STEP- Based Assembly Feature Recognition Using Attribute Adjacency Graph for Prismatic Parts

Dr.Hussein Salem Ketan
Engineering College/Mechanical Engineering Department / University of Baghdad
Email:hussketan@yahoo.com
Aseel Jameel Haleel
Engineering College/Mechanical Engineering Department / University of Baghdad

Received on: 11/7/2012 & Accepted on: 7/3/2013

ABSTRACT

This paper introduces the concept of STEP AP203(STandard for Exchange of Product model data) an ISO standard as a neutral format for exchange of CAD model data between different CAD/CAM systems, and how STEP AP 203 data is stored and how the feature information can be extracted and recognized from STEP file. In this paper a hybrid (graph and rule) based approach is used to recognize the features of mechanical prismatic parts. The Attribute Adjacency Graph (AAG) and Attribute Adjacency Matrix (AAM) approaches are used to recognize the form features, and rule based approach is used to recognize assembly features.The proposed methodology in this paper has been completely implemented by designing an integrated system called STEP-based Assembly Sequence Planning (ST-ASP) system. The (ST-ASP) system is built by using Visual Basic 6.0 supported by Solid works 2011 package and implemented on (HP Pavilion dv6) PC. The (ST-ASP) system is directed to 3D prismatic parts. The form features explored in this system include both depression and protrusion features, and the assembly mating relations explored in this system include; against, fit, and insert which is used in recognize assembly features. Finally the system has been tested to carry out a case study to demonstrate the feasibility of the proposed methodology.

Keywords:STEP, feature recognition, form feature, assembly feature, mating relations, attribute adjacent graph (AAG).

تمييز السمات التجميعية باعتماد نموذج تبادل البيانات القياسي للمنتج بأستخدام مخطط صفات التجاور للاشكال الموشورية

الخلاصة

هذا البحث يوضح مفهوم نموذج تبادل البيانات القياسي للمنتج كواحد من معايير الايزو10303 كصيغة قياسية لتبادل بيانات انظمة التصميم المعان بالحاسوب بين مختلف انظمة التصميم والتصنيع المعان بالحاسوب, وكيفية حفظ بيانات المنتج في نموذج تبادل البيانات القياسي للمنتج(AP203) وكيفية استخلاص معلومات السمات من نموذج تبادل البيانات القياسي للمنتج وتمييزالسمات.سمات الشكل للاشكال الموشورية تم تمييزها بأستخدام مخطط صفات التجاور ومصفوفة صفات التجاور اما السمات التجميعية تم تميزها بأعتماد (rule basedapproach).المنهجية المقترحة في هذا البحث تم تنفيذها بالكامل بواسطة تصميم نظام متكامل يدعى (ST-ASP), تم بناء نظام (-ST

(ASP بأستخدام لغة (Visual Basic 6.0) وبرنامج (SolidWork2011) وتم تنفيذه على حاسبة شخصية (HP Pavilion DV6) بنظام تشغيل (Windows7).في هذا البحث تم تمييز سمات الشكل (المقعرة والمحدبة) والعلاقات التجميعية(التقابل, التطابق, التداخل) المستخدمة لتمييز السمات التجميعية. اخيرا تم اختبار النظام على منتج ميكانيكي افتراضي لاستعراض جدوى المنهجية المقترحة.

INTRODUCTION

During the last two decades much effort has been devoted to the area of feature technology to ease the problem of Computer Aided Design (CAD) integration with downstream activities.Features give a higher conceptual meaning to component characteristic by dissecting component geometry into recognizable and meaningful forms, and this is very important in the manufacturing context as they can capture the higher level engineering content of a part. Therefore, features are considered the communication medium between design and manufacture. Feature technology is categorized by two popular approaches is: Automatic Feature Recognition (AFR) and Design by Feature [1]. Assembly features have been used by several researchers to improve the efficiency of the assembly planning process, but there have been significant variations in the definition of an assembly feature. Holland and Bronsvoort (2000) define an assembly feature as" all assembly specific information within modeling and planning". Their definition consists of two types of assembly information, one is generic level information used to handle a component and other is the instance level information about the connections between components [2]. Shyamsundar and RajitGadh (2001) define assembly feature as "a property of an assembly unit (AU) with respect to (or in the context of) other component(s), which provides assembly related informationrelevant to the design, manufacture or function of the product assembly" [3]. Zha and Du (2002) define assembly features as "particular form features that affect assembly operations, which are defined by connectors" [4]. Chan and Tan (2003) define an assembly feature as "the elementary connection feature containing mating relationsbetween the components" [5].Hamidullahet. al. (2006) define assembly features as "a connection between two form features on mating parts, associated with assembly intents. Assembly intents include assembly and/or mating relations, assembly operation (e.g. fastening operation, and fusion operation), and other assembly attributes (e.g. feature's position and orientation etc.)"[3]. Assembly feature transform, interpret CAD model to provide information to assembly planning for automated information interpretation process [6]. A number of standardshave been developed for transferring product data, such as IGES and STEP. Thesestandards specify the format and contents of physical files; STEP provides a representation and exchange standard of product data in a formal manner interpretable by computers[7, 8].

METHODOLOGY

The proposed methodology to recognize assembly feature composed of three components:parts representation, form features Recognition, and assembly feature recognition as shown in the Figure (1).

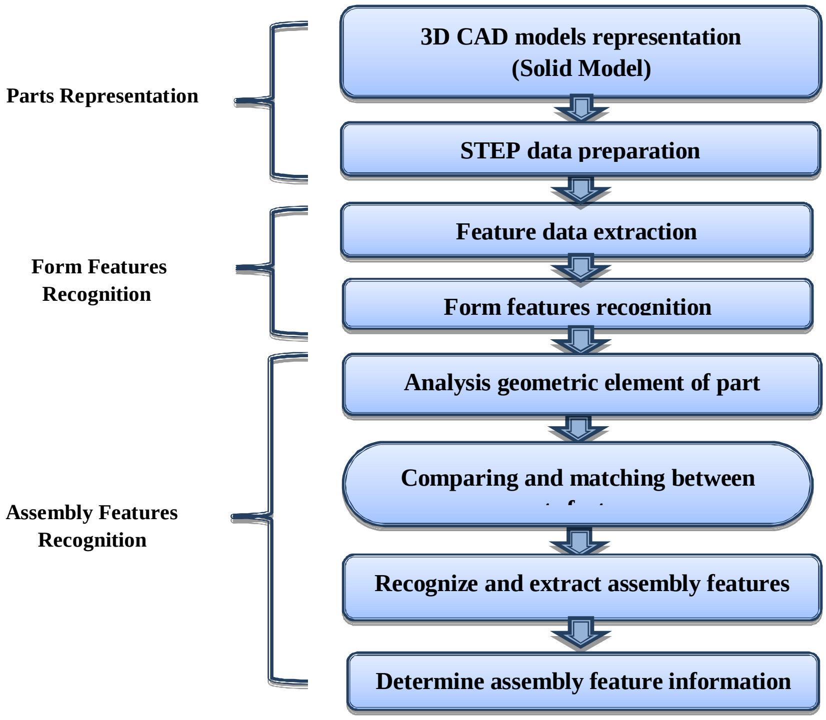


Figure (1) Flowchart of Proposed Methodology Structure.

Parts Representation

In the first stage of the proposed methodology, the parts are designed through CAD software (Solid Works) and they are represented as solid models. Most of the CAD systems utilize some form of B-Rep as their internal representation. The detailed representation specific to each of these systems is different. The standards are developed to transfer geometric data between CAD and CAM systems that have different internal representations[9].

