Home Monitoring of Hypertensive Patients through Intelligent Dialog System

Ivano Azzini, MSc, Toni Giorgino, MSc,
Silvana Quaglini, PhD, Carla Rognoni, MSc, Mario Stefanelli, PhD
Medical Informatics Laboratory,
Dept. of Computers and Systems Science, University of Pavia, 27100 Pavia, Italy
Daniele Falavigna, MSc, Roberto Gretter, MSc
Istituto per la Ricerca Scientifica e Tecnologica
ITC – irst, 38050 Panté di Povo, Trento, Italy

Abstract

Recent advances in automatic speech recognition and related technologies allow the computers to carry on conversations by telephone. We based on the knowledge represented in a widely accepted guideline for the care of hypertension to develop a system that allows physicians to manage a complete medical record for hypertensive patients. The care delivery professional can enter, manipulate and review the record through a conventional GUI. Patients are allowed to dial a toll-free number and enter a subset of the data by themselves. They interact with an automatic speech recognition and synthetisation engine whose behavior is determined by an intelligent agent based on their condition and treatments, which are stored in the database. The aim of the dialog is to allow patients to efficiently enter the data that need more frequent monitoring, such as blood pressure values and habits. The dialog system is flexible towards a user-friendly and adaptive interaction.

1. Introduction and setting

The telephone has been used in medicine since its appearance [1], but only in the last years its use has been integrated with the most recent telecommunication tools. References [1] and [2] describe the use of Dual Tone Multi Frequency (DTMF) systems and similar technologies as an alternative and supplement to a visit performed by a physician.

The success of these systems, particularly in checking and advising patients with hypertension, is described in [3,4]. The natural evolution of DTMF systems in this domain is the most recent mixed initiative spoken dialogue system [5]. Reference [6], for example, describes an attempt in this direction. The challenge of this innovative methodology consists in the design and development of an intelligent (adaptive) dialogue system (IDS) that collaborates with a user for the accomplishment of a task [7].

Following the motivations exposed above, in the framework of the E.U. project "Homey" we built an intelligent (adaptive) dialogue system to manage and monitor patients with essential hypertension. We have chosen a group of patients in care at the "San Matteo Hospital" in Pavia, Italy; each patient would normally have to meet his physician approximately every 2-4 weeks. The goal of the periodical examination is to monitor blood pressure values, habits, and other variables, and estimate certain standard risk indicators. The physician takes account of these results in order to prescribe or modify the pharmacological therapy and to possibly prompt the patient to make changes to his life style.

In the dialog system proposed, the patients periodically call a dedicated telephone number and engage a dialogue with the system, which talks and interacts with them to acquire clinical data, monitor his style of life and ask about the presence of side effects. We tried to keep the dialogue as close as possible to the interaction between a physician and a patient; it also gives advice, by issuing alerts and prompts as appropriate, to keep her (or maintain her) in a class of low risk. As suggested in [8], we have extracted our domain knowledge from a set of world-widely accepted guidelines for the hypertension and dyslipidemia [9, 10, 11]. The system is interfaced to a database that records the detailed medical history of the patients being followed.

We have also formulated a mathematical model (which could be considered as a generalization of the one presented in [12]) that describes and legitimates the existence of our dialogue systems; nevertheless, in this paper we have set the attention on the architecture and the process that we followed for the development of this medical IDS system.

2. Architecture

Figure 1 depicts a simplification of the data flow inside our application. Two actors are allowed to enter data into the supporting database. In the first place, the physician uses a conventional (graphics, keyboard and mouse) interface to store and update data in a database.

On the other hand, the patient is also allowed to enter these data that she can acquire at home by herself and transfer it to the care providing centre, by the means of a telephone.

To accomplish this task, the patient is instructed to dial a toll free number once every a chosen amount of time. When this happens, she is connected to a call center; the dialog is then directed by the *dialog manager* of the SPINET speech recognition system [13]. The dialog manager executes a *dialog description* written in a high level language and reads a *state vector*, which encodes the patient's status. The interaction with the user is then carried on in accordance to both the state vector and the dialog description.

The state vector is represented by a set of discrete variables; some of them control the dialog flow (e.g. if the user is not a smoker, she will not be further asked about the number of cigarettes she smokes). Other fields are output variables, which may be strings, integers, or Boolean values. When the telephone is finally hung up, the values of the output variables reflect what the user has uttered during the call.

