# DATABASE AND GRAPHICAL OBJECTS IN ENGINEERING DESIGN

**Algirdas SOKAS**

Vilnius Gediminas Technical University,
Sauletekio al. 11, LT10223 Vilnius, Lithuania
email: algirdas.sokas@ fm.vtu.lt

**Abstract.** This paper introduces graphics programming in the Computer Aided Design environment. Using Visual Basic for Applications programming language methods, algorithms and procedures the following design problems can be solved: attaching information from database to a graphical object being drawn, reading information from certain graphical objects of a drawing and collecting, storing, presenting this information in database formats, carrying mathematical operations with read information. The algorithm has been formed to attach database record information to graphical object which gives the procedures how to: select one graphical object record in the database, create extended data of new graphical object, draw programming graphical object depending on extended data and scale in the drawing. The algorithm has been created to read the information from graphical objects and to collect extended data in the database records: select graphical objects in the drawing, take the first object, verify whether an object has extended data and collect extended data in the database record. The following phenomena are presented: procedure fragments and results dependent on selected record in the database of graphical object parameters; creation of extended data for new graphical object; and information transfer to another database. The following questions have been answered how the prepared procedures work to collect drawing information in the database with different number of graphical objects and with and without sorting procedure. The example of a facade drawing and database with doors information with program written in Visual Basic for Applications language has been given. The general purpose of this paper is to contribute to a debate about the possibility of Visual Basic for Applications graphics programming.

**Key Words:** data exchange technology, extended data, graphics programming, Visual Basic for Applications.

## 1. Introduction

Many engineers draw drawings, collect graphical objects, store information, present projects and carry calculations. Today personal computer with new software tools is irreplaceable work instrument for a specialist. Designers often work with one computer-aided design (CAD) tool. But today in the CAD environment engineer can program and use other applications [ 1, 2, 3] as well.

Worldwide, the most widespread computer-aided store and access information tool is database. First industrial program [4] of database named Information Management System (IMS) was created by International Business Machines (IBM) in 1968. It is a good example of a successful database management system based on the hierarchical data model [5]. Edgar Frank Codd, a mathematician from IBM created first relational database model in 1969 [6].

Initially, some database systems such as Dbase, FoxPro, Clipper, Paradox and Clarion in operating system DOS were used for personal computers. Nowadays, Microsoft Office program Access is a very popular database in Windows system.

Graphical system AutoCAD (Automated Computer Aided Drafting and Design) [7, 8, 9] is widely used in the world because it has open architecture and programmers can understand many system files. In the system environment a user can not only operate other programming languages using standard drawing and design commands but can also create his own functions. In the AutoCAD environment we can program with Visual Basic for Applications language [10, 11, 12].

ISSN 1644-9363 / PLN 15.00 © 2005 PTGiGI

Integrated CAD environment and data exchange capability enlarge designers' possibilities a lot. Computer users wanted a technology that would enable automated designing. Microsoft made open database connectivity (ODBC) technology in 1991 [15, 16]. It was the first universal method to access database.

Microsoft created object linking and embedding database (OLE DB) technology in 1996 [17, 18]. By means of this technology we can use functional capacity of one application without leaving the other program.

Modern data exchange technology is ActiveX [14, 19, 20]. This technology allows exchanging information easily with AutoCAD application and other ActiveX enables applications as database [21]. Microsoft Windows technology ActiveX data objects (ADO) uses e-mail and Internet for databases and for files systems. First version of ADO was created in 1996.

Formulation of problem for solving these design problems:

- Attach information from database to a graphical object being drawn;
- Read information from certain graphical objects of a drawing and collect, store and present this information in database.

## 2. Programming with objects from database

The computer can only produce what it has been fully instructed to do. Programming of graphical objects can produce many different graphical results depending on parameters. We can prepare these parameters using another application that is different from the one we draw in. it refers to such construction component type database such as beam, wall, column, slab [16, 18]. We can draw in the CAD system but access parameters in the database [22, 23].

