TOOLS AND TECHNIQUES

Building a Speech Interface to a Medical Diagnostic System

Smadar Shiffman, Alice W. Wu, Alex D. Poon, Christopher D. Lane, Blackford Middleton, Randolph A. Miller,* Fred E. Masarie, Jr.,* Gregory F. Cooper,* Edward H. Shortliffe, and Lawrence M. Fagan

Stanford University
***University of Pittsburgh**

A diagnostic system for supporting medical decisions must do more than reach the correct diagnosis. Quick Medical Reference,[1] DXplain,[2] and Oncocin[3] all provide reasonable decision support, but none of them is extensively used in routine clinical settings, partly because of the effort required to learn and to use these systems. Doctors are generally reluctant to use interfaces that require extensive typing. This resistance partially explains why existing clinical decision-support systems are not more widely used.

One important factor that can influence physicians to accept or reject a system is the quality of the user interface, as reflected by convenience and ease of use.[4] Accordingly, an interface for medical applications should

- minimize interruptions in routine patient care;
- give users a sense of control by allowing short interactions that convey the logic of how the application works; and
- use "intuitive" input techniques that avoid typing.[5]

Although systems for physicians have employed pointing devices such as light pens, touch screens, and mice, researchers have long suggested that speech recognizers would provide the most natural (and hence most acceptable) approach to clinical data entry.

Developments in speech recognition technology have made it reasonable to consider building tailored speech interfaces for medical systems. Recent applications in other domains include a system that uses speech recognition to provide telephone banking services[6] and a talk-and-point interface to an airborne warning and control system.[7] Medical products also exist, including a speech-controlled system for generating radiology reports.[8] Although these applications demonstrate the feasibility of speech interfaces, neither speech recognition technology nor language-understanding techniques have matured enough to support interfaces approaching the scope and complexity of human discourse. Current speech systems impose language restrictions on users, such as a limited vocabulary size or constrained grammars. Even if we accept these constraints, recognition in today's systems is imperfect. Ideally, a speech interface should let users communicate with the computer effectively, efficiently, and comfortably, despite restrictions and limitations imposed by the technology.

Although the underlying speech recognition technologies clearly need further basic and applied research, current methods do invite experiments on incorporating speech into data-management and decision-support systems. We decided to assess how best to integrate existing commercial speech systems with computing environments designed for physicians. We examined the performance and roles of both

0885/9000/91/0200-0041 $1.00 © 1991 IEEE

A speech interface should let users communicate with the computer effectively and comfortably, despite restrictions imposed by the technology.

continuous-speech and single-word recognition products. Our initial goal, and the one that continues to dominate our experiments, was to develop a speech interface to a medical diagnostic system that is easy to use and hence appealing to physicians.

Previously, we experimented with graphics and continuous speech using Oncocin, a decision support tool developed at Stanford for clinical oncology.[9] One aspect of this work was to build a speech interface to a spreadsheet-like display of a patient's medical record during treatment. Underlying the graphic display is an expert system for determining appropriate adjustments to complex treatment regimens. Although we built a speech-based interface to that program, we realized that spoken input is best used when the input terminology is too complex to be navigated with short graphical menus.[10,11]

We used the lessons of that work to design an interface for QMR-DT,[12] an evolving decision-theoretic version of Quick Medical Reference, which performs diagnostic reasoning about diseases in internal medicine.[13] QMR can be used as an electronic textbook, a low-level consultant, or a diagnostic consultant for complex cases. Both QMR and QMR-DT employ a knowledge base and terminology derived from a large and well-known predecessor, Internist-1.[14] QMR-DT encompasses the part of QMR's functionality that provides differential diagnoses for a set of patient characteristics, but it uses a different algorithm to compute diagnoses.

Our work includes two programs that integrate off-the-shelf speech technology with programs that manipulate medical terminology. The Term Identifier program uses an isolated-word, speaker-dependent speech product to provide an interface for entering medical findings into QMR-DT. Frame Browser is an auxiliary program used by the developers of Term Identifier to examine frame structures (which were later used in Term Identifier). We also used Frame Browser to experiment with a continuous-speech system.

Our interface design was constrained by requirements from three sources:

(1) the function of the diagnostic system (namely, to provide possible diagnoses for a set of given patient characteristics),
(2) the nature of medical terminology, and
(3) the limitations of speech technology.

We designed the interface to facilitate the system's basic functions while adequately representing the complexity of medical terminology to permit natural interaction.

Current speech technology

The speech systems available for integration with application products differ in various ways.[15] Speaker-independent systems recognize speech from any new speaker, whereas speaker-dependent systems require a special session for training the system first. Although training can be tedious for users, it enables adaptation to user-specific attributes and, therefore, improves the recognition rate.

All systems use some kind of speaker model that characterizes the speech that can be recognized. Speaker-independent systems typically use generic models, while speaker-dependent systems use user-specific models. Storing data for each speaker can require considerable disk space compared to that used by generic speaker models.

