Framework for Computer Numerical Control Machine Code Generation from Vector Data

*Yasir Nawazish Ali, Syed M. Jafar Rizvi, Ghulam Mustafa, Muhammad Imran
Graphical Interface Development Division (GIDD)
System Operation and Software Development Department (SOSD)
Instrumentation Control and Computer Complex (ICCC), Islamabad, Pakistan
*yasir_ali863@yahoo.com, jafarrizvi@yahoo.com

***Abstract* — This Paper presents the framework for Computer Numerical Control (CNC) Machine code generation from the vector data. Two standard vector data formats are Drawing Exchange Format (DXF) and Initial Graphics Exchange Specification (IGES). Data exchange between these formats has been implemented as first phase in this project; when parameter of an entity in one format is different to other then required parameter is computed using geometric relations and when an entity is not present in targeted format, it is converted into closest equivalent entity in targeted format. In second phase, a Windows Presentation Foundation (WPF) based drawing editor is implemented to manipulate entities. In the final phase, software generates CNC machine standard G-code, which is used as input for CNC simulator.**

Keywords – Vector Graphics; Drawing Exchange Format; Initial Graphics Exchange Specification; Computer Numerical Control; G - Code.

I. INTRODUCTION

A. Computer Aided Design (CAD)

Computer-Aided design (CAD) is used for computer-based tools that assist engineers, architects and other design professionals in their design activities. CAD is normally used in designing of tools and machinery used in manufacturing, drafting and designing of all types of buildings.

B. Raster Graphics

Raster graphics is the most common graphics format in use today. Bits are the main building blocks of this format. Normally data is written in sections in this format. Header section is the first section that holds all environmental variables and then data section comes, where every pixel is represented by number of bits. In short, if user wants to show lot of colors then file would have larger size [1]. Examples are Bitmap, JPEG, TIFF and many others.

C. Vector Graphics

Vector graphics uses geometrical primitives such as “Line”, “Arc”, “Poly-line”, “Polygon” and many other instead of pixel information. All complex drawings are made by using these basic primitives. Vector graphics take lesser information to draw an entity on canvas. For example, if circle is to be drawn in vector graphics, then it requires center point and radius, whereas raster graphics file will contain all points that come on its locus. One more advantage of vector graphics file is that as they are stored in the form of mathematical equations therefore, various transformation operations, such as, translation, rotation, scaling can be performed on entities. Moreover, any combination of translation, rotation and scaling transformations can be applied to selected entities

In short, CAD applications require files that will not blur if they are scaled, zoomed, or printed. So, due to these features vector graphics files are used in CAD applications [2]. Examples are Drawing Exchange Format (DXF), Initial Graphics Exchange Specification (IGES), Stereo Lithography (STL) and many more.

Typical primitive entities that are used in vector graphics file are as described in Table I.

TABLE I. BASIC ENTITIES

Basic Primitives			
Ellipse	Line	Text	Spline
Circle	Hyperbola	Arc	Solid

D. Machine Code (G & M Codes)

CNC machine is run through programmed code (Machine Code), which is written in G-code programming language. G-code programming language consists of Preparatory functions (G-codes) and Miscellaneous functions (M-codes). G-code instructs the machine tool to perform actions, such as: G-00 is used for rapid move, G-01 is used for controlled feed move in a straight line, G-02 is used for clockwise circular interpolation, G-03 is used for counter clockwise circular interpolation etc. Similarly M-codes are used for various functions like M-02 is used for end of program; M-07 is used to turn on coolant A etc.

