Jurnal Mekanikal
Disember 2003, Bil. 16, 31 - 46

FEATURE EXTRACTION FROM STEP AP224 FILE SETS

Azwan Iskandar Azmi
School of Information Technology
Universiti Utara Malaysia
06010 Sintok
Kedah

Jamaludin Mohd Taib
Zulkepli Muhammad
Faculty of Mechanical Engineering
Universiti Teknologi Malaysia
81310 UTM Skudai
Johor

ABSTRACT

Feature technology transforms the low-level geometrical definition of a 3D model into high level product definition, aiming for full automation of downstream manufacturing activities. Today, feature has been implemented in STandard for Exchange of Product Model specifically in Application Protocol 224 (STEP AP224) which utilizes manufacturing feature definition based on prismatic and turning parts. Hence, this research is carried out to study how STEP AP data is stored and how the features can be extracted. The extracted features are then used to automate the generation of the NC Code for the milling process. This work is limited to isolated milling features on 2.5D parts or prismatic parts.

Keywords: *Feature Technology, NC Code generation, neutral file, Computer Aided Design (CAD), Computer Aided Manufacturing (CAM).*

1.0 INTRODUCTION

Over the past decades, part modeling in CAD system has been utilizing feature technology as the key to integrate design and manufacturing. The methodologies proposed for creating high level feature definition design have been taking these two approaches: (i) feature based design and (ii) feature recognition [1]. The former is implemented in CAD, whilst the latter is used in CAM. Nevertheless, they have a lot of similarities in defining features.

Due to the nature of most neutral files such as ACIS, PARASOLID and IGES, these neutral files only employ the boundary representation (B-Rep) model for communication or translation purposes of dissimilar CAD systems. Furthermore, re-input, translation or re-interpretation of part design data that used the feature technology will usually happen while using CAM, which is actually wasting the feature information of the design embedded on the CAD system [2].

Currently, STEP standard is the only neutral file format that utilizes feature technology as the key for file transfer and sharing. In addition, STEP file format applies object oriented database method to map the relationship within the file data structure. This is unique to STEP as opposed to other types of neutral file. Thus, this research aims to study the characteristics of STEP file especially the STEP AP224 file. A system has been developed to extract the feature definition and to use this information to automatically generate the NC code. This paper will emphasize on the method of extracting the feature information of the STEP AP224 file. However, this paper will not cover the methods to write the NC Code. The methods to write NC Code have been covered by the first author in his thesis [3].

The paper is organized as follows. Section 2 reviews the literature on feature technology. Then, section 3 discusses STEP as a neutral file format as well as its important characteristics. Section 4 proposes the algorithm and detail discussion of the developed algorithm. Discussion and conclusions are presented in Section 5.

2.0 LITERATURE REVIEW

The impact of feature technology on downstream manufacturing activities is still unverified despite the fact that feature technology has been a major topic of research for the past two decades. Since the earliest work on feature technology by Kyprianou [4], the research has grown substantially. Initial efforts emphasized on the attention to recognize isolated features [5,6] followed by recognition of interacting features [7,8]. Despite the fact that feature recognition system is capable of recognizing interacting features, its implementation on downstream activities has not been carried out. Further research now include manufacturability of the features [9,10].

Feature technology research has matured adequately and has been implemented in STEP AP224 [11]. Feature technology research has shifted to the development of translator to accommodate STEP AP224. Some of the related work in this field is as carried out by Bhandarkar [12] and Han *et al* [13].

However, translation of the STEP AP224 files has not been addressed adequately. Study on the characteristics of the STEP AP 224 is essential as to develop the approach to extract data information of the feature. Initial work in this area, as reported in this paper, is limited to isolated features which are holes and slots. Even though it is limited to these two features, the proposed algorithm to extract feature information can be implemented to other types of isolated feature.

3.0 STEP DATA EXCHANGE FORMAT

STEP provides a standardization method to define all data necessary for the description of a CAD model throughout the life cycle of the model. STEP file is in ASCII file format and it covers product related information along with mechanism and definition for product data exchange and sharing.

