Advancement of Assistive Technology
G. Anogianakis et al. (Eds.)
IOS Press, 1997

The Intelligent Indexing Function: Sound Access to Healthcare Distributed Databases

P. Angelidis, G. Nentidis, S. Maglavera, G. Anogianakis
Biotrast S.A., 111 Mitropoleos, 54622, Thessaloniki, GREECE

Keywords: database management; healthcare information systems; searching algorithms

Abstract. The Intelligent Indexing Function is a mechanism, based on artificial intelligence techniques, which allows location and access to information in a distributed database system in a fast, sound and accurate way. Despite that the link delivered within the European Community DG XIII Health Telematics INSIDE project (TP1150) and implemented within the corresponding software package, installed in an Infocenter serving elderly people, it was developed in such a manner, so as to be directly usable to any application requiring linking between databases for a distributed information delivery system and it can be directly applied to any type of Health Care Provision and Rehabilitation service.

1. Introduction

We present here a link between databases for a distributed information delivery service. The work is part of the INSIDE project, within the European Commission's Health Telematics Section of the Fourth Framework Programme on Research and Development [1]. Although the link has been developed for an Infocenter serving elderly people it can be directly applied to any type of Health Care provision service.

The general objective of INSIDE is to provide an integrated information delivery service to the elderly through the use of an intelligent system which gives access to available health care and socially related relevant information distributed among the various structured and non structured data banks [2]. An *InfoCentre* connected both to a *Local* and *Wide Access Network* is the core of the system. It makes use of a private database, which contains *pointers* to existing information sources both locally and outside the area. User access to the system is provided through the telephone to professional operators, who receive, identify and process the requests.

INSIDE makes use of a network based technology [3]. The system consists of a distributed database, connecting and managing different local sites spread over the territory. Moreover, every local site has access to other relevant and previously existing information sources.

2. User requirements: Information flow for an elderly request

The answering process to solve the elderly requests involves several agents that collaborate to send the final result to the elderly. The information flow can be seen in figure 1, where the possible active agents are: the elderly, the InfoCentre operator, the

health or social professionals and the elders' GP. The media to exchange the information depends on the agents and also on the request characteristics. The media can be voice, electronic documents, computer displays and mailed reports. The sources of information are either the professionals or the Infocentre databases.

The operator interacts both with the elder, using voice communication, and with the system, in order to retrieve the required information. The operator translates the natural language of the user question into the defined categories. This process is guided by the user interface, providing the operator with some kind of electronic questionnaire where the request can be matched. The output of the system query will be displayed on the terminal screen, and the operator has to translate it into natural language to give the information to the caller. This is an iterative process because the contents of the search can modify or expand the user request. When the operator detects that he/she is not allowed to give the specific answer, the system generates a detailed report of the request which is sent to an electronic mailbox for pending requests, waiting for a professional attention. The professional use the system to obtain the required information, but in some cases he/she has to verify the elder needs contacting the elder's GP. The professional gives the final answer to the elder by phone.

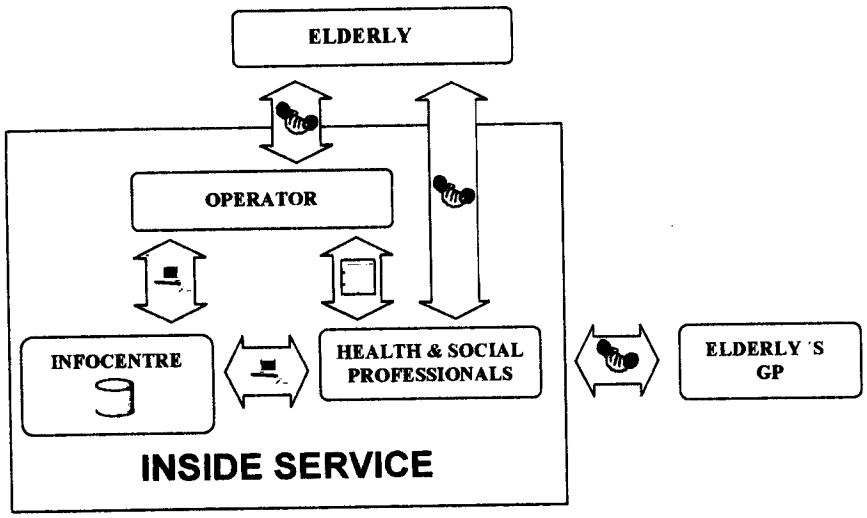


Figure 1. Information flow for an elderly request at the InfoCentre

3. Software Description

The advantages of a distributed system are ensured through a software core which manages the information flow so that transparency, fragmentation and replication transparency are achieved automatically, without any user intervention. We call it *The Intelligent Indexing Function.*

The Intelligent Indexing Function is a mechanism which allows the operator to access the information in a remote database transparently, i.e. as if it was information stored in the INSIDE database.