II. SYSTEM ARCHITECTURE

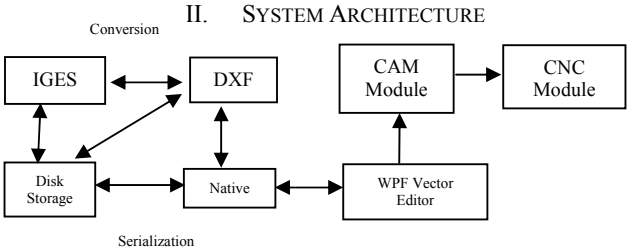

Figure 1. Main System Architecture

Main System Architecture is shown in Figure 1. In Figure 1, IGES and DXF blocks are two vector graphics formats and user can exchange data between these formats. Disk storage block means user can store respective data on hard disk. For example, if arrow comes to Disk storage from DXF that means user can export model in DXF format. And if arrow comes to DXF from Disk storage, then user can import DXF model from hard disk. Graphics editor has its own native format for the internal storage of graphics entities. The graphics entities are displayed using Windows Presentation Foundation (WPF) graphics format on the drawing canvas. The entities on the canvas can be saved either in DXF format or in the native format. Blocks IGES, DXF, Disk Storage and Native shows input part of software. User can import vector graphics file to its (Windows Presentation Foundation) WPF vector graphics editor. To achieve this, vector graphics file (DXF or IGES) is read and then converted into the native file format, then exchange this file into different vector graphics format (DXF or IGES) and draw new entities in the native format. After this user can save them on disk storage either in native format or vector graphics format. In WPF vector graphics editor user can perform different operations like transformation on entities, drawing new entities and many others functions. WPF is new technique supported by Microsoft; Windows Vista is developed by this technique. After designing model user can generate machine code (G– code) by using CAM module. In this module G–code of respective entities would be generated, means line can be converted to G–01 and circle or circular arc can be converted into either G–02 or G–03 and finally machine code would be given to CNC module that can show how machine's tool could perform action if machine code would be given to it.

III. Drawing Exchange Format (DXF)

DXF is a vector graphics file format, developed by Autodesk as their solution for enabling data interoperability between AutoCAD and other programs. DXF was originally introduced in December 1982 as part of AutoCAD 1.0 [3].

DXF support all basic primitives like "Line", "Arc", "Circle" and many more. It supported two file formats ASCII and binary. DXF doesn't support complex entity that's why most competitors use Drawing Graphics (DWG) as their basic file format.

In this format, every variable has group code that describes about its nature and its respective value. DXF is composed into number of sections like "Header section", "Object section", "Entity section" and many more. Header section holds the environmental variables like its version number; "Class section" holds information of all system defined classes whose instances appear in entity section and many more. In "Table section" system defined tables like APPID (Application ID) table, layer table are residing, "Block section" holds entities that appear in block section. Entity section is important section and it holds all graphical object like line, arc, circle and others in it, Object section holds all non-graphical objects and "Thumbnail Image section" is the last section that is obsolete now, it holds preview of the drawing [4].

IV. Initial Graphics Exchange Specification (IGES)

The official title of IGES is Digital Representation for Communication of Product Definition Data, first published in January, 1980 by the National Bureau of Standards as NBSIR 80-1978. A CAD user can exchange product data models in the form of circuit diagrams, wire frame, and freeform surface or solid modeling representations. IGES supports entities like line, arcs, spline curves, NURB curves, surfaces 3D-solids and many more entities [5].

IGES file is composed of 80-character ASCII records, a record length derived from the punch card era. Text strings are represented in 'Hollerith' format, the number of characters in the string, followed by the letter 'H', followed by the text string, e.g., '4HSLOT' is equivalent to SLOT. The file is divided into five sections, indicated by a character (S, G, D, P, or T) in 72nd column. Where, 'S' denotes start section (a free form comment section), G denotes Global section (an ordered set of file section), D has the meaning of Directory entry section (entity definition record), P is equal to Parameter entity section and T denotes Terminating section.

"Directory Entry section" and "Parameter section" are important sections of this file format. In "Directory section", entities and their attributes like color, width are declared whereas, in "Parameter section" these entities are defined and different parameters of these entities such as if entity is a circle, then its center point, radius are defined there.

V. Vector Graphics Entities

Line, arc, circle and many more entities are supported in these file formats. This section will tell user how these entities are stored in DXF and IGES file format.

A. Line

Line can be described as a shape, which contains infinite number of points. In Euclidean geometry, exactly one line can be found that passes through any two points. Line provides shortest connection between two points.

1) Line in DXF: If in entity section string "0" and "LINE" come then it's the start of the Line code in DXF file. After these lines group code and there respective value would be written in the way as defined in Table II.

