mobile computing
clinical information systems
decision making support systems

Piotr RADLIŃSKI*, Jerzy SAS**

USER INTERACTION SIMPLIFICATION METHOD FOR HEALTHCARE MOBILE APPLICATIONS

In the paper an evaluation and optimisation framework for medical data access systems user interface is presented. User interface design seems to be of crucial importance for clinical data access applications acceptance, in particular for the applications running on handheld computers where the interface tools are significantly limited. The ease of program use depends strongly on the correct user interface design and on the algorithms which try to predict the user decisions and interactions made in the process of data access or modification. If the program is able to correctly predict the user actions and fetch him reasonable defaults then the number of interface actions which the user must do is significantly reduced. The method presented here focuses on typical functions available in clinical mobile data access systems: medication prescriptions and diagnostic and laboratory tests orders. The user interaction with an application is considered as the sequence of decisions. Using the records stored in the hospital database, the algorithm finds the most probable decisions at the subsequent stages of the interaction and uses it as defaults presented to the user. In this way instead of entering the data from the keyboard the user can much faster select it from the list.

1. INTRODUCTION

The mobile access to patient data offers the physicians many advantages, most important of which are: instant and immediate access to diagnostic and treatment up-to-date data at the point of care, immediate availability of treatment decisions made by a physician, reduced risk of mistakes introduced on the path from medical decision maker to its execution point, reduction of rework and in result - reduction of healthcare costs. Constantly decreasing prices of handheld computers and the wireless LAN hardware make mobile medical data access systems widely available for hospitals and other healthcare institutions. Increasing performance of handhelds - currently comparable to typical desktop PSs - enables their usage not only as front-end displays of replicated data (as e.g. described in [6]) but as fully functional processing units with instant access to database servers maintaining broad range of clinical data ([8]).

Having at the disposal such hardware and application development software tools it is easy to implement the mobile counterpart of any typical hospital information system with the functionality

* Chair of Systems and Computer Networks, Wroclaw University of Technology
** Computer Science Department, Wroclaw University of Technology, 50-370 Wroclaw, Wyb. Wyspianskiego 27

desired in the mobile mode. Limitations of user interface tools available on palmsize computers cause however that the user dialog with the mobile application must be designed in the specific way. There are two most important limitations of the palmsize computer user interface distinguishing it from typical desktop PC:

- lack of physical keyboard, instead the inconvenient virtual keyboard is available in most operating systems running on palm computers
- limited capacity of the display area.

From our experiences in developing and putting into practice the mobile hospital data access system it follows that the physicians accustomed to flexible and convenient interface of typical Windows applications strongly complained to the user interface style imposed by the PocketPC OS. The most critical factor was the number of elementary actions, which a user must do to perform typical operations available in the mobile system: access to selected patient records, prescribing medications, ordering laboratory and diagnostic examinations. Due to limited amount of data that can be shown on the screen and lack of physical keyboard, entering data to palm application is much more complicated as in the case of similar application running on PC.

In the paper the method is described which predicts the information a user enters to the mobile application at the patient bed. The data values most probable in the current context are loaded into selection lists (combo boxes). The current context is defined by the data entered so far and the patient state attributes. In result, instead of entering the data textually using inconvenient virtual keyboard or selecting it from long selection lists, in most cases the user can select correct option by simply clicking a item in the combo box. The method applies probabilistic approach widely used in pattern recognition or decision making support ([3]). The sequence of data entry actions necessary to perform the operation in the system are treated as the sequence of decisions. At each stage of such process the program predicts the set of most frequently used decisions in the context determined by the decision (entered data) at the earlier process stages. The contents of database of the hospital information system is used to find the set of most probable decisions. The found decisions set is presented to the user as the contents of short selection list combo box.

The concept described here has been partly implemented, tested and put into clinical practice in the HIPOKRATES-MOBILE system based on palms size computers running PocketPC or Windows CE operating system. It is the mobile counterpart of the complex clinical information system HIPOKRATES used in more than hundred healthcare units.