Speech systems also differ in the amount of continuity they allow in the input speech. Some systems allow continuous speech, whereas others require that the user utter isolated words, one at a time. (One-word phrases or compound terms are typical, but short, multiple-word phrases are also possible.) Continuous speech is more difficult to process than isolated-word speech: Word boundaries are difficult to identify, speech segments affect one another, and function words (articles, prepositions, and pronouns) are usually articulated poorly. However, continuous-speech recognition lets users speak to the computer naturally; speaking a single word at a time is less natural.

The vocabularies supported by speech systems vary in size from 10 words to more than 35,000 words. Using a large vocabulary causes difficulties in maintaining accuracy, but using small vocabularies can impose unwanted restrictions on the naturalness of the communication. In addition, for speech systems that use a grammar, increasing the number of constraints imposed by the grammar increases the accuracy of word recognition but reduces the naturalness of the communication. Typical grammars include finite-state networks of allowable sentences (based on phrase structure rules) and trigram grammars (grammars that estimate the probability of a word's being in a sentence, given the two previous words).

Design considerations

We were aware that adding new modalities to an existing interface might decrease system integrity or create integration discontinuity.[16] In our experience, when the Voice Navigator speech system is linked to common Apple

Macintosh applications, the resulting interface can be less comfortable to use than the original graphic interface. Specifically, it is difficult to control mouse operations — moving the cursor to a certain location on the screen, for example — with speech. Developers can achieve good interface integrity by addressing the requirements imposed by the interaction modality (speech, in our case) at the beginning of the design process. We chose to develop a speech interface for a program that had no interface as yet (QMR-DT), so that we could consider speech-related issues from the start.

QMR-DT performs inference on a probabilistic reformulation of the Internist-1 knowledge base, updated in 1985.[14] This knowledge base includes more than 500 diseases, around 3,500 patient findings (including physical findings, test results, and demographic descriptors), and many links defining causal, temporal, and logical interrelationships between diseases. Because QMR-DT had been conceived as an experiment in adapting QMR and its ad hoc scoring scheme to use more formal probabilistic techniques, its developers had spent almost no effort building an attractive interface to it. When they performed experiments, they ran the program in batch mode. As with Internist-1, the input to the program is a set of findings, and the output is a differential diagnosis of the leading disease hypotheses, ranked by their probabilities. Since QMR-DT would eventually require a physician's interface but brought to our collaboration no preconceived notions about its form, it gave us an ideal opportunity to build a speech interface from scratch using existing speech technologies.

We needed a speech-oriented method that would enable easy input of complex finding names, such as *stomach endoscopy longitudinal laceration cardiac location*. This task is difficult because each medical decision support program needs a controlled vocabulary but a physician can specify a single medical term in many ways.[17] The absence of a standard medical language requires users to speculate what terms the system knows. Therefore, an interface to a medical diagnostic program should recognize various alternate expressions for medical concepts in the knowledge base and should identify (for user confirmation) the intended finding for any expression entered. Without adequate training, users might think that free-form natural language input is possible. However, all speech interfaces operate under restrictions that are dictated by a specific vocabulary, and some systems require predefined word ordering. Because of the current status of speech recognition technology, speech systems typically attain 97-percent accuracy at most in word recognition.[15]

A speech interface should compensate for these limitations by guiding user interactions. The nature and form of such guidance became the primary focus of our investigations. We used an underlying organization of relevant medical terms to adequately represent the complexity of medical expression while limiting possible user interaction. Medical terms typically encompass several concepts. For example, words describing clinical observations generally include the body part associated with the observation, the duration of the observation, the observed severity, and so on. We based our organization scheme on concepts that characterize findings both explicitly and implicitly.

Continuous speech is more difficult to process, but systems that require the user to speak a single word at a time are less natural to use.

The classification scheme

We examined several classification schemes that could provide orientation stages for the input process: the QMR finding hierarchy, which is used for navigating through QMR to locate a desired finding;[18] the Current Medical Information and Terminology classification;[19] the Unified Medical Language System semantic network;[20] and frame data structures developed by the University of Pittsburgh's UMLS project to describe Internist-1/QMR manifestations.[21] We chose to use UMLS frames to orient and guide speech input because of their direct relation to the findings stored in the Internist-1 knowledge base. We used these frame data structures to tailor an intuitive finding classification to the needs of a speech interface.

Pittsburgh's UMLS project defined a set of knowledge frames to facilitate the electronic translation of terms between medical vocabularies. The frames can be used to encode general medical terminology in a standardized format.

The frame data structures were defined at two levels of representation: generic frames (also known as concept frames) and instantiated frames. Generic frames provide templates for describing general information about medical findings in an organized format. Specifically, these structures define the set of allowable attributes and values used in the descriptions of findings in the Internist-1, QMR, Help,[22] and DXplain vocabularies. For example, the generic-frame structure for the concept *abdominal pain* contains all the information that a physician might use to describe or modify the concept.

