AUTOMATIZED FORMATION OF SPECIFICATIONS IN ENGINEERING DRAWINGS

Algirdas SOKAS

Vilnius Gediminas Technical University
11 Sauletekio ave., LT2040 Vilnius, Lithuania
email: algirdas.sokas@fm.vtu.lt

Abstract. We can analyse the engineering graphics result computer-aided design (CAD) file as a database of design information that is developed over time. Each software application has its own native file format. A text-only or American Standard Code for Information Interchange (ASCII) file uses nothing more than the standard 256 characters. For design information analysis we can use some kind of graphic exchange formats. These are standardied formats that capture graphic information in the same way that a text file captures text data. The Drawing Interchange Format (DXF) developed by Autodesk in 1982 is popular for analysis CAD information. Formulation of problem. Prepared methods and Visual Basic Application (VBA) procedures for solve these design problems: creates matrix with graphical objects' names and numbers, in order to find a list of similar graphical objects; creates elements' specifications in the drawing. The following procedures are presented: procedure fragment which calculates the number of graphical objects and writes objects' names to a vector string, the next procedure finds the same names of graphical objects in vector and calculates their number; this new information is written to matrix, in which the first column has names of graphical objects from drawing database (DDB) and the second column has the corresponding number of these objects. The presented examples of procedures show that it is possible to control names and numbers of graphical objects in the drawing. Using this technology in the engineering drawings, we can automatically design specifications of drawing elements.

Key Words: graphics programming, exchange formats, drawing database, Visual Basic, application

1. Introduction

In 1960, a young Massachusetts Institute of Technology graduate student, Ivan Sutherland, decided to do his Ph.D. thesis on the application of computer graphics to engineering design. The project was called Sketchpad and created in 1963. First computer-aided design (CAD) product Line Drawing System was created by David Evans and Ivan Sutherland in 1969 [1].

Now we can analyse the engineering graphics result CAD file as a database of design information that is developed over time. Each software application has its own native file format. Other file formats are standardised so they can be read by many different applications. A text-only or American Standard Code for Information Interchange (ASCII) file uses nothing more than the standard 256 characters.

For design information analysis we can use some kind of graphic exchange formats. These are standardized formats that capture graphic information in the same way that a text file captures text data. Initial Graphic Exchange Specification (IGES) format [2] was developed first and published in 1980. The Drawing Interchange Format (DXF) [3] developed by Autodesk in 1982 is popular for analysis of CAD information. Relatively new Standard for the Exchange of Product Model Data (STEP) format [4] was published as international standard ISO 10303 in 1994.

The DXF format changes with each new release of AutoCAD. For the first 10 releases Autodesk did nothing to allow an older version of AutoCAD to read drawings made with newer versions of the software. The binary DXF for AutoCAD 10 was developed by Autodesk only in 1988 [5].

ISSN 1644-9363 / PLN 15.00 © 2004 PTGiGI

In the AutoCAD environment we can program with C, AutoLISP and Visual Basic (VB) languages. Now VB is Microsoft object-oriented programming language. Visual Basic for Applications (VBA) is another Basic language version, which is now included to many Microsoft programs as Word, Excel, PowerPoint, Access. Since AutoCAD 14 we can use VBA for graphics programming. 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.