To prepare VBA project [21], first we have to create an instance of the application and to declare a variable that will represent the connection with other application, line (1). In the second line (2) database record set is declared. Secondly, we have to create a set of statements with declared variables.

```
Dim con As ADODB.Connection                                      (1)
Dim rec As ADODB.Recordset                                       (2)
          Set con = New ADODB.Connection                         (3)
           With con                                              (4)
                    .Provider = "Microsoft.Jet.OLEDB.4.0"        (5)
                    .Open "C:\My Documents\ Door.mdb"            (6)
                    End With                                     (7)
Set rec = New ADODB.Recordset                                    (8)
```

The new keyword in the third line (3) starts a new session of database, adds provider (5) and (6) opens database from disk with name "Door.mdb". Line (8) sets new records.

## 3. Algorithms

Let's assume that in the database we have *n* graphical objects, *n* records (rows) with number and text information of these graphical objects. Let's suppose that in the database we have *m* fields (columns) with information of graphical objects. All geometrical and other information about a given graphical object is found in one record of database. All given information such as name, producer of all graphical objects is found in the fields of database.

Therefore, programming procedure for attaching information from database record to a graphical object being drawn is needed. Let's form an algorithm to attach database record information to a graphical object:

1. Select one graphical object record in the database;
2. Create extended data of a new graphical object;
3. Draw programming graphical object depending on extended data and scale in the drawing.

Figure 1 shows programming connections with databases and graphical objects.

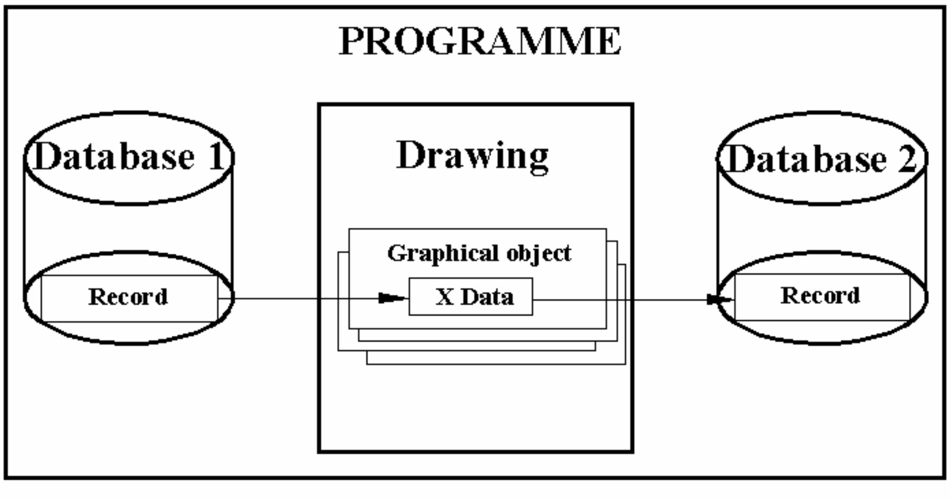


Figure 1: Programming connections with databases and graphical objects

Graphical object parameters are available in the database and then drawing can be done in CAD system. Let's assume we have k graphical objects in the drawing. Some of the objects are with extended data. There is p number of such objects. So we need programming procedures for reading information from certain graphical objects of a drawing and collecting, storing and presenting this information in the database formats.

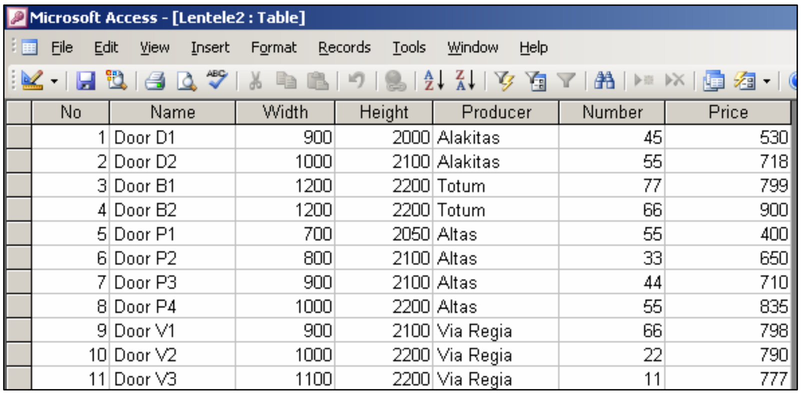
Let's form an algorithm to read the information from graphical objects and to collect extended data in the database records:

1. Select *k* graphical objects in the drawing
2. Take the first object
3. Verify whether the object has extended data
4. If it does not have go to position 2 (*k*-*p*); if it has go to position 5 (*p*)
5. Collect extended data in the database record.