The state vector and the database at this point have to be synchronized and prepared for the next call. This is achieved in two steps: when a call ends, the state vector is parsed and its content is translated and stored in the database. The records inserted in this way by telephone calls are stored along with those taken during visits, so that the physician is able to review them. The values entered by the patient are distinguished by those taken during encounters because a marker appears near them on the graphical interface (figure 2).

On the other hand, the database holds some values that should affect the dialog (e.g.: whether the patient has been prescribed to follow a diet, or the date of the next visit). When a new call is set up for a certain patient, these values are extracted from the database and the corresponding state vector is prepared. We have given the name "adaptation agent"

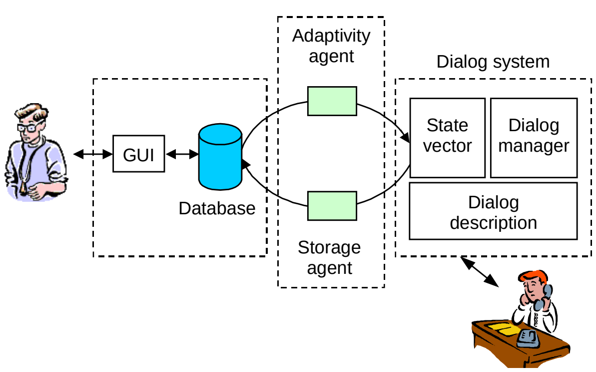

Figure 1: Architecture overview

to the software that performs this task. The name stems from the fact that this agent will also perform some of the adaptations to make the dialog more effective and comfortable. The four parts sketched in this section will be detailed below.

2.1 The database

The database was implemented using Microsoft Access 2000. It is made up of approximately 30 tables that store the actual data (plus 45 storing numeric codes), about 250 queries, 70 macros and 50 forms and subforms. Data are entered in the tables through a graphic user interface built for this purpose. The interface consists of a set of forms described below:

1. *Main menu* – provides access to the database functionalities.
2. *Personal and pathological anamnesis* – for the storage of personal data and familial and pathological anamnesis; the latter is divided into cardiovascular and non-cardiovascular diseases.
3. *Life style and risk factors* – lists patients' life habits and allows to calculate three risk indicators: global ten-years risk (Framingham), risk for hypertensive patients and risk for dyslipidemic patients. The risk factors are computed following standardized rules suggested in the guidelines.
4. *Drugs, physical examination and prescribed clinical tests* – this form is made up of seven pages: one is used to record prescribed drugs and side effects possibly observed; two pages record the outcomes of the physical examination, three pages record the outcomes of the clinical tests and finally one displays the medical report. The two pages pertaining to the physical examination allow the physician to store the data acquired during the visit, e.g. weight, systolic and diastolic pressure, heart rate and other information pertaining to the heart, chest, lower limbs, abdomen, peripheral pulse rates, vascular murmurs and neurological examination (see figure 2). The three pages regarding the exams collect the outcomes of instrumental (e.g. EKG, echocardiography, echodoppler, Holter monitoring, etc.) and laboratory tests. Among the latter, those pertaining to the lipid function tests are of particular importance. The medical report allows the recording of the outcomes of the visits, the suggested therapy and date and time of the next scheduled visit.
5. *Queries* – allows to perform queries on the database, for example, to obtain the list of patients to be visited during the week or the list of tests prescribed by the physician but not carried out yet.
6. *Report printout* – allows printing a summary of patient information, including personal data, cardiovascular disease history, physical examination, etc. Besides this mandatory information, a set of optional data may also be displayed, like the outcomes of a set

of exams that can be individually chosen from the same interface.

2.2 The dialog

One dedicated server with adequate hardware is connected to the telephone network and runs a commercial "call center" software, which answers phone calls and routes them to the voice recognizer. When one call arrives, the dialogue software is started. After authenticating the user, it reads from the state vector a set of discrete variables encoding the *user model*, the *patient model*, and the *target* (or dialog) *function* for a specific patient.