2. PROBLEM FORMULATION

Let us consider the program which makes possible to perform a set of independent operations $\{O_1, O_2, ..., O_n\}$, $n >= 1$. In a medical information systems such operation can be e.g.:

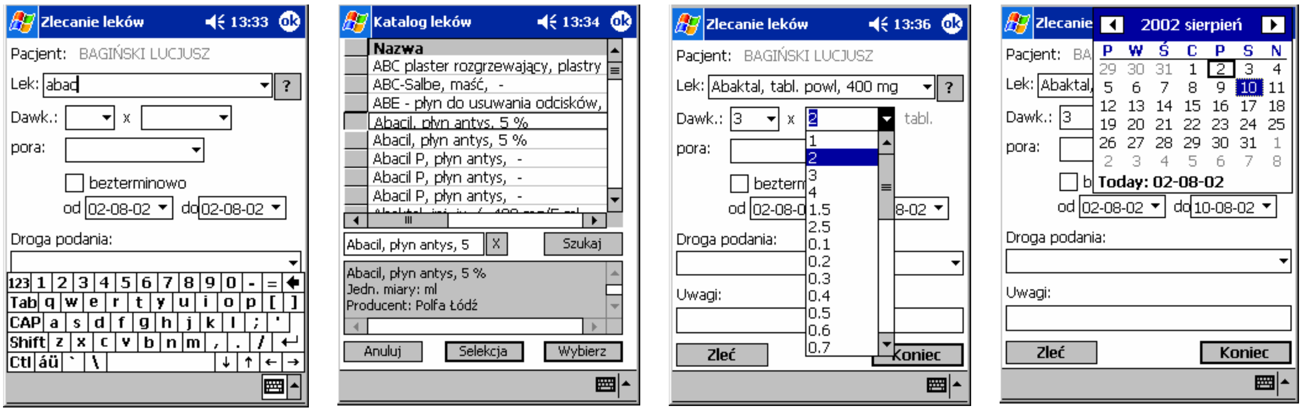
- displaying results of laboratory examinations for given patient for given period,
- prescribing the medication for a patient,
- ordering the laboratory test etc.

Each operation O_i consists of a certain number of elementary stages $\{o_{i,1}, o_{i,2}, ..., o_{i,m(i)}\}$ where $m(i)$ denotes the count of stages of O_i. The elementary stage consists in entering a data element to the application. In the case of medication prescription such stages will be:

- selection of the patient (usually by name),

- selection of the medication,
- determining the unit dose,
- determining how many times per day the medication will be administered,
- determining the period of application (in days).

At each stage the user have to do a number of basic user interface (UI) actions. The type and count of basic UI actions depend on applied user interaction style and used UI controls. The implementation of the exemplary operation: medicals prescription in HIPOKRATES-MOBILE system is illustrated on the Fig.1.



a) entering medication name with the virtual keyboard

b) medication selection from the list box

c) dosage selection from the combo box

d) determining medication administering period

Fig.1. Medication prescribing in HIPOKRATES-MOBILE system running on Compaq iPAQ 3876

Let's assume that we are dealing with the application built using typical UI controls supported by the mobile computer operating system: combo boxes (data entered by selection from the popup list or with the virtual keyboard), edit fields (data entered only by typing on the virtual keyboard), action buttons, check boxes and radio boxes. The following basic UI actions can be distinguished:

- click on the combo box to display selection list,
- selection list scroll by dragging the slider,
- click on the desired element in the selection list,
- click to open the virtual keyboard,
- click on a key on the virtual keyboard,
- click to close the virtual keyboard
- click to set focus on the edit field,
- click to change the state of radio or combo box,
- click on an action button.

Each of these actions seem to be equally troublesome for the user except of selection list scrolling. In this case we assume as the basic action an action, which scrolls a selection list by such count of items as the length of the list window. If e.g. the combo box window contains 5 elements and the desired element is on the 23rd position in the list then the user must perform 4 basic scrolling actions until the desired position is displayed.

We will assume that the data that need to be entered at the stage $o_{i,j}$ come from the finite set of elements $d_{i,j} = \{d_{i,j,1}$, $di,j,2$, ..., $d_{i,j,n(i,j)}$ $\}$, where $n(i,j)$ is the number of possible data variants at j-th stage of i-th operation. We can treat the data selection in the stage $o_{i,j}$ as the decision the user makes to perform the operation. The decision depends on various attributes of the item, which the decision concerns and on the decision made at the earlier stages of the operation. For instance, the selection of the medication depends on the patient disease, examination results and previous treatment of the patient. The unit dose depends on all these factors and on the medication selected at the previous stage.