In other words, the result of the routing is guided not only to the available service providers in the INSIDE database but also to the selected remote database's ones as well

nad this is done transparently to the operator. However, in order to offer more flexibility and to reduce unavoidable search delays this procedure is operator driven, i.e. it is only be invoked on the operator's decision for additional information. Moreover, supplementary search fields are available to the operator for additional flexibility, as for example

- name (e.g. starting with A)
- location
- type of organization

The main characteristics of the mechanism are its effectiveness, transparency and integrateability into the INSIDE software. The function implementing the mechanism accesses the external databases in such a way that none major modification is required in the INSIDE database structure, nor the user interface software.

Although, only one table would suffice to store the external source information, it was decided to store it in a structure similar to the one of the INSIDE database. This was decided as a matter of compliance with the INSIDE design philosophy, but also for practical reasons: this way the user interface didnot have to be modified at all. The structure includes tables in categories a) service provider geographic location and b) service providers information and is shown in Table 1. These tables are automatically created from the core software and they are transparent both to the user and the database. The main features of the software are:

a) always returns result
b) automatic area filtering of the results
c) platform independent
d) integrateable at the GUI or database level
e) direct
f) intelligent
g) transparent
h) user controlled, additional masking criteria

The algorithm was developed in a DLL form and it was embedded in the GUI. The algorithm, according to the overall system technical specifications, is provided with a keyword (primary need service) and returns all service providers that offer this service. In order to find a service provider based on the keywords, an initial search is performed in all the keywords in the table that holds all service provider categories and products. This algorithm finds all words in the categories file that match with the keyword, and then finds all category descriptions that contains any of the matches.

As soon as the correct category has been found, the corresponding tables are used to find all service providers that belong into the certain category and/or produce a certain product. And from those service providers we can choose for example those that are in Genova.

The search within the remote database is achieved using Direct Routing. Direct routing defines an intelligent searching mechanism which allows the operator to access the keywords list independently of the basic routing structure.

This algorithm implement artificial intelligence techniques from the pattern matching area [4]. Specifically, it involves string matching techniques [5] by finding where a certain relatively short series of data exists in a long series of data through identification of objects with most similar ones.

Its main characteristics is the high degree of freedom it offers and its high speed. The function implementing the algorithm manages to locate the closest solution in an

intelligent and accurate way, maintaining on the same time the important attribute of being fast.
Its main features are:

a) best match
b) always returns result
c) measurable match (error definable)
d) multiple re-searching steps
e) platform independent
f) integratable at the GUI or database level
g) fast and reliable

4. Description: Practical Experiment of the Algorithm

This searching algorithm mainly does the following:

a) It loads all the keywords into memory if desired or otherwise it just opens the table of the keywords.
b) It splits the user's phrase into words
c) For each word in the user's phrase, it tries to find a list of the best matched words that exist in the keyword list.
d) It searches all the keyword list to find all keywords (phrases) that contain any of the word matches found.

The above steps are explained more analytically in the following.

Step (a) is the initialization of the procedure. Here the language that is going to be used is selected and also if a cache for the keywords will be also used. Since the algorithm for matching the strings must forgive misspellings and misplaced letters in the string, an index can not be used. This means that all the keywords in the database must be searched sequentially. For a few hundred words this isn't going to slow down the whole procedure. But if more than a thousand exist then the algorithm gets considerably slow. So, this algorithm gives the opportunity to load the keywords into a buffer which makes things faster. The memory used for this, isn't much. Considering that each keyword can be up to 50 characters, a buffer of a 1000 words will be 50000 big, a number which is not so big for windows programs. And since the whole procedure is considerably faster when keywords are loaded into memory, it is a very good approach. However caching comes as an option.

Step (b) splits the phrase into words, by recognising the different words within a phrase and storing them in a sequential table.

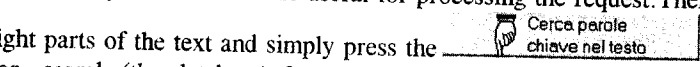
Step (c) is actually the algorithm that scans two words to find out if they match in any way. For that step at the beginning we used an algorithm that matches two words by the way they sound. This approach didn't work very well because the most important letter of the word was the first one. So if the user misspelled the first letter then a big error was produced, even though they were other parts of the strings that were similar. We then decided to take another approach, which leaded us into the final procedure that we use now. This one scans the two words from the start and counts all the common letters it finds. This may lead to an error if a letter is misplaced into the string, so the same procedure is followed from the end of the string. The total number of characters found to match both directions, is kept and stored. After that all the numbers gathered from the search, are checked to find out the maximum one. The word that produced this number and all the words that match the original one with less that three letters than the best match (meaning that the number they produced is smaller to the maximum by three), are

returned as best matches. This algorithm has very good results even when the keywords are typed wrong or extra letters have been typed into the string by mistake.

But still another step was added to improve the overall searching performance. Adding step (d) had as a result the algorithm to produce more result keywords because it now returns all the keywords that contain any of the words found to match. Even though it may give as a result strings that may seem dissimilar, it was placed there so that the user can get a list of keywords even if he or she remembers only one word of the full keyword phrase under investigation. For example if he or she remember the word "assistance", a whole list of keywords will be returned which contain the word assistance so that the user may have the opportunity to locate exactly what he or she is looking for. The important and attractive feature of the algorithm is that the list will be produced even if the user types "ssistance" or "assiastance" or any other misspelled similar word.

