6th International Conference on Applied Informatics
Eger, Hungary, January 27–31, 2004.

Programming Graphical Objects And Spreadsheet In Engineering Design

Algirdas Sokas

Department of Engineering Graphics, Vilnius Gediminas Technical University
e-mail: algirdas.sokas@ fm.vtu.lt

Abstract

This article introduces you to graphics programming in the AutoCAD environment. Using Visual Basic Application (VBA) programming language prepared methods, algorithms and procedures for solving these design problems: reading information from certain graphical objects of a drawing and collecting, storing, presenting this information in spreadsheet formats, fulfilling mathematical operations with read information, and with information from spreadsheet updating the graphical objects in the drawing. The general purpose of this paper is to contribute to such a debate about possibility of Visual Basic Application graphics programming.

Key Words and Phrases: Graphics programming, extended data, Visual Basic Application, data exchange technology.

1. Introduction

For Many engineers draw drawings, collects graphical objects, store information, present projects and executes calculations. Today personal computer with new software tools is irreplaceable work instrument for a specialist; it increases work productivity. 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].

In the world most widespread computer-aided calculations tool is spreadsheet. Den Briklin and Bob Frankston created first spreadsheet program [3] named VisiCalc (full name - visible calculation) in 1979. They were graduates of Massachusetts Technological Institute.

Then there were other spreadsheet systems as Lotus 1-2-3, Super Calc, Quattro Pro and Microsoft Works in operating system DOS. Now Microsoft Office program Excel is very popular spreadsheet in Windows system.

Graphical system AutoCAD [4, 5, 6] is widely used in the world because it has open architecture and programmers can understand many system files. In the system's environment user can operate other programming languages using standard draw and model commands but can create own functions.

In the AutoCAD environment we can program with Visual Basic language. Visual Basic (VB) is "Microsoft" object-oriented programming language. Visual Basic for Applications is another Basic language version which is now included to many "Microsoft" programs as Excel and Access. Since AutoCAD 14 we can use VBA for graphics programming [7, 8, 9]. The main difference between VBA and VB is that VBA runs in the same process space as AutoCAD, providing an AutoCAD-intelligent and very fast programming environment.

Integrated CAD environment and data exchange capability [10, 11] enlarge designers possibilities greatly. In the article [12] we found that in the XXI century personal computers were with personal software.

1.1. Formulation of problem

Prepare methods, algorithms and procedures for solving these design problems:

— Read information from certain graphical objects of a drawing and collect, store, present this information in spreadsheet formats,

— Fulfilling mathematical operations with the read information,

— With information from spreadsheet update the graphical objects in the drawing.

1.2. Data exchange technology

We can do data copy and cut operations in many Windows systems applications. That we can do among different programs. For example in this article figures are created in the AutoCAD environment and then copied and pasted into this Word document. This operation is done by hands and it uses computer's special memory. It is data exchange technology by hands.

Computer users wanted technology that fulfills copy and paste without hands. Company "Microsoft" created dynamic data exchange (DDE) technology where two programs can exchange data between themselves. One application was client (receiving of information) and the other server (source of information).

Later "Microsoft" created object linking and embedding (OLE) technology. Using this technology we can use functional capacity of one application but not leaving the other program.

Modern data exchange technology is ActiveX [13, 14]. This technology allows exchanging information easily with AutoCAD applications and other ActiveX enabled applications as Microsoft Excel [15].

2. Programming with objects from spreadsheet

The computer can only produce what it has been fully instructed to do. Programming graphical objects [16] can produce many different graphical results depending from parameters. We can prepare these parameters in another application that is different from the one we draw in. We can draw in the CAD system but change parameters in the spreadsheet [17].

To prepare VBA project [15], firs we have to create an instance of the application and to declare a variable that will represent the other application, line (1).

```
Dim ExcelApp As Excel.Application                    (1)
Dim ExcelWorkbook As Object                          (2)
Dim ExcelSheet As Object                             (3)
```

In the second line (2) declare Excel workbook and in the third line (3) declare workbook sheet as object. Second, we have to create a set of statements with declared variables.

```
Set ExcelApp = New Excel.Application                 (4)
Set ExcelWorkbook = Excel.Workbooks.Add              (5)
Set ExcelSheet = Excel.ActiveSheet                   (6)
ExcelSheet.Name = "Info"                             (7)
```

The new keyword in the fourth line (4) starts a new session of Excel, adds workbook (5) and activates sheet (6) with name “Info”, line (7).

