Business Graphics Interface to Databases

Kuan-Tsae Huang **Moshe M. Zloof**

IBM Thomas J. Watson Research Center

P. O. Box 218

Yorktown Heights, New York 10598

Abstract

Recently, due to the relatively inexpensive availability of displays and printers, graphic interfaces are increasingly playing an important role in analyzing stored or computed data. The integration of graphics and computerized databases is thus becoming important for the professional end user. This paper describes a method to generate graphics from relational databases which is part of the Office-by-Example (OBE) language. This method allows professionals who are non-programmers to express to desired graphical output without the help of a professional programmer.

INTRODUCTION

Although computer graphics has been in existence for many years, the high cost of displays and output equipment prevented these systems from being popular to a large number of users. In recent years, the combination of falling prices of raster color displays, memory, and processor and output devices have contributed to the significant progress in computer- generated graphics for business applications (1). The data for those business graphics are normally generated from various data bases of business departments, such as marketing, manufacturing, accounting, finance, etc. This data and its graphic representation play a major role in the process of management decision making.

As computers become directly available to the management echelon of an organization, it is becoming clear that: first, the interaction with the computer must be straight forward, namely that for most cases there is no need for the intervention of a professional programmer. Second, users must have easy, dynamic access to local and corporate data bases. By dynamic we mean an on-line, up-to-date version of the data. Third, is that the professional himself can generate various graphs from this on-line data.

Currently, there are a number of graphic software packages in the market- place that can generate various kinds of graphs from retrieved or analyzed data. These packages can generally be classified into two categories: First, there are those packages which supply a set of subroutine librar- ies. The user formulates programs by supplying data and calling the sub- routines with appropriate parameters to generate the graphs. Certainly, for a user to use these packages he/she must be a professional program- mer. Secondly, are those standalone software packages which provide the use of commands and menus for specifying features of the desired graph. In most of these packages, the user has to supply the values of the data to the system directly, lacking a dynamic link to a local or central database. Furthermore, once the graph is generated it is almost impossible to edit it interactively, such as the ability to modify colors, scales, titles, etc., directly on the output screen.

This short paper describes a method for a non-programmer-professional to directly generate graphics from a stored database system. These graphics are restricted to the domain of business graphics, e.g., various bar graphs, pie graphs and histograms. In addition, the output of these graphics can be itself stored as a new object in the database and retrieved at a later time. This

paper is organized as follows: In Section 2 we give a short overview of the Office-by- Example (OBE) language (2,3), which is being developed at IBM Thomas J. Watson Research Center. In Section 3 we describe the graphic interface. In Section 4 we describe the interactive editing facility for modifying the generated output. We will terminate the paper by a conclusion section.

OVERVIEW OF OFFICE-BY-EXAMPLE

Office-By-Example (OBE)(2,3), is a two-dimensional language and system which attempts to mimic manual procedures of business and office systems. OBE is a superset and natural extension of the Query-By-Example (QBE) database management system (4). QBE, a relational database system, is an IBM product that is used in many applications. OBE, on the other hand, is a Research project currently in various stages of architecture and development.

The OBE system has the QBE relational database management system as its base.It attempts to combine and unify aspects of:

- WORD PROCESSING
- DATA PROCESSING
- REPORT WRITING
- GRAPHICS
- IMAGES
- ELECTRONIC MAIL

The programming style of OBE is the same as that of QBE: direct programming within two-dimensional pictures of business objects. The data objects in OBE include tables, letters, menus, forms, reports, charts, graphs, and images. These objects can be manipulated on the screen in a very flexible manner. Each object can be displayed, moved, copied, and scrolled within a single window (program). It can also be stored, retrieved, and updated to and from a database. The users are able to manipulate objects and text within the objects by using expand, erase, copy, move, scroll, locate, zoom and pushdown functions. In addition, the users can easily customize a variety of menus for them- selves or others, so that one need not know more about the system than to point at menu selections that have been designed especially for them. It is expected that QBE users will require very little additional training to be able to use OBE. Currently OBE is written to support various terminals (IBM 3277, 3278 and the 3279 color terminal), and with little effort can support an all points addressable terminal. The color graphics are produced on the 3279 color display.