STEP was first issued in 1994 and this standard categorizes the CAD data based on Application Protocol (AP). APs are used to define the collection of information required for the Base Exchange unit. Each AP is known as a formal document that describes the activities of the lifecycle of a CAD model. Currently, there are a few APs that are actively used. For examples, AP 201 is for explicit draughting, AP 203 for 3D design and the latest one, AP238 is for STEP-NC Manufacturing.

Of particular interest in this paper is the STEP AP224. It is an ISO STEP application protocol that defines the manufacturing information using manufacturing features to machine discrete mechanical parts.

3.1 STEP AP224-Mechanical Product Definition using Machining Features

STEP AP224 defines the model ranging from low geometrical definition to high level product definition. Topology and geometrical information of BReps model is the low-level definition and manufacturing and machining features are the high level product definition.

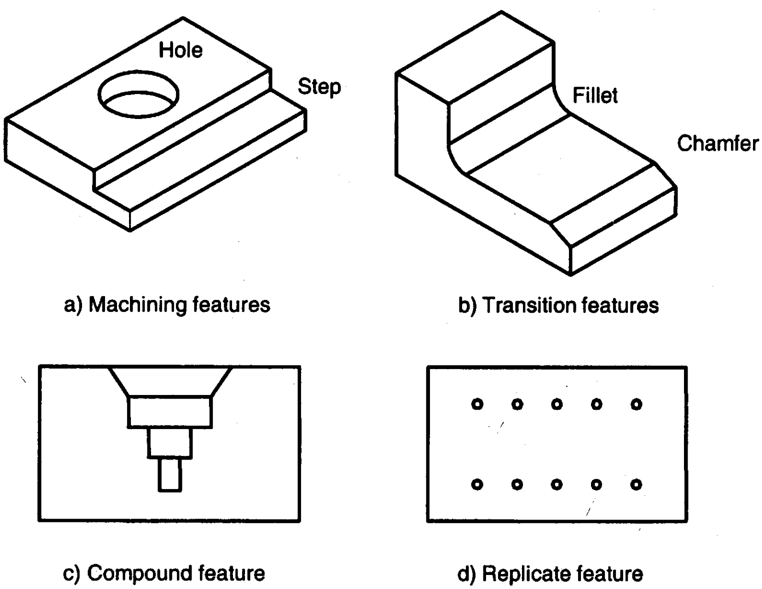

Figure 1 The defined machining and manufacturing feature groups of STEP AP224 [11]

In STEP AP224, four groups of manufacturing features are defined as illustrated in Figure 1:

a. Machining features
b. Transition features
c. Compound features
d. Replicate features

The main interest of this paper is feature definition, which is based on manufacturing features associated with the milling and turning processes. Manufacturing and machining features are two different things. Manufacturing features are the portions of a part with some manufacturing significance [1] where different manufacturing features will use different sets of manufacturing processes associated with them. In this case, the machining features are those features in which the portion of the workpart will be removed by certain machining operations [1].

3.2 STEP AP224 Data Structures

STEP file is described by a structured language called EXPRESS. EXPRESS maps the relationship between information by binding and inheritance data storage. A portion of STEP file that defines the plane surface is shown below:

```
#212=CARTESIAN_POINT('',(3,0.0,5.0));
#213=DIRECTION('',(0.0,0.0,1.0));
#214=DIRECTION('',(1.0,0.0,0.0));
#215=AXIS2_PLACEMENT_3D('',#212,#213,#214);
#216=PLANE('',#215);
```

In this example, line #216 states that plane surface has two entities, label as denoted by symbol '' and axis2_placement_3D denoted by #215. Since this plane surface is not labeled, there is no string inside the single quotation. The other entity is inherited from line #215 which is axis2_placement_3D. As the data is further retrieved, by referring to line 215, axis_placement_3D has 4 entities, label, Cartesian point, axis direction and reference direction. All the attributes except for the initial one is inherited as specified.

