

PRACTICE OF NON-GRAPHICAL OBJECT XRECORD IN ENGINEERING DESIGN

Algirdas SOKAS

Vilnius Gediminas Technical University
Sauletekio al. 11, LT-10223 Vilnius, Lithuania
e-mail: algirdas.sokas@fm.vgtu.lt

Abstract. The problems of automated design system with using object xrecord are researched in the article. Two parts of drawing are separated: graphical objects (lines, circles, polygons, ellipses, and so forth) and non-graphical objects (layers, linetypes, dimstyles, selectionsets, xrecord etc). A question, how non-graphical objects xrecord is used for graphical programming, is analyzed. Literature about non-graphical objects usage for designing engineering systems is presented, and examples using Visual Basic for Application programming language are referred. Finally, the conclusions are made.

Keywords: graphic programming, non-graphical objects, Visual Basic Application.

1. Introduction

Graphical objects, also known as entities, are the visible objects (lines, circles, polygons, ellipses etc.) that make up a drawing. To create these objects use the appropriate add method. To modify or query these objects, use the methods or properties of the object itself. Each graphical object has methods that allow an application to perform most of the AutoCAD editing commands such as copy, move, offset, rotate, mirror and so forth [1].

Non-graphical objects are the invisible (informational) objects that are part of a drawing, such as layers, linetypes, dimstyles, selectionsets, xrecord etc. To create these objects use the add method of the parent collection object. To modify or query these objects, use the methods or properties of the object itself. Each non-graphical object has methods and properties specific to its purpose; all have methods for setting and retrieving extended data (xdata), and deleting themselves [2, 3].

The AcadDictionary object represents a container object for storing and retrieving objects, with associated string keywords by which they are referenced. A dictionary can contain any type of object including other dictionaries. You can create new dictionaries, add entries to an existing dictionary, and get the keyword for a given object or the object for a given keyword. You can change the object associated with a given keyword or rename the keyword for a given object. However, it does not perform type checking of entries [4].

The AcadXRecord object is used to store and manage arbitrary data. Xrecords are similar in concept to xdata but are not limited by size or order. They use standard AutoCAD group codes, the values of which are all below 1000, which means in addition to all the normally used data types, xrecords are capable of storing object IDs, allowing ownership of many other objects. Xrecord objects are saved with the drawing and can be directly accessed by other Visual Basic and LISP programs. Xrecords are created using the add method of the dictionary object and accessed programmatically through the item method of the dictionary object. The only basic operations that can be performed on xrecords are adding, getting and deleting. There is no mean to update (modify) an xrecord other than to delete and re-create it.

2. Programming with object xrecord

Let us find out programming focus of using an object xrecord by testing a simple example. Draw circles and write their circumference values to a separate xrecord object. Drawing and xrecord information are presented in Fig. 1. Next procedure finds the sum of values in the xrecord object. The result is presented in Fig. 2.

At first, dictionary and xrecord variables are declared.

```
Dim Zodynas As AcadDictionary
Dim Irasas As AcadXRecord
```

In the procedure given values are set for the variables and xrecord is opened.

```
Set Zodynas = ThisDrawing.Dictionaries("My dictionary")
Set Irasas = Zodynas.GetObject("Circumference")
Irasas.GetXRecordData Xkodas, Xduom
```

Dimensions of code and data vectors are tested.

```
If VarType(Xkodas) And vbArray = vbArray Then
  Size = UBound(Xkodas) + 1
  ReDim Preserve Xkodas(0 To Size)
  ReDim Preserve Xduom(0 To Size)
Else
  Size = 0
  ReDim Xkodas(0 To Size) As Integer
  ReDim Xduom(0 To Size) As Variant
End If
```

