GRAPHICS INTERACTION IN DATABASES

C. Frasson M. Erradi
Département d'Informatique et Recherche Opérationnelle
Université de Montréal, C.P. 6128, Montréal, Québec H3C 3J7

ABSTRACT

This paper is intended to highlight the evolution of interaction techniques between users and database systems. We examine the evolution of database query languages and emphasize the importance of the graphical interface taking in account the availability of new functionalities and the users demand. Then, using examples in a medical environment, we present the characteristics of an interactive graphical language based on icons.

RESUME

Cet article est destiné à mettre en relief l'évolution des techniques d'interaction entre les utilisateurs et les systèmes de bases de données. On examine aussi l'évolution des langages de requêtes dans les bases de données et l'on souligne l'importance de l'interface graphique en considérant la disponibilité de nouvelles fonctionnalités et la demande actuelle des utilisateurs. On présente ensuite, au moyen d'exemples médicaux, les caractéristiques d'un langage d'interaction graphique utilisant des icones.

KEYWORDS: database query languages, interaction techniques, icons, graphics and pictures.

INTRODUCTION

The evolution of database systems was characterized by an increase of the power of data manipulation languages. At the beginning, programming languages were the only means available for interacting with the data. In fact, the method used was complex, long and intended for programmers or specialists. At that time the end-user was not given much consideration. Progressively, the need for a more powerful interaction became necessary. The interface particularly evolved with the relational model allowing non procedural languages [1], [6], [21].

Availability of relational database systems, word processing and others software on microcomputers permitted to reach a new class of users. They are mostly non specialized users who generally wish a ready-to-use and simple system which does not require technical knowledge.

However, professionals requirements also contributed to a more sophisticated interface which might lead to better productivity. In addition, new types of data such as graphics or images, which convey a great quantity of information, are more and more used in various domains.

Finally, recent techniques like the mouse, bit-mapped displays and multiple-window screen management enable users to directly and easily perform their tasks.

In that context, a graphics interface now appears useful. Going further, an icon is sufficiently expressive to be understood by large group of users and used for communicating with databases.

In this paper we highlight the evolution of interaction techniques and justify the present importance of a graphics interface in databases. Then we present the main characteristics of a language which uses icons as a support to easily manipulate various types of data: structured data (numeric or alphanumeric) and unstructured data (text, image, voice) which will be more and more required in an end-user environment. Examples are given in a medical context.

THE EVOLUTION OF QUERY LANGUAGES.

Programming languages progressively evolved to be independent of the type of the computer, of the external memories and of the access methods. The concept of physical and logical independence, introduced by Date [7], and the need for a centralized data management were at the origin of the databases systems.

The development of architectures and languages has successively gone through hierarchical, network and relational models.

a. Hierarchical [15] and network [9] languages are intended for programmers and even for programmers who are familiar with the process of navigating through the database.

Before formulating a query it is necessary to define a set of procedures in order to "communicate" with the portion of the database choosen by the user: for instance, a program communication block (PCB), in IMS, or a schema and a subschema, in a network type language. In addition, procedures to find the data obey a "top-down-left-right" process (for hierarchies) or a list processing approach (network). Generally they have to be included in a host language (COBOL, PL/1, FORTRAN,...).

b. Several high-level data manipulation languages have been designed for the relational model. Four different strategies can be distinguished:

- Languages based on relational algebra which manipulate relations using a set of operators (union, difference, intersection, product, join, projection, selection) and which are procedural because we must know how to get what we want. However the underlying concepts are important.
- Relational calculus is non-procedural and represents a desirable property but it is hard to understand and use.
- Transform-oriented languages are non-procedural languages which provide easy-to-use structures for expressing what is desired in terms of what is known. This approach is represented by SEQUEL [6] and other similar languages which served as a basis for industrial products [24], [21].

The fourth class of relational languages is graphic. They provide users with a picture of the relations to manipulate. The user fills in tables on the screen, entering an example of what he wants in the appropriate place. Query-by-Example (QBE) [27] and CUPID [19] are the most representative of those languages. We give some examples of QBE in the next section.