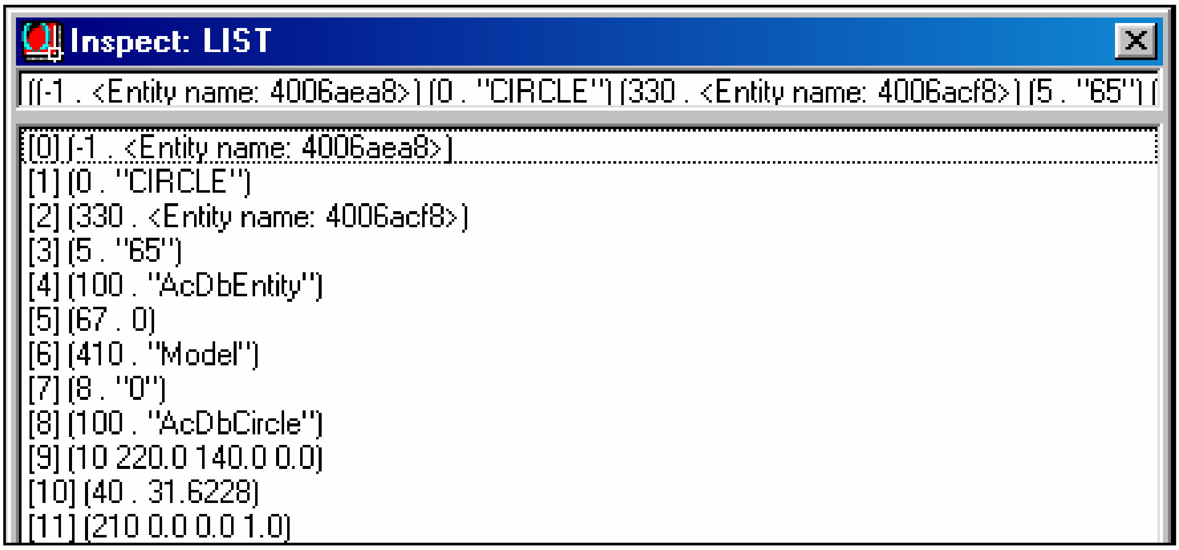
2. Formulation of problem

For a designer it is urgent to draw a graphical object, but also information connected with that object is needed. Firstly, geometrical parameters; secondly, object material; and finally, other information connected with the graphical object (cost, supplier, producer, etc.). The first articles [6] about engineering drawings are written about specifications, materials and amounts.

Formulation of problem. Prepared methods and VBA procedures for solve these design problems: creates matrix with graphical objects' names and numbers, in order to find a list of similar graphical objects; creates elements' specifications in the drawing.

3. Drawing database

All information about drawing in the AutoCAD system is in the drawing database (DDB). We will study it in DXF format, which is in many graphical systems. Data that describe entity is a list. It is made of different DXF group codes. Each such group is separated by brackets also forms a list from code, dot and meaning. Code defines property, dot is a distinctive sign, and meaning is property's parameter. For example, a list (0 . "CIRCLE") informs that code equals to zero and defines entity type, meaning is entity name (Fig 2).



```
((-1 . <Entity name: 4006aea8>) (0 . "CIRCLE") (330 . <Entity name: 4006acf8>) (5 . "65") (
```

```
[0] (-1 . <Entity name: 4006aea8>)
[1] (0 . "CIRCLE")
[2] (330 . <Entity name: 4006acf8>)
[3] (5 . "65")
[4] (100 . "AcDbEntity")
[5] (67 . 0)
[6] (410 . "Model")
[7] (8 . "0")
[8] (100 . "AcDbCircle")
[9] (10 220.0 140.0 0.0)
[10] (40 . 31.6228)
[11] (210 0.0 0.0 1.0)
```

Figure 1: Drawing database

Draw a circle and get such DXF format:

```
((-1 . <Entity name: 4006aea8>) (0 . "CIRCLE")
(330 . <Entity name: 4006acf8>) (5 . "65")
(100 . "AcDbEntity") (67 . 0) (410 . "Model")
(8 . "0") (100 . "AcDbCircle") (10 220.0 140.0 0.0)
(40 . 31.6228) (210 0.0 0.0 1.0)
```

In the first row code "–1" indicates entity name, "0" - entity type, "330" – program handle, "5" – entity handle, "100" – subclass data marker, "67" – space model or paper, "410" – layout name, "8" – layer name, "10" – centre of a circle, "40" – radius of a circle, "210" – z axis direction.

Knowing DDB structure and DXF file codes, we can create programming methods for creating elements' specifications in the drawing.

4. VBA procedures

We will realize graphical objects' accounting technology by programming with Visual Basic Application language for AutoCAD. An object is the main element of the programming language VBA. We will go deep into the technology of the graphical objects' management.

Procedure fragment which calculates the number of graphical objects and writes objects' names to a vector string:

```
Number = ThisDrawing.ModelSpace.Count
For i = 0 To Number - 1
  Set grafObj = ThisDrawing.ModelSpace.Item(i)
   v(i + 1) = grafObj.ObjectName
Next I
```

Names of graphical objects in the drawing database are presented by a code and entity name, for example, list (100 . "AcDbCircle"). Algorithm for sorting graphical objects are in the drawing (Fig 2).