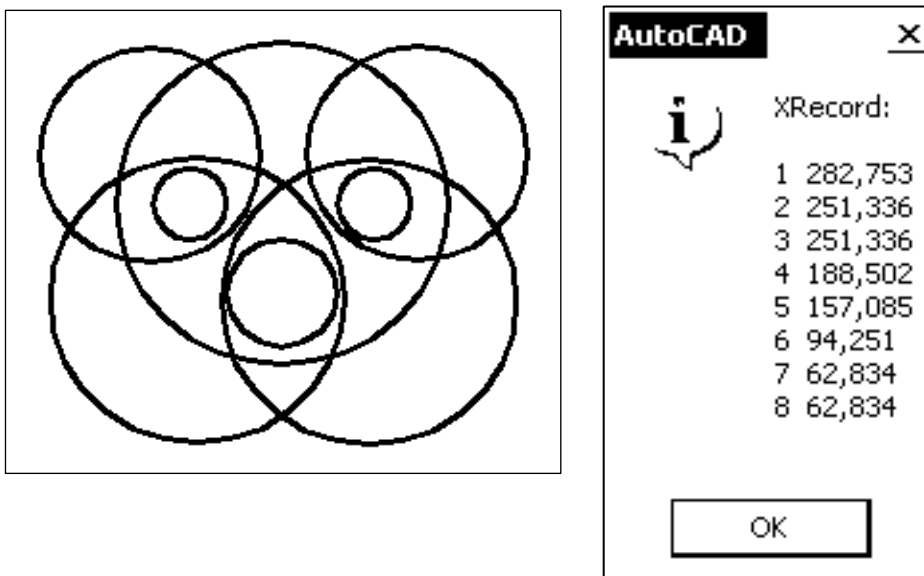


Fig. 1: Drawing and xrecord information

The data is set.

$$Xkodas(Size) = 0$$

$$Xduom(Size) = 2 * Radius * 3.1417$$

Next procedure calculates the sum of the xrecord information.

```

For i = 0 To Size
    Duom = Xduom(i)
    suma = suma + Val(Duom)
Next
MsgBox suma

```

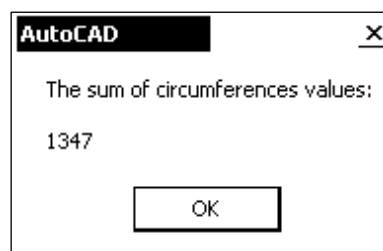


Fig. 2: Sum of numeral information

3. There are two methods to delete objects from xrecord and drawing

We show xrecord list number of a graphical object, which we want to erase from the list and drawing. Based on the list number we programmatically select object ID number, which is also in the xrecord. Both methods use ID number of graphical objects from xrecord (Fig. 3). Program code, which finds object ID number based on given list number, is presented.

```

i = Val(InputBox("What number to erase? ", , a)) - 1
kodas = xkodas(i)
i1 = InStr(1, xduom(i), ";")
duom = CDBl(Mid(xduom(i), 1, i1 - 1))
i1 = InStr(1, xduom(i), "=")
numberID = CLng(Mid(xduom(i), i1 + 1, 9))

```

Graphical object can be erased from xrecord and the drawing when we have object ID number. First method uses ID numbers vector of all graphical objects in the drawing. At first, objects ID number vector is formed.

```

objsk = ThisDrawing.ModelSpace.Count
For j = 0 To objsk - 1
    Set object = ThisDrawing.ModelSpace.Item(j)
    v(j + 1) = object.ObjectID
Next j

```

Then concrete object ID is found and erased.

```

For j = 0 To objsk - 1

```

```

If v(j + 1) = numberID Then
  Set object = ThisDrawing.ModelSpace.Item(j)
  object.Delete
End If
Next j

```

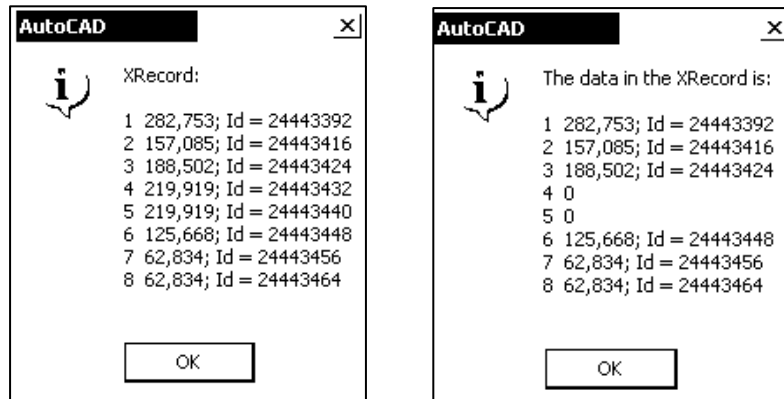


Fig. 3: Automatically found information

Program procedure results for designing truss node and updating drawing is shown in Fig. 4.