In the database we can sort records by name objects and add objects with the same name. We can prepare summary report of graphical objects in the drawing.

## 4. Access information in database

Programming with Visual Basic for Applications language in the AutoCAD environment [10, 11] and access parameters in the database table. Figure 2 shows Microsoft Access table with drawing information.



| No | Name | Width | Height | Producer | Number | Price |
|---|---|---|---|---|---|---|
| 1 | Door D1 | 900 | 2000 | Alakitas | 45 | 530 |
| 2 | Door D2 | 1000 | 2100 | Alakitas | 55 | 718 |
| 3 | Door B1 | 1200 | 2200 | Totum | 77 | 799 |
| 4 | Door B2 | 1200 | 2200 | Totum | 66 | 900 |
| 5 | Door P1 | 700 | 2050 | Altas | 55 | 400 |
| 6 | Door P2 | 800 | 2100 | Altas | 33 | 650 |
| 7 | Door P3 | 900 | 2100 | Altas | 44 | 710 |
| 8 | Door P4 | 1000 | 2200 | Altas | 55 | 835 |
| 9 | Door V1 | 900 | 2100 | Via Regia | 66 | 798 |
| 10 | Door V2 | 1000 | 2200 | Via Regia | 22 | 790 |
| 11 | Door V3 | 1100 | 2200 | Via Regia | 11 | 777 |

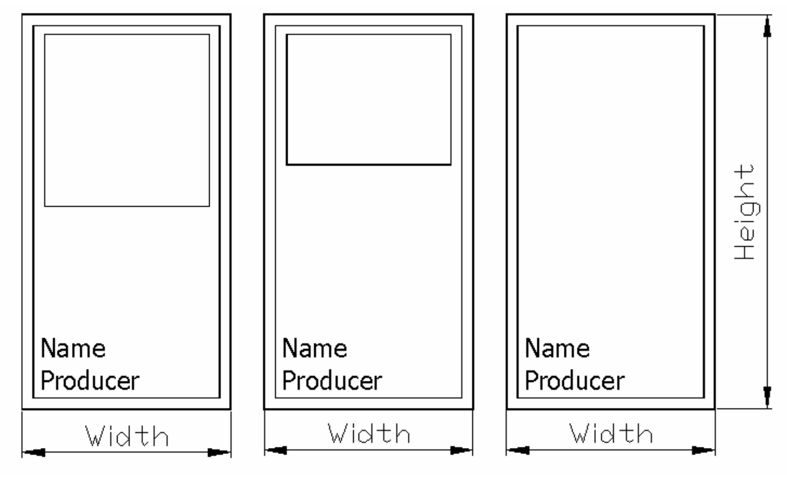
Figure 2: Database table and drawing information

We can select one graphical object record in the database (Figure 2) of graphical object parameters:

```
rec.Open "Lentele2", con, adOpenKeyset, adLockBatchOptimistic                      ( 9)
ListBox1.Clear                                                                     (10)
Do Until rec.EOF                                                                   (11)
ListBox1.AddItem rec(0) & " " & rec(1) & "  " & rec(2) &  "  " & rec(3) & "  " & rec(4)(12)
rec.MoveNext                                                                       (13)
Loop                                                                               (14)
```

In the ninth line (9) database table "Lentele2" is open depending on connection variables. In the next lines one graphical object record in the database of graphical object parameters is selected.

Prepare database with geometrical and other information about given graphical object. Information of graphical objects in database (Figure 3) can be easily edited and new records can be introduced.



## 5. Drawing with information from database

Parameters can be prepared in other application than we draw. We draw in the CAD system but parameters are stored in the database.