The CAD software generates and provides the geometrical information of the parts design in the form of STEP AP203 file that is then used as a standard format that provides the proposed methodology the ability to communicate with the different CAD/CAM systems. Various CAD systems support STEP AP203 and have their own internal conversion program to convert the CAD data to STEP AP203 format, such as SolidWorks, CATIA, and PRO/ENGINEER. In this paper the SolidWorks Package is used to represent parts.

Form Features Recognition

The feature information extraction from STEP file is to extract geometrical and topological information of the designed parts from data segment. The data structure of STEP file is tree-like which is composed of level of entity, face, edge and point. STEP file consists of two parts: the head file and data section. The head file includes administration and attribute information such as author, date and so on, which begins with the key word "HEADER". The data section is the principal part of STEP file, which begins with “DATA” and ends with “ENDSEC” [10]. The data

section includes a lot of instantiated objects of entities types defined by EXPRESS. The statement in data segment is a series of entities used to describe the information of product. The structure of sentence is shown as follows: #entity identification = entity name (attribute 1, attribute2, …).The entity identification is an integer generated randomly, the entity name is a series of words such as ADVANCED FACE, CARTESIAN POINT and so on, the attribute may be value, string, three coordinate, true-false value and so on. STEP AP203 stores the 3D model data in B-Rep format. The structure of STEPAP203 data is shown in Figure (2).

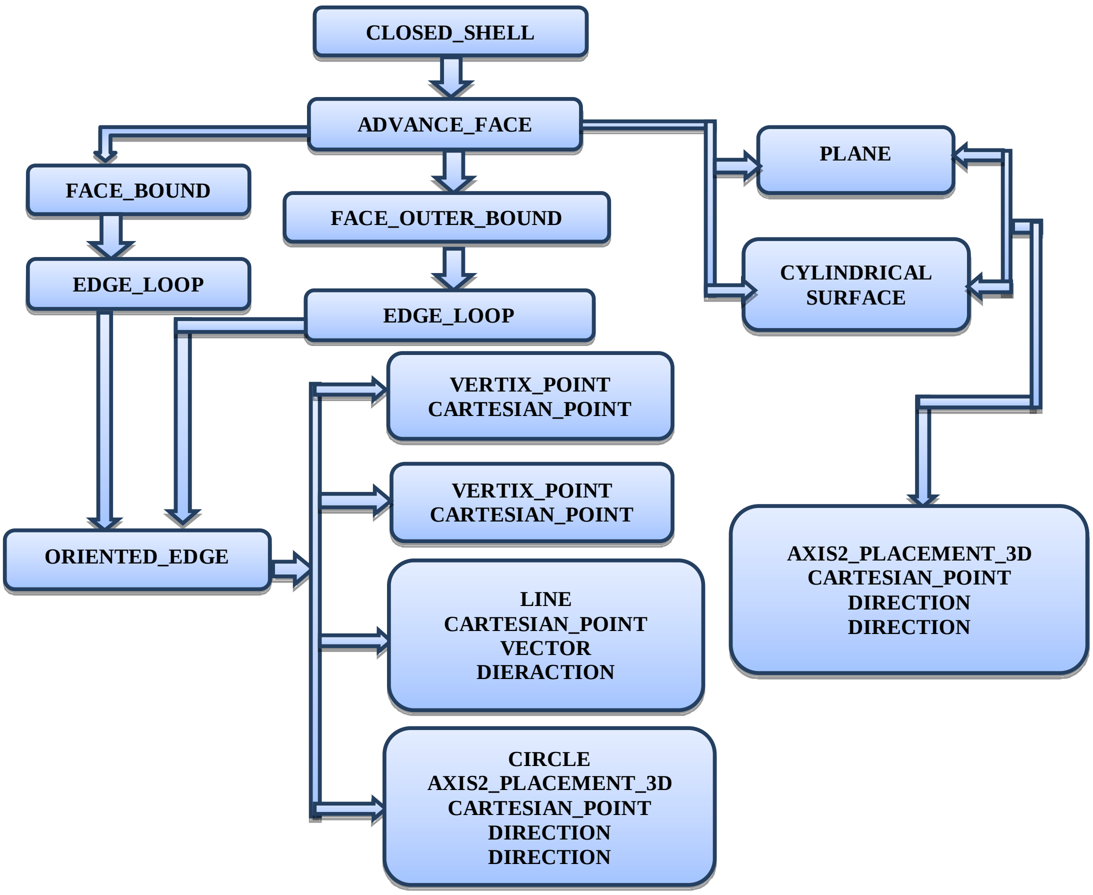
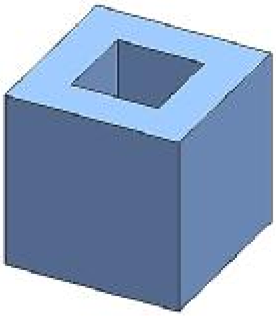


Figure (2) STEP AP 203 Structure.

The STEP file tracing logic is implemented as shown in Figure (3). The tracing of the STEP file is not just read from line 1, then line 2, then line 3. The reality is that, after reading line 1, reader may need to refer to line#266, then to line#342 and so on. According to the hierarchy of the STEP format, the most upper level is the "CLOSED_SHELL" so the keyword can be searched. In Figure (3) some data extracts of STEP AP203 File are represented. As shown in line #266, CLOSED-SHELL specifies all the surfaces which form the component. In this example, the component is a box with blind pocket which has eleven flat surfaces which are

referred to Lines (#342, #367, #364, #356, #339, #349, #373, #375, #338, #369, #343). In this case, all are flat surfaces as indicated in lines (#21, #25, #31, #35, #47, #59, #75, #83, #89, #97,#103). The 3D model STEP files are accessed and the relevant geometric and topological information are obtained, the geometric data of the model is used for subsequent steps of the form feature recognition.



```
. . . . . .
#21 = PLANE ( 'NONE',  #170 )
. . . . . .
#25 = PLANE ( 'NONE',  #138 )
 . . . . . .
#31 = PLANE ( 'NONE',  #136 )
. . . . . .
#35 = PLANE ( 'NONE',  #161 )
. . . . . .
#47 = PLANE ( 'NONE',  #184 )
. . . . . .
#59 = PLANE ( 'NONE',  #168 )
. . . . . .
#75 = PLANE ( 'NONE',  #140 )
. . . . . .
#83 = PLANE ( 'NONE',  #163 )
. . . . . .
#89 = PLANE ( 'NONE',  #149 )
. . . . . .
#97 = PLANE ( 'NONE',  #148 )
. . . . . .
#103 = PLANE ( 'NONE',  #139 )
. . . . . .
#266 = CLOSED_SHELL ( 'NONE', ( #342, #367, #364, #356, #339, #349, #373, #375, #338, #369, #343 ) )
. . . . . . . .
#270 = FACE_BOUND ( 'NONE', #263, .T. )
. . . . . . . .
#272 = FACE_OUTER_BOUND ( 'NONE', #257, .T. )
#273 = FACE_OUTER_BOUND ( 'NONE', #260, .T. )
. . . . . . . .
#275 = FACE_OUTER_BOUND ( 'NONE', #264, .T. )
#276 = FACE_OUTER_BOUND ( 'NONE', #267, .T. )
. . . . . . . .
#282 = FACE_OUTER_BOUND ( 'NONE', #259, .T. )
. . . . . . . .
#287 = FACE_OUTER_BOUND ( 'NONE', #261, .T. )
. . . . . . . .
#294 = FACE_OUTER_BOUND ( 'NONE', #262, .T. )
. . . . . . . .
#297 = FACE_OUTER_BOUND ( 'NONE', #256, .T. )
. . . . . . . .
#299 = FACE_OUTER_BOUND ( 'NONE', #268, .T. )
. . . . . . . .
#302 = FACE_OUTER_BOUND ( 'NONE', #258, .T. )
. . . . . . . .
#304 = FACE_OUTER_BOUND ( 'NONE', #265, .T. )
. . . . . . . .
#338 = ADVANCED_FACE ( 'NONE', ( #272 ), #21, .F. )
#339 = ADVANCED_FACE ( 'NONE', ( #270, #273 ), #25, .F. )
. . . . . . . .
#342 = ADVANCED_FACE ( 'NONE', ( #276 ), #31, .F. )
#343 = ADVANCED_FACE ( 'NONE', ( #275 ), #35, .T. )
. . . . . . . .
#349 = ADVANCED_FACE ( 'NONE', ( #282 ), #47, .T. )
. . . . . . . .
#356 = ADVANCED_FACE ( 'NONE', ( #287 ), #59, .F. )
. . . . . . . .
#364 = ADVANCED_FACE ( 'NONE', ( #294 ), #75, .F. )
. . . . . . . .
#367 = ADVANCED_FACE ( 'NONE', ( #297 ), #83, .F. )
. . . . . . . .
#369 = ADVANCED_FACE ( 'NONE', ( #299 ), #89, .F. )
. . . . . . . .
#373 = ADVANCED_FACE ( 'NONE', ( #302 ), #97, .F. )
. . . . . . . .
#375 = ADVANCED_FACE ( 'NONE', ( #304 ), #103, .F. )
```