2.1. Algorithms

Assume in the drawing we have n graphical objects. Part of the objects is with extended data. There is p number of such objects. Information in the extended data is t types. One graphical object can have all t types of information. So we need programming procedures for reading information from certain graphical objects of a drawing and collecting, storing, presenting this information in spreadsheet formats, attaching information from spreadsheet to a graphical object being drawn.

Form algorithm to read the information from graphical objects and to collect extended data in spreadsheet:

1. Select n graphical objects in the drawing;
2. Take first object from beginning;
3. Verify whether object has extended data;
   if does not have, go to position $2(n-p)$;
   if has, go to position $4(p)$;
4. Collect extended data in spreadsheet.

So in the (Fig 1) $n = 7$ graphical objects are shown: four circles with radius and areas presented with $A1 \ldots A4$ variables, three rectangular with areas presented with $S1 \ldots S3$ variables. Meanings are written to object's extended data. Types of extended data are $t = 2$, objects with extended data are $p = 7$.

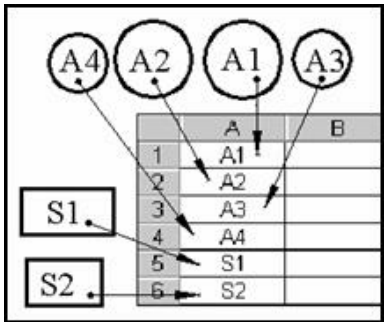


Figure 1: Graphical objects extended data and spreadsheet

Form an algorithm to attach spreadsheet information to graphical object:

1. Edit extended data in spreadsheet;
2. Select all p CAD objects with extended data;
3. Attach information to CAD objects.
4. Update graphical object.

2.2. List information in spreadsheet

Programming with Visual Basic for Application language in the AutoCAD environment [7,8,18]. Procedure results for exporting extended data of graphical objects to spreadsheet are shown in Fig 2.

Procedure fragment for exporting extended data of graphical objects in spreadsheet:

```
Dim iRowNum As Integer, Header As Boolean                          (08)
Dim objEntity As AcadEntity                                        (09)
Dim tipas As Variant, ddd As Variant                               (10)
iRowNum = 1                                                        (11)
For Each objEntity In objModelSpace                                (12)
  With objEntity                                                   (13)
    If StrComp(.EntityName, "AcDbCircle", 1) = 0 Then              (14)
      .GetXData "", tipas, ddd                                     (15)
      On Error Resume Next : Err.Clear                             (16)
      objWorksheet.Cells(iRowNum + 1, 1).Value = iRowNum - 1       (17)
      objWorksheet.Cells(iRowNum + 1, 2).Value = Round(ddd(2), 2)  (18)
      objWorksheet.Cells(iRowNum + 1, 3).Value = Round(ddd(4), 3)  (19)
      iRowNum = iRowNum + 1                                        (20)
    End If                                                         (21)
  End With                                                         (22)
Next objEntity                                                     (23)
```

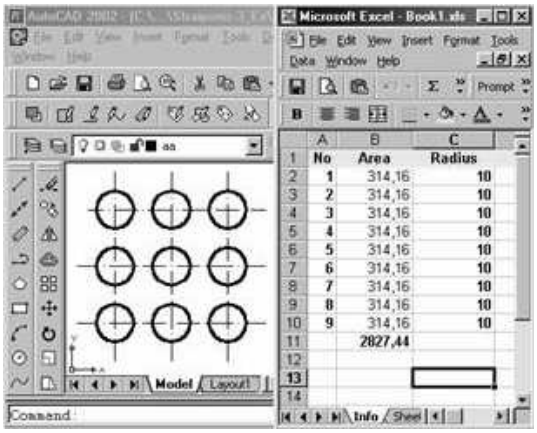


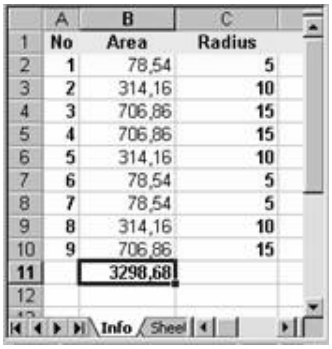
Figure 2: Example CAD and spreadsheet

In the (8-10) lines variables are declared. In the second line begin procedure forms spreadsheet first row. In the (12-23) lines for all graphical objects in the drawing which are circles and have extended data, extended data is exported to spreadsheet.