The dialog function determines which data must be acquired coherently with the evolution of the pathology based on the knowledge represented by the user and patient models. To this purpose, the dialog function is changed according to the information contained in the database. This enables the whole system to behave according to the long-term adaptation scheme suggested in [9]. In this system the long-term adaptation for each patient takes place *after* she has completed the conversation with the system. The system is therefore able to adapt itself to the evolution of the disease, new patient habits or features, changed physician directions, etc.

We are also considering a different kind of adjustment strategy, which we have called *short-term* (or *punctual*) *adaptation*. This kind of adaptation takes place *during* the course of a dialogue. The system is able to handle a range of user's behaviors, as it will be shown in the second example below. Short-term adaptation is managed by the code contained in the dialogue description.

The two kinds of adaptations, together with the use of a mixed initiative strategy [5], allow the system to conduct dialogues of the following shape (translated from the Italian language):

System: Tell me your heart rate, please.
Patient: I have not measured it, but today my weight is almost 85 kilos.
System: So your heart rate is missing, and your weight is 85 kilos?
Patient: My weight is 84.
System: So your heart rate is missing, and your weight is 84 kilos?
Patient: Yes.
System: I would like to remind you that your physician prescribed a diet. Your weight has increased by 2 kilograms since last call.

The system executes the following pseudo-code in the background:

Check if the weight acquired is a reasonable value (e.g. not over 200 kilos):
- False ⇒ repeat the question to the patient and try to acquire a new corrected value for the weight.
- True ⇒ read from the patient model the information regarding "weight acquired in the last call" and "diet".

Compute the time elapsed since the last call and calculate a maximum allowed weight gain (loss)

Check if the new weight is beyond the limit calculated above
- True ⇒ repeat the question to the patient and try to acquire a new corrected value for the weight.
- False ⇒ check if the "diet" field for the patient is set to "Yes".
  - True ⇒ remind to the patient: "you are on a diet!". Notify the patient whether his weight has increased / unchanged / decreased since the last call.

Store in the database the following data:
Weight := "64"
Heart Rate := "Missing Data"

Go to the next dialogue step according to the target function for this call.

The dialogue above exhibits long-term adaptation because it is using some knowledge about the patient deriving from previous calls.

On the other hand, the system short-term adaptability can be seen, for example, in the following case. After certain significant questions, a special set of grammars is activated in the background. One of them is triggered when the name of a malaise is possibly uttered. If this happens, the system reads the field "prescribed drugs" from the patient model and activates an investigation on the possible side effects for those drugs. This investigation is conducted with a different level of detail in relation to the severity of the side effect.

The system has other linguistic and medical characteristics. Prompts are for example formulated in the masculine or feminine gender depending on the sex of the user; vocal prompt of different complexity are used in relation to his ability, etc.

2.3 Interaction between dialogue and the database

The dialogue description is expressed in a simple procedural-like language. This language enables the developers to choose the succession of questions and to associate the desired grammars or bigram to each of them. (A grammar describes the set of allowable answers, e.g. yes/no). An interpreter, part of the SPINET speech recognitions system, executes the dialogue description during telephone calls. When the user hangs up the telephone, all the answers gathered are stored in a text file representing the state vector. The data are to be ultimately stored in the Homey database along with those entered by the physician conventionally via keyboard and mouse.

This merging, however, poses a problem. Well-designed databases hold their data in a strongly structured form: care is taken to create tables storing numeric codes for strings; also, records are inserted with different modalities depending on the context. For example, one single telephone call has to be logged exactly once, while in the same call the user may report zero, one or more observed side effects; each of them should be stored as an individual record. Each parameter is also strongly typed, so a string cannot be inserted in place of an integer value or a date, or vice versa.

On the contrary, the course of a dialogue does not enforce the same fixed structure; it is therefore not trivial to insert the collected data in a pre-existing database. The difficulty arises from two facts. First, even if the set of data asked is known before the call is set up, there is always the possibility of a number of exceptions to take the precedence over the intended dialogic sequence. Fields may be missing because of recognition errors, answers not foreseen in grammars, lack of willingness to answer, interrupted phone calls and a number of events which are "exceptional" and should not be ignored. The dialogue description reports these events as strings or empty values instead of the expected items. It is important to keep track of these events for future calls: if the conversation was interrupted, one wishes to ask the missing answers during the next call; if instead there was a recognition error, one wants to notify the developers; and just do nothing if the user said "I don't know".