The number of basic UI actions the user have to do while making decision at the stage $o_{i,j}$ depends on the actual physician's decision $d_{k,i,j}$ and the configuration of UI controls at the stage $o_{i,j}$ We assume that the program can dynamically change the UI controls configuration depending on selected patient and previous decisions. We restrict further considerations to selection lists, assuming that selecting from list is the main and most convenient method of data entering on palm size computers. In this case the UI dynamic reconfiguration consists in filling the selection lists with the sequence of decisions $[d_{k(1),i,j}, d_{k(2),i,j}, ..., d_{k(l),i,j}]$ ordered by its estimated probability in the current context. The symbol l here denotes the minimum of $n(i,j)$ and assumed capacity of the list.

Let us denote by $C(i, j, a(s_i), d_{k(1),i,1}, d_{k(2),i,2}, ... d_{k(j-1),i,j-1})$ the rule used to set automatically UI configuration at the stage $o_{i,j}$. The context defined by the attributes a of the patient s_i subject to the operation and the sequence of the previous decisions $[d_{k(1),i,1}, d_{k(2),i,2}, ... d_{k(j-1),i,j-1}]$. Let $L(i,j, d_{k,i,j}, C)$ denotes the number of basic UI actions that the user have to do at the stage $o_{i,j}$ assuming that his actual decision is $d_{k,i,j}$ and the UI configuration set automatically by the program is C.

Our aim is to find out such UI configuration rule C which will minimize the total count of basic UI actions necessary to complete the operation O_i. We will apply probabilistic approach. Let us assume that the attributes characterizing the state of a patient subject to operation, operation index i and the sequence of decisions $[d_{k(1),i,1}, d_{k(2),i,2}, ..., d_{k(m(i)),i,m(i)}]$ are observed values of the multidimensional vector of random variables $\boldsymbol{X}$

$$\boldsymbol{X} = [\boldsymbol{A}, \boldsymbol{I}, \boldsymbol{D_1}, \boldsymbol{D_2}, ... \boldsymbol{D_M}] \tag{1}$$

where:

A - vector of patient attributes being considered,

I - index of the operation being performed,

$M = max(m(i), i = 1, 2, ..., n)$ - maximal count of a single operation stages,

$D_j, j=1, 2, ..., M$ - index of the decision made at j-th stage of the operation I.

For given UI configuration rule C and for each observed value $[a, i, d_1, d_2, ..., d_M]$ of the random variable X we can calculate the total count of basic UI actions necessary to perform completely the operation i as:

$$q(a,i,d_1,d_2,...,d_M,C) = \sum_{j=1}^{M} L(i,d_j,C(i,j,a,d_1,d_2,...d_{j-1})) \, . \tag{2}$$

Because q is e function of random variable it is also the random variable and we can calculate the mean value of Q:

$$E(Q(C)) = \sum_{i=1}^{n} \sum_{a\in\Lambda} \sum_{d_1\in\Delta_1} ... \sum_{d_M\in\Delta_M} q(a,i,d_1,d_2,...,d_M,C) p(a,i,d_1,d_2,...,d_M) \, , \tag{3}$$

where $p(a,i,d_1, d_2, ..., d_M)$ is the probability of the random vector $<a ,i, d_1, d_2, ..., d_M>$ and $\Delta_1, ...,\Delta_M$ denote the domains of possible decisions at the stages $1, ..., M$ respectively.

The value calculated according to the formula (3) represents the average number of basic UI actions, the user has to do to perform the operation available in the interactive program. The lower this value is, the more convenient and faster is the program usage. We can treat $E(Q(C))$ as a measure of UI ease of use. $E(Q(C))$ depends also on the UI configuration rule C applied in the program. By appropriately constructing the algorithm for the rule C we can reduce the mean value $E(Q(C))$ and in this way make the program more user friendly. We can define the optimal UI configuration rule C^* defined as the rule which minimizes the formula (3), i.e.

$$E(Q(C^*)) = \min_C E(Q(C)) \tag{4}$$

3. USER INTERACTION OPTIMISATION BY FINDING THE SET OF MOST PROBABLE DECISIONS

Direct application of the formula (4) to optimise the user interface is unfortunately impossible in most cases. This is because the probabilistic distribution of random variable $\boldsymbol{X}$ is unknown. The probabilistic properties of $\boldsymbol{X}$ strongly depend of the characteristic features of patients population in given hospital, on the physicians habits and therapeutic methods applied in the hospital. We can however use the patients records stored in the hospital database to at least roughly approximate the properties of the random variable X. Additionally we will decompose the formula (3) into groups of components corresponding to various elementary stages. We then find near-optimal rule C for UI configuration for each elementary stage $o_{i,j}$.