TABLE II. Line in DXF

Group Code	Meaning	Sample
10	1st Point's X	4.02
20	1st Point's Y	2.12
30	1st Point's Z	3.20
11	2nd Point's X	4.50
21	2nd Point's Y	2.32
31	2nd Point's Z	3.60

Sample values tell us that line is to be drawn that has the first point and second point as (4.02, 2.12, 3.2), (4.50, 2.32, 3.6) respectively.

2) Line in IGES: If entity number 110 (line) is found in directory section then it is the line entity and its different parameters such as status, color are defined there.

110 2802 0 1 0 0 0 0 D 905
110 0 8 1 0 0 0 0 D 906

This sample states that at line # 2802 of the parameter section, the user can find definition of line entity and its color # would be 8. And parameters of line like its starting point, ending point would be defined in parameter section as defined below,

110,-1.0D0,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0; 905P 2802

This sample states that it is line # 2802 of parameter section and its declaration in directory entry section could be find at line # 905 and first point of line is (-1, 0, 0) and ending point would be (1, 0, 0).

B. Circle

In Euclidean geometry, a circle is the set of all points in a plane at a fixed distance, called the radius, from a given point, the centre.

1) Circle in DXF: If in entity section, a string "0" appears followed by the string "CIRCLE", then it is the start of circle code in DXF file. After these lines group code and there respective value are written in the way as defined in Table III.

TABLE III. CIRCLE IN DXF

Group Code	Meaning	Sample
62	Color Number	225
39	Line Thickness	1.0
10	Center Point's X value	6.8466
20	Center Point's Y value	6.6639
30	Center Point's Z value	0.0
40	Radius of the Circle	10.0

Sample values tell us that circle is to be drawn that has the radius 10 units and center point as (6.8466, 6.6639, 0) and AutoCAD Color Index (ACI) as 225.

2) Circle in IGES: If entity number 100 (circle) is found in directory section, then it is circle entity and its different parameters such as status, color are defined there and parameters of circle entity such as starting point, ending point and center point are defined in parameter section.

C. Solid

Solid is an entity that is used to represent the polygon in drawing, solid has some rules in drawing polygon and that are if line 1 that consists of point 1 and point 3 and line 2 that consists of point 2 and point 4 has intersection point in range of these points then single polygon is divided into two triangles, first triangle consists of point 1, 2 and intersection point and second triangle consists of point 3, 4 and intersection point. And if two lines don't intersect then polygon is in shape of rectangle. Example of solids is given in Figure 2 and Figure 3.

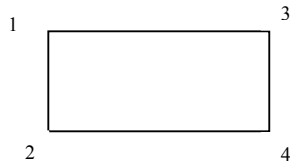

Figure 2. Solid (Rectangle)

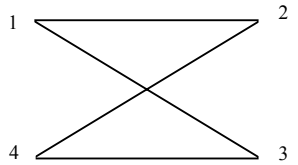

Figure 3. Solid (Tie)

1) Solid in DXF: If in entity section string "0" and "SOLID" come then it's the start of solid code in DXF file. After these lines group code and there respective value would be written in the way as defined in Table IV.

TABLE IV. SOLID IN DXF

Group Code	Meaning	Sample
10	1st Point's X	35.0
20	1st Point's Y	50.0
30	1st Point's Z	0.0
11	2nd Point's X	25.0
21	2nd Point's Y	50.0
31	2nd Point's Z	0.0
12	3rd Point's X	37.5
22	3rd Point's Y	25.0
32	3rd Point's Z	0.0
13	4th Point's X	37.5
23	4th Point's Y	50.0
33	4th Point's Z	0.0

Sample value describes that 4 points are defined for solid if lines intersect each other than two polygons will formed each having 3 vertices and if not intersect than one polygon of 4 points will form.

D. Ellipse

Ellipse is the locus of points on a plane where the sum of distances from any point on the curve to two foci is constant.