Instantiated frames are organized descriptions of specific Internist-1/QMR findings. These frames contain only those portions of the generic frame template that are relevant to a specific finding. For example, the instantiated frame *abdomen pain acute* contains only the information that is relevant to the duration of abdominal pain. Some of the

Generic frame

Abdominal pain

Generic frame: Last edited on 7/17/1987 by Masarie

Allowable status:	Presence or absence
Normal status:	Absent
Site:	Abdominal topographical site

Method(s)

Name:	Gastrointestinal symptom
Reliability:	3

Qualifier(s)

- Severity
- Time duration qualitative
- Time duration quantitative
- Influence on abdominal pain
- Abdominal pain radiation
- Abdominal pain quality
- Pattern of occurrence

Legal values for two qualifiers

Time duration qualitative

Exactly one of:

- Acute
- Subacute
- Chronic

Pattern of occurrence

Exactly one of:

- Intermittent
- Recurrent
- Continuous
- As single occurrence
- Transient
- Paroxysmal
- Persistent
- Progressive

Figure 1. The generic frame *abdominal pain* and the legal values for two of its qualifiers.[21]

attributes that modify generic concepts are the presence or absence of a finding, the location where the finding was detected, the methods used to detect the finding, and the quality of the finding.

Figure 1 shows the generic frame *abdominal pain* and the legal values for two of its qualifiers. Figure 2 shows two instantiations of the generic frame. The status that designates abnormal situations (*presence* or *absence*) is *present* in both instantiations. The method for detecting both findings is having the patient report a *gastrointestinal symptom*. The site for the finding *abdomen pain epigastrium recurrent attack <s> hx* is the *epigastrium*, which is characterized in the generic frame as an *abdominal topographical site*. The finding *abdomen pain acute* has no specific location associated with it. The qualifier or attribute that describes the finding *abdomen pain acute* is *time duration qualitative* with the value *acute*. The qualifier that describes the finding *abdomen pain epigastrium recurrent attack <s> hx* is *pattern of occurrence* with the value *recurrent*.

Frame Browser

Before we could build an effective interface to QMR-DT, we

- learned about the vocabulary that can be used in describing medical findings,
- examined ways for frame structures to help users specify Internist-1 terms, and
- learned about the characteristics of and relationships among frames.

We created the Frame Browser tool to help with these tasks. We also used it for experimenting with a continuous-speech system. Users can manipulate the program by mouse and keyboard, by speech, or both.

Frame Browser runs on a Next workstation interfaced to a Speech Systems, Inc. DS200 system. The SSI system recognizes continuous speech and is speaker independent. It uses several generic speaker models (for example, male and female models) and does not require a training session for each user. The system has a vocabulary of more than 38,000 words (including root forms and inflections) and requires a dictionary and a grammar for each set of sentences it is expected to recognize. Our configuration includes one dictionary (about 700 words) and three grammars that are switched according to the context.

Frame Browser displays generic and instantiated frame structures on request. Users can utter a subset of more than 500 frame names into the program. This number is limited by the medical terms appearing in the dictionary that SSI supplies with its equipment (see Figure 3 for a sample).

Users enter speech into the system by speaking into a headset while pressing a button. The program first checks whether a command was uttered. Saying the name of a command activates it and opens all the menus leading to that command if they are not open already. A command does not have to appear on the screen as a menu option to be activated by speech.

If the program does not identify an utterance as a command, it interprets it as a frame name. The program determines whether an utterance refers to a generic frame or an instantiated frame according to the active window. Utterances that designate medical findings cause the corresponding frame to be displayed. For example, after the user says "Abdominal pain," the frame for *abdominal pain* appears in the active window.

In our preliminary testing, a physician and a nurse spoke more than 500 frame names into the program in a fairly quiet environment. The speakers read the names once from a list they had not seen before the testing session. They waited an average of 0.7 seconds for the program to decode each utterance, and about 90 percent of the utterances were recognized correctly. We did not notice any significant difference in the recognition of medical terms compared to standard English words. While users can employ continuous speech to enter findings with a reasonable recognition time, the accuracy rate suggests that we must pay special attention to resolving cases of misrecognized utterances.

Our work on continuous speech entry assumes that users know the specific wording for any finding in the knowledge base. However, this is rarely true in real clinical situations, especially for new users. We are now investigating input techniques in which the user does not have to know the exact expressions for all findings.

The developers of Term Identifier used Frame Browser to quickly associate generic and instantiated frames and to easily access the legal values of frame attributes. The program was also useful in assessing how well given generic-frame fields differentiate among instantiations of the generic frame. For example, a generic-frame field that has a similar value in many instantiations, like the field *status* with the value *present* (see Figure 1), is not a good discriminator. A field that characterizes few instantiations, like the field *pattern of occurrence*, or a field that has a different value for each instantiated frame, are good discriminators.

Instantiated frames

Abdomen pain acute

Instantiated frame: Last edited on 10/14/1987 by Masarie
Concept name: Abdominal pain
Status: Present

Method: Gastrointestinal symptom
Reliability: 3

Qualifier(s)
Time duration qualitative: Acute

Abdomen pain epigastrium recurrent attack <s> hx

Instantiated frame: Last edited on 7/29/1987 by Masarie
Concept name: Abdominal pain
Status: Present
Site: Epigastrium

Method: Gastrointestinal symptom
Reliability: 3

Qualifier(s)
Pattern of occurrence: Recurrent

Figure 2. Two derivative instantiations of the generic frame *abdominal pain*.[21]

The Term Identifier interface

We designed the Term Identifier program as an interface to QMR-DT so that users could enter findings quickly and naturally. The program's diagnostic engine takes as input a list of Internist-1 findings and produces as output a differential diagnosis. One difficulty in entering findings is that the Internist-1 knowledge base has only one wording for each of the 3,5000 findings. Term Identifier guides users in entering these findings. The interface is especially useful in cases where users know little about the way a finding is represented in the Internist-1 knowledge base.