Figure (3) Data Extracts of STEP AP203 File for Box with Blind Pocket.

The information extract from STEP AP203 file is converted into AAG to recognize the feature from the solid model of part. An Attribute Adjacency Graph (AAG) is defined as a graph in which nodes correspond to the part faces and arcs correspond to part edges. Each face of the feature is represented by a node, and every pair of nodes represents adjacent faces linked by an arc. The graph can be defined as G = {N, C, A} where N is the set of nodes, C is the set of connection arcs between nodes and A is the set of attributes which denote the kind of connection. The arcs of the AAG graph are labeled by binary digits (0 and 1). The attribute values attached to the arcs are determined by the convexity of the edge. If the faces sharing an edge form a concave angle, the attribute of arc is set to 0, and if two adjacent faces form a convex angle, the attribute of arc is set to 1.

The information extracted from STEP AP203includes the set of face, set of edge, set of vertex, and set of the relationship between geometric and topological entities. A set of face and a set of edge are converted to N and C respectively, the conversion producer is introduced as follows:

1. Create AAG nodes. Check the number of ADVANCED_FACE in the list of extracted data, if the number of ADVANCED_FACE in this list is N, then the number of nodes in AAG is N.
2. Create AAG arcs. Every edge in EDGE_LOOP corresponding to every ADVANCED_ FACE is examined to find whether the EDGE_ LOOP shared the same edge with other EDGE_ LOOP. If these two faces share the same edge, the arc between of these two faces is generated. Every edge in a manifold solid is shared by two faces. The edge will have a particular orientation when being part of one face, and the opposite orientation when being part of the other. This is because the external loops of all the faces are stored in an anti-clockwise sense in STEP. For example, in Figure (4), edge E has an orientation of V1 →V2 when being part of face F1, but the same edge will have an orientation of V2 → V1 when it is part of face F2.
3. The normal vector of these two adjacent faces and the shared edge between faces are recorded to compute the convexity and concavity of the arc. The normal vector of the face (N) is in the form (ai+bj+ck).The Edge Angle (Θ) is measured anticlockwise between N1 and N2 .If the Edge Angle is more than 180^o (Figure (4a), the edge is a Convex Edge. When it is less than 180^o (Figure 4b), the edge is a Concave Edge. Angle (Θ) can be calculated using the equation (1):

$$\Theta = COS^{-1}\left[(N_1 . N_2) / (|N_1| \, |N_2|)\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Where:
N1 : The normal to the first face
N2 : The normal to the second face
|N1| : The magnitude of N1
|N2| : The magnitude of N2

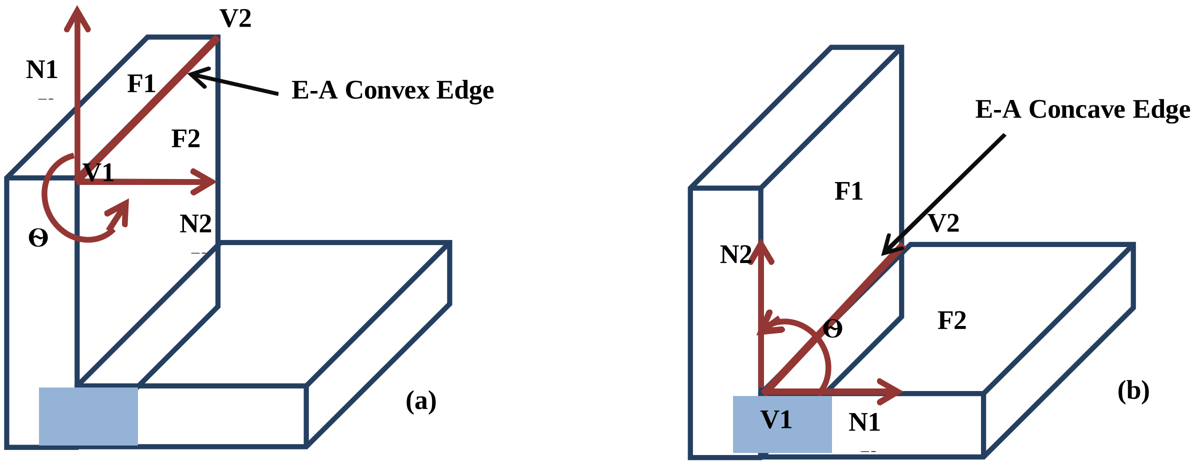


Figure (4) Edge Classification (a) Convex Edge (b) Concave Edge.

In this paper the graph based approach is represented in the form of an Attribute Adjacency Matrix (AAM), the topological information can be represented in the adjacency matrix format according to the graph based approach. The form features taxonomy considered in this work is shown in Figure (5). If the matrix of a particular predefined feature matches that of the predefined template form feature matrix in the library, a form feature is identified and extracted out.