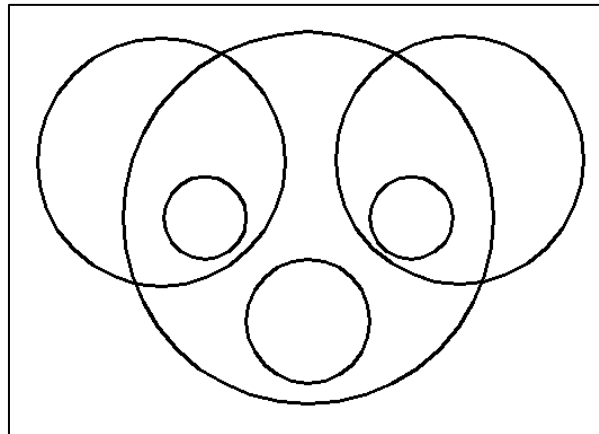


Fig. 4: The drawing after erasing two objects

The other method uses non-graphical object selectionset. A list of all graphical objects in the drawing is formed programmatically and the object needed to be erased is selected based on ID number.

```

t1(0) = 0: t1(1) = 0: t1(2) = 0
t2(0) = 420: t2(1) = 297: t2(2) = 0
Set sel = ThisDrawing.SelectionSets.Add("new_sel")
sel.Select acSelectionSetWindow, t1, t2
For Each object2 In sel
  If object2.ObjectID = numberID Then
    object2.Delete
  End If

```

```
Next object2  
ThisDrawing.SelectionSets("new_sel").Delete
```

The presented program procedures results show that it is possible to control non-graphical object xrecord and use it to solve graphical problems.

4. Conclusions

It is very important to use non-graphical objects for programming rational drawing systems. Programming examples of using non-graphical object xrecord are presented.

Two objects erasing methods in the drawing are presented, when only the number of an object in the list is given. The first method uses ID numbers vector of all graphical objects of the drawing whereas the second method uses non-graphical object selection set.

References

- [1] Sokas A.: *Creation of graphical objects' properties and methods. Engineering Graphics. Proceedings of the Eighth International Conference BALTGRAF-8. Tallinn, TUT, 2006, 101-105.*
- [2] *AutoCAD 2002 DXF Reference Guide.* Autodesk, Inc. 2001, 182 p.
<<http://www.autodesk.com/techpubs/autocad/dxf/>>.
- [3] Gonzalez M.P., Lores J., Granollers A.: *Enhancing usability testing through datamining techniques: A novel approach to detecting usability problem patterns for a context of use.* Information and Software Technology. 2008, Vol. 50, 547-568.
- [4] Sutphin J.: *AutoCAD 2006 VBA. A Programmer's Reference.* Apress, 2006, 777 p.

WYKORZYSTANIE NIEGRAFICZNEGO OBJEKTU XREKORD W PROJEKTOWANIU INŻYNIERSKIM

W artykule omawia się zagadnienia projektowania automatycznego z użyciem obiektu xrecord. Obiekty graficzne (linie proste, okręgi, wielokąty, elipsy, etc.) i niegraficzne elementy systemu (warstwy, typy linii, style wymiarowania, zbiór parametrów wyboru, xrecord, etc.) są oddzielnymi częściami. Analizowana jest istota użycia xrecordu w programowaniu graficznym. Podano przykłady użycia obiektów niegraficznych zrealizowanych w języku Visual Basic for Application do projektowania systemów inżynierskich oraz przedstawiono wnioski.