Term Identifier runs on a Macintosh under the Hypercard development tool. It uses Voice Navigator XA by Articulate Systems, an isolated-word, speaker-dependent system. The program requires a training session for each user, including several repetitions of each utterance that Term Identifier is expected to recognize. Voice Navigator XA allows a vocabulary of 1,000 utterances at a time. Like Frame Browser, Term Identifier can be manipulated by mouse and keyboard, by speech, or both.

Presently, the program's database of findings is limited to the set of instantiated frames. Thus far, only about 1,200 of the 3,500 Internist-1 findings have corresponding instantiated frames defined for them; therefore, the set of findings that Term Identifier can handle is limited to that number. However, once the remaining instantiated frames are defined, Term Identifier will be able to import the frames

Generic frame names

Somnolence
Spleen calcification
Spleen size
Stature short
Stomach aspirate fungus culture
Stomach aspirate gross inspection
Stomach external pressure deformity of imaging technique
Straight left heart border
Subclavian bruit
Sudden death family history
Temperature

Instantiated frame names

Fever
Fever intermittent
Growth retardation
Headache severe
Heart catheterization aortic pulse bifid
Heart catheterization pulmonary venous drainage into right heart
Heart catheterization right atrium v wave increased
Heart output decreased
Heart output increased
Heart radioisotope scan pericardial density
Hoarseness
Jaundice

Figure 3. Sample frame names that can be entered into Frame Browser using speech.

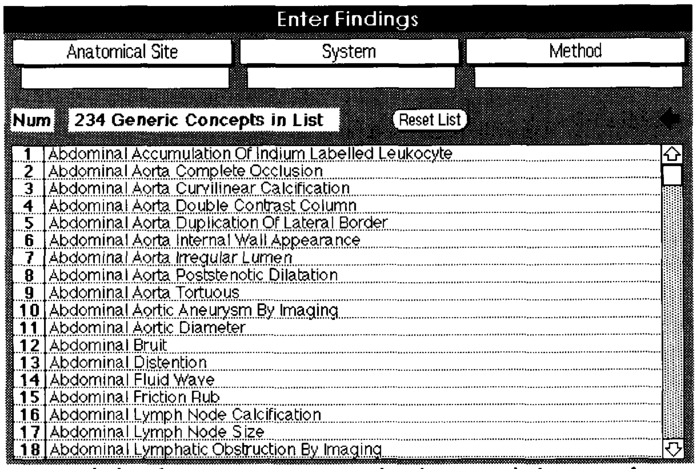


Figure 4. The list of generic concepts presented to the user at the beginning of an input session.

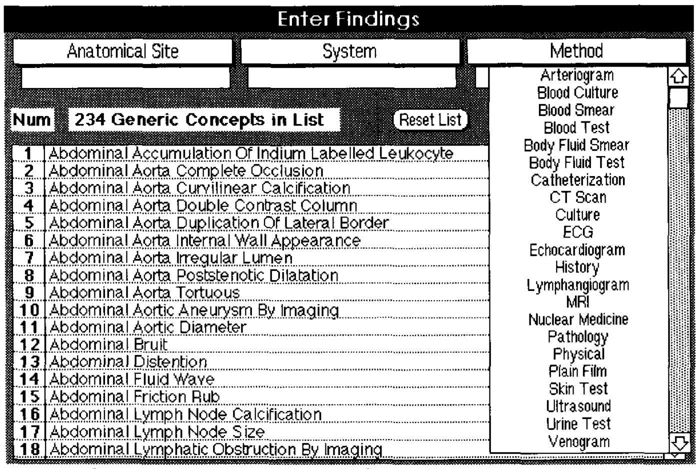


Figure 5. The methods known to Term Identifier.

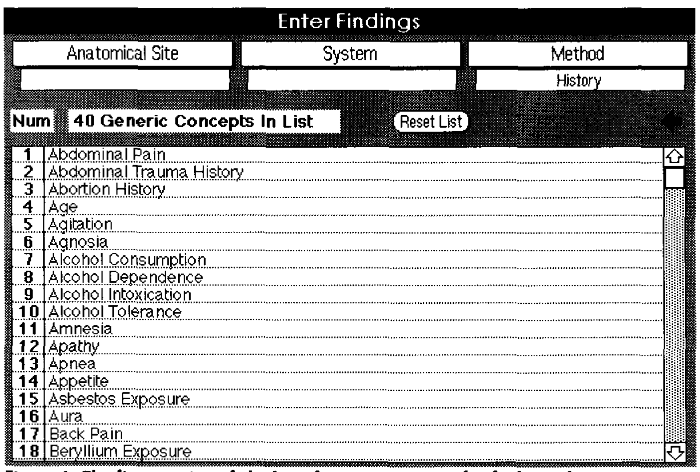


Figure 6. The first portion of the list of generic concepts for findings that are detected by taking a patient's history.

directly and thus encompass the full set of Internist-1 findings.