We can rearrange the formula (4) in such way that the components for various values of j,i indices will be grouped together. The group of components for the same value of i and j represent the average count of basic UI actions at the stage $o_{i,j}$. The C rule in these components depends only on the attribute vector a and the set of earlier decisions $d_{i,k}$, $k =1, 2, ..., j-1$. As it was stated in the previous section we only consider UI reconfiguration by loading selection lists to be used at the current elementary stage. The following cases may occur:

- the actual user decision is just in the combo box - 0 basic UI actions are necessary,
- the actual user decision is among first k elements on the selection list - 2 basic UI actions are necessary (k is the number of items that can be displayed in the list box selection window)
- the user decision is on the n-th position on the list - $Ent(n/k) + 1$ basic UI actions are necessary.

The UI configuration rule that minimizes the expected count of basic UI actions necessary at this stage will be following:

- estimate the conditional probabilities : $p(d_{k,i,j} \mid a, d_1, d_2, ..., d_{j-1})$ of all possible decisions $d_{k,i,j}$ at $o_{i,j}$ stage, where $d_1, ..., d_{j-1}$ are the decisions made at the earlier stages of the operation,
- find the decision $d_{i,j}^*$ with the highest estimated conditional probability and place it as default in the combo box used at current stage,

- sort remaining decisions by their estimated probabilities in descending order and place them in the selection list in this order.

In this way the most probable decision requires no basic UI actions. The set of the most probable next *k* decisions require two UI actions. Less probable decisions need more basic UI actions, because they are located in the further part of the selection list are more scrolling is required to access then. However they occur relatively rarely.

The conditional probabilities used in the presented rule can be estimated from the data gathered in the hospital information system database using simple estimator:

$$: p(d_{k,i,j} \mid a, d_1, d_2, ..., d_{j\text{-}1}) = \frac{n(k,i,j,a, d_1, d_2, ..., d_{j\text{-}1})}{\sum_{l \in \Delta_{i,j}} n(l,i,j,a, d_1, d_2, ..., d_{j\text{-}1})}, \qquad (5)$$

where $n(l,i,j,a, d_1, d_2, ..., d_{j\text{-}1})$ is the number of decisions d_l made at the stage $o_{i,j}$ recorded in the hospital database in the context of patient attributes a and earlier decisions $d_1, d_2, ..., d_{j\text{-}1}$. In practice it sometimes happens that the count of recorded decisions made at the stage $o_{i,j}$ in the context defined by $a, d_1, d_2, ..., d_{j\text{-}1}$ is not sufficient to reliably estimate the required probability. In such way some elements of the context may be omitted.

4. EXPERIMENTAL RESULTS

The method described in the previous section has been implemented and provisionally tested using the clinical data stored in the database of HIPOKRATES hospital information system. HIPOKRATES is the complex information system which gathers and maintains the wide range of data necessary for typical hospital functioning including: patient examination and treatment records, medications administration, accounting data and many others. The medication prescribing operation was analysed as the typical candidate for mobile implementation. The following data from the database were used in the experiment:

- patient's main disease, age, sex, weight - as elements of the attribute vector *a*,
- the contents of the catalogue of medications used in the hospital,
- the set of medication prescription records containing the medication ATC code, unit dose, count of daily doses, period of administering.

The age and weight attributes were discretized for the sake of experiment. The medication prescription operation consisted of four elementary stages: medication selecting, determining the unit dose, determining the daily repetition and determining of the administering period. The patient selection was excluded from optimisation because it is hard to find any reasonable rule for this stage. We assumed the selection lists window sizes as in the actual implementation of HIPOKRATES -MOBILE system running on Compaq iPAQ 3876 palm size computers:

- medications list box - 8 items,
- unit dose combo box - 13 items,
- daily repetition combo box - 13 items.

The administering period was determined by selecting the first and the last day of the period from the calendar tool. In this case selection lists were not used, merely the most probable defaults are placed in the date edit fields. The database used in the experiment comes from one of the hospitals using the system and contains 2745 patients records, 6790 records in the medications catalogue and 8709 prescription records. For short, we will further call the part of database contents used in the experiment - the learning set.