2.3. Change information in drawing

We can easily edit information of graphical objects and calculate other parameters in spreadsheet. Results for changed information of graphical objects in spreadsheet are shown Fig 3. The next procedure fragment selects graphical objects with extended data, imports information from spreadsheet and changes old information in the drawing:

```
For Each objEntity In ThisDrawing.ModelSpace                    (24)
With objEntity                                                  (25)
If StrComp(.EntityName, "AcDbCircle", 1)= 0 Then                (26)
    .GetXData "", tipas, ddd                                    (27)
  On Error Resume Next : Err.Clear                              (28)
ddd(2)=ExcelSheet.Cells(iRowNum + 1, 2).Value                   (29)
ddd(4)=ExcelSheet.Cells(iRowNum + 1, 3).Value                   (30)
    .SetXData tipas, ddd                                        (31)
  iRowNum = iRowNum + 1                                         (32)
End If : End With : Next objEntity                              (33)
```



	A	B	C
1	No	Area	Radius
2	1	78,54	5
3	2	314,16	10
4	3	706,86	15
5	4	706,86	15
6	5	314,16	10
7	6	78,54	5
8	7	78,54	5
9	8	314,16	10
10	9	706,86	15
11		3298,68	
12			

Info Sheet

Figure 3: Spreadsheet with drawing information

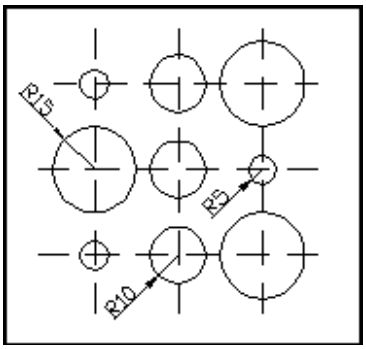


Figure 4: Changed drawing information

In the (24-33) lines for all graphical objects in the drawing, which are circles and have extended data, information is imported from spreadsheet and old extended data is changed.

Procedure results for importing information from spreadsheet and updating drawing are shown in Fig 4.

We can prepare parameters in another application than we have to draw. We can draw in the CAD system but change parameters in the spreadsheet.

3. Example. Changed spacer holes

For a mechanic engineer drawing an engine spacer is an often done work. This work is done faster if special programming procedures for visualizing graphical objects are used. The book [6] explains how to prepare an engineering drawing with AutoCAD and programming language AutoLISP. Lets fulfil engine spacer drawing and prepare spreadsheet with holes information with program written in Visual Basic Application language.

In the engine spacer we have some problems with different dimensions and materials; however, we won't talk about that. In the engine spacer we have few different holes; their properties describe program's procedures and can be called from the menu (Fig 5). We will solve only problems dealing with automation formation spreadsheet of holes radius (Fig 3) according to results of a drawn spacer of an engine and possibility to change drawing according to new radius.

Program's menu and procedure results for importing information from spreadsheet and updating drawing are shown in Fig 4. Procedure results for importing different information from spreadsheet and updating drawing are shown in Fig 6.

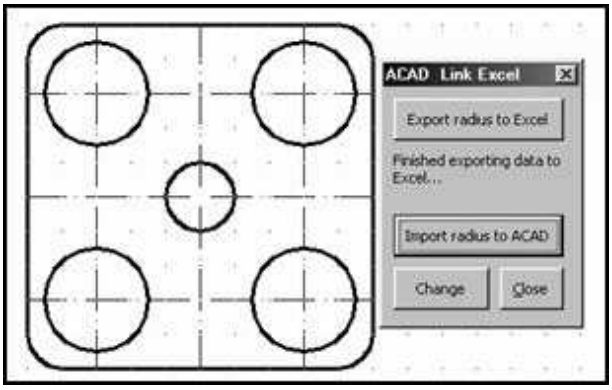


Figure 5: Drawing and program's menu

Integrated CAD environment (AutoCAD and Excel spreadsheet) and data exchange capability as ActiveX technology with Visual Basic Application programmers language enlarge designers possibilities.