STEP file begins with the keyword ISO-10303-21 and is followed by Header. Header basically contains the descriptions of the file, such as the name of the file, the date and time of stamp, author's name, organization name that generates the file and the type of file schema or name. The list below shows the header section:

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('AP224 File'),'2;1');
FILE_NAME('97057_611223-5_c.224','1997-08-
01T11:08:30 05:00',('RPTS
Operator'),('RPTS'),'RPTS MP 6.0','PTC
Pro/ENGINEER Version 18.0','RPTS Operator');
FILE_SCHEMA(('FEATURE_BASED_PROCESS_PLANNING'));
```

Then, it is followed by the Data Section. This section is divided into two groups; the geometrical and topological, and form feature Data. The geometrical and topological data defines geometrical entity and topological elements. This data is similar to other application protocols that have topology and geometrical BReps data, such as AP203 and AP214. The data ends with the highest topology in B-Reps, which is solid. In STEP, the solid is equivalent to MANIFOLD_SOLID_BREP entity. Therefore, this entity can be inferred as the end of the topological and geometrical data.

The other data that is unique to STEP AP224 is the manufacturing feature definition. A feature in STEP AP224 is defined by the sweeping of a profile along a given path and its end profile condition. Each feature is described by its geometrical and parametric shape. The geometrical shape will inform the type of profile, path and end profile condition. Whilst, the parametric shape will establish the parameter for each profile, path and end profile condition. For instance, a through hole is described by sweeping of a closed circular profile swept along a linear path and the end condition is 'through'. Whilst for the parameter shape will ascertain the radii of the circular profile and the distance of the sweeping. These data is stored through binding and inheritance. Sample of these elements from the STEP file can be seen in the system architecture section.

Due to the method of storing the data, which is by binding and inheritance, frequent line referring is always required. It becomes more cumbersome when the reference line has to be referred to the previous line, causing the reader to read backwards. Most programming languages, such as FORTRAN, C and C++, actually read line by line, starting from the first to the following line. Due to the nature of data mapping in STEP, backward line reading is required. However, backward reading cannot be carried out by most programming languages unless the reading process is terminated and a new reading process is carried out. Hence, backward reading which results from the characteristics of STEP file is not the solution in writing an efficient source code.

This becomes worst when a line can be a mixture of numbers and string. This causes every line to be handled as strings. In order to differentiate between numbers and strings, the system has to truncate certain string and convert them to numbers. The truncating and converting process requires the system to fully understand every single line. Retrieving the data using this method is impossible especially when the attributes vary as the position of the number changes.

The solution to this is to search the keyword within the line. If the keyword is found, the system will compare another keyword to retrieve the geometry of the features. For example, the following is taken from STEP file to define the height of the stock.

```
#482=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()M
EASURE_WITH_UNIT(LENGTH_MEASURE(12.50),#15)QUALIFIED_REPRESENT
ATION_ITEM((#474,#475,#476))REPRESENTATION_ITEM('height'));
```

The first keyword is REPRESENTATION_ITEM('height') and the second, LENGTH_MEASURE(. In order to retrieve the height, the system will compare the whole line (called *descr* in this sample program) with the second keyword. The comparison is done letter by letter until the whole second keyword is detected. When this happens, the system will save the letter until it reaches ')'. The programming is as shown below:

```
if (   strstr (descr, first_kw) != NULL &&
       strstr(descr, sec_kw) != NULL)    //
       {
       for (int i = 0 ; i < len_descr ; i++ )
             {
             int checker = 0;
             if (descr[i] == sec_kw[0])
                   {
                   for (int j = 0 ; j < len_sec ; j++ )
                         {
                         checker = 1;
                         if (descr[i+j] != sec_kw[j])
                               checker = 2;
                         }
                   }
             if (checker == 1 )
                   {
                   while ( descr[i+len_sec+counter] != ')')
                         {
                         numeric_string[counter]                    =
                         descr[i+len_sec+counter];
                         counter = counter + 1;
                         }
                   numeric_string[counter+1] = '\0';
                   i = i+len_sec+counter;
                   }
             }
       }
```

Both keywords appear several times in STEP file, for instance, when retrieving the stock height and the height of a hole. Therefore, using only the keyword to retrieve the geometrical properties of stock and each feature can cause confusion. Therefore, the position of information is used for guideline.

The data extraction method used in this research is a rule and hierarchical-based approach. The procedure for reading the STEP AP224 file is summarized as an algorithm as shown below:

<u>Extract Base Shape and Features Geometric Data Algorithm</u>

```
Read/ Open STEP file
Read the file
Retrieve 'base shape'
      Record length, width, height; continue;
Retrieve feature information
```