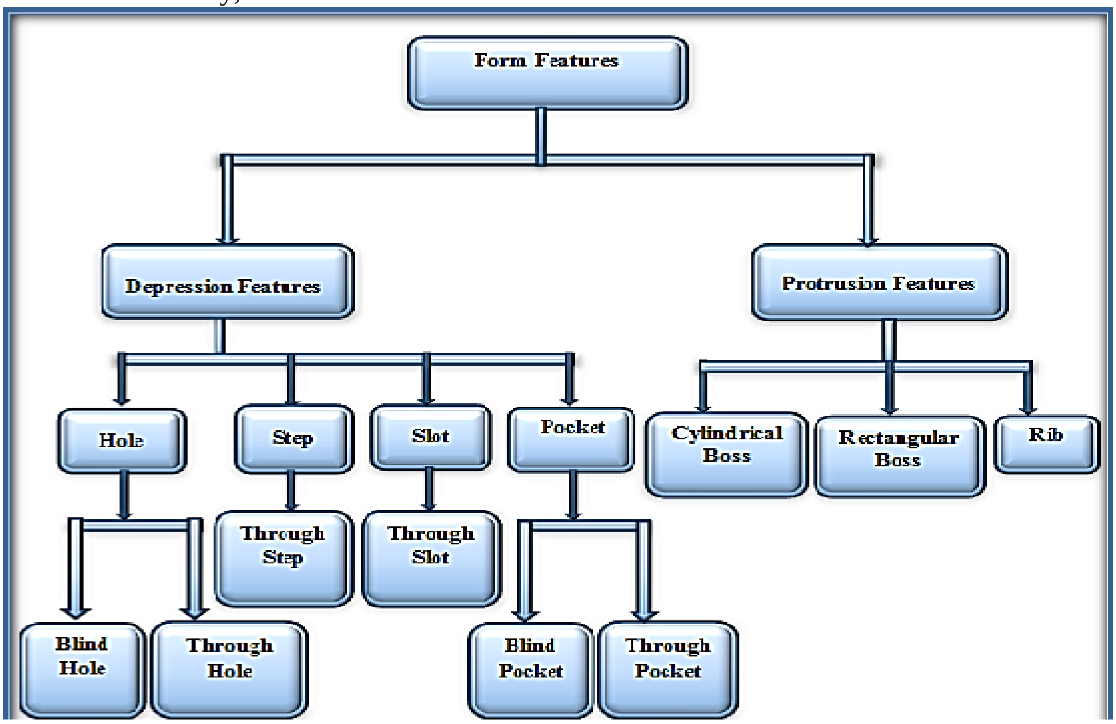


Figure (5) Form Features Taxonomy.

The attribute adjacency matrix is a binary matrix whose elements are 0's and 1's.The row\column number in the matrix corresponds to the faces in the part. The a_{ij} Value shown in the matrix represents:

$$a_{ij} \begin{cases} 0 & \text{i and j shared a concave edge} \\ 1 & \text{i and j shared a convex edge} \end{cases}$$

In the previous works of Joshi and Chang (1988) were the first to propose the Attributed Adjacency Graph (AAG) [11], Subsequently Trika and Kashyap (1994) [12], Yuhaniz and Bin Haron (2002) [13], and Sunil and Pande (2010) [14], used AAG and their work is limited to recognize only depression features. In this paper both depression and protrusion features will be recognized. The following steps will be followed to recognize the form feature:

1.After constructing AAG of the whole part based on the data extracted from STEP file, AAM of the whole part is constructed whose rows and columns are equal to the number of the faces in the part and then the adjacencies faces are found and the attribute are entered in to the cell of the matrix a_{ij}.

2. The feature is represented as sub matrix from the part matrix. To extract feature matrix from the part matrix we classify the faces in the part matrix into two types of faces (feature faces and base faces). Feature faces represent the faces belonging to the feature and they have attribute (0), while the base faces represent the faces adjacent to the feature faces and they have attribute (1).

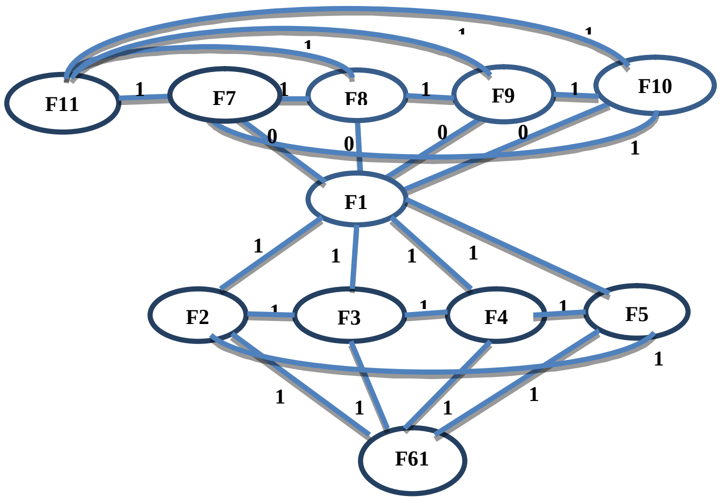
3. Extraction procedure of the feature matrix from the part matrix includes the following steps:

a. Checking the entire columns in the part matrix and choosing each column that contains attribute (0). Only the columns that have less than or equal to four relations will be chosen, these columns will represent the feature faces.

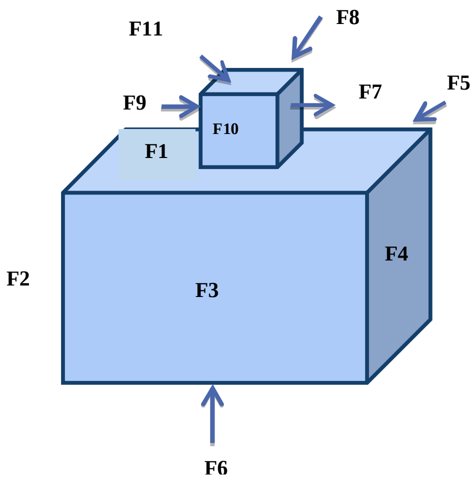
b. For each column we check the cells of the column to determine their relations with the rows, the row with attribute (1) represents the base face, we choose only the face that has less than or equal to four relations with other faces.

c. The chosen faces from (a &b) will be arranged in the new matrix and the Adjacency Counts (C_A) for each row in the feature matrix is determined. The Adjacency Count (C_A) included the calculation of the number of (0) and (1) attributes in each row where, A = (0), B = (1).

4. After calculation (A) and (B), (C_A) is determined [C_A = (A) + (B)]. Adjacency Count value and feature matrix will be used to matching the recognized feature with feature pattern to find out the type of feature. Figure (6) illustrates example of 3D solid model of box with rectangular boss feature and the AAG, AAM of box with rectangular boss and feature matrix for rectangular boss.



(b)AAG of Box with Rectangular Boss



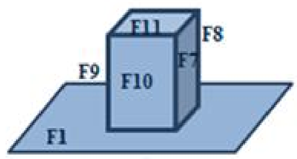
(a) Box with Rectangular Boss

I / j	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
F1		1	1	1	1		0	0	0	0	
F2	1		1		1	1					
F3	1	1		1		1					
F4	1		1		1	1					
F5	1	1		1		1					
F6		1	1	1	1						
F7	0							1		1	1
F8	0						1		1		1
F9	0							1		1	1
F10	0						1		1		1
F11							1	1	1	1	

(c) AAM of Box with Rectangular Boss

i / j	F7	F8	F9	F10	F11
F7		1		1	1
F8	1		1		1
F9		1		1	1
F10	1		1		1
F11	1	1	1	1	

(e) Rectangular Boss Feature Matrix



(d) Rectangular Boss Feature Model

A=0 , B=1

$$C_A = (A) + (B)$$

$$C_A = (0) + (16) = 16$$

The rule for recognition rectangular boss is:

j \ i	F7	F8	F9	F10	F11	C_A A	B
F7		1		1	1		3
F8	1		1		1		3
F9		1		1	1		3
F10	1		1		1		3
F11	1	1	1	1			4

(f) Rectangular Boss Feature Matrix with Adjacency Count (C_A)

