

Automatic generation of parametric drawings using DXF and Visual Basic

Alcnia Zita Sampaio¹, Antnio Aguiar da Costa²

Abstract - The definition of drawings based on parametric shapes is a fast way to create similar drawings. In technical fields, such as Civil Engineering, the elaboration of design documentation involves a great number of drawings that are identical in configuration, the values of some of the dimensions the only changes that need to be made. It is, then, important for engineering students to know how to implement software that could help to obtain, quickly, the same kind of drawings needed in projects of a specific area in engineering. Among examples of this type of drawing are: the cross section of a column, the details of the iron reinforcements in a slab or the section of a metal beam. This subject was introduced into the undergraduate Civil Engineering curriculum as part of the Computer Aided Drawing course. With this module the students acquire knowledge about a type of drawing format file, *DXF*, which is frequently used as an integral part of several engineering computer applications. The data structure is presented and they learn how to define drawings, directly writing files following the specific *DXF* data structure. They use the *Visual Basic* language to develop small computer programmes to allow the generation of basic technical drawings. This programming language has a friendly interface which is very easy both to understand and to use. The students first elaborate the automatic generation of lines and circular arcs and then develop their own application. The paper shows the most relevant aspects of the teaching process and some results obtained by the students.

Index Terms - Education, Engineering, Design drawings, *DXF* files.

INTRODUCTION

In engineering there are innumerable computer applications which turn to the definition of drawings or the creation of 3D models based on parametric sketches. For example, one design system contains the delineation of polygons completed from the identification of the type of element and the request for its dimensions, or a program for calculation, makes use of the definition of the metallic elements, classifying then by class, HEA or IPE, and permitting their specification through the attribution of numeric values to the geometric parameters associated with the selected cross section. Since this type of resource is very frequently used by professional engineers, it was thought that it would be of interest to include the teaching of the basics of parametric design in the curriculum of the first year of Civil Engineering

as part of the Computer Assisted Drawing course [1]. This knowledge is seen as complementary to the study of technical drawing. Students already familiar with the use of graphic systems in the definition of conventional plan projections and 3D models, where the same types of configurations are used repeatedly, have a better understanding of the need to develop applications based on parametric drawing concepts.

The methodology used in the teaching of this subject was structured according to the following phases:

- The introduction of parametric design, taking as an illustrating example the rectangle;
- The presentation of a drawing file format with an intuitive structure of simple data and use generalised to graphic applications;
- Recourse to an object-oriented programming language for the definition of interfaces and calculation algorithms;
- Implementation of didactic applications with the support of the teacher, which serves to draw together the taught concepts;
- Finally, the students are asked to create a more complex, individual application related to his/her Engineering speciality.

In this process, the students have recourse to knowledge they have already acquired, from studies in:

- **Technical Drawing:** the ability to identify conventional representations and manipulate computer graphic systems for design definition;
- **Programming:** the skill of organizing and structuring problem resolution and the knowledge of object-oriented language.

Student satisfaction is clearly shown in their engagement with the development with their own application. This final piece of individual work is evidence of:

- A good assimilation of the concept of parametric drawings;
- The correct understanding of the structure of data of the type of drawing files used;
- An understanding particularly suited to a problem of the automatic generation of designs based on parametric models.

With this training, students are enabled to develop, throughout their education, and beyond, in their professional practice, situations of repetitive representation, which are frequent in Engineering.

¹ Alcnia Zita Sampaio, Technical University of Lisbon, Dep. Civil Engineering and Architecture, zita@civil.ist.utl.pt

² Antnio Aguiar da Costa, Technical University of Lisbon, Dep. Civil Engineering and Architecture, amcosta@civil.ist.utl.pt

THE EXAMPLE OF THE RECTANGLE

A commercial graphic system allows the representation of a rectangle through:

- The selection of the **rectangular** entity;
- The indication of two numerical values corresponding to the associated geometric **parameters** (length, a , and width, b , or diagonal, d and angle α , Figure 1).