```
    if (feature == 'round_hole')
          {
          Record hole placement and diamater;
          Record hole depth and bottom condition;
          }
    else if (feature == 'slot')
          {
          Record slot placement;
          Record slot length, width and height;
          }
Retrieve material information
          Record Material_Designation;
```

4.0 SYSTEM ARCHITECTURE

The overall architecture of the system, from the STEP AP224 feature definition as the input to the NC code generation, is illustrated in Figure 2. It is a two-level process; i) the feature is extracted from the STEP AP224 file sets, ii) process planning procedure mainly for NC Code generation.

The input of the system is limited to prismatic part with machining features. Since there are a number of features available to be studied, this paper will focus only on hole and slot. This is because of the fact that the procedure to extract other types of features will use similar approach since the characteristics of STEP AP224 in storing the data is also similar.

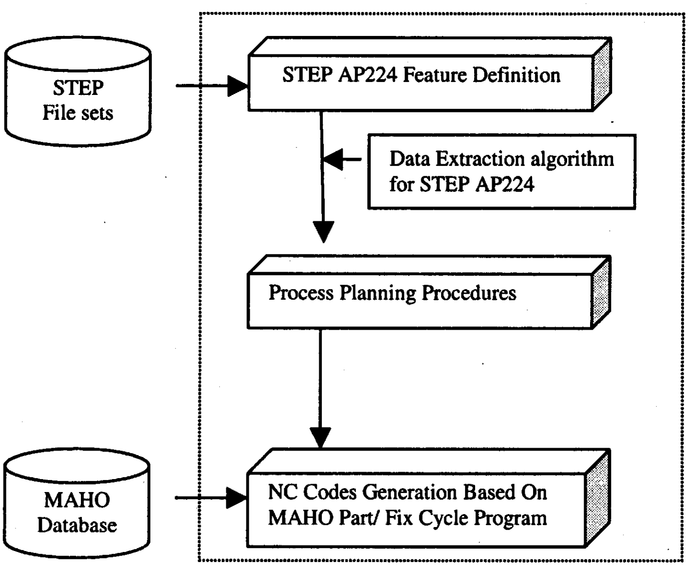

Figure 2 System Architecture

Once the hole and slot feature information is extracted, both feature information will undergo process planning. The process planning consists of defining the machining method, tool size and tool type. The tool size and type are retrieved from the MAHO database. With this information, the NC Code can be written. However, this stage is not covered in this paper. The following sections will elaborate on the feature extraction.

4.1 Feature Extraction

The following sections discuss the method to search stock, hole and slot from STEP AP224 file sets.

a) Stock

In STEP AP224, stock is called block shape. The following is the portion of STEP AP224 to define the stock.

```
#482=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()M
EASURE_WITH_UNIT(LENGTH_MEASURE(12.50),#15)QUALIFIED_REPRESENT
ATION_ITEM((#474,#475,#476))REPRESENTATION_ITEM('height'));
...
#491=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()M
EASURE_WITH_UNIT(LENGTH_MEASURE(225.00),#15)QUALIFIED_REPRESEN
TATION_ITEM((#483,#484,#485))REPRESENTATION_ITEM('length'));
...
#500=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()M
EASURE_WITH_UNIT(LENGTH_MEASURE(100.00),#15)QUALIFIED_REPRESEN
TATION_ITEM((#492,#493,#494))REPRESENTATION_ITEM('width'));
#501=BLOCK_SHAPE_REPRESENTATION('base_shape',(#473,#482,#491,#
500), #34);
```

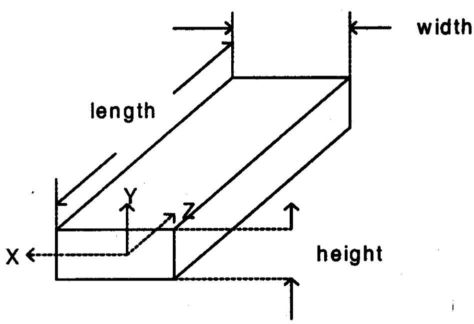

Figure 3 Stock geometrical properties set by STEP AP224

The important data for stock is its geometrical properties; width, length and height, which is illustrated in Figure 3. The keywords used here are as follows.

BLOCK_SHAPE_REPRESENTATION : prismatic part (milling process)

REPRESENTATION_ITEM('length') : length of the stock
REPRESENTATION_ITEM('width') : width of the stock
REPRESENTATION_ITEM('height') : height of the stock

In detecting the prismatic part, only one keyword is used as the information required from the line is just the type of model. Whilst, for the geometrical properties, the second keyword is LENGTH_MEASURE (in addition to the REPRESENTATION_ITEM).

b) Hole

The presence of feature is indicated by application elements called INSTANCED_FEATURE and the key word used here is MACHINING_FEATURE( )ROUND_HOLE( ). The sample is as follows.