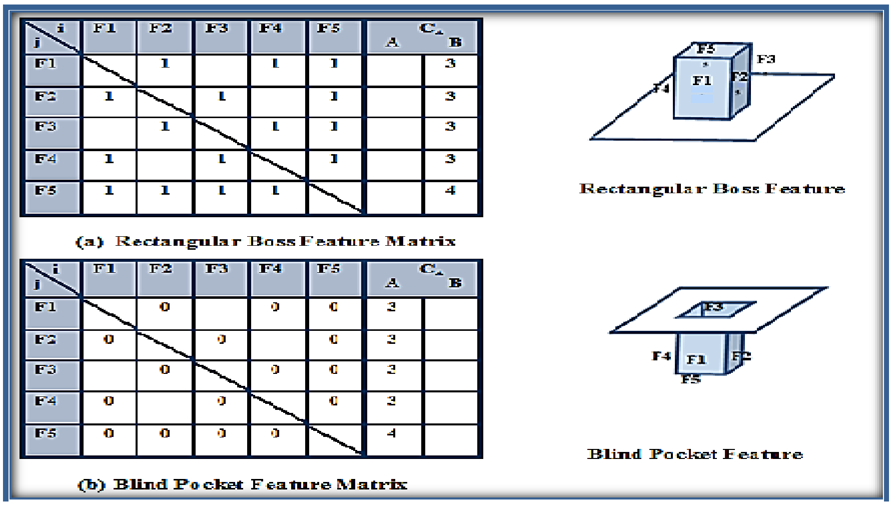
Figure (6): (a) Box with Rectangular Boss, (b) AAG of the Box with Rectangular Boss, (c) AAM of Box with Rectangular Boss (d) Rectangular Boss Feature Model, (e) Rectangular Boss Feature Matrix. (f) Rectangular Boss Feature Matrix with Adjacency Count

Assembly Features Recognition

Assembly feature is defined as "matching pair of form features associated with assembly mating relations" Assembly mating relations include (e.g. against, fit, insert, align). The assembly feature can be represented mathematically as:

$$AF= (F_i \cup F_j) \cup R_{ij} \ldots\ldots\ldots\ldots\ldots \ (2)$$

Where: (AF) shows an assembly feature, (F_i, F_j) represents the matching form features, (R_{ij}) shows assembly relation between the matching features.The matching between form features can be determined by examination of the form feature type and form feature matrix, if the first form feature is depression feature and the second form feature is protrusion feature and attributes of the first feature matrix are opposite to the attributes of the second feature matrix, then these features can be matched. Figure (7) illustrates the matching between two form features (rectangular boss and blind pocket).



j \ i	F1	F2	F3	F4	F5	C_A A	B
F1		1		1	1		3
F2	1		1		1		3
F3		1		1	1		3
F4	1		1		1		3
F5	1	1	1	1			4

(a) Rectangular Boss Feature Matrix

j \ i	F1	F2	F3	F4	F5	C_A A	B
F1		0		0	0	2	
F2	0		0		0	2	
F3		0		0	0	2	
F4	0		0		0	2	
F5	0	0	0	0		4	

(b) Blind Pocket Feature Matrix

Figure (7) Matching Form Feature Matrices.

The mating constraints considered in this work are fits, against and insert.Given two components having two features F1and F2, f_i represents the i^{th} face in F1 and f_j represents the j^{th} face in F2. The general rules used to automatically find mating surfaces among components shown as follows:

Rule 1

IF the faces type of (f_i) and (f_j)is planer AND the outward normals of two faces opposite each other (N_i= -N_j) AND (P_i, P_j) lies on the same plane THEN the mating type is Against. Where N is the normal vector of plane and P is a point on the plane.

Rule 2

IF the faces type of (f_i) and (f_j) is cylindrical AND satisfies two conditions that the axial center lines are collinear AND radiuses are equal (r_i=r_j) THEN the mating type is Fit.This rule can be applied to planar faces like the mating between boss and pocket.

Rule 3

IF the faces type of (f_i) and (f_j) is planer AND any two adjacent vertical planar faces of F1 are coincident with the corresponding vertical planar faces of F2 AND both features have the same volume space THEN the mating type is Insert.
Figure (8) illustrates proposed algorithm for recognition and extraction form and assembly features.

ST-ASP SYSTEM ARCHITECTURE

The proposed ST-ASP system has the ability to extract the information from the product description; this information includes geometry, geometric relationships, and dimensions. The system has the ability to recognize features and extract the parameters of the features. ST-ASP system is developed using Visual Basic 6.0 on HP Pavilion DV6 PC; this system is supported bySolid Works 2011 Package. ST-ASP system consists of two main modules: Design module and assembly feature module. Through the design module the system links automatically with Solid Works package to drawing the mechanical parts and then save the designed parts as STEP AP203 file which represents the input to the assembly feature module. The assembly feature module consists of three sub modules; features data extraction which extracts the necessary information for features recognition from STEP file, form features recognition based on Attribute Adjacency Matrix (AAM) approach, and assembly features recognition which is based on matching form features and mating relations. Figure (9) shows the architecture of ST-ASP system.

THE ST-ASP SYSTEM TESTING

A case study is carried out and detailed to demonstrate the feasibility of the proposed methodology and the capability of the ST-ASP system.Figure (10) shows case study designed by SolidWorks package; Figures (11), (12), (13) show the outputs result from assembly feature module window.