Second, the data asked and stored during the dialogue are subject to logical dependencies, such as not being asked because there is no need. At a certain point, for example, the patient is asked if she had been well. If the answer is affirmative, she will not be further asked about side effects she might have suffered from.

This variety of outcomes is reflected in a wide variability in the intermediate results that the dialogue description stores in the state vector. The complexity that we have sketched may be summarized in the need of two elements that are absent in the dialogue description output, but in our view are necessary to have a truly adaptive dialogue. The information we have to add is:

- A tree of the dependencies between the fields, i.e. which ones will not be requested depending on which other. Each leaf of the tree is one of the fields that may be possibly asked at call-time. Forks in the tree may be "switched off" depending on values of chosen fields or other conditions (i.e. date, time of last call etc.) to enable or disable the lower branches.
- A set of attributes associated to each leaf of the tree. These attributes are used to preserve state information between calls, such as whether the corresponding information (leaf) was successfully obtained, whether it is to be asked the next time.

The interaction between the dialogue description and the database is handled by a set of Java classes. These programs are the agents that transform the data acquired vocally at call-time into the well-structured records of the database. These programs also apply some consistency checks and maintain the set of attributes discussed above synchronized with the data resulting from the last call.

Interestingly, the components of our system have roles somewhat analogous to those foreseen by the Model-View-Controller paradigm often employed in Web programming [14]. The analogy is significant because a further development of the project foresees the creation of a Web interface to the same database. This interface may reuse and take benefit of the already developed Model component (the database and the corresponding access logic). At present, we are also studying the possibility of further extending the accessibility of the data using other web-based protocols so they can be accessed by wireless mobile devices or multimodal (voice enabled) browsers.

3. Future developments

The separation between database, attributes, and call data will allow the programs to develop from simple procedural transformation programs into agencies ca-

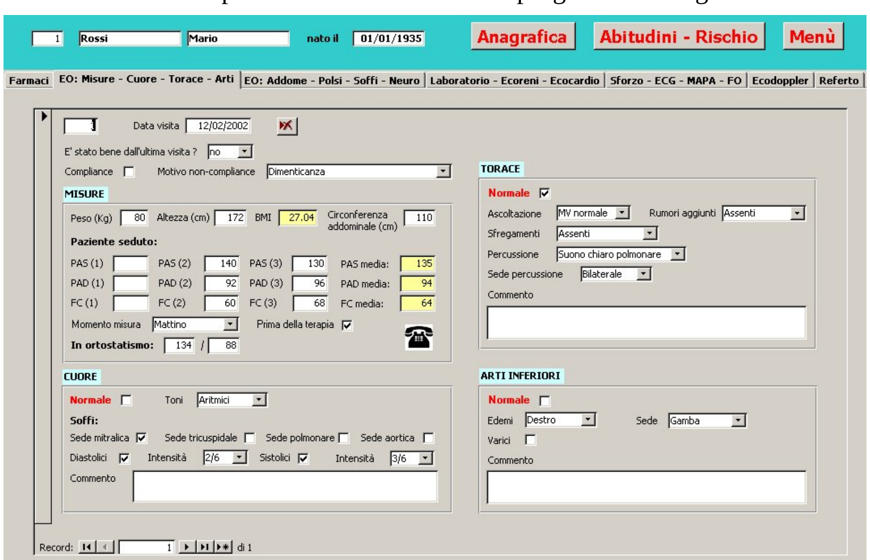

Figure 2: The graphical interface as shown to the physician. The telephone icon appears in the form because a part of the data was entered by the patient via the spoken dialog system.

pable of accomplishing more complex tasks, like rule-based or machine learning techniques.

A general goal of speech research is to increase the user friendliness of human-machine interaction. Properly designed graphical user interfaces with traditional (keyboard and mouse) input devices offer few chances of the computer misunderstanding human's commands; spoken dialogues conducted by computers are less fortunate in this respect; automatic speech recognition is prone to recognition errors. These errors can only be detected and corrected by the means of confirmation prompts, which are annoying because they ask the user to present the same information twice or more; also, requesting these confirmations lengthens the duration of the conversation.