Beginning with all 1,200 findings, Term Identifier reduces this set progressively as users provide information about the finding they are looking for. To help users add information about a finding, the program uses a classification based on the frame data structures discussed earlier. The reduction process narrows the scope of the findings presented to users in two stages. The first stage helps users select an appropriate generic concept for the finding. The second reduction stage helps users select a finding within that generic concept.

At the beginning of an input session, the program displays the complete list of generic concepts, as shown in Figure 4. The list reduces as the user provides information about the finding: the *anatomical site* of the finding, the *system* likely to be affected by the finding, and the *method* by which the finding is discovered. At most, the user will have to provide all three items, but providing just one or two items will usually reduce the list enough to select a generic frame directly.

An example. Let's walk through the steps for specifying the finding *abdomen pain epigastrium unrelieved by antacid* using speech input (the example can also be performed with a mouse or keyboard). We can reduce the full list of generic concepts by indicating the medical technique by which the finding is discovered (for example, through physical examination of the patient or through a laboratory test). By saying "Method," we obtain the list of methods recognized by Term Identifier, as shown in Figure 5. Since abdominal pain symptoms are likely to be discovered by a patient's own complaints, it is appropriate to say "History" at this point. The program then finds all the generic frames that can be discovered by going over a patient's history. Many findings have more than one method of detection; thus, for some generic concepts, any of several methods could be indicated. In this case, saying "History" makes the program display a list of 40 generic concepts, part of which is shown in Figure 6.

We can reduce the list further by specifying the anatomical site where the finding occurs. If we say "Abdomen," Term Identifier will discern that an anatomical site has been specified, as opposed to a system or method. If the utterance is ambiguous (that is, if it is applicable to several categories), each choice is highlighted on the display, and we must say one of the category names. In this case, the word "abdomen" suits only one category, and the list now reduces to three generic frames. Figure 7 shows the reduced list.

The first stage of the reduction process is complete when we select the generic concept that best represents the finding. In this example, we say "Select one" to select the first line on the list — *abdominal pain*. Once the generic concept is selected, the frame for the concept is displayed (see Figure 8). The attributes that form the generic frame are displayed at the top.

In the second reduction stage, the program represents findings as instantiations of the generic-concept frames. Most generic frames cover five or fewer instantiated frames, and the user can just choose from the short list. However, some generic frames have a long list of instantiations, making a choice difficult. For example, the generic frame *abdominal pain* has 30 instantiations. We can reduce the list by supplying more information about the finding. When we select one of the generic-frame concepts by saying "Epigastrium," the program reduces the list for *abdominal pain* to only five findings, as shown in Figure 9. We choose a finding from the new list by saying "Select," followed by the appropriate line number.

Entering a finding such as *abdomen pain epigastrium unrelieved by antacid* requires five utterances. However, in many cases where the list of instantiated frames is short, the second reduction stage is immediate, so entering a finding can take as few as three short utterances. Having selected a finding, we specify whether it is positive or negative (that is, whether or not it occurs in the patient — it is often important to know that a specific key finding does not occur in a patient). We can also view the list of diseases that can be associated with the finding. After specifying a set of findings, we can request from QMR-DT a differential diagnosis for the case.

Advantages and limitations. In our testing of Term Identifier, two physicians entered about 100 findings into the program in a fairly quiet environment. During two tests, they waited an average of less than 0.5 seconds for an utterance to take effect.

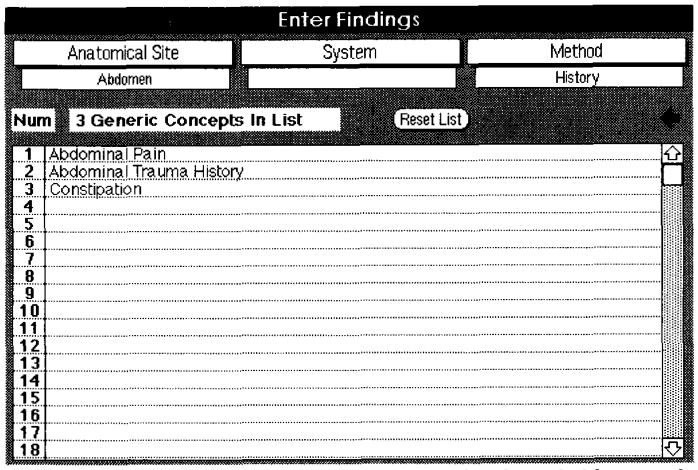


Figure 7. The list of generic concepts after the user specifies the method *history* and the anatomical site *abdomen*.

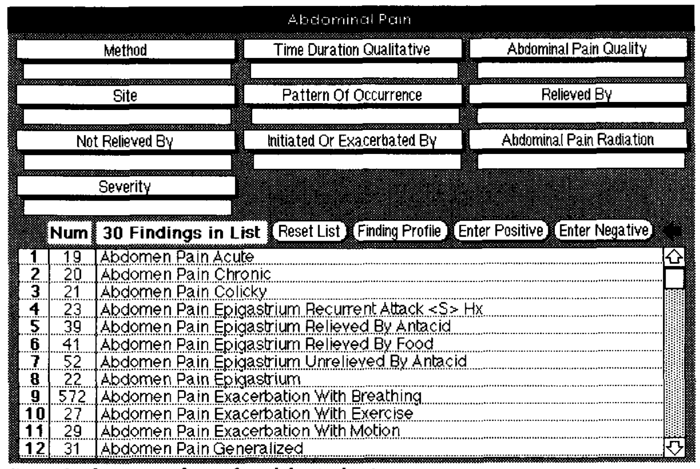


Figure 8. The generic frame for *abdominal pain*.