Except for the two languages mentionned above, the needs of the non-specialist have been neglected before the eighties. Even SEQUEL required to be included in a host programming language. After 1980, the availability of microcomputers [12] increased the need for a more sophisticated user interface and new languages were developed with this goal: Dbase III [8], knowledge-Manager [16] Microrim [20], for instance. They first provide a very simple way and powerful functions for interacting with the data. Then, additional primitive operations make the languages more powerful, self-contained, providing for better productivity of programmers. Correspondingly, a new layer has been added to SQL in order to be directly used by end-users.

THE IMPORTANCE OF THE GRAPHICAL INTERFACE.

The advantage of a graphical query language was first noted by Zloof with QBE [27]. A query may be built up in any order the user likes which is important because the perception of a problem and the solution are generally different for different users. When an end-user understands his own needs, he wishes a direct and visual interaction with the computer. He wants to query without having to worry about procedural details. CUPID [19], implemented on top of a predicate calculus language, is another graphical language in which the user builts queries by light-pen manipulation of a set of symbols.

However, the interest in a graphical interface was not really emphasized before the beginning of the eighties when two forces converged: new developments of researches in graphics and image processing, and the availability of new interactive facilities.

- The Pictorial approach.
  There exists two major types of pictorial processing.

First, the pattern recognition and image processing community (PRIP) uses pictorial information for specific applications such as various LANDSAT processing (geography, cartography, earth resources analyses...), medical imaging, shape and fingerprint recognition,... An original image is transformed into another image (more distinct), a simplified form (sketch) or a symbolic description.

Second, the computer graphics applications lead to an (interactive) design of pictures from descriptions.

Both approaches aim at utilization of database systems, and particularly relational systems, for supporting the descriptions.

Figure 1 shows the different categories of data the user can interact with.

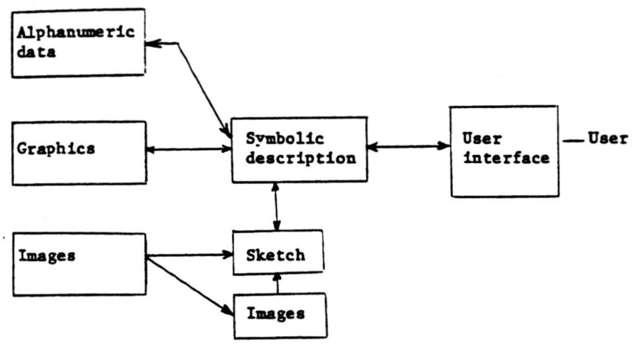


Figure 1

The need for integrating various types of data in a database system resulted in a vast effort of the different communities.

An interesting survey of the subject is given in [25]. In [17] Kunii and Harada emphasize the need for graphical interaction in CAD. In [3] Chang S. and Kunii present and discuss different approaches to pictorial database design. A fundamental step for enhancement of the man-machine communication is presented by Chang and Fu [4], [5] as an extension of QBE: Query by Pictorial-Example. The user can manipulate images or introduce pictorial examples in order to formulate pictorial queries. To illustrate the following discussion, we will consider a simple relational model concerning patients and teeth:

Patient (Pat-id, name, address, phone, age)

Pat-tooth (Frame, Pat-id, tooth-id, tooth-name)

Jaw (Frame, Tooth-id, Jaw-id)

Disease-desc (Tooth-id, Disease, tooth, date)

Pictures of teeth (for instance X-Ray) are taken separately, and a jaw presents a general view of all the teeth of a patient. They are identified by a frame number.

Q1: Give the name and age of corresponding patients having a cavity on the molar-37.

In QPE (as in QBE) the user enters Patient, Pat-tooth and Disease-desc as the table-names and fills in the columns.

Patient	Pat-id	Name	Address	Phone	Age
	X	P.			P.

Pat-tooth	Frame	Pat-id	Tooth-id	Tooth-name
		X.	Y	

D sease-desc	Tooth-id	Disease	Tooth	Date
	Y	carie	Molar -37	

Q2: Find the name of the patient whose tooth is pointed on the screen. This request can only be formulated in QPE.