The next procedure selects one graphical object record with $m=5$ fields in the database of graphical object parameters and enables creation of extended data of new graphical objects:

```
Dim DataType(0 To 1) As Integer                                        (15)
Dim Data(0 To 1) As Variant                                            (16)
FunctionDoor TextBox2, TextBox3, objDoor                               (17)
If objDoor.ObjectName = "AcDbPolyline" Then                            (18)
        DataType(0) = 1001: Data(0) = "DoorData"                       (19)
        DataType(1) = 1000                                             (20)
                        Data(1) = TextBox1 & ";" & CStr(TextBox2)& _
         ";" & CStr(TextBox3) & ";" & TextBox4                         (21)
        objDoor.SetXData DataType, Data                                (22)
End If                                                                 (23)
```

In the first two lines (15, 16) variables are declared. The third line (17) is function of which we can draw graphical objects depending parameters from database (Figure 5). Lines (18-23) begin procedure of creating extended data of a new graphical object.

We've solved the first designing problem how to attach information from database to a graphical object being drawn.

## 6. Collecting drawing information in the database

Next design problem is to read information from certain graphical objects of a drawing and collect, store and present this information in database formats.

We can select graphical objects with extended data and write information in a matrix and sort by name [13]. The next procedure finds the same names of graphical objects in matrix and calculates their number. This new information is written to other matrix MM, in which the first column has names of graphical objects from drawing and the second column has the corresponding number of these objects. This new information is necessary to be written to database. Write matrix MM information to other database:

```
For i = 1 To objects_count                              (24)
   rec2.AddNew                                          (25)
         rec2.Fields("Code").Value = “D” & i            (26)
         rec2.Fields("Name").Value = MM(i, 1)           (27)
         rec2.Fields("Count").Value = MM(i, 2)          (28)
   rec2.Update                                          (29)
Next i                                                  (30)
```

The first and last lines (24, 30) present the beginning and the end of a cycle. The second line (25) creates new record with a set of new information. Lines (26-29) give instructions to database. That is other database Door2 (Figure 1) where drawing information is stored.

## 7. Analysis of procedures

The two questions have been asked: how the prepared procedures work for collecting drawing information in the database with different number of graphical objects and with and without sorting procedure and how much time is necessary to carry out all algorithms to create database. The results of experiments are shown in Figure 4.

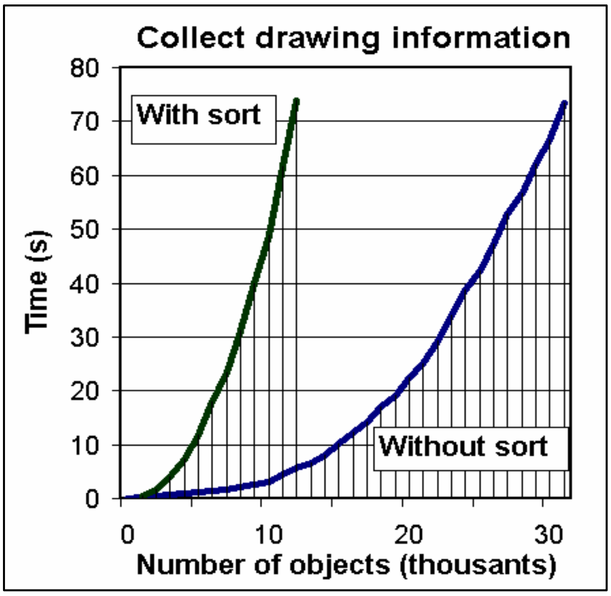


Figure 4: Comparison of the results of prepared procedures speed

We have selected graphical objects with extended data and calculated number of objects and performed all area set of experiments. The methods have been compared in time (seconds) of performance calculating thousands uniform graphical objects with PC Pentium III, 600 MHz, RAM 200 MB.

The first set of experiments involved sorted and calculated number of objects. 5000 uniform graphical objects were sorted and calculated in 11.8 seconds, 10000 objects were calculated in 48.88 seconds, 12000 objects were calculated in 73.98 seconds.