```
#508=(CHARACTERIZED_OBJECT('','')FEATURE_DEFINITION()INSTAN
CED_FEATURE()MACHINING_FEATURE()ROUND_HOLE()SHAPE_ASPECT(''
,'',#43,.T.));
```

The geometric shape of hole is circular closed profile that is swept along a linear path. This shape is stated in the file sets as follows.

Hole

```
#519=FEATURE_COMPONENT_DEFINITION('','');
#520=PRODUCT_DEFINITION_SHAPE('','',#519);
#521=CIRCULAR_CLOSED_PROFILE('circular_profile','',#520,.F.);
...
#533=SHAPE_DEFINITION_REPRESENTATION(#532,#531);
#534=FEATURE_COMPONENT_DEFINITION('','');
#535=PRODUCT_DEFINITION_SHAPE('','',#534);
#536=PATH_FEATURE_COMPONENT('linear path','linear',#535,.F.);
```

After the hole is detected, line #511 states the position of the hole using Cartesian coordinate system. The portion of STEP AP224 file to locate the position of the hole is as follows:

```
#509=PROPERTY_DEFINITION('','',#508);
#510=PRODUCT_DEFINITION_SHAPE('','',#508);
#511=CARTESIAN_POINT('',(112.5,-50.00,12.50));
```

The other geometries that are important for machining the hole are the depth and diameter of the hole. The key words are bolded in the example. The following shows a related portion of the STEP AP224 files for definition of hole diameter and its depth.

Diameter of the hole feature

```
#530=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()M
EASURE_WITH_UNIT(LENGTH_MEASURE(75.00),#15)QUALIFIED_REPRESENT
ATION_ITEM((#522,#523,#524))REPRESENTATION_ITEM('diameter'));
```

Hole depth

```
#541=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()M
EASURE_WITH_UNIT(LENGTH_MEASURE(12.50),#15)REPRESENTATION_ITEM
('depth'));
```

The final attribute for the hole is its bottom condition which can either be through or with other bottom condition such as spherical, flat with radius bottom, flat with bottom taper, flat or conical (refer to Figure 4). The keyword is HOLE_BOTTOM.

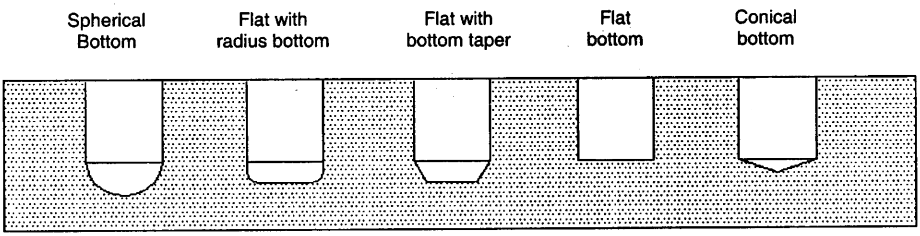

Figure 4 Bottom condition for hole (besides 'through')

The related portion from STEP AP224 that defines the bottom condition of the hole is as follows.

Hole bottom condition

```
#553=HOLE_BOTTOM('bottom_condition','through',#552,.F.);
#554=SHAPE_ASPECT('','bottom_condition_occurrence',#510,.T.)
#555=FEATURE_COMPONENT_RELATIONSHIP('','hole_bottom
usage',#553,#554);
```

c) Slot

The following will elaborate the method to retrieve a slot in a similar way as the hole feature; slot detection, geometry and lastly attributes. The keywords are bolded.

Slot (explicit geometric shape)