Pat-tooth	Frame	Pat-id	Tooth-id	Tooth-name
		X	@	

Patient	Pat-id	Name	Address	Phone	Age
	X	P. Y			

This illustrates a capability of QPE. For more details about the syntax see [4].

Q3: Show the complete description of the jaw of patient John. Also, this query can be formulated in QPE only, using the DISPLAY (DIS) function.

Patient	Pat-id	Name	Address	Phone	Age
	X	John			

Pat-tooth	Frame	Pat-id	Tooth-id	Tooth-name
		X	Y	

Jaw	Frame	Tooth-id	Jaw-id
Dis.		Y	

- New-interactive facilities.

  Graphical aids evidently enhance the interaction. However, there is an increasing need for more sophisticated user interfaces [14], [22]. A user-interface based on interactive techniques, like the mouse and the bit-mapped display [10], [23] enables users to perform their tasks more directly and easily [26].

An important aspect of human behaviour is that it can understand the meaning of a symbolic picture (an icon) at a glance unlike the meaning of a page of alphanumeric data (even in a structured form). In various domains (medicine, scientific applications, mechanic, traffic control, ...) icons are used for an easy communication with group of users. Taking in account this capability of the humain brain to process picture more efficiently than text we designed [13] an interactive system based on icons.

AN ICON-BASED INTERFACE.

Our approach uses an iconic interface for directly handling both alphanumeric data and images. This interface is made up of a set of objects familiar to the user and of a set of operations which can be performed on these objects. We distinguish three areas on the screen (figure 2).

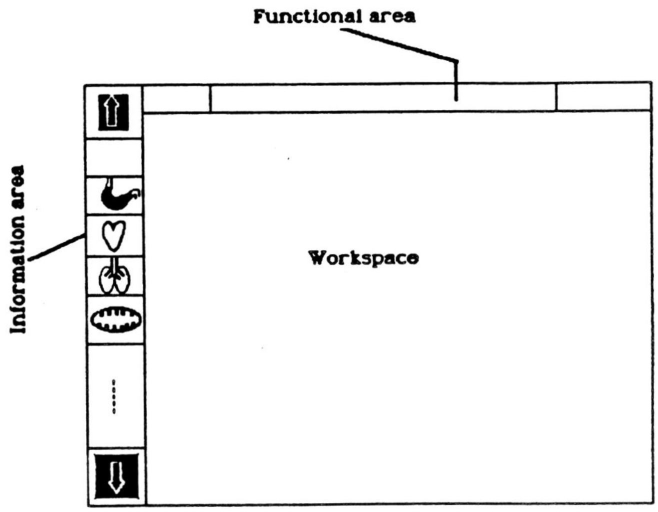


FIG.2 - The areas of interface

The information area can contain the different kind of objects to be manipulated and also the properties associated with an object. Objects represented on figure 2 concern a stomach, a heart, lungs and a jaw. The properties appear in the information area by cliking on the selected object while this one appears in the workspace. For instance, diseases such as cancer, pleuristy, pneumonia (which can affect lungs), or cavity, crawn, bridge,... (which can affect the teeth of a jaw). Notice that properties can also be represented by icons.

The functional area contains the types of operations associated with the objects. We distinguish icons commands (creating or modifying icons), retrieval and functions.

The workspace area is a window where an object (selected by cliking on the corresponding icon) can be manipulated using commands in the functional area.

Figure 3 shows a jaw of a patient with a list of properties associated in the information area. They represent diseases of teeth and the icons represented here are actually used