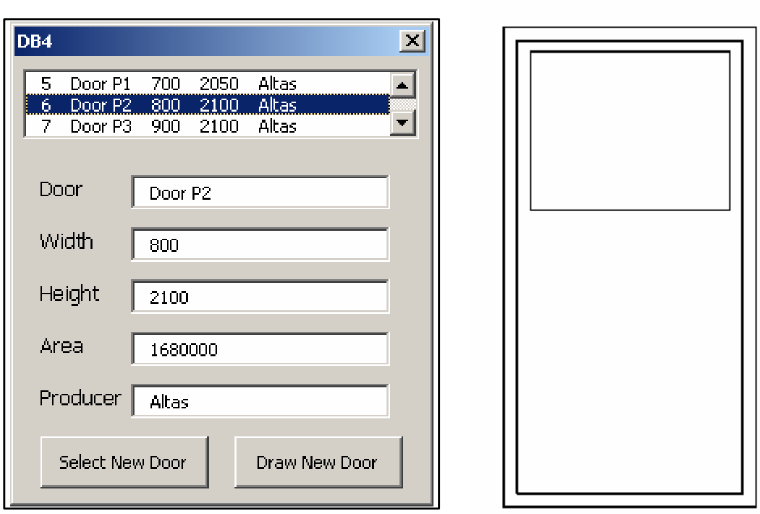
The second set of experiments involved calculated number of objects. 5000 uniform graphical objects have been calculated in 1.18 second, 10000 objects have been calculated in 3.07 seconds, 20000 objects have been calculated in 22.42 seconds and 30000 objects have been calculated in 73.29 seconds.

After 72 seconds 12000 objects have been sorted and calculated, but 30000 objects have only been calculated.

## 8. Example

For an engineer a task of drawing a facade and sections of a doors is a common routine. This work is done faster if special programming procedures for visualizing graphical objects are used. The book [9] explains how to prepare an engineering drawing with AutoCAD and programming language AutoLISP. Let’s draw a facade and section drawings and prepare database with doors information with program written in VBA language.

This paper does not discuss sections drawings with some problems with different dimensions and materials. In the sections drawings we have few different doors; their properties are described by program’s procedures and can be called from the menu (Figure 5).



DB4

```
5   Door P1   700   2050   Altas
6   Door P2   800   2100   Altas
7   Door P3   900   2100   Altas
```

Door: Door P2

Width: 800

Height: 2100

Area: 1680000

Producer: Altas

Figure5: Programs menu and drawing

At this stage only problem dealing with automation formation doors in the facade and sections according getting information from database and updating drawing will be solved. Program's menu and procedure results from getting information from database and updating drawing are shown in Figure 5.

When drawing doors in the facade and sections of a building we specify one point of the doors cone. When we want to see all the variants of doors, we can create graphical menu in which slides would show objects themselves.

Integrated CAD environment (AutoCAD) and MS Access database, and data exchange capability as ActiveX technology with Visual Basic for Applications language enlarge possibilities in engineering design.

## 9. Conclusions

The paper presents methods, algorithms and procedures for solving such designing problems as: attaching information from database to a graphical object being drawn, reading information from certain graphical objects of a drawing and collecting, storing and presenting this information in database formats, carrying mathematical operations with read information.

An algorithm has been created to attach database record information to graphical object which gives the procedures of the following steps: select one graphical object record in the database, create extended data new graphical object, draw programming graphical object depending on extended data and scale in the drawing. An algorithm has been formed to read the information from graphical objects and to collect extended data in the database records which gives the procedures of the following steps: select graphical objects in the drawing, take first object from beginning, verify whether object has extended data, collect extended data in the database record.

The following problems have been presented: procedure fragments and results dependent on selected record in the database of graphical object parameters; creation of extended data for new graphical object; and information transfer to another database. The procedures have been prepared to work out how collect drawing information in the database with different number of graphical objects and with and without sorting procedure.

The example door drawing and database with doors information with program written in VBA language has been given.

**References**

[1] Kunigahalli R., Russell J. S.: *Framework for development of CAD / CAC systems*. Automation in Construction, Vol. 3, No 3, 1995, p. 327-340.

[2] Kim I., Liebich T., Maver T.: *Managing design data in a integrated CAAD environment: a product model approach*. Automation in Construction, Vol. 7, No 1, 1997, p.35-53.

[3] Khemlani L., Timerman A., Benne B., Kalay Y. E.: *Intelligent representation for computer-aided building design*. Automation in Construction, Vol. 8, No 1, 1998, p. 49-71.

[4] Karpova T.: *Database: models, designs, realization*. Sankt-Peterburg "Piter" 2002, 304 p. (in Russian).