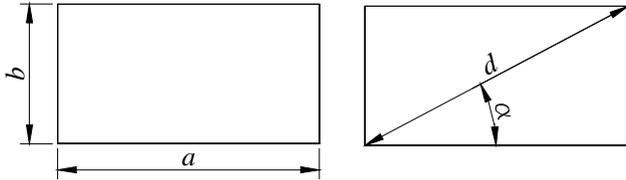


FIGURE 1
PARAMETRIC REPRESENTATION OF THE RECTANGLE.

The specific situations as defined are immediately shown in the visualisation window of the system (Figure 2).

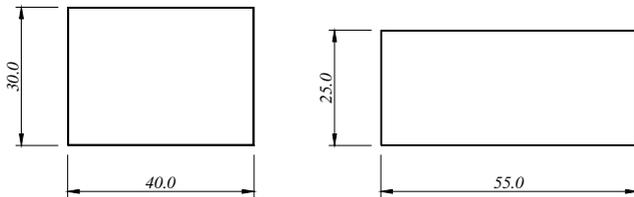


FIGURE 2
REPRESENTATION OF PARTICULAR CASES OF A RECTANGLE.

The graphic system has therefore a parametric entity which allows specific cases of rectangular forms to be obtained rapidly, without having to provide the coordinates of each of the vertices or the dimensions of the sides of which it is made up.

I. The structure of data

Accessing the structure of the data relative to one of the shown rectangles, through the use of the system *AutoCAD*, it can be seen that its composition includes (Figure 3):

- The identification of the represented figure, a polygonal line defined on the plane (x,y) , *LWPOLYLINE*, and closed;
- The coordinates (x,y,z) of the four vertices of the rectangle.

LWPOLYLINE Layer: "0"			
Space: Model space			
Handle = 2B			
Closed			
Constant width	0.0000		
area	1200.0000		
perimeter	140.0000		
at point	X= 0.0000	Y= 0.0000	Z= 0.0000
at point	X= 40.0000	Y= 0.0000	Z= 0.0000
at point	X= 40.0000	Y= 30.0000	Z= 0.0000
at point	X= 0.0000	Y= 30.0000	Z= 0.0000

FIGURE 3
LIST OF THE DATA OF A RECTANGLE.

To obtain these data the graphics system must be associated with a cartesian coordinate system (x,y,z) and it

must include algorithms which transpose the data onto the parametric format (a,b) thus structuring a matrix of coordinate vertices. The algorithms for drawing on the plane are:

$$\begin{aligned} x(1) &= 0.0 & y(1) &= 0.0 \\ x(2) &= a & y(2) &= 0.0 \\ x(3) &= a & y(3) &= b \\ x(4) &= 0.0 & y(4) &= b \end{aligned} \quad (1)$$

II. DXF format of the drawing file

The list of data included in Figure 3 can be reorganised according to the *Drawing Exchange Format (DXF)* format of the drawing file. Figure 4 included an excerpt from the *DXF* file of the drawing of the same rectangle.

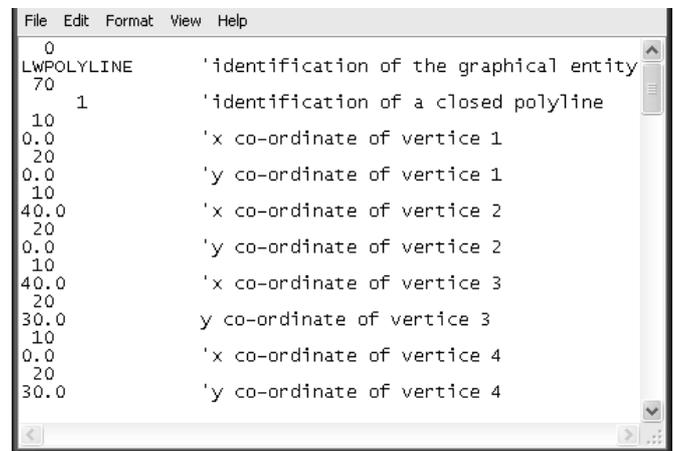


FIGURE 4
EXCERPT FROM THE DXF FILE FOR A RECTANGLE.

Initially, the *DXF* drawing file format was developed by *Autodesk* (USA) for the *AutoCAD* system as a way of enabling the transfer of drawings between this system and other *Computer Aided Design (CAD)* products or data processing programs [2]. Today, the majority of programs which involve graphic representation are able to create and interpret *DXF* files.