5. Evaluation

The Intelligent Indexing Function was integrated within the INSIDE GUI to assist operators of the system to locate the requested information from the elderly calls quickly and securely and to provide links with external sources of information through the Direct Routing mechanism. The software has been gladly accepted by the operators who consider it «a powerful tool in processing rapidly a request». Measurements in the Infocentres showed reduction of mean calling duration to almost the one third upon introduction of the Intelligent Indexing Function [6]. In figure 2 we show the screen the operator sees while processing a call (extracted from the Italian version of the program).

On the left there are two windows. On the top one the operator enters parts of his/her discussion with the elderly he/she considers useful for processing the request. Then he/she is able to highlight parts of the text and simply press the 

button, which stands for «search (the database) for key-words in the (highlighted) text». This ways the intelligent searching mechanism starts and looks for related keywords as described previously.

6. Conclusion

We presented the Intelligent Indexing Function, a mechanism which allows location and access to information in a remote database in a fast, sound and accurate way. Despite that the mechanism delivered within the European Community DG XIII Health Telematics INSIDE project (TP1150) and implemented within the corresponding software package, it was developed in such a manner, so as to be directly usable to any application requiring linking between databases for a distributed information delivery system.

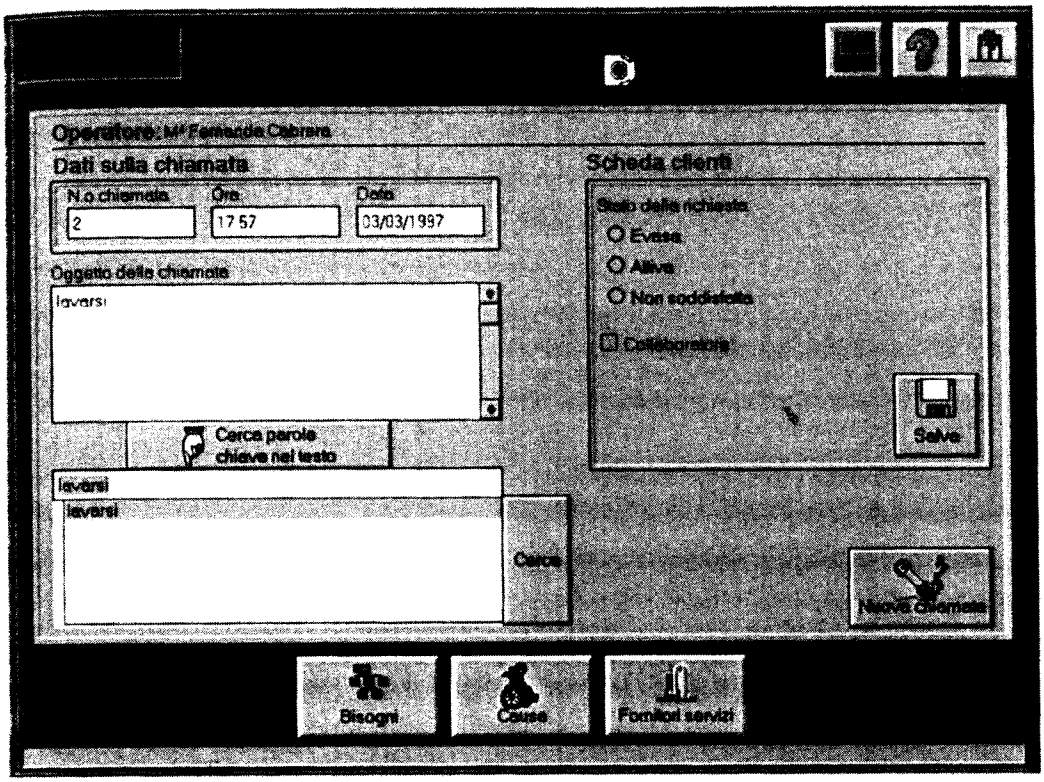


Figure 2: The screen the operator sees while processing a call

References

[1] Anon, Technology initiative for disabled and elderly people. Pilot action synopses, CEC, Report EUR 15023 EN, 1993.

[2] Czaja, S.J. et al., Computer communication as an aid to independence for older adults, Behaviour & Information Technology, vol.12, no.4, 1993, pp. 197-207.

[3] TIDE Programme 1150, INSIDE, Deliverable D2, Service Functional Specifications, July 1995.

[4] Aho, A.V., Hopcroft, J.E. amd Ullman, J.D., The design and analysis of computer algorithms, Addison-Wesley, 1974.

[5] Knuth, D.E., Morris, Jr. J.H. and Pratt, V.R., Fast pattern matching in strings, SIAM Journal of Computing, vol. 6, np. 2, pp. 323-350, 1977.

[6] TIDE Programme 1150, INSIDE, Deliverable D5, Test Bed Run Results, Dec. 1996.