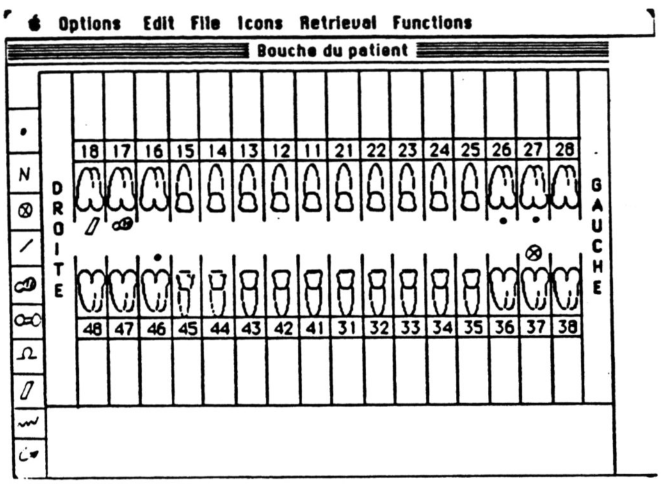


Figure 3

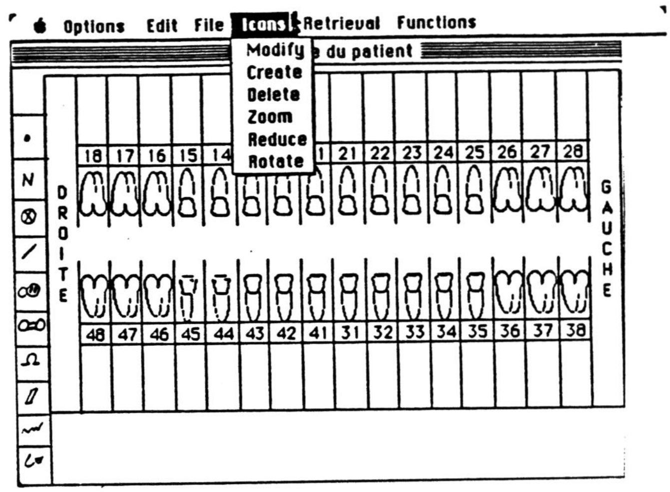


Figure 4

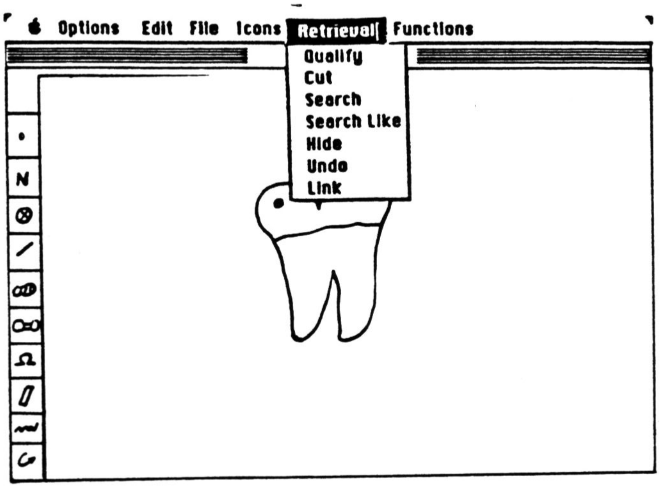


Figure 5

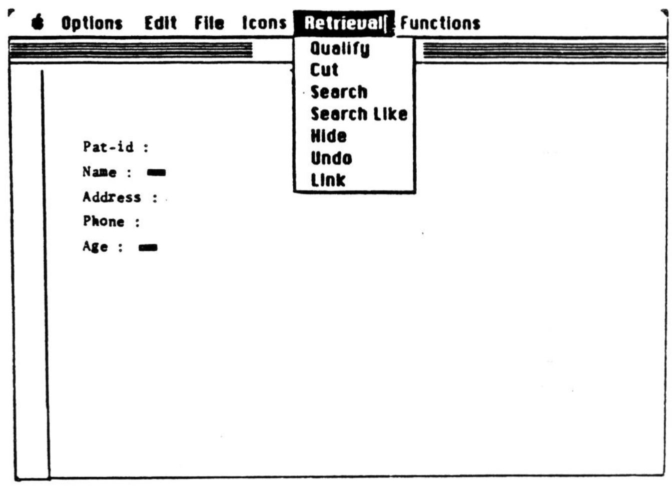


Figure 6

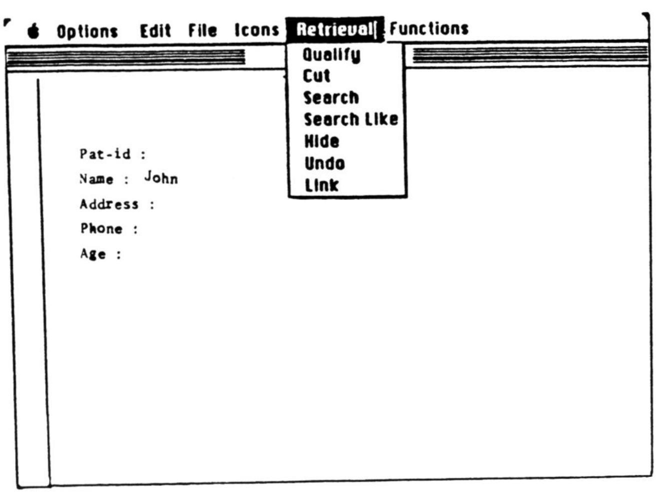


Figure 7

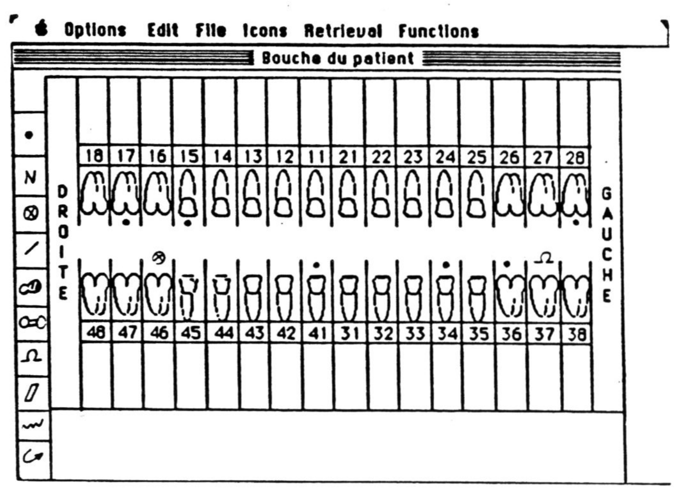
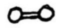


Figure 8

by the dentists. For instance, the meaning of those icons is the following:

• : cavity	 : fixed bridge
N : open contact	 : chronicle periapical periodontis
⊗ : crown	 : present root canal
/ : absent tooth	 : radicular resorption
 : present filling	 : rotation

The user can select properties and assign them to teeth using only the mouse. The resulting object is modified or can serve as a model for a retrieval as we will see in the next examples.

Teeth can also be choosen independently from the information area as indicated on figure 5 which represents molar-37.

Figures 4 and 5 exhibit some commands available in the functional area.

Let us see how the queries mentionned above can be formulated in such a system.

Q1: We select the molar-37 in the information area and assign a cavity by clicking successively on the corresponding icon (information area) and on the tooth (figure 5). Then, we select the SEARCH LIKE and LINK options in the retrieval commands in order to bind the tooth with other objects. Clicking on the patient icon - this icon is not shown here and looks like:

will bring the attributes of a patient into the workspace. The results will be obtained after a quick selection (by clicking) on Name, Age and on the SEARCH command. (figure 6).

Notice that all these operations are performed very quickly with the mouse. As in QBE a condition box is available for relational comparisons.

Q2: this query is similar to the previous manipulation, starting with a tooth of a patient.

Q3: here, we first select "John" by clicking on the patient icon and filling in the attribute name with the value "John" (figure 7). We select the SEARCH option and we bind this occurence using LINK.

Selecting the jaw icon, we obtain the result shown in figure 8 which represents John's jaw.

The SEARCH LIKE command retrieves similar pictures. As we can see on figure 3 the properties assigned to each tooth mean: "exists somewhere on the tooth". However, the precise location can be obtained by retrieving the corresponding tooth alone.

Other examples in a medical context are given in [13]. Icons are used as a simplified support for image and textual manipulation. An important fact is that it is not always necessary to consider a precise and complete image of the object to understand the meaning of the underlying information (objects are generally familiar to the user). However a simplified and direct manipulation is more important.