The *DXF* file can be created by *AutoCAD*'s own system, by selecting the *dxf* option when the drawing is to be saved as shown in Figure 5.



FIGURE 5
SAVING A FILE IN DXF FORMAT.

However, what is proposed here is the creation of an application which generates *DXF* files so that they have to be read by the *AutoCAD* system or other system compatible with that file format.

It will be noticed that for a given type of graphic element, the corresponding *DXF* structure is constant, only varying in the values of the coordinate vertices. Thus, it is possible to program the automatic representation of a given

element, by defining an algorithm associating the coordinates to the variables, which will be established for each particular case (Figure 6).

```

File Edit Format Help
write " 0"
write "LWPOLYLINE"
write "70"
write " 1"
for i=1 to 4
  write "10"
  write x(i)
  write "20"
  write y(i)
next i

```

FIGURE 6
INSTRUCTIONS FOR WRITING DATA TO DXF FILES.

III. Interface

This concept for programming *DXF* files by type of graphic elements requires the creation of an interface which allows the interaction with the application so that numeric values can be attributed to parameters. That is to say, that in the case of the rectangle an interface similar to that shown in Figure 7 must be defined.

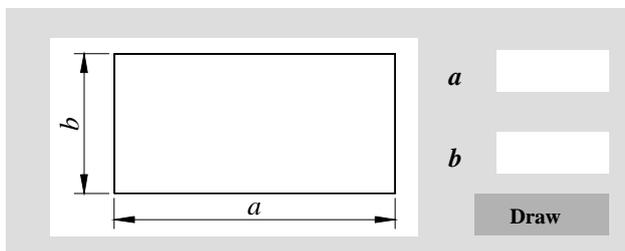


FIGURE 7
SYSTEM INTERFACE FOR THE GENERATION OF RECTANGLES.

It is possible, by means of the interface, to indicate the corresponding values of the parameters a e b , in any particular case, followed by the calculation of the values of the variables for the automatic generation of rectangles, using the expressions of transposition algorithms (1), the coordinates of the 4 vertices, which are written to a text file, with a *dxf* extension, using the programming instructions shown in Figure 6. The *Draw* button, shown in Figure 7, allows the connection with *AutoCAD* and, therefore, the visualisation of the rectangle which corresponds to the values attributed to the parameters a and b .

IV. Programming Language Visual Basic

The programming language used in the definition of the interface, in the calculation algorithms and when writing them to a *dxf* file is *Virtual Basic*, as it provides a direct method of object-based programming which allows the generation of interfaces, to be easily understood which is

essential if interaction with application based on parametric drawing is to be promoted [3].

AUTOMATIC GENERATION OF GRAPHIC ENTITIES

The generation of the rectangle is based on the creation of a *dxf* file related to the *LWPOLYLINE* element. In order to define other shapes of interest to engineering it is necessary to have recourse to others type of graphic entities: the straight *LINE* and the circular *ARC*. Each one of these elements has its own data structure, in the *DXF* format. Figure 8 shows the list of data relevant to two particular cases.

```

File Edit Format View Help
0
LINE 'identification of the graphical entity
8
0 'identification of the layer
10
5.0 'x co-ordinate of the initial vertice
20
15.0 'y co-ordinate of the initial vertice
11
40.0 'x co-ordinate of the final vertice
21
75.0 'y co-ordinate of the final vertice

File Edit Format View Help
0
ARC 'identification of the graphical entity
8
0 'identification of the layer
10
35.0 'x co-ordinate of the center of the arc
20
18.0 'y co-ordinate of the center of the arc
40
35.0 'radius of the arc
50
25.0 'initial angle
51
54.0 'final angle

```

FIGURE 8
STRUCTURE OF DXF DATA OF LINE AND ARC ENTITIES.

The automatic generation of these entities is achieved by considering as variables:

- For a *LINE* entity, the coordinates $x(1)$ and $y(1)$, from the starting point and the coordinates, $x(2)$ and $y(2)$, from the end point of the straight segment;
- For the *ARC* form, the coordinates (x,y) of the centre of the arc $(xc$ e $yc)$, the radius (r) and the initial and final angles $(ang1$ and $ang2)$.

