Arbor, Michigan, all in experimental psychology.*

Linda A. Roberts *is a technical manager of the User Interface Design Group in AT&T Global Business Communications Systems (GBCS), Middletown, New Jersey. Her organization provides user interface design and human factors support for premises equipment, products, and platforms. She joined AT&T in 1982. She has B.Mus. and a B.Ed., both* *in music, from the University of Toronto, Toronto, Ontario; and an M.S. degree and a Ph.D., both in cognitive psychology, from Rutgers University, New Brunswick, New Jersey.*

Blake L. Wattenbarger *is a technical manager of the Consumer Services User Interface Design District at AT&T Bell Laboratories in Holmdel, New Jersey. His group provides user interface design and human factors support to Consumer Communications Services (CCS). He joined AT&T in 1970. He has a B.A. degree in physics and mathematics from Oklahoma City University, and an A.M. degree in psychology, an A.M. degree in mathematics, and a Ph.D. in experimental psychology, all from the University of Michigan in Ann Arbor.*