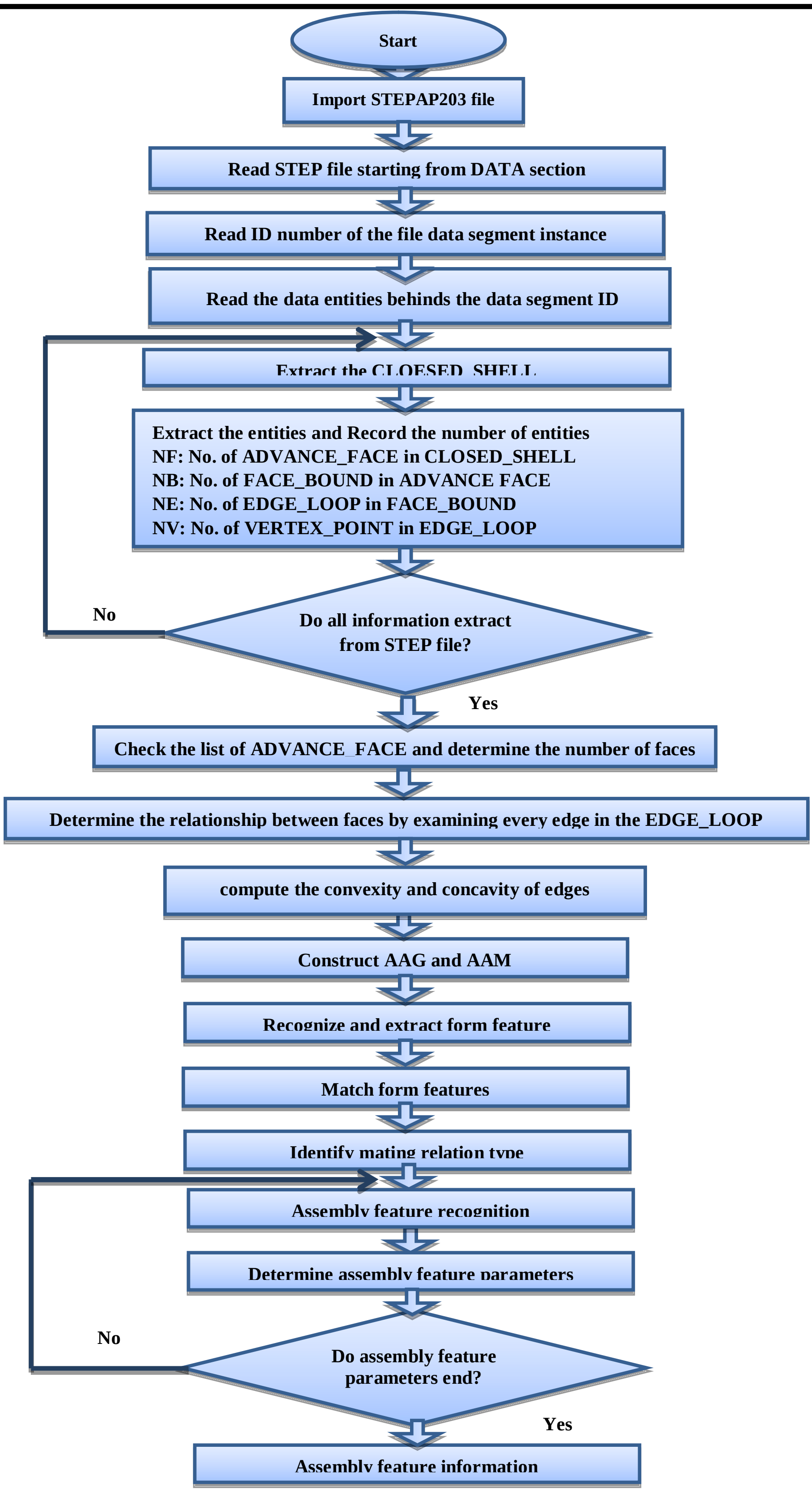


Figure (8) Flowchart of Proposed Algorithm for Recognition and Extraction Form and Assembly Feature.

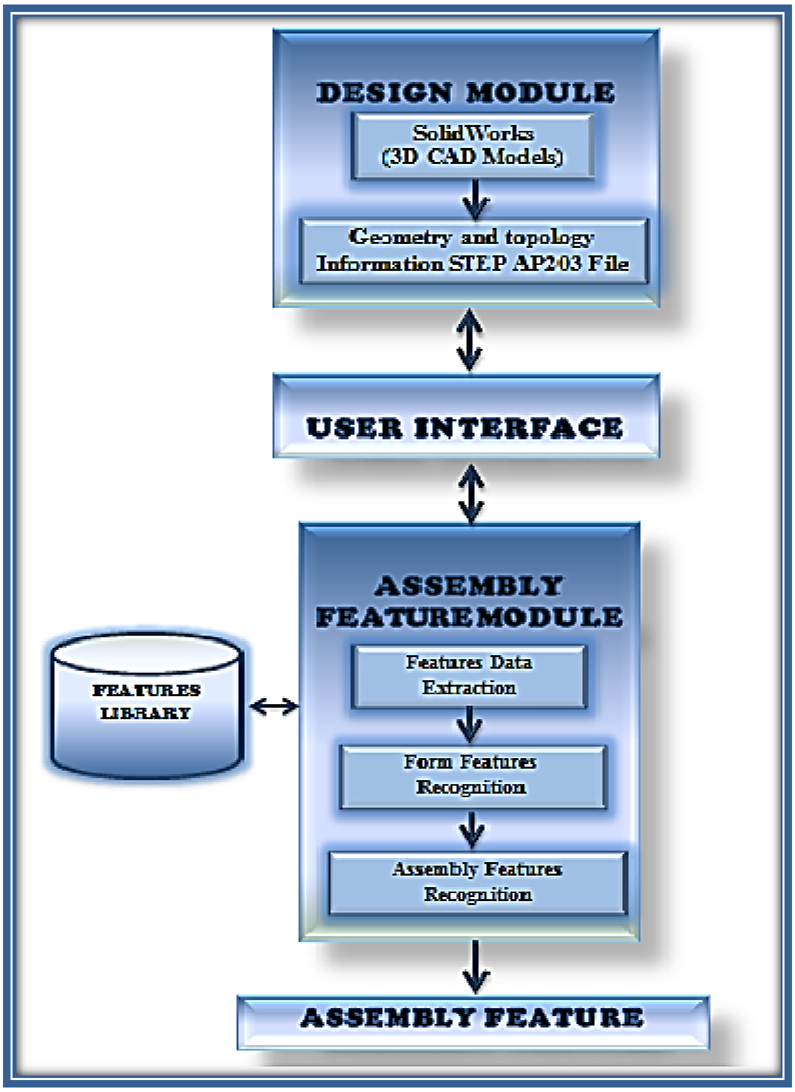


Figure (9) Architecture of Proposed (ST-ASP) System.