```
#1154=SHAPE_DEFINITION_REPRESENTATION(#1136,#1153);
#1155=FEATURE_COMPONENT_DEFINITION('','');
#1156=PRODUCT_DEFINITION_SHAPE('','',#1155);
#1157=SQUARE_U_PROFILE('','swept shape',#1156,.F.);
...
#1201=PRODUCT_DEFINITION_SHAPE('','',#1200);
#1202=PATH_FEATURE_COMPONENT('linear
path','linear',#1201,.F.);
```

Slot Feature Detection

```
#1135=(CHARACTERIZED_OBJECT('','')FEATURE_DEFINITION()INSTANCE
D_FEATURE()MACHINING_FEATURE()SLOT()SHAPE_ASPECT('','',#43,.T.
));
```

Similar to hole, line #1138 states the Cartesian of point and two vectors. The Cartesian point is actually the point that is shown in Figure 5. The first vector is direction and the second vector is in the reference direction. The second vector is equal to the orientation of the slot.

```
#1138=CARTESIAN_POINT('',(300.0,-27.5,12.5));
#1139=DIRECTION('',(-1.0,0.0,0.0));
#1140=DIRECTION('',(0.0,-1.0,0.0));
#1141=AXIS2_PLACEMENT_3D('orientation',#1138,#1139,#1140);
```

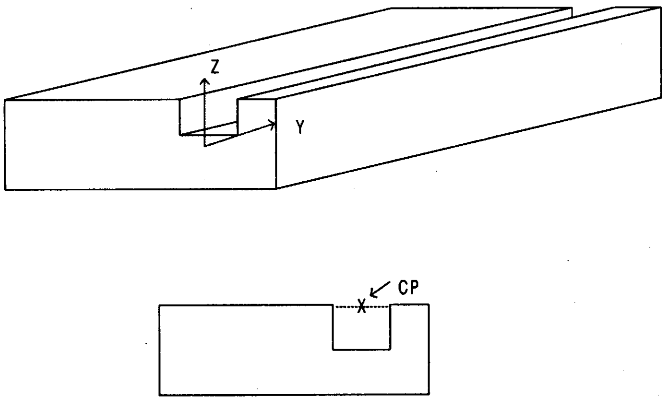

Figure 5 The reference point for Cartesian Coordinates

Slot is created from sweeping process of a profile that is called SQUARE_U_PROFILE as shown in the following line;

```
#1157=SQUARE_U_PROFILE('','swept shape',#1156,.F.);
```

After the profile is defined, the system searches the parameter shape of the profile. The related geometry of the profile is shown in Figure 6.

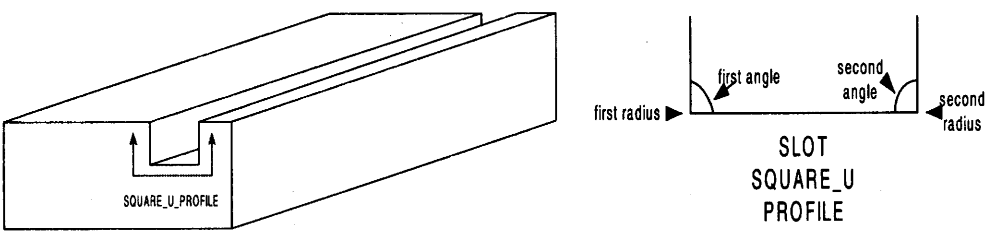

Figure 6 Geometry of the slot

The list of program below illustrates the search for the parameter shape. The keywords used to identify the parameter shape are bolded.

Depth

```
#1150=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(12.5),#15)QUALIFIED_REPRESENT
ATION_ITEM((#1142,#1143,#1144))REPRESENTATION_ITEM('depth'));
```

Width

```
#1166=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(25.0),#15)QUALIFIED_REPRESENT
ATION_ITEM((#1158,#1159,#1160))REPRESENTATION_ITEM('width'));
```

First angle

```
#1175=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_AN
GLE_MEASURE(90.0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()QUALIFIED
_REPRESENTATION_ITEM((#1167,#1168,#1169))REPRESENTATION_ITEM('
first angle'));
```

Second angle

```
#1184=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_AN
GLE_MEASURE(90.0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()QUALIFIED
_REPRESENTATION_ITEM((#1176,#1177,#1178))REPRESENTATION_ITEM('
second angle'));
```

First radius

```
#1189=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0),#15)REPRESENTATION_ITEM(
'first radius'));
```

Second Radius

```
#1194=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0),#15)REPRESENTATION_ITEM(
'second radius'));
```

Distance

```
#1211=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(300.0),#15)QUALIFIED_REPRESEN
TATION_ITEM((#1203,#1204,#1205))REPRESENTATION_ITEM('distance'
));
```

The last element is the end condition called SLOT_END. Open slot end means the slot is through from one side to another side of the stock.