DXF data files will be written according to the instructions included in Figure 9 for the *LINE* and *ARC* elements.

GENERATION OF DRAWINGS COMPOSED OF SEVERAL ENTITIES

A *DXF* file can contain the definition of a set of entities. The structure of the data of a drawing composed of a series of straight lines and arcs presents, in sequence, the

identification of each element, according to the *DXF* format and each one represented with its particular values.

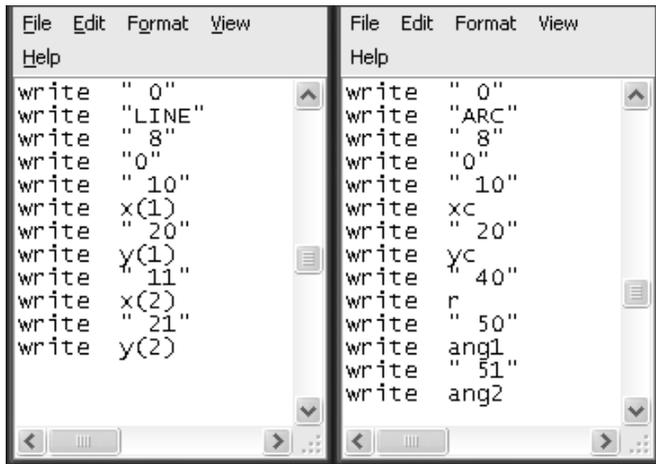


FIGURE 9

INSTRUCTIONS FOR WRITING DATA FOR LINE AND ARC ELEMENTS.

Thus, for example, the outline of a drawing included in Figure 10 is recorded in *DXF* format as shown in Figure 11.

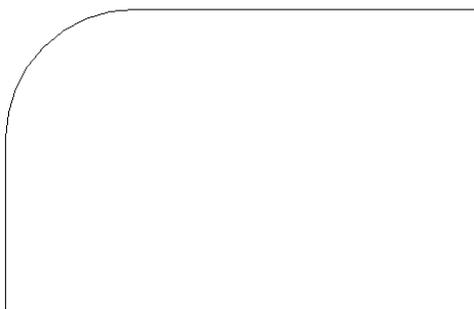


FIGURE 10

DRAWING CONSISTING OF THREE ELEMENTS.

In the list shown the following elements are easy to identify:

- The LINE element is represented by the coordinates of (2.0,2.0) and (2.0,6.0);
- Followed by, the ARC element, defined by its centre (3.5,6.0), radius value as 1.5 and initial 180° and final 90° angles;
- And finally, another LINE entity that is limited by the points (3.5,7.5) and (5.5,7.5).

The concept of the definition of a drawing through a sequential list of geometric data relating to each of the graphic elements of which it is made up is the basis of the creation of the majority of parametric models. The implementation of the computer application based on parametric configurations, requires, then, that programming routines be established which bring about the production of the variables associated with the parameters and that then they proceed with the writing to a text file, in the *DXF* format appropriate to each form.