GRAPHICS INTERFACE

In order to generate a graph in OBE, the user had to first identify the database fields from which the graph is being generated. This data can come from fields in a single database table or from fields in multiple tables. In our system, tables are stored in the Query-By- Example (QBE) relational database system. The procedure by which a user formulates a graph is as follows: (1) the tables are displayed on the screen in a two-dimensional manner. The user then displays an appropriate "graphics template" e.g., if the user wants to generate a vertical bar graph, he displays a vertical bar graph template. (See Figure 1). Next, the user places example elements (variables) in the appropriate fields from which the data is to be drawn. Identical example elements are then entered by the user in the appropriate spaces in the template. If, for example, a red bar is desired, the example element is placed in the template's red square. Having established the links, the user places the command "P." (stands for print or display) at the heading of the template. Upon processing this program, the system will produce a corresponding graph.

Let's illustrate this by the following example. Let's assume that a manager is interested in seeing the trend in gross and net earnings since 1976. Assuming the data is stored in a table EARNINGS with attributes (COMPANY, YEAR, GROSS, NET) and the manager is seeking a vertical bar graph describing a particular company (say ABC) gross and net earnings since 1976. The formulation of such a program is shown in Figure 1 and the desired output graph is shown in Figure 2.

INTERACTIVE GRAPHICS EDITING

Having generated the output graph on the screen, the user can edit it interactively in a very flexible manner, e.g., one can modify the width and spacing of all bars by pressing a single function key; the vertical scale can be changed by expanding or shrinking the vertical coordinate; the bar colors can also be modified; headings, footings and comments can also be added. The edited output can be either stored in the database for future use or can be superimposed into text or image objects. Figure 3 shows the change in the vertical scale. Figure 4 shows an example of a change in the bar's width.

CONCLUSION

We have presented here an easy method for interfacing graphics to database management systems. Its advantage is that users who are non-programmers can write programs to directly pull data from large corporate databases and generate and edit the graphs interactively.

The elegant, two-dimensional programming approach of Query-By-Example and Office-By-Example, as was shown in this short example, makes the formulation of such graphics programs easy and straight forward.

(1) W. C. House, "Interactive Computer Graphics Systems" Petrocelli Books, Inc., 1982.

(2) M. M. Zloof, "QBE/OBE: A Language for Office and Business Automation", COMPUTER Magazine, IEEE, Inc., 1981

(3) M. M. Zloof, "Office-By-Example: A Business Language that Unifies Data and Word Processing and Electronic Mail", IBM Systems Journal, Vol. 22, No. 3, 1982, pp. 272-304

(4) M. M. Zloof, "Query-By-Example", AFIPS Conference Proceedings, National Computer Conference 44, 1975, pp. 431-438