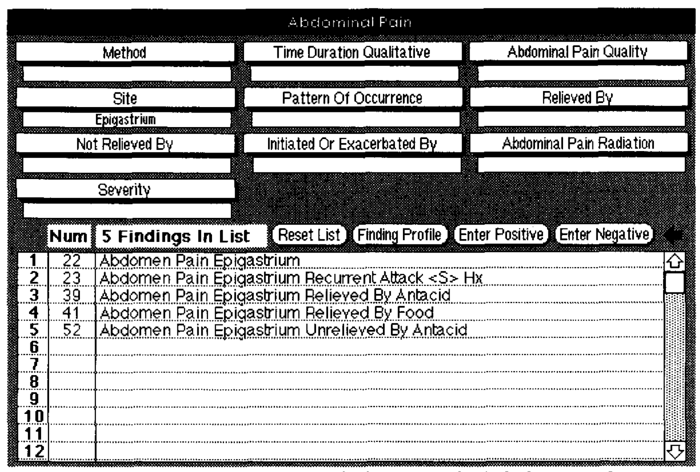


Figure 9. The final list of abdominal pain findings, out of which the user selects a finding.

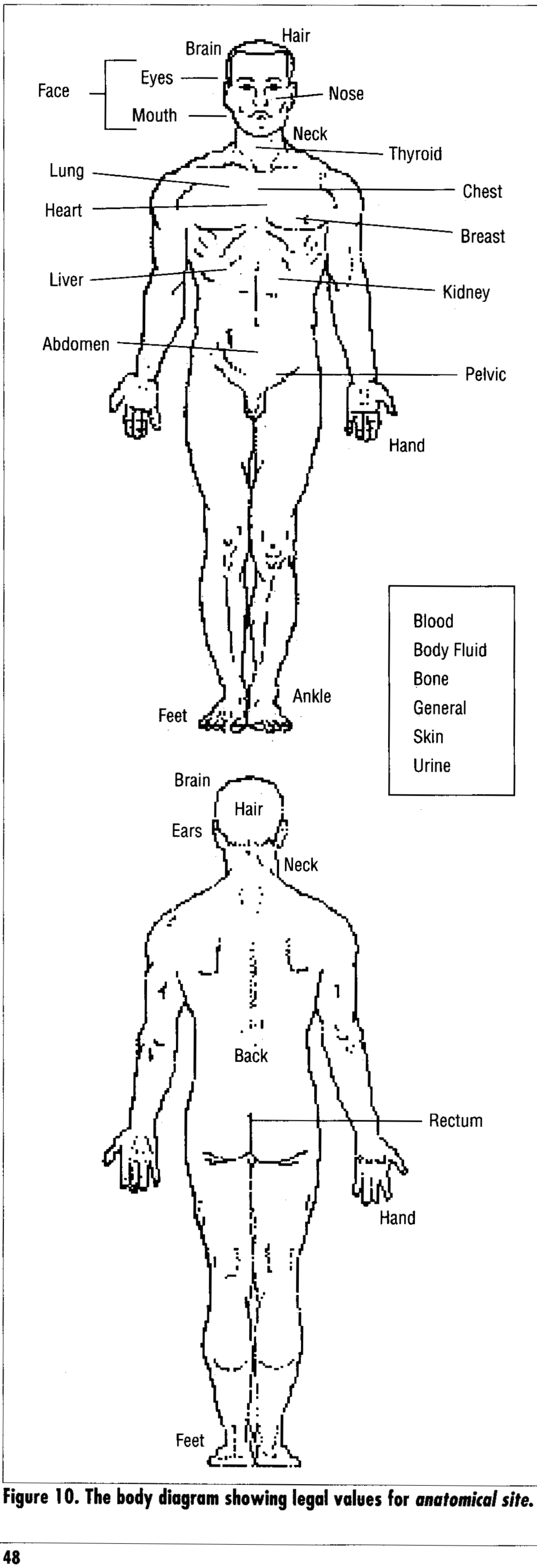


Figure 10. The body diagram showing legal values for *anatomical site*.

The first test measured the accuracy of the system's speech component. The physicians entered more than 70 findings, specifying each finding once by saying a predefined sequence of utterances (which represented a path for a finding). In 83 percent of the cases, the interaction was smooth, and all utterances were recognized correctly. Of all the utterances (more than 250) used to specify the full set of findings, nearly 90 percent were recognized correctly.

After the first test, the physicians spent some time getting acquainted with the system. We asked the physicians to enter 20 findings for which a path was not predefined, and we noted each path as the physicians tried to find it.

The second test checked how easy it was for a user to get to a given finding. We gave the physicians a case description from which they had to extract and enter 12 finding names. It took about five steps to specify a finding.