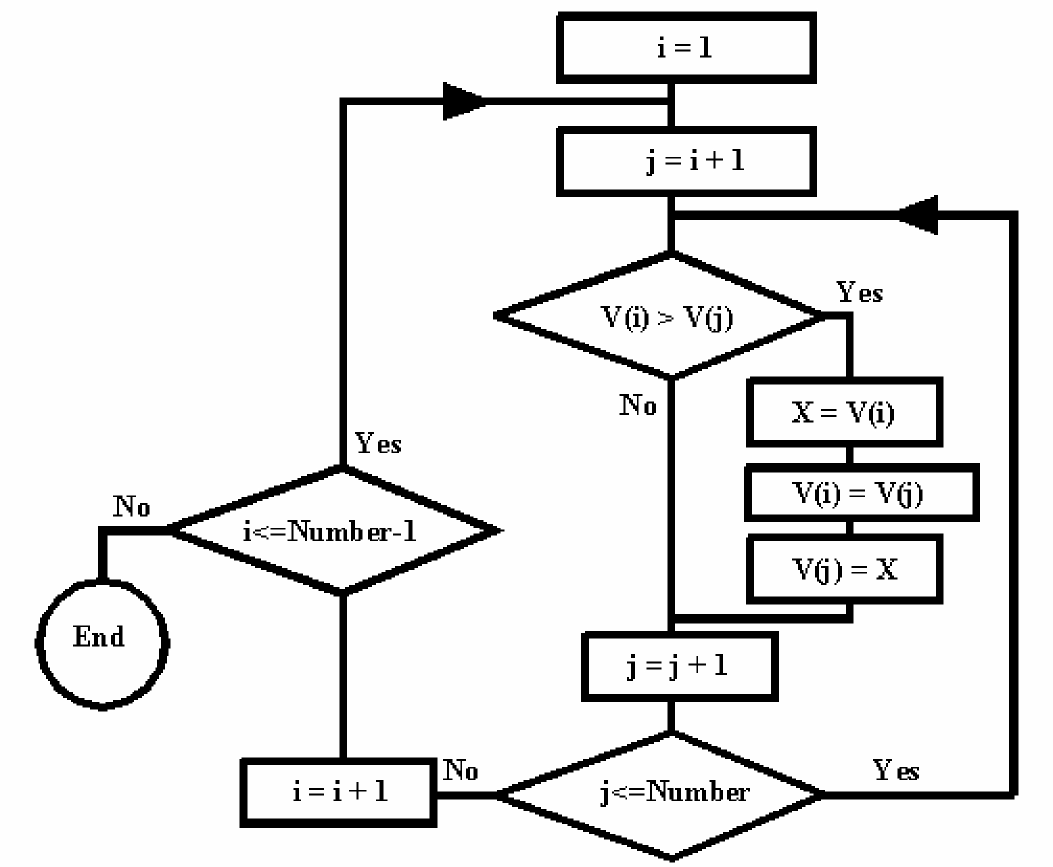


Figure 2: Algorithm for sorting graphical objects in the drawing

Procedure fragment for sorting names of graphical objects in the drawing:

```
For i = 1 To Number - 1
   For j = i + 1 To Number
     If Asc (Right (Left (v ( i ), 5), 1)) > Asc(Right (Left (v ( j ), 5), 1)) Then
         x = v( i )
         v( i ) = v( j )
         v( j ) = x
     End If
   Next j
Next i
```

The fifth sign ASCII code from names of graphical objects are compared in the third row of the procedure fragment. The names in the vector string are sorted according to this comparison.

The next procedure finds the same names of graphical objects in vector **V** and calculates their number k. This new information is written to matrix [Y], in which the first column has names of graphical objects from DDB and the second column has the corresponding number of these objects. Algorithm for calculating graphical objects and formatting a matrix are shown in Fig 3.