TABLE V. ELLIPSE IN DXF

Group Code	Meaning	Sample
10	Center point's X	35.0
20	Center point's Y	50.0
30	Center point's Z	0.0
11	Major axis vector's X	2.012
21	Major axis vector's Y	-0.011
31	Major axis vector's Z	0.0
40	Minor to Major ratio	0.2778
41	Start Angle	0.0
42	End Angle	6.28

1) Ellipse in DXF: In DXF File, ellipse entity can be picked up when following statement in entity section "0" and

"ELLIPSE" encounters. After these lines, group code and there respective values are written as defined in Table V.

Sample value describes that center point is (35.0, 50.0, 0.0), major axis vector is (2.012, 0.011, 0.0), minor to major ratio is 0.2778 and it is complete ellipse.

2) Ellipse in IGES: if entity number 104 (ellipse) is found in directory section, then it is conic entity and its parameters such as its status, color are defined in this section and ellipse's parameter such as starting point, ending point and six quadratic coefficients A, B, C, D, E and F are defined in parameter section.

VI. Vector Graphics Entities Exchanger

IGES and DXF support different number of entities in their format. And they have different parameters for the same entity; means in one format circle has parameters like start point, end point and center point and in other format circle has parameters such as center point and radius. So, first task is to check whether this entity is supported in other format or not if yes then convert its parameters to targeted format's parameter and if not then find its closest match and replace it with its match. Different entities are exchanged in the following headings.

A. Circle

In IGES format, circle has parameters such as center point, start point and end point. And in DXF it has center point and radius. If IGES file is an input file and DXF is targeted one then radius can be calculated from distance formula,

$$rad = \sqrt{(s.x - e.x)^2 + (s.y - e.y)^2} \quad (1)$$

Where, s and e is respectively start and end point. And if DXF file would be inputted one and IGES is targeted one then start point and end point would be calculated using center point.

$$(s.x, s.y) = (c.x + rad, c.y) \quad (2)$$

Where, s and c is start and center point of circle.

B. Arc

Arc entity in IGES file contain center, start and end point whereas, DXF file contain center point, radius, start and end angle. Start angle (θ_2), end angle (θ_1) and radius (rad) can be calculated by these formulas.

$$\theta_1 = \tan^{-1}\left(\frac{e.y - c.y}{e.x - c.x}\right) \quad (3)$$

$$\theta_2 = \tan^{-1}\left(\frac{s.y - c.y}{s.x - c.x}\right) \quad (4)$$

$$rad = \sqrt{(e.x - c.x)^2 + (e.y - c.y)^2} \quad (5)$$

And if DXF file is inputted file then these formulas can be used to calculate start and end point.

$$(s.x, s.y) = (rad * \cos\theta_2, rad * \sin\theta_2) \quad (6)$$

$$(e.x, e.y) = (rad * \cos\theta_1, rad * \sin\theta_1) \quad (7)$$

C. Solid

In DXF format solid consists of four points P_1, P_2, P_3 and P_4 whereas, solid is not an entity in IGES format. So, lines are solid's best match in IGES format. User has to check whether it is rectangular in nature or not. It can be checked by the position of intersection point. Equation of line comprises of P_1 and P_3 can be defined as in [8].

$$y = m_1 * x + c_1 \quad (8)$$

Where m_1 is the slope of line and c_1 is intercept of the line and equation of line comprises of P_2 and P_4 can be defined as in [9].

$$y = m_2 * x + c_2 \quad (9)$$

Equate these equations and get intersection point, if it lies within the range of minimum x, maximum x, minimum y and maximum y then solid is tie in nature otherwise it can be rectangular. Figure 5 depicts how software exchange different graphics format.

VII. Drawing Editor's Graphical Environment

The drawing or graphical editor is a window based Integrated Development Environment (IDE), having basic drawing features similar to commercial products, such as AutoCAD and Pro/E. The 2D graphics canvas is used for vector-graphics based drawings made up of graphics entities such as line, polyline, arc, circle, etc. The graphical environment has a single document interface, which means that only one drawing can be opened at one time. The native 'GID' format files can be opened and saved for later use. The drawing can be imported from AutoCAD's standard DXF format into native 'GID' format. The drawing can be converted to standard G-code instructions and then using G-code, toolpath simulation may be carried out. The editing of drawing entities may be done by changing the properties of individual entities using the properties editor. The properties editor allows changing some properties, while other properties are computed based on given properties and those properties are not editable. The coordinate values are displayed in the status bar of the graphics environment. Figure 4 depicts the software's WPF drawing editor.