Experienced users might find it more natural to say finding names directly, bypassing the two reduction stages described above. However, Term Identifier cannot recognize finding names because most are long phrases, and the Voice Navigator only accepts utterances of eight continuous syllables at most. The scope of medical terms that Term Identifier can handle might be expanded to include predefined and custom-made synonyms for various finding names.

There is a trade-off between the number of steps needed to specify a finding and the number of values out of which a user selects an item to reduce the set of possible findings. If the list of legal values for an attribute is small, each value might refer to a large number of findings, making the final choice difficult. Another level of selection might be necessary to narrow the set of findings. When the list of legal values is large, each value typically refers to a small set of findings, although it might be difficult to select a value.

Long lists are useful for speech interaction because the number of reduction steps can be minimized while maintaining the possibility of quick item selection. If the items on a long list are displayed mnemonically, users can identify options easily and select an item quickly by saying its name. In contrast, using a mouse to select an item from a long list might be less comfortable if the user has to drag the mouse a long way on the screen. We preferred to minimize the number of steps needed to specify a finding, and to have long lists of specific values presented in a clear and manageable way to allow easy selection. For example, the list of values for *anatomical site* is presented on a body diagram, where legal values are associated with locations (see Figure 10). We are also pursuing ways to represent other lists of legal values in a way that makes selection easy.

The Term Identifier interface lets users enter findings easily into the QMR-DT program. We must still evaluate the interface's value in clinical settings in a formal study. Although the interface allows reasonably smooth communication with the diagnostic program, a continuous-speech interface might allow more natural communication. Our work with Frame Browser indicates that a continuous-speech interface could be developed for the diagnostic system to let experienced users to say finding names directly.

We plan to develop an interface to QMR-DT that will let users enter findings using natural-language utterances. The interface will have to map the wide range of allowable input phrases to Internist-1 findings. Limitations in the current technology would restrict input to only a subset of the English language. To avoid wasteful input efforts, the interface will have to convey to users what kind of utterances are acceptable in terms of content and phrase structure.

We believe that speech interfaces will eventually perform well and be affordable to a degree that will make them an attractive solution for encouraging the use of clinical decision support systems. Since some inaccuracy in the recognition process is inevitable, however, we are also exploring the use of intuitive graphics to resolve errors. The integration of speech and graphics holds great promise for engineering effective human interfaces to decision support tools.

Acknowledgments

We thank Nomi Harris and Clifford Wulfman for their assistance in building portions of the Term Identifier program. We thank Kevin Johnson, John Egar, and Jeanette Polascheck for their help in testing the programs. We also thank Tom Rindfleisch and Edward Feigenbaum for their help in making this project possible.

This work has been supported by the National Library of Medicine (LM-04864 to Stanford University, RCDA K04-LM-00084 to Randolph Miller), and by Sumex core research (LM-05208). The UMLS research was supported by NLM contract N01-LM-6-3522 to the University of Pittsburgh and the University of Utah. Computer facilities were provided by the Sumex-Aim Resource for Artificial Intelligence in Medicine and through equipment grants from Speech Systems, Clinical Information Advantages, and Apple Computer.

References

1. R.A. Bankowitz et al., "A Computer-Assisted Medical Diagnostic Consultation Service: Implementation and Prospective Evaluation of a Prototype," *Annals of Internal Medicine*, Vol. 110, No. 10, 1989, pp. 824-832.

2. G.O. Barnett et al., "DXplain: An Evolving Diagnostic Decision-Support System," *JAMA: J. Am. Medical Assoc.*, Vol. 258, No. 1, 1987, pp. 67-74.

3. D.M. Hickam et al., "A Study of the Treatment Advice of a Computer-Based Cancer Chemotherapy Protocol Advisor," *Annals of Internal Medicine*, Vol. 103, No. 6, 1985, pp. 928-936.

4. L.E. Perreault and G. Wiederhold, "System Design and Evaluation," in *Medical Informatics: Computer Applications in Health Care*, E.H. Shortliffe et al., eds., Addison-Wesley, Reading, Mass., 1990.

5. E.H. Shortliffe, "Computer Programs to Support Clinical Decision Making," *JAMA: J. Am. Medical Assoc.*, Vol. 258, No. 1, 1987, pp. 61-66.

6. R. Nakatsu, "Anser: An Application of Speech Technology to the Japanese Banking Industry," *Computer*, Vol. 23, No. 8, Aug. 1990, pp. 43-48.

7. M.A. Salisbury et al., "Talk and Draw: Bundling Speech and Graphics," *Computer*, Vol. 23, No. 8, Aug. 1990, pp. 59-65.

8. A.H. Robbins, D.D. Horowitz, and M. Srinivasan, "Speech-Controlled Generation of Radiology Reports," *Radiology*, Vol. 164, No. 2, 1987, pp. 569-573.

9. E.H. Shortliffe, "Medical Expert Systems — Knowledge Tools for Physicians," *Western J. Medicine*, Vol. 145, No. 6, 1986, pp. 830-839.

10. C.E. Wulfman et al., "Three Experiments in Applying Continuous-Speech Recognition Technology to Oncology Record Keeping," Tech. Report KSL-90-67, Knowledge Systems Laboratory, Stanford University, Palo Alto, Calif., 1990.