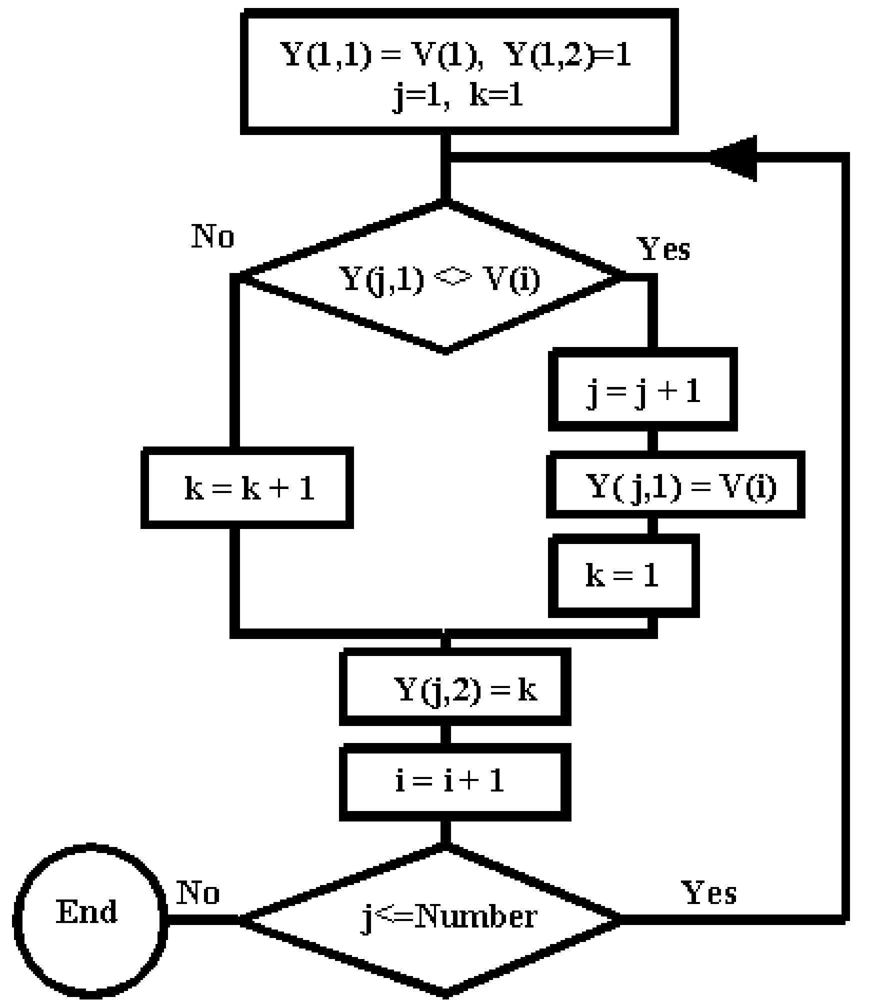


Figure 3: Algorithm for calculating graphical objects and formatting a matrix

Procedure fragment realize presented algorithm:

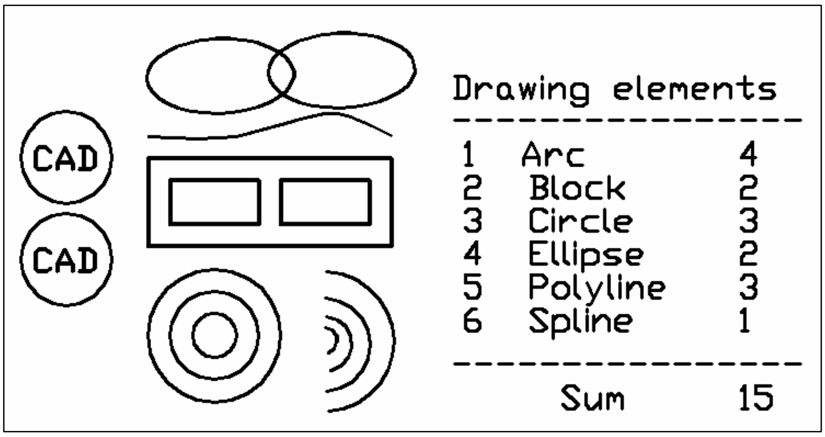
```
j = 1 : k = 1
y(1, 1) = v(1) : y(1, 2) = 1
For I = 2 To Number
  If Asc(Right (Left( y( j, 1), 5), 1)) <> Asc(Right (Left( v( i ), 5), 1)) Then
      j = j + 1 :   y( j, 1 ) = v( i ) :   k = 1
   Else :    k = k + 1
   End If :   y( j, 2) = k
Next i
mj = j
```

Here mj – number of different graphical objects, j – index of an object, k – number of the same objects corresponding to each j.