```
#1219=FEATURE_COMPONENT_DEFINITION('','');
#1220=PRODUCT_DEFINITION_SHAPE('','',#1219);
#1221=SLOT_END('','open',#1220,.F.);
#1222=FEATURE_COMPONENT_DEFINITION('','');
```

d) Material

Apart from the stock and feature information, the system has to identify the material and unit used.

```
#19=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#816=MATERIAL_DESIGNATION('1020 STEEL PER MIL-S-7952',(#815));
```

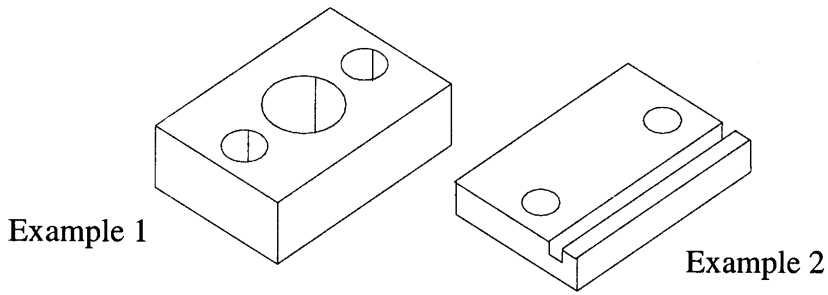

Figure 7 CAD models for illustrative examples

5.0 DISCUSSIONS AND CONCLUSION

The system developed is implemented in a C language environment. It reads STEP AP224 to extract information on stock, features and material specifications. Then, it writes the NC Code. This system successfully automates the NC Code generation based on features. Two sample models were used here as shown in Figure 7. The sample NC Codes generated are given in the appendix and have been tested in a MAHO CNC simulator available in the Production Laboratory, Faculty of Mechanical Enginering, Universiti Teknologi Malaysia. Simulation results showed that the generated NC codes run without any tool collisions with the stock and there is also no unwanted machining during the simulation.

However, this paper only emphasizes on the algorithm in reading the STEP AP224 file sets. The first file sample consists of 854 lines and the second, 1837 lines. This system successfully reads the file sets a single time and successfully retrieves all the rèquired information. The reasons are as follows

i. Reading the data according to the position of the data in the file sets. The first data is unit, followed by stock data, feature and material.

ii. Using the keywords to retrieve data. C language reads the file line by line. Every line is treated as a string and by detecting the keywords, only relevant part of the line is retrieved.

This study has been limited to 2.5D parts or prismatic parts with isolated features such as round holes, pockets and slots. The system should be further developed to include other features on both prismatic and turning parts.

REFERENCES

1. Shah, J.J. and Mantyla, M., 1995, *Parametric and Feature-Based CAD/ CAM-Concepts, Techniques and Applications.* Wiley, New York.
2. Karadkar, B.J. and Pande, S.S., 1996, Feature based automatic *CNC code generation for prismatic parts. International Journal of Computers in Industry.* Vol. 28, pp 137-150.
3. Azwan Iskandar Azmi, 2002, *Study on the implementation of STEP AP224 in process planning,* M.Sc Thesis, Universiti Teknologi Malaysia.
4. Kyprianou, L. K., 1980, *Shape Classification in Computer Aided Design*, PhD Thesis, University of Cambridge, Cambridge
5. Choi, B. K., 1982, *CAD/CAM compatible and tool-oriented process planning system*, Ph.D. Thesis, Purdue University, West Lafayette, Indiana.
6. Henderson, M. R., 1984, *Extraction of feature information from three dimensional CAD data*, PhD Thesis, Purdue University, Indiana.
7. Corney, J. R., 1993, *Graph Based Feature Recognition*, Ph.D. Thesis, Heriot-Watt University.
8. Sakurai, H., 1995, Volume decomposition and feature recognition: Part 1 - Polyhedral objects, *Computer Aided Design*, Vol. 27, No. 11, pp 833-843.
9. Han, J. H., 1996, *3D geometric reasoning algorithms for feature recognition*, Ph.D. Thesis, University of Southern California.
10. Vandenbrande, J. H., 1990, *Automatic recognition of machinable features in solid models*, Ph.D. Thesis, University of Rochester.
11. ISO, 2001, *Mechanical Product Definition for Process Planning using Machining Features 2nd Edition*
12. Bhandarkar, M.P. and Nagi, R., 2000, STEP-Based feature extraction from STEP geometry for agile manufacturing, *International Journal of Computers in Industry. Vol.* 41, pp. 3-24.
13. Han, J. H, M. Kang and H Choi, 2001, STEP-based feature recognition for manufacturing cost optimization, *Computer Aided Design*, Vol 33, pp 671-686.

APPENDIX

NC codes for Example1:

```
PM
N9004
N1 G18        (XZ Plane)
N2 G51        (Cancel offset of G52)
N3 G53        (Delete stored zero datum shift)
N4 G52        (Automatic shift value of zero datum shift)
N5 G71        (Metric Input System)
N6 G98 X-100.000 Y-12.500 Z-100.000 I325.000
J112.500 K200.00  (Framing window plane)
N7 G99 X0.000    Y-12.500 Z0.000    I225.000 J12.500  K100.00
      (Framing workpiece)
N8  M8        (Coolant on)
N9  T2  M6    (Tool change to machine hole1)
N10 G42
N11 F201.655   S763.845   M3
N12 G89   (Z-12.500) B2   R37.500   I60   J-1   K2.5
N13 G79   X112.500   Y0.000   Z50.000
N14 G0    Y12.500
N15 T4    M6        (Tool change to machine hole2)
N16 G42
N17 F63.662   S795.775  M3
N18 G89   (Z-12.500)   B2   R15.600   I60   J-1   K2.5
N19 G79   X187.500   Y0.000   Z50.000
N20 G0    Y12.500
N21 T6    M6        (Tool change to machine hole3)
N22 G42
N23 F63.662   S795.775  M3
N24 G89   (Z-12.500)   B2   R15.600   I60   J-1   K2.5
N25 G79   X37.500   Y0.000   Z50.000
N26 G0    Y12.500
N27 M9              (Coolant off)
N28 M5              (Spindle stop)
N29 T0  M6    (No tool change)
N30 M30             (Program stop)
```

Tool Library For Pocket Milling of the features

```
T2  R12.500
T4  R10.000
T6  R10.000
```

NC codes for Example2:

```
PM

N9004
N1 G18        (XZ Plane)
N2 G51        (Cancel offset of G52)
N3 G53        (Delete stored zero datum shift)
N4 G52        (Automatic shift value of zero datum shift)
N5 G71        (metric Input System)
```

```
N6 G98 X-100 Y-25.000   Z-100    I400.000  J125.000   K250.000
      (Framing window plane)
N7 G99 X0     Y-25.000   Z0       I300.000  J25.000    K150.000
      (Framing workpiece)
N8  M8       (Coolant on)
N9  T2  M6   (Tool change to machine hole1)
N10 G42
N11 F233.767  S2387.326  M3
N12 G89  (Z-25.000)   B2        R8.300   I60   J-1   K2.5
N13 G79  X281.200     Y0.000   Z75.000
N14 G0   Y25.000
N15 T4  M6   (Tool change to machine hole2)
N16 G42
N17 F233.767  S2387.326  M3
N18 G89  (Z-25.000)   B2        R8.300   I60   J-1   K2.5
N19 G79  X18.700      Y0.000   Z75.000
N20 G0   Y25.000
N21 T6   M6 (Tool change to machine slot1)
N22 G42
N23 F467.534  S1909.861  M3
N24 G87 X320.000 Y25.000 Z12.500  B2 R10.000  I60   J-1   K2.5
N25 G79  X150.000   Y0.000   Z27.500
N26 G0   Y12.500
N27 M9              (Coolant off)
N28 M5              (Spindle stop)
N29 T0  M6  (No tool change)
N30 M30             (Program stop)
```

Tool Library For Pocket Milling of the features

```
T2 R4.000
T4 R4.000
T6 R5.000
```