Some aspects of the present system have been implemented on a Macintosh, in Mac-Forth, using file access techniques. Extension to a relational system is under way together with other types of objects.

CONCLUSION.

The system we have designed is intended for users interested by the principle of "what you get is what you see". It uses an object oriented interface and a direct manipulation. The interaction is controlled by the system so that only sequences of authorized operations are allowed.

Preliminary tests have shown an easy and rapid manipulation. We are now examining its extension to other domains of application.

ACKNOWLEDGMENTS.

We would like to thank Jan Gecsei for his comments and the different students who have participated to the preliminary tests. This research was supported by the NSERC under grant A0196.

Références

1. Astrahan M & Ap. "System R: A relational approach to database management". ACM TODS 1,2, June 1976.

2. Bachman C.W. "The programmer as Navigator" CACM 16,11, November 1973.

3. Chang S.K., Kunii T. "Pictorial Database systems", Computer, November 1981.

4. Chang, N.S., Fu, K.S., "Query by Pictorial Example", IEEE. Trans. Soft. Eng. SE6(1980).

5. Chang, N.S., Fu, K.S., "Picture Query Languages for Pictorial Database Systems", Computer 14, 23-33 (1981).

6. Chamberlin, D., "A summary of user experience with SQL data sublanguage", Proc. International Conference on Data Bases, Aberdeen, Scotland, July 1980.

7. Date, C.J., Hopewell, P., "File definition and logical data independence", Proc. ACM SIGFIDET Workshop on data description access and control, 1971.

8. Dbase III [TM] "Dbase III Reference Guide" Ashton Tate, 1984.

9. Data Base Task Group of CODASYL. Programming language commitee Report, April 1971, available from BCS, ACM.

10. Ellis C., Bernal M. "Officetalk-d, an experimental office information system." Proc ACM, SIGOA conf., 1982.

11. Er-radi, M., Frasson, C., "Image Database Systems", Dép. I.R.O., Université de Montréal, Rapport interne, no. 155, Octobre 1984.

12. Frasson, C., "Bases de données sur microordinateurs". Publication 498. Département d'I.R.O., Université de Montréal, Janvier 1984.

13. Frasson, C., Erradi, M., "An Icon-Based Language for Applications in Medicine", Département I.R.O., University of Montreal, Internal report no 158, January 1985.

14. Herot C., "Graphical user interfaces" Proc of the NYU symposium on user interfaces, May 1982.

15. IMS/VS "Information management system / virtual storage" general information Manual GH20-1260, IBM, White Plains N.Y. 1979.

16. Knowledge Man, "Reference guide" Micro Data Base Systems, Lafayette, 1983.

17. Kunii T., Harada M. "SID: A system for interactive design" AFIPS Conf. Proc. 1980.

18. Mapper 5 et 6 "Reference guide". Sperry Inc., Mississauga, Ontario, 1983.

19. McDonald, N., Stonebraker, M., "CUPID, The Friendly Query Language". Proc. ACM Pacific Conference, San Francisco, (April 1975).

20. Rbase 4000, 6000 "Reference guide". Microrim Inc., Bellevue, 1984.

21. Relational Software Inc., ORACLE Introduction version 1.3 - Available from RSI, 3000 Sond Hill Road, Menlo Park, California, 94025.

22. Schneiderman B., "The future of interactive systems and the emergence of direct manipulation "Proc. of the NYU symposium on user interfaces, May 1982.

23. Smith D., Irby C., Kimball R., Verplank B., Harshem E. "Designing the star user interface. Byte vol 7, no 4 1982.

24. SQL/Data System general Information - IBM Document GH24-5012-0, 1981.

25. Tamura H., Yokoya N. "Image database systems: a survey". Pattern Recognition, Vol 17, no 1, 1984.

26. Williams G. "The Lisa Computer System "Byte vol 8,2, 1983.

27. Zloof, M.M., "Query by Example". Proc. National Computer Conference, Vol. 44. AFIPS Press 1975.