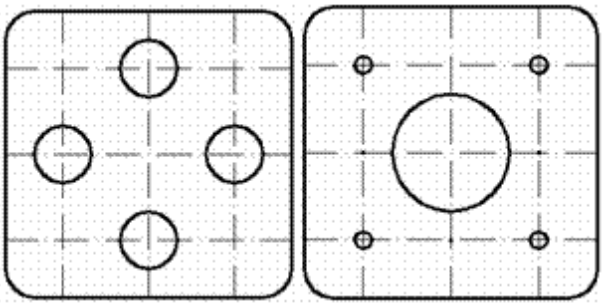


Figure 6: Drawings with changed information

4. Conclusions

Prepare methods, algorithms and procedures for reading information from certain graphical objects of a drawing and collect, store, present this information in spreadsheet formats, fulfill mathematical operations with read information, and with information from spreadsheet update the graphical objects in the drawing. Prepare example engine spacer drawing and spreadsheet with holes information with program written in Visual Basic Application language.

References

[1] Kunigahalli R., Russell J. S. Framework for development of CAD / CAC systems. Automation in Construction, Vol. 3, Num. 3 (1995), 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, Num. 1 (1997), 35-53.

[3] Agafonov A. Borning VISICALC. Computer press (Kompiuter press), Num. 7 (1990), 65-67 (in Russian).

[4] Terpugov A., Kuznecov. AutoCAD rise the designer idea. CAD and Graphics (SAPR i grafika), Num. 4, (1999), 5- 7 (in Russian).

[5] Sampaio A.Z. Geometric modeling of box girder desk for integrated bridge graphical system. Automation in Construction, Vol. 12, Num. 1 (2003), 55-66.

[6] Čiupaila L. Applied graphics in building engineering (Taikomoji grafika statybos inđinerijoje). Vilnius: Technika, 2002. 312 p. (in Lithuania).

[7] Sutphim J. AutoCAD 2000 VBA. Wrox, 1999. 674 p.

[8] Cottingham M. Mastering AutoCAD VBA. Sybex, 2001. 656 p.

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

[10] Sun M., Lockley S. R. Data exchange system for an integrated building design system. Automation in Construction, Vol. 6, Num. 2 (1997), 147-155.

[11] Whyte J., Bouchlaghem N., Thorpe A., McCaffer R. From CAD to virtual reality: modeling approaches, data exchange and interactive 3D building design tools. Automation in Construction, Vol. 10, Num. 1 (2000), 43-55.

[12] Evgenev G. CAD in XXI century: problem of relation form and content. CAD and Graphics (SAPR i grafika), Num. 12 (1999), 50-54 (in Russian).

[13] Mazurin A., Mazurina C. ActiveX for AutoCAD R14: a practical guide. Objects methods. CAD and graphics, Num. 5 (1998), 92-96 (in Russian).

[14] Lalani S., Chandak R. ActiveX programmer's library (Biblioteka programista ActiveX). Minsk: Popuri, 2000. 624 p. (in Russian).

[15] Kuzmenko V.G. VBA 2002. Moskov: Binom, 2002. 624 p. (in Russian).

[16] Monedero J. Parametric design: a review and some experiences. Automation in Construction, Vol. 9, Num. 4 (2000), 369-377.

[17] Hicks B.J., Culley S.J. An integrated modelling environment for the embodiment of mechanical systems. Computer-Aided Design, Vol. 34, Num. 6 (2002), 435-451.

[18] Sokas A. Operating of the graphics primitive applications. Proceedings of the Lithuania Science and Industry Conference. Engineering and Computer Graphics. Kaunas: Technologija, 2002. 93-97 (in Lithuania).

Postal address

Algirdas Sokas
Department of Engineering Graphics
Vilnius Gediminas Technical University
Sauletekio al. 11, LT-2040 Vilnius
Lithuania