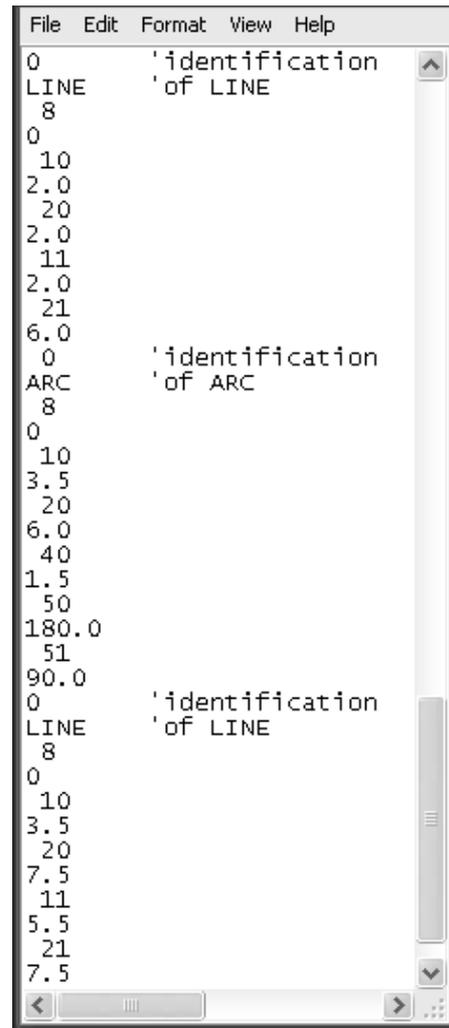


FIGURE 11

STRUCTURE OF DXF DATA FOR A SEQUENCE OF ENTITIES.

The program for the generation of drawings composed of a set of graphic elements will bring about the connection between each of the routines of the elements involved whenever it is necessary to write the corresponding *DXF* definition in the file of the drawing that it is creating.

APPLICATIONS DEVELOPED BY STUDENTS

By using the teaching examples presented to them, the students learned a simple and intuitive data structure and a language which allowed them to define friendly interfaces and the necessary calculation algorithms. Using this knowledge as a base, they were able to define their own applications. Figure 12 includes one of the interfaces implemented by the students.

The interfaces include the parametric drawing selected by the student and a set of boxes for the introduction of numeric values corresponding to each of the parameters associated to the drawing shown. Each of the boxes is identified by its respective label.

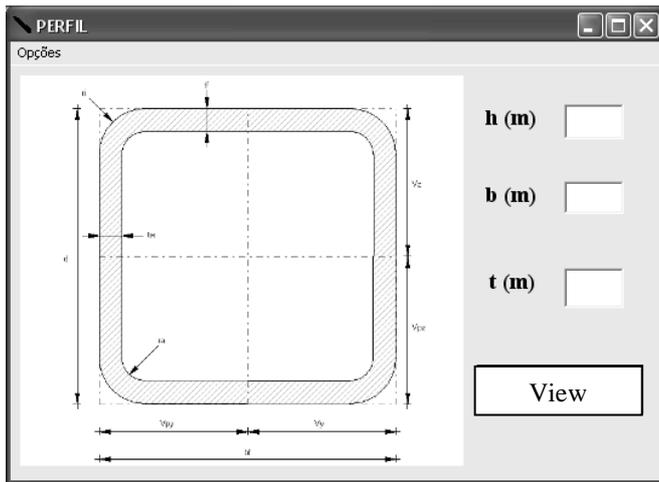


FIGURE 12
INTERFACE OF THE APPLICATION DEVELOPED BY A STUDENT.

Figure 13 shows the interface, in which can be seen the attribution of some of the values to the parameters of the metallic cross section represented in it.

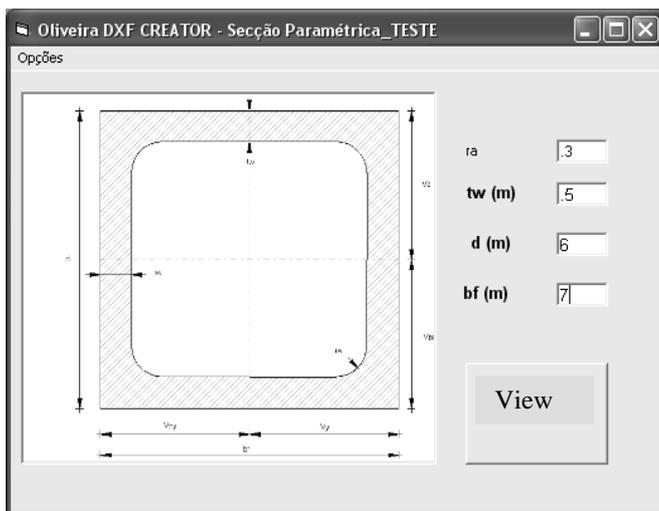


FIGURE 13
APPLICATION SHOWING VALUES ATTRIBUTED TO PARAMETERS.

In addition, the students included in their applications, alert message windows associated with common mistakes made in parametric drawings, so that the forms obtained by this process would correspond to realistic situations. For example, in Figure 14 the box shown indicates that the attributes associated to the parameters are incorrect or that the relationships between some of the parameters were defined using intervals too close together.