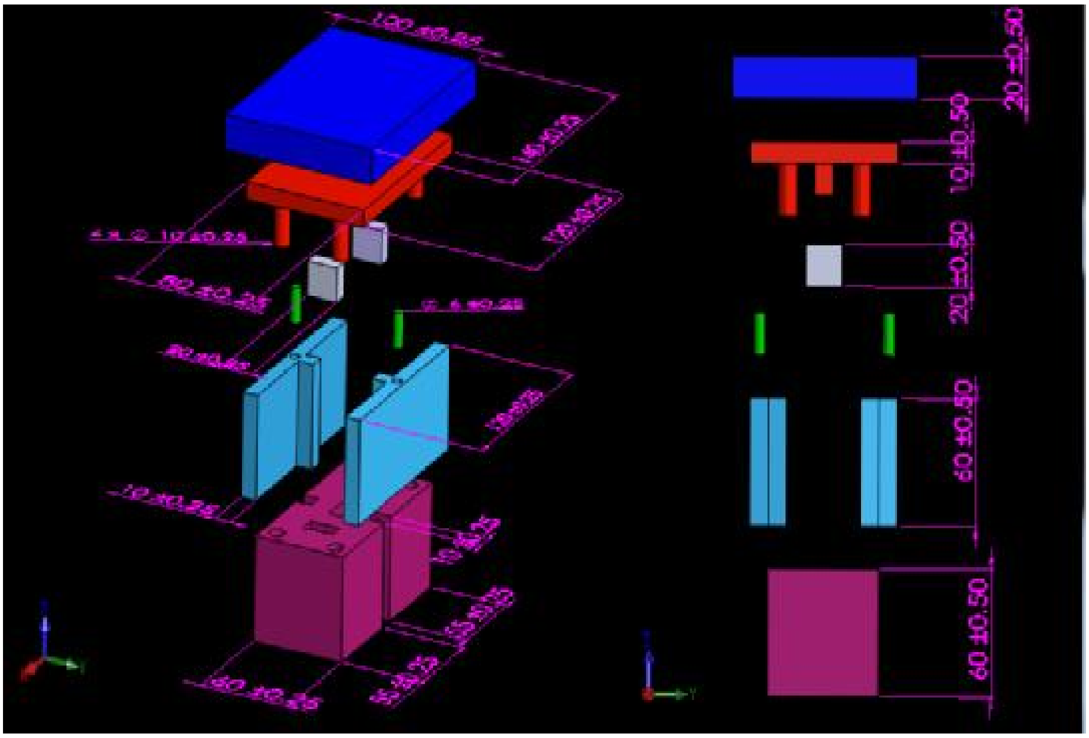
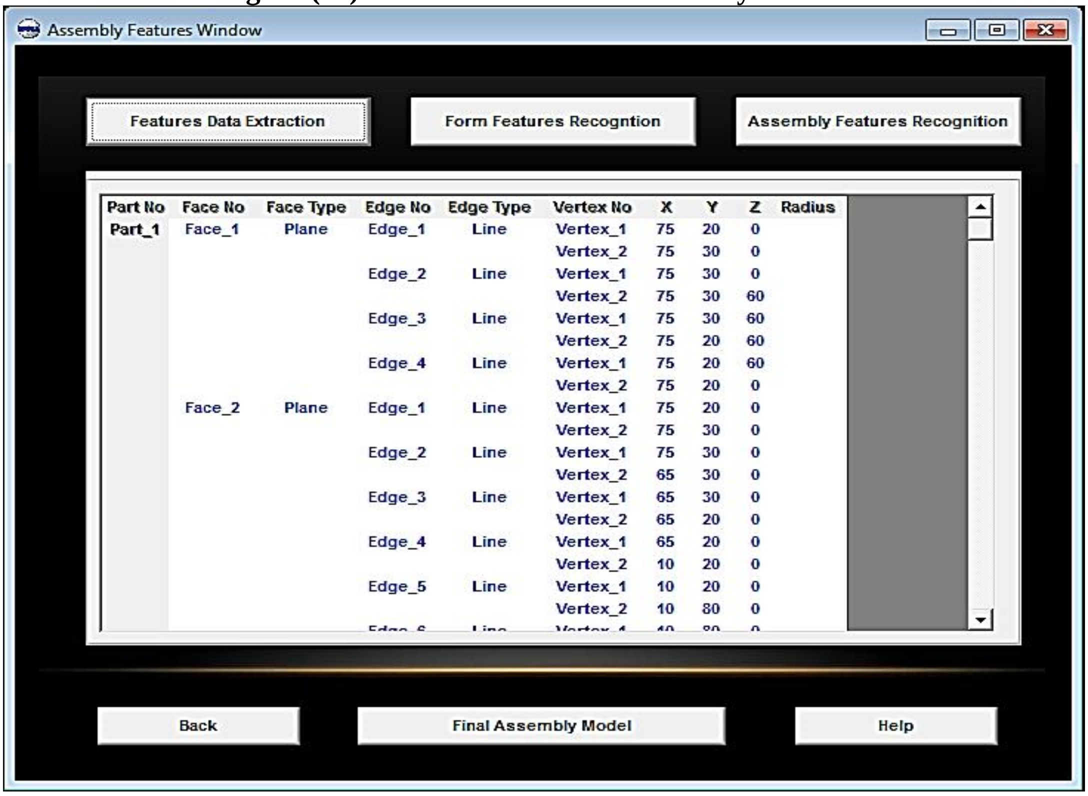
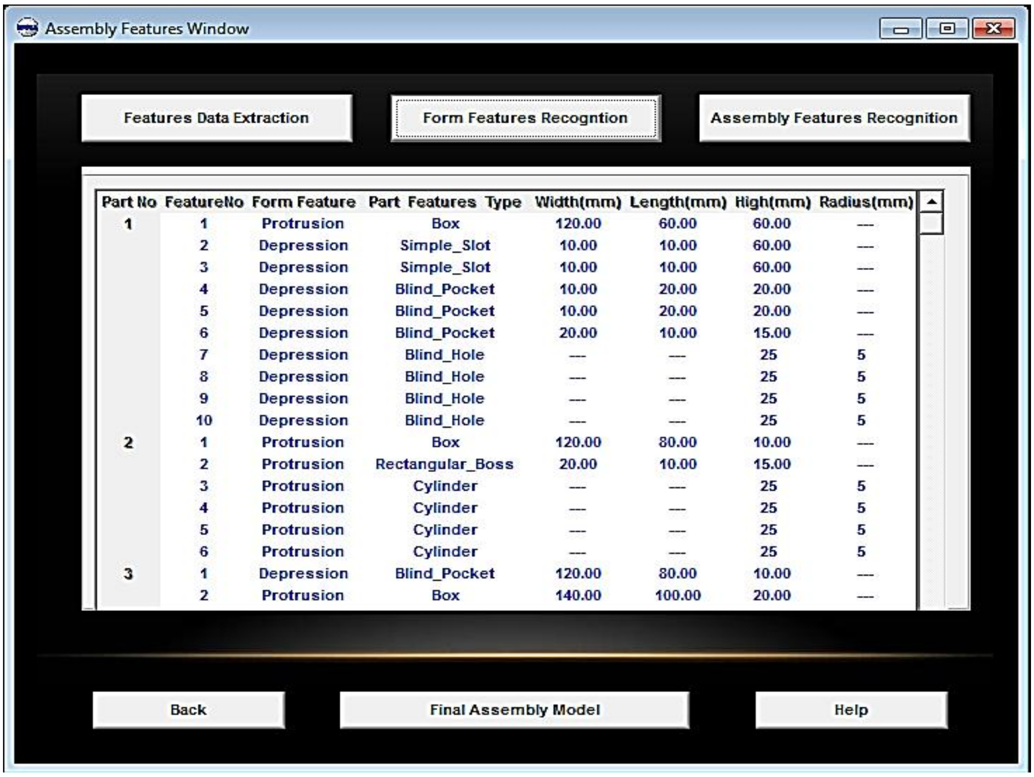


Figure (10) 3D CAD Models of Case Study.



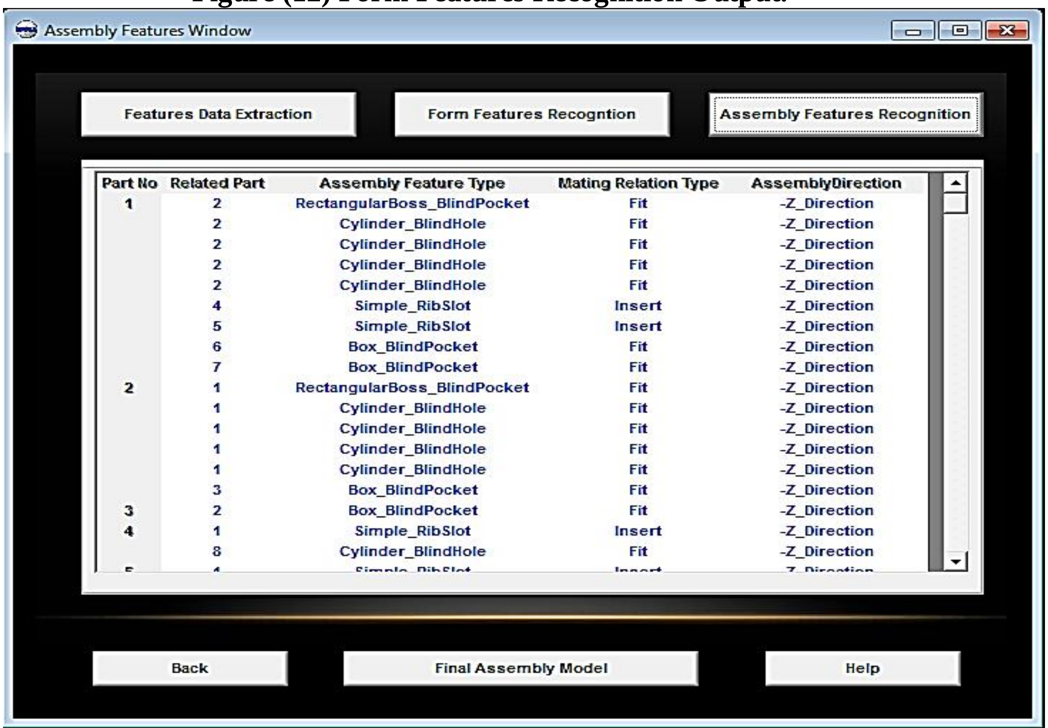
Part No	Face No	Face Type	Edge No	Edge Type	Vertex No	X	Y	Z	Radius
Part_1	Face_1	Plane	Edge_1	Line	Vertex_1	75	20	0	
					Vertex_2	75	30	0	
			Edge_2	Line	Vertex_1	75	30	0	
					Vertex_2	75	30	60	
			Edge_3	Line	Vertex_1	75	30	60	
					Vertex_2	75	20	60	
			Edge_4	Line	Vertex_1	75	20	60	
					Vertex_2	75	20	0	
	Face_2	Plane	Edge_1	Line	Vertex_1	75	20	0	
					Vertex_2	75	30	0	
			Edge_2	Line	Vertex_1	75	30	0	
					Vertex_2	65	30	0	
			Edge_3	Line	Vertex_1	65	30	0	
					Vertex_2	65	20	0	
			Edge_4	Line	Vertex_1	65	20	0	
					Vertex_2	10	20	0	
			Edge_5	Line	Vertex_1	10	20	0	
					Vertex_2	10	80	0	