The aim of the experiment was to compare the described method with the simplest method which loads all possible decisions to the selections lists sorted alphabetically. Also the modification of the simplest method was tested, which moves to the selection list beginning the medications recently administered for a patient being processed. This modification was suggested by the physicians using the HIPOKRATES system. It is based on the belief, that frequently medications are prescribed repeatedly. In order to investigate how the method performance is influenced by the learning set size, the experiment was repeated for various counts of prescription records used to determine the contents and order of the selection lists. In the experiment, entering of prescription data was simulated using assumed characteristics of the dialog controls in the mobile application. We used "leave one out" method. The prescriptions from the database were used as test cases. For each test case remaining prescription records from the database were used as e learning set. For each prescription and each tested UI configuration method the count of basic UI actions was calculated. Finally for all tested methods the average basic UI operation counts were calculated. The results are shown in the table below.

Learning set size	Average basic UI operations count - alphabetical lists sort	Average basic UI operations count - alphabetical lists sort with recently administered medications	Average basic UI operations count - finding the most probable decisions
2000	16.7	16.9	15.2
4000	16.9	17.1	10.7
6000	16.7	16.7	9.7
8709	16.8	17.2	8.5

The proposed method significantly reduces the average count of basic UI actions in relation to the simplest method where no decision prediction is made. The method which places in the beginning of the selection list the medications recently administered for the patient behaves even worse. This is probably because actually the medications are administered twice rarely, the user opens the combo box but not finds there the needed item (additional two basic UI actions) and just now opens the medication catalogue.

5. CONCLUSIONS, FURTHER WORKS

The method presented in the paper makes it possible to simplify the user interaction with typical medical data access and entry applications. The aim of the method is to reduce the count of elementary interface actions, which a user have to do while interacting with a program. The aim is achieved by predicting user decisions. Simplification of the user interaction with the program in especially important in the case of applications running on mobile devices where the interface tools are strongly limited. Although the described concept can be applied in any area of application it seems to be particularly useful in medical mobile applications for data access and entering. The hospital information system database can be used as the source of data for user decision prediction. The simulation experiment carried out on the real clinical database proved the advantages of the proposed method in comparison with the methods that do not apply prediction. The average count of elementary UI actions is reduced almost twice, provided that the sufficiently large learning set is available.

The efficiency of the described method can be probably increased. In our investigations we did not carried out the thorough feature selection. We arbitrarily assumed that only disease, age, sex and body weight are the patient attributes considered when finding the most probable medications to be prescribed. Probably, taking into account also the results of diagnostic and laboratory tests and the medications administered to the patient earlier would improve the correctness of prediction. Another way that can lead to the method improvement is the usage of physicians knowledge about relations between patient attributes and recommended medications. The knowledge can be presented in the form of logical rules and can be combined with the implicit knowledge extracted from the learning data in the database (as e.g. proposed in [2]).

The method was practically applied in the mobile counterpart of HIPOKRATES hospital information systems. The HIPOKRETES-MOBILE application runs on any palm-size computer running PocketPC operating system. The mobile application communicates on-line with SQL database server via wireless LAN.

BIBLIOGRAPHY

[1] HUZAR Z., KURZYŃSKI M., SAS J., Problem-oriented database for combined diagnostic algorithms (in polish), II National Conference "Knowledge engineering and expert systems", Vol. II pp. 181-188. Techn. Univ. of Wroclaw Press

[2] HUZAR Z., KURZYŃSKI M., SAS J., Rule-Based Pattern Recognition With Learning, Techn. Univ. of Wroclaw Press, Wroclaw, 1994

[3] KURZYŃSKI M., Pattern Recognition. Statistical Methods. Techn. Univ. of Wroclaw Press, Wrocław, 1997

[4] PUCHALA E., Design of an expert system for medical decision making support (in polish), Journal of Medical Informatics and Technologies, Vol. 5, Nov 2000

[5] PUCHAŁA E., System of telemedicine services designed for family doctors' practices. Telemed J. 2000, Vol 6, No. 4

[6] RADLINSKI P., SAS J., Handheld Based Hospital Information System for Fast Data Access at Patient's Bed, Journal of Medical Informatics and Technologies, Vol. 2, Nov 2001

[7] THERRIEN C.W., Decision Estimation and Classification, John Wiley & Sons, Inc., 1989

[8] TRĄBKA W., Hospital Information Systems, University Press "Vesalius", Kraków 1999