VIII. Machine Code from Vector Graphics Entities

Machine code (G-Code) can also be generated from these vector graphics entities.

A. *Line*

G – 01 is the machine code that defines controlled feed rate move in straight line and is more equivalent to vector graphics entity; Line. Its basic syntax is;

G 01 X1.02 Y2.04 Z0.0

Literal after X, Y, Z shows point where current tool position would be transferred. We also know that line has two endpoints, starting and terminating point. If current tool position is either on any endpoint then uses G–01 command and give other endpoint as its parameter and if tool position is not on any endpoint then check which endpoint is closer to current tool position. After this G–00 command could be used to first reach to closest endpoint of line and then uses G–01 command and give other endpoint as its parameter. In this way vector graphics entity (Line) is converted its equivalent machine code.

B. *Circular Arc*

G–02 is the machine code that defines clockwise circular interpolation and G–03 means counter clockwise circular interpolation. Circle and Circular Arc would be converted into machine code by using G–02 and G–03 machine codes. Its basic syntax is;

G03 X1.000 Y2.000 I-0.7500 J1.0000

Literals after X, Y is the point where tool position could be transferred and I, J is dx and dy which could be added to current tool position till final point could come. If current tool position is on start point of circular arc then G–03 is the command to represent circular arc and start point '–` center point is the parameter in G–03 command and if terminating point is on current tool position then G–02 command could be used.

And if current tool position is neither on starting point nor on terminating point then check which point is closer to current tool position then uses G–00 command to first rapid movement of tool position and then uses either G–02 or G–03 to represent Circular Arc or Circle entity.

Figure 6 shows tool bar and menu bar that generates G-code and draw that G-code on drawing canvas to mimics CNC tool's behavior.

IX. Conclusion and Future Enhancements

There are two types of graphics images; Photographic and Models. Drawing of machine parts and building drawings are best represented by vector graphics. Vector graphics file format does not store pixel-by-pixel information and more accurate than raster graphics. Project objectives are to exchange vector graphics file from one file format to another and generate machine code from these vector graphics entities. Firstly, data is retrieved from each file format in form of entities and then find this entity in other format if found then convert it and if not then find appropriate entity match in other format and convert it. And in final phase, G-code of this entity is generated automatically from the opened drawing in the WPF graphics editor. The generated code is then input to CNC Module. The CNC Module contains a simulator for G-code [6], which verifies the generated G-code before it could be sent for actual machining.

In future software will incorporate more entities from each file format and also enhance its working to other vector graphics file formats like STL (stereo lithography) and will incorporate more machine (G and M) codes to enhance CNC simulator functionality.

References

[1] WikiPedia (the Free Encyclopedia) , "Raster Image", WikiPedia, USA, 17th May, 2007. URL:http://en.wikipedia.org/wiki/Raster_image

[2] L. Leurs, "Comparison between Raster and Vector Image", Prepressure pages, USA, January 2007. URL:http://www.prepressure.com/image/bitmapvector.htm

[3] Wikipedia (the Free Encyclopedia), "Drawing Exchange Format", WikiPedia, USA, 20th May, 2007. URL:http://en.wikipedia.org/wiki/DXF

[4] Auto Desk, "DXF References", Auto Desk, USA & CANDA, 2007

[5] Andres lglisies, "Computer Aided Geometric Design and Computer Graphics", Department of Mathematics and Computational Science, University of Cantabria, USA. URL:http://personales.unican.es/iglesias/CD-CG/IndustrialFormats.pdf

[6] Muhammad Imran et al., "Computer Numerical Control Software Featuring Buffer-Board Simulation and Tool-path Simulation", INMIC, Karachi, 2008.