The translation agents introduced would be to potentially able to increase the caller comfort. The first and most obvious advantage of putting "intelligence" inside them would be to shorten the dialogue duration. One way to do this would be to shorten the prompts spoken to users considered "expert" (those who have already called a certain number of times, who supposedly know with some approximation what questions to expect). Also, it would be possible to extract statistics on the confirmation prompts' outcomes. Those users that tend to be understood by the system frequently enough could have the number of such prompts reduced. Alternatively, the step-by-step prompts could be replaced by a more general summary confirmation, like "…is all of the above information correct?" In case of a negative answer, the dialogue description could fall back to question-by-question confirmations. A further source of adaptation could provide some learning of user's answers, to avoid repeatedly asking selected questions whose answers have been constant in some chosen period of time (e.g. "do you still do swimming?").

In short, the software agents may adapt to the history of calls of each user. Since the physicians have a dedicated graphical interface to the data, it is also conceivable to extend it so that it can be used to customize the course of the dialog. Physicians would then be able to tune specific aspects of the data set to be collected for each patient. We could then say that every call has an associated "intent" that defines the scope of the interaction. The intent can be partly chosen by the system (e.g.: to collect data, to fill in missing values from a previous call, or to remind the patient of the next scheduled visit); or it can be chosen by the practitioner (e.g.: how often to monitor, whether and how often to ask about life habits).

4. Conclusions

We have described an infrastructure for monitoring hypertensive patients. The system keeps the detailed clinical records about the patients being followed in a database; the data may be updated either directly by the physician, or by the patients themselves, via a dialogue on the telephone with an intelligent system. In this way they avoid the inconvenience of going to the S. Matteo hospital to record those values which can easily measured at home. The physician is able to review all the information entered, along with the derived risk indicators, to make informed decisions.

Acknowledgements

This research has been partially funded by EU under the Fifth Framework IST project "HOMEY", number IST-2001-32434.

References

[1] Friedman RH, Stollerman JE, Mahoney DM, Rozenblyum L. The virtual visit: using telecommunications technology to care of patients. J Am Med Inform Assoc 1997; 4:413-425.

[2] Friedman RH. Automated telephone conversations to assess health behavior and deliver behavioral interventions. J Med Syst 1998 Apr; 22(2):95-102.

[3] Rogers MAM, Small D, Buchan DA et al. Home monitoring service improves mean arterial pressure in patients with essential hypertension. Ann Intern Med 2001; 134:1024-1032.

[4] Friedman RH, Kazis LE, Jette A et al. A telecommunications system for monitoring and counseling patients with hypertension. Impact on medication adherence and blood pressure control. Am J Hypertens 1996 Apr; 9(4 Pt 1):285-292.

[5] McTear MF. Spoken dialogue technology: enabling the conversational user interface. Submitted to ACM Computing Surveys.

[6] Singh M, Barnett J, Singh MP. Spoken dialogue interfaces for patient care.

[7] McRoy SW, Ali SS, Restificar A, Channarukul SS. Building intelligent dialog system. In the ACM SIGART publication intelligence 1999 Spring; 10(1).

[8] Azzini I, Falavigna D, Gretter R, Lanzola G, Orlandi M. First steps toward an adaptive spoken dialogue system in medical domain. EUROSPEECH 2001, Aalborg, Denmark, September 2001.

[9] The sixth report of the joint national committee on prevention, detection, evaluation, and treatment of high blood pressure. NIH Publication No. 98-4080, November 1997.

[10] 1999 World Health Organization – International society of hypertension guidelines for the management of hypertension: guidelines subcommittee. Journal of Hypertension 1999, 17:151-183.

[11] Executive Summary of the Third Report of the National Cholesterol Education Program (NCEP) Expert Panel on Detection, Evaluation, and Treatment of High Blood Cholesterol in Adults (Adult Treatment Panel III). *JAMA.* 2001;285:2486-2497

[12] Levin E, Pieraccini R, Eckert W. A stochastic model of human-machine interaction for learning dialogue strategies. IEEE Transactions on speech and Audio Processing 2000 Jan; 8(1):11-23.

[13] Falavigna D, Gretter R, Orlandi M. A mixed language model for a dialogue system over the telephone. Proceedings of ICSLP 2000, Beijing, China, 16-20 October 2000.

[14] Gamma E, Helm R, Johnson R, Vlissides J. Design patterns elements of reusable object-orientated software. Addison-Wesley; 1995.