11. E.A. Issacs et al., "Designing a Spoken Interface for Medical Decision-Support Systems," Tech. Report KSL-90-69, Knowledge Systems Laboratory, Stanford University, Palo Alto, Calif., 1990.

12. M. Shwe et al., "A Probabilistic Reformulation of the Quick Medical Reference System," *Proc. 14th Conf. Computer Applications in Medical Care*, IEEE Computer Society Press, Los Alamitos, Calif., 1990, pp. 790-794.

13. R.A. Miller et al., "The Internist-1/Quick Medical Reference Project — Status Report," *Western J. Medicine*, Vol. 145, No. 6, 1986, pp. 816-822.

14. R.A. Miller, H.E. Pople, and J.D. Myers, "Internist-1: an Experimental Computer-Based Diagnostic Consultant for General Internal Medicine," *New England J. Medicine*, Vol. 307, No. 8, 1982, pp. 468-476.

15. K.F. Lee, *Automatic Speech Recognition: The Development of the Sphinx System*, Kluwer Academic Publishers, Boston, 1989.

16. C.E. Wulfman et al., "Integration Discontinuity: Interfacing Users and Systems," Tech. Report KSL-88-12, Knowledge Systems Laboratory, Stanford University, Palo Alto, Calif., 1988.

17. B.L. Humphreys and D.A.B. Lindberg, "Building the Unified Medical Language System," *Proc. 13th Conf. Computer Applications in Medical Care*, IEEE Computer Society Press, Los Alamitos, Calif., 1989, pp. 475-480.

18. R. de Bliek, F.E. Masarie, Jr., and R.A. Miller, *QMR User Manual*, Univ. of Pittsburgh, Pittsburgh, Penn., 1988.

19. *Current Medical Information and Terminology*, American Medical Association, Chicago, 1981.

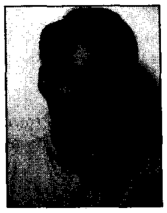
20. A.T. McCray, "The UMLS Semantic Network," *Proc. 13th Conf. Computer Applications in Medical Care*, IEEE Computer Society Press, Los Alamitos, Calif., 1989, pp. 503-507.

21. F.E. Masarie and R.A. Miller, "UMLS Progress Report, Mapping of Medical Knowledge Representation: Internist-1, Help, and Mesh, Unified Medical Language System," tech. report, Division of Extramural Research, Nat'l Library of Medicine, Bethesda, Md., 1987.

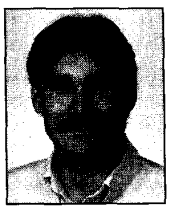
22. T.A. Pryor et al., "The Help System," *J. Medical Systems*, Vol. 7, No. 2, 1983, pp. 87-102.

Smadar Shiffman is a scientific programmer at the Section on Medical Informatics at Stanford University. Her interests include user interface design, integrating speech recognition technology into user interfaces to medical systems, and natural language processing.

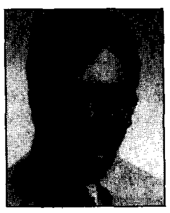
Alice W. Wu is currently pursuing her MS in computer science at Stanford University. Her concentration is in software systems design, with emphasis on user interfaces and human-computer interaction.



Alex D. Poon is an undergraduate at Stanford, where he will receive a BS in computer science in 1991. His current research involves comparing single-word to continuous-speech recognition for gathering computerized medical data.



Christopher D. Lane is a member of the technical staff for the Sumex-Aim resource at Stanford University. His work focuses on intercomputer communication in support of medical computing activities.

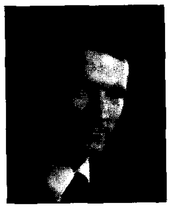


Blackford Middleton is a fellow in the Division of General Internal Medicine and an MS candidate in Health Services Research at the Stanford University School of Medicine. His research interests include designing and evaluating medical decision support tools.

Randolph A. Miller heads the Section of Medical Informatics and teaches medicine at the University of Pittsburgh. He is also principal investigator for the university's UMLS Project and director of the Training Program in Medical Informatics, sponsored by the National Library of Medicine.

Fred E. Masarie, Jr., was on the research faculty of the School of Medicine at the University of Pittsburgh from 1986 to 1990. He is now helping to start the Camdat Corporation, a medical decision support software company in Pittsburgh.

Gregory F. Cooper is assistant professor of medicine in the Section of Medical Informatics in the University of Pittsburgh's Department of Medicine. His research focuses on methods for using decision theory to develop expert systems that assist in medical decision making.



Edward H. Shortliffe, professor of medicine and computer science, heads both the Division of General Internal Medicine and the Medical Information Sciences Training Program at Stanford's School of Medicine. His research interest is in developing integrated decision support tools for clinicians.

Lawrence M. Fagan is a senior research scientist in the Section on Medical Informatics at Stanford and associate director of the Medical Information Sciences Training Program. His interests include the design of medical decision support systems and interfaces, therapy planning, and temporal reasoning.

The authors can be reached in care of Smadar Shiffman, Section on Medical Informatics, Medical School Office Building 215, Stanford, CA 94305-5479; e-mail shiffman@sumex-aim.stanford.edu.