Appendix 1: Software's Screen Shots

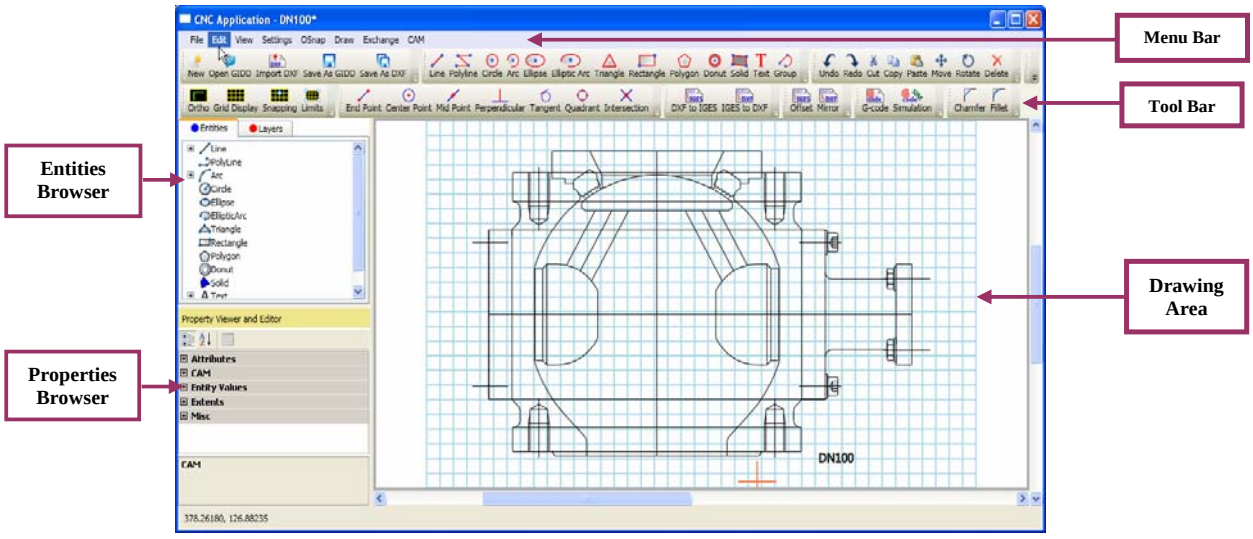

Figure 4. WPF Vector Graphics Editor

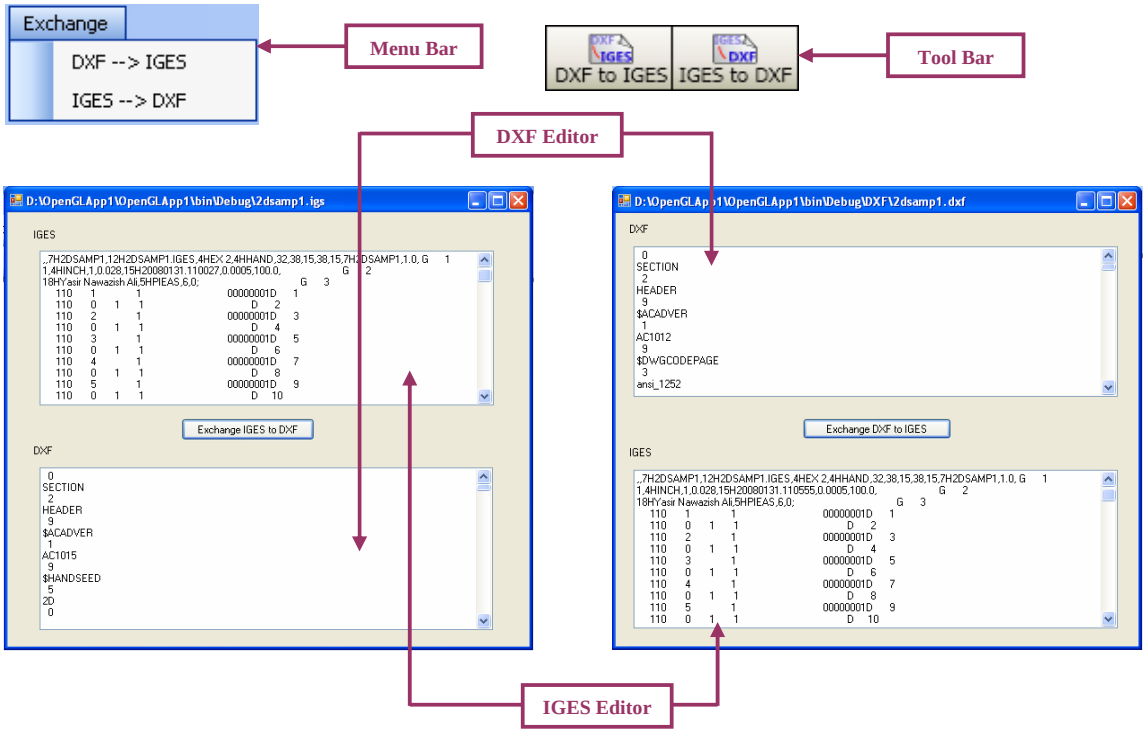