[5] Anumba C. J.: *Data structures and DBMS for computer-aided design systems*. Advances in Engineering Software, No 25, 1996, p. 123-129.

[6] Codd E. F.: *A relational model of data for large shared data banks*. Communications of the ACM (CACM), No 13 (6), 1970, p. 377-387.

[7] Sampaio A. Z.: *Geometric modelling of box girder desk for integrated bridge graphical system*. Automation in Construction, Vol. 12, No 1, 2003, p. 55-66.

[8] Miles J. C., Gray W. A., Carnduff T.W., Santoyridis I., Faulconbridge A.: *Versioning and configuration management in design using CAD and complex wrapped objects*. Artificial Intelligence in Engineering, Vol. 14, No 3, 2000, p. 249-260.

[9] Čiupaila L.: *Applied graphics in building engineering*. Vilnius "Technika" 2002, 312 p.(in Lithuanian).

[10] Cottingham M.: Mastering AutoCAD VBA. "SYBEX" 2001, 656 p.

[11] Sokas A.: *Engineering graphics and Visual Basic*. Engineering graphics BALTGRAF-6.Proceedings of the Sixth International Conference, Riga RTU 2002, p. 82-87.

[12] Sokas A.: *Graphical objects and information in engineering design*. Biuletyn Polskiego Towarzystwa Geometrii i Grafiki Inžynierskiej, Zeszyt 13, 2003, str. 54-56.

[13] Sokas A.: *Automatized formation of specifications in engineering drawings*. The Journal of Polish Society for Geometry and Engineering graphics, Vol. 14, 2004, p. 58-63.

[14] Sokas A., Vilkevič V.: *Drawing and data exchange technology in engineering design*. Engineering Graphics BALTGRAF-7. Proceedings of the Seventh International Conference, Vilnius "Technika", 2004, p. 104-110.

[15] Reed P.: *Using ODBC to Access Nontabular Data*. DBMS *and* Internet Systems, Vol. 9, No 4, 1996 (http://www.dbmsmag.com/9604d54.html).

[16] Hadikusumo B. H. W., Rowlinson S.: *Integration of virtually real construction model and design-for-safety-process database*. Automation in Construction, Vol. 11, No 5, 2002, p. 501-509.

[17] North K.: Programming with OLE and ActiveX. DBMS *and* Internet Systems. Vol. 9, No 12, 1996 (http://www.dbmsmag.com/9611d14.html).

[18] Dawood N., Sriprasert E., Mallasi Z., Hobbs B.: *Development of an integrated information resource base for 4D/VR construction processes simulation*. Automation in Construction, Vol. 12, No 1, 2002, p. 123-131.

[19] Mazurin A., Mazurina C.: *ActiveX for AutoCAD R14: a practical guide*. Objects CAD and graphics, No 5, 1998, p. 92-96 (in Russian).

[20] Lalani S., Chandak R.: *ActiveX programmer's library*. Minsk "Popuri" 2000, 624 p.(in Russian).

[21] Gunderloy M.: *Visual Basic. Developer's Guide to ADO*. San Francisco "SYBEX" 2000, 336 p.

[22] Chau K. W., Cao Y., Anson M., Zhang J.: *Application of data warehouse and Decision Support System in construction management*. Automation in Construction, Vol. 12, No 2, 2002, p. 213-224.

[23] Park J. H., Storch R. L.: *Pipe-routing algorithm development: case study of a ship engine room design*. Expert Systems with Applications, Vol. 23, No 3, 2002, p. 299-309.

## BAZA DANYCH I OBIEKTY GRAFICZNE W PROJEKTOWANIU INŻYNIERSKIM

Praca dotyczy oprogramowania grafiki inżynierskiej w środowisku CAD. Używając języka Visual Basic for Applications (VBA) opracowano metody, algorytmy i procedury do rozwiązywania następujących problemów: przywiązanie informacji z bazy danych do narysowanego obiektu graficznego; odczytanie informacji z obiektów graficznych narysowanych, zachowanych w bibliotece; prezentacja tych informacji w różnych formatach danych; wykonanie operacji matematycznych na odczytanych danych. W pracy zamieszczono fragmenty procedur realizujących powyższe procesy oraz przykład rysowania fasady budynku z wykorzystaniem języka VBA.