The student can program conditions of relationships between parameters and associate maximum and minimum values to a certain dimensions of the cross sections For example, in Figure 14, the parameters cannot be attributed negative values.

All the interfaces include a text box, *View*, which indicates the beginning of the generation of *DXF* files and its later reading through the *AutoCAD* system. When the user clicks on the box, calculations of the values of the variables

included in the *dxf* formats of each element are allowed and written to the drawing file. It also automates the access to *AutoCAD* for the reading of the file that was created.

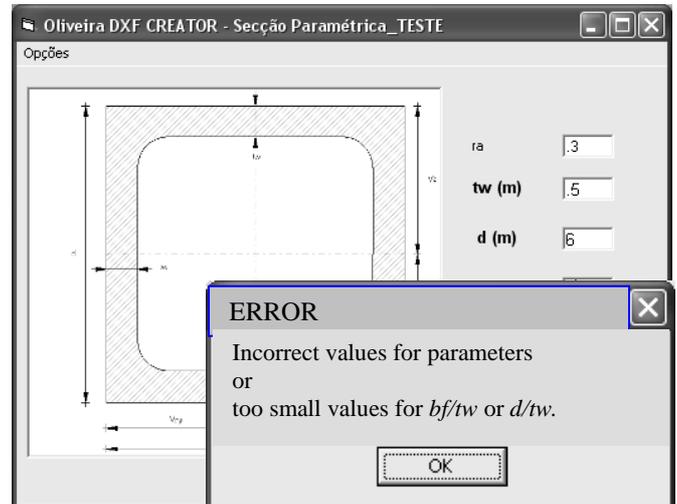


FIGURE 14
ALERT WINDOW FOR ERRORS ASSOCIATED WITH PARAMETERS.

LEARNING

The students are able to define drawings and models which can be represented by the elements described and whose data structure has been presented and applied. This teaching gave them a broad base for understanding the concept of parametric design and the structure of the *DXF* format, frequently used in many computer applications with some graphic component. The extrapolation for other 2D elements and even for 3D entities is immediate. They were provided with a bibliography that has sufficient details to enable them, individually, to create their own applications made up of other types of elements, such as a *CIRCLE*, a *POLYLINE* and in 3D modelling, the *3DFACE* entity.

During the presentation of this module, the students also learned how to modify and amplify the initial computer application (with reference to the rectangle) so that they were able to adjust the final program they developed to other parametric drawings, reusing part of the programming already carried out and adding other graphic elements or other calculation algorithms.

CONCLUSIONS

This paper presents teaching methodology related to a process for the generation of drawings based on parametric 2D models which is based on the creation of *DXF* files. This is within the context of engineering education with the aim of enabling students to develop computer applications which allow the automatic generation of drawings on the plane. The interest of this subject lies in the frequency with which the students, in different areas of specialisation, during their studies and later as engineers, have to produce, time and again, drawings with identical configuration. With this in mind, this subject matter was included in the Computer Assisted Drawing module of their course.

Both the concept of parametric drawing and the *DXF* format from the drawing file are introduced through a simple example of application, the rectangle. This learning progresses to the identification of the other two elements, LINE and ARC, and to the composition of more complex drawings based on these entities. The learning cumulates in the development of an application by individual students through which he/she demonstrates that he/she has acquired the necessary skills and knowledge to carry it out.

The evaluation of the teacher concerning the interest which this subject matter stimulated and maintained and the degree of understanding achieved was that it was clearly more than satisfactory.

REFERENCES

- [1] Sampaio, A.Z., Costa, A. Generation of parametric drawings in DXF format using the programming language Visual Basic, (in portuguese) Report ICIST, DTC, Lisbon, (in process).
- [2] Jones, F.H., Martin, L., *The AutoCAD database book - Accessing and managing CAD drawing information*, Ventana Press, 4^a ed., E.U.A., 1991.
- [3] Campos, L., Vilar, S., Lúcio, L., *Programação em Visual Basic 6*, FCA – editora de informática, 6^a ed., Lisboa, Janeiro 1999.