Figure 5. Vector Graphics Exchange Software

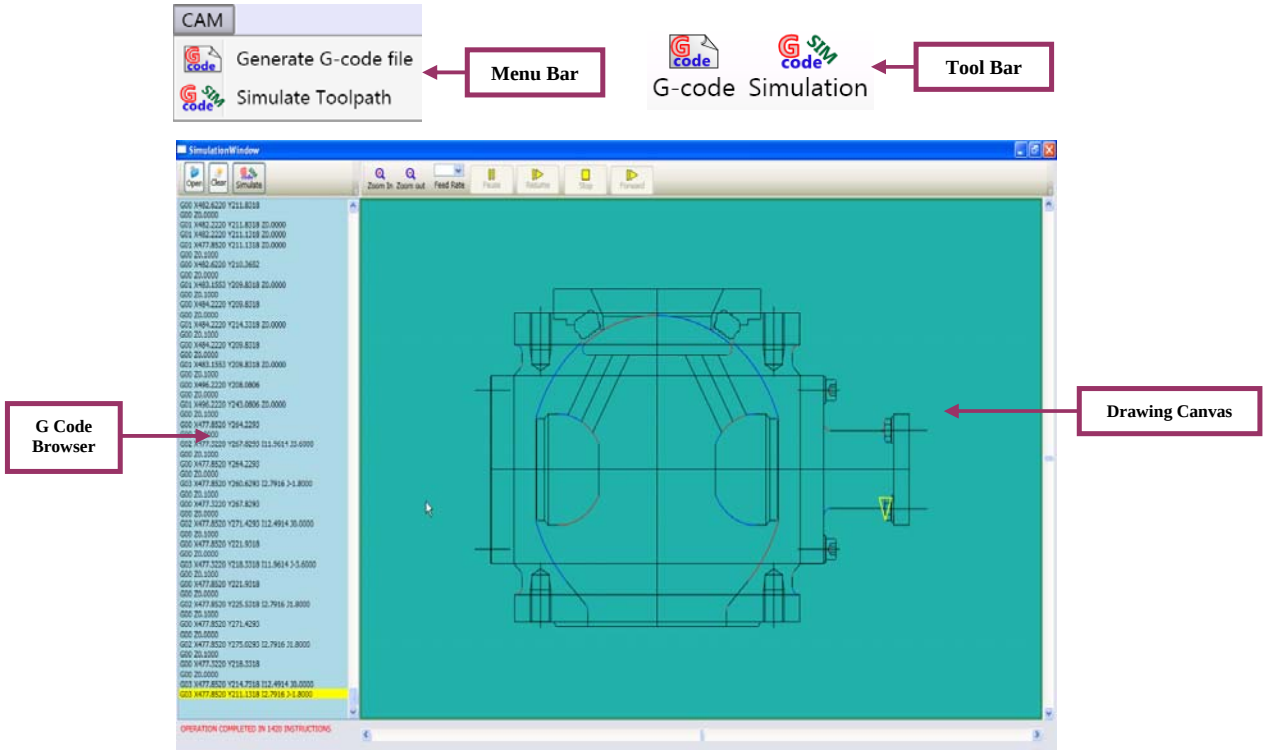

Figure 6. Vector Graphics CAM Module