The next procedure fragment shows how the names of graphical objects from DDB can be changed to other useful names:

```
For j = 1 To mj
    Select Case y(j, 1)
         Case "AcDbArc": y(j,1) = "Arc"
         Case "AcDbCircle": y(j,1) = "Circle"
         Case"AcDbEllipse":y(j,1)="Ellipse"
         Case"AcDbBlockRefere":y(j,1)= "Block"
         Case "AcDbPolyline":y(j,1) = "Polyline"
         Case"AcDbSpline": y(j,1) ="Spline"
    End Select
Next j
```

We can print the new information. The result is presented in Fig 4.



Drawing elements

1	Arc	4
2	Block	2
3	Circle	3
4	Ellipse	2
5	Polyline	3
6	Spline	1
	Sum	15

Figure 4: Drawing and specification

Printing procedure fragment for graphical objects in the drawing:

```
For j = 1 To mj
  If Asc(y (j, 1)) <> 0 Then
         it(0) = pt(0) + 5
         it(1) = pt(1) - 15 - j * 15
         it(2) = pt(2)
```

```
            StrText = j & "  " & y( j, 1)
            Set objText = ThisDrawing.ModelSpace.AddText(StrText, it, 10)
            objText.Update
            it(0) = pt(0) + 150
            StrText = y(j, 2)
            Set objText = ThisDrawing.ModelSpace.AddText(StrText, it, 10)
            objText.Update
        End If
    Next j
```

Presented examples of procedures show that it is possible to control names and numbers of graphical objects in the drawing. Using this technology in the engineering drawings, we can automatically design specifications of drawing elements.

5. Conclusions

- the main formats of drawings and DXF history are presented.
- VBA procedure fragments for defining number of graphical objects in the drawing, sorting objects and calculating number of same objects, changing names of objects, and printing a list are presented.
- algorithm to calculate graphical objects and format matrix are presented.
- automated design of specifications from graphical objects in the drawing are presented.

References

[1] Smith A. R.: *The Stuff of dreams*. Computer Graphics World, July 1998. (http://cgw.pennnet.com/Articles).

[2] Smith B.: *Initial graphics exchange specification*. (IGES), version 4.0. Warrendale, PA: Society of Automotive Engineers, 1988.

[3] AutoCAD 2002 DXF Reference Guide. Autodesk Inc. 2001. 182 p. (http: //www. autodesk.com/techpubs/autocad/dxf/)

[4] International Organization for Standardization. STEP (ISO 10303) on a page, URL. (http://www.iso.ch/iso/)

[5] Walker J.: *The Autodesk file. Bits of history, words of experience*. Switzerland, Neuchatel, 1994. (http://www.fourmilab.ch/autofile/)

[6] Masters E.H.:*The engineering design functions which the computer can most effectively aid*. Computer-Aided Design, Vol 25, No 2, 1993, 130-135.

AUTOMATYCZNE TWORZENIE SPECYFIKACJI OBIEKTÓW W RYSUNKACH INŻYNIERSKICH

Możemy analizować obiekty graficzne otrzymane drogą operacji CAD jako dane informacyjne i przetwarzać je. Każdy program komputerowy (aplikacja) posiada własny oryginalny format pliku. W plikach typu ASCII wykorzystuje się nie więcej niż 256 standardowych znaków. Do analizy informacyjnej możemy używać pewne wymienne rodzaje formatów graficznych, które sprowadzają się do tekstowych plików danych. Wśród nich najbardziej popularnym jest tekstowy format DXF.

Reviewer: Henryk GLIŃSKI, PhD, MSc

Received June 17, 2004