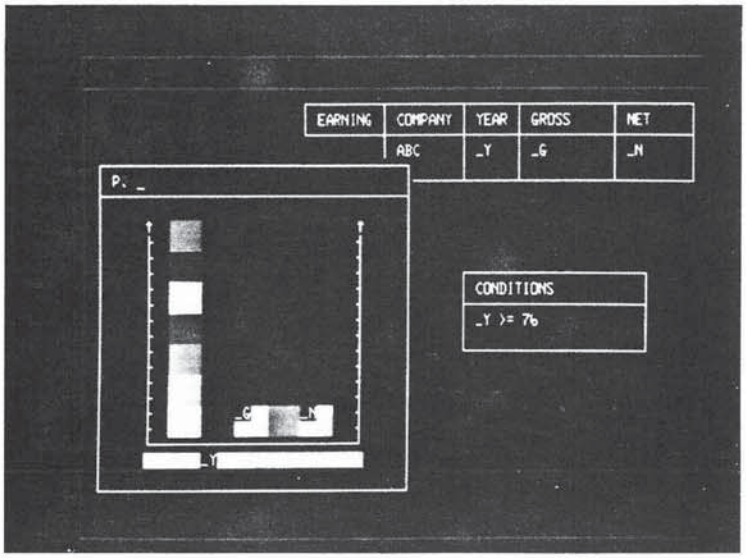

Figure 1 The User Created Program

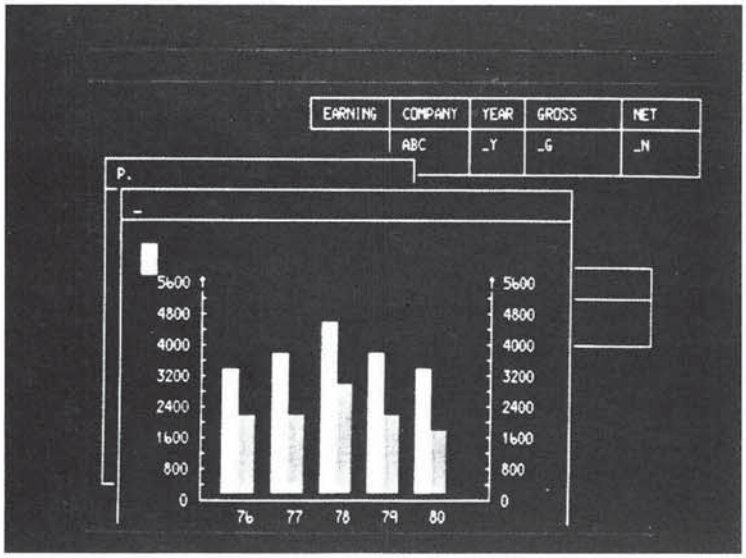

Figure 2 The Output Graphics

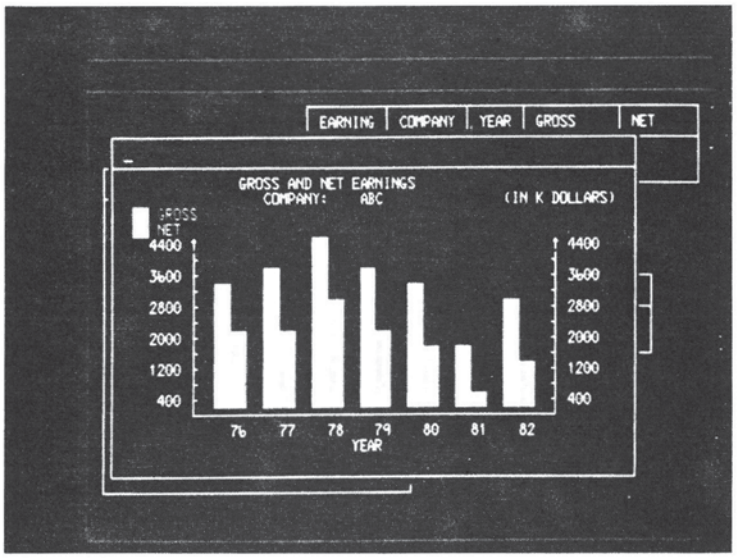

Figure 3 Change in The Vertical Scale

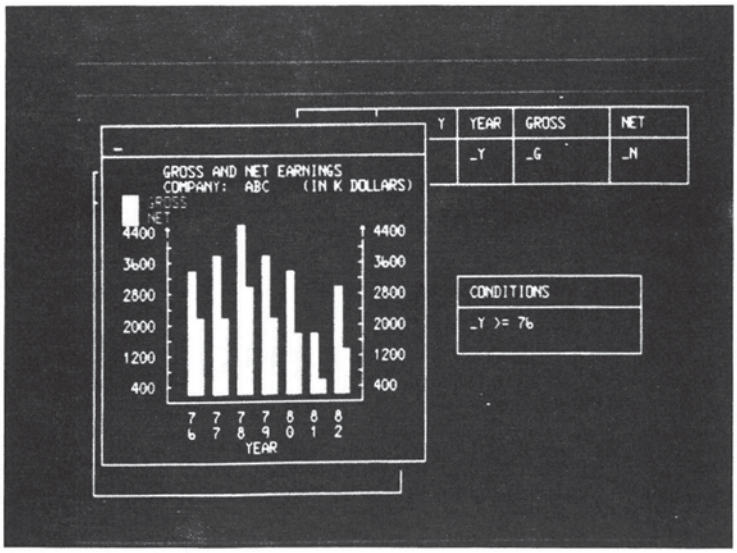

Figure 4 Change in The Bar Width