Figure (11) Features Data Extraction Output.



Part No	FeatureNo	Form Feature	Part Features Type	Width(mm)	Length(mm)	High(mm)	Radius(mm)
1	1	Protrusion	Box	120.00	60.00	60.00	---
	2	Depression	Simple_Slot	10.00	10.00	60.00	---
	3	Depression	Simple_Slot	10.00	10.00	60.00	---
	4	Depression	Blind_Pocket	10.00	20.00	20.00	---
	5	Depression	Blind_Pocket	10.00	20.00	20.00	---
	6	Depression	Blind_Pocket	20.00	10.00	15.00	---
	7	Depression	Blind_Hole	---	---	25	5
	8	Depression	Blind_Hole	---	---	25	5
	9	Depression	Blind_Hole	---	---	25	5
	10	Depression	Blind_Hole	---	---	25	5
2	1	Protrusion	Box	120.00	80.00	10.00	---
	2	Protrusion	Rectangular_Boss	20.00	10.00	15.00	---
	3	Protrusion	Cylinder	---	---	25	5
	4	Protrusion	Cylinder	---	---	25	5
	5	Protrusion	Cylinder	---	---	25	5
	6	Protrusion	Cylinder	---	---	25	5
3	1	Depression	Blind_Pocket	120.00	80.00	10.00	---
	2	Protrusion	Box	140.00	100.00	20.00	---

Figure (12) Form Features Recognition Output.



Part No	Related Part	Assembly Feature Type	Mating Relation Type	AssemblyDirection
1	2	RectangularBoss_BlindPocket	Fit	-Z_Direction
	2	Cylinder_BlindHole	Fit	-Z_Direction
	2	Cylinder_BlindHole	Fit	-Z_Direction
	2	Cylinder_BlindHole	Fit	-Z_Direction
	2	Cylinder_BlindHole	Fit	-Z_Direction
	4	Simple_RibSlot	Insert	-Z_Direction
	5	Simple_RibSlot	Insert	-Z_Direction
	6	Box_BlindPocket	Fit	-Z_Direction
	7	Box_BlindPocket	Fit	-Z_Direction
2	1	RectangularBoss_BlindPocket	Fit	-Z_Direction
	1	Cylinder_BlindHole	Fit	-Z_Direction
	1	Cylinder_BlindHole	Fit	-Z_Direction
	1	Cylinder_BlindHole	Fit	-Z_Direction
	1	Cylinder_BlindHole	Fit	-Z_Direction
	3	Box_BlindPocket	Fit	-Z_Direction
3	2	Box_BlindPocket	Fit	-Z_Direction
4	1	Simple_RibSlot	Insert	-Z_Direction
	8	Cylinder_BlindHole	Fit	-Z_Direction
5	4	Simple_RibSlot	Insert	-Z_Direction

Figure (13) Assembly Features Recognition Output.

CONCLUSIONS

The main contributions of this paper include building a simple methodology for recognition and extraction form and assembly features from STEP file. STEP standard simplifies the extraction of faces, edges, and vertices because the structure of STEP file is similar to EXPRESS language, this low level information is useful to determine part form features. The recognition algorithms works on received STEP AP203 file format and generates an output feature file that can be interfaced with downstream planning activities such as assembly planning. The proposed form features algorithm has the capability to recognize both depression and protrusion features comparing with old form features algorithms that based on AAG. Graph and matrix of a feature are simplified by reducing the number of faces in a part volume so that feature representation complexity is greatly reduced. Also, for each feature type there exists standard AAM stored in the database of computer used as a signature of feature for recognizing that feature type. The developed ST-ASP system was success to recognize (blind and through holes, blind and through pocket, cylindrical and rectangular bosses, through step and slot, rib, and cylindrical and rectangular pins) features for prismatic parts with high efficiency.

REFERENCES

[1].Abu, R., and TAP, M.D., "Attribute Based Feature Recognition for Machining Features" JurnalTeknologi, 46(A), 87–103, UniversitiTeknologi Malaysia, 2007.

[2]. Holland, W.V., and Bronsvoort W. F., "Assembly Features In Modeling and Planning" Robotic and Computer Integrated Manufacturing, 16, 277-294, Elsevier Science, 2000.

[3].Hamidullah, Bohez E., and Irfan M. A., "Assembly Features: Definition, Classification, and Instantiation" 2nd International Conference on Emerging Technologies, Peshawar, Pakistan, IEEE-ICET, 2006.

[4].Zha, X.F., and Du H., "A PDES\STEP Based Model and System for Concurrent Integrated Design and Assembly Planning" Computer Aided Design, 34, 1087-1110, Elsevier Science Ltd., 2002.

[5].Chan, C.K., and Tan, S.T., "Generating Assembly Features onto Split Solid Models" Computer-Aided Design, 35, 1315–1336, ElsevierLtd., 2003.

[6].Sung, R. C. W., Corney, J. R., and Clark D. E. R., "Octree Based Recognition of Assembly Features" Design Engineering TechnicalConference and Computer Information in Engineering Conference,ASME, 2000.

[7].Yifei, T., Dongbo L., Changbo L., and Minjian Y., "A Feature Extraction Based Process Planning System" International Journal of Advance Manufacturing Technology, 38:1192–1200, Springer-Verlag London Limited, 2008.

[8].Amaitik, S. M., "Development of A STEP Feature-Based Intelligent Process Planning System for Prismatic Parts" PhD. Thesis, Department of Mechanical Engineering, Middle East Technical University, 2005.

[9].Abouel, Nasr E., and Kamrani A. K., "Computer-Based Design and Manufacturing: An Information-Based Approach" Springer Science+ Business Media, LLC., 2007.

[10]. Pan, C., "Integrating CAD Files and Automatic Assembly Sequence Planning" Ph.D. thesis, Iowa State University, 2005.

[11].Joshi S., and Chang T., "Graph Based Heuristics for Recognition of Machined Features from a 3D Solid Model" Computer-Aided Design, 20 (2):58-66, Elsevier Science Ltd, 1988.
[12].Trika S.N., and Kashyap R.L., " Geometric Reasoning for Extraction of Manufacturing Features in Iso-oriented Polyhedrons" IEEE Transactions on Pattern Analysis and Machine Intelligence, 16 (11) 1087–1100, 1994.
[13].Yuhaniz S. S., and Bin Haron H., "Improving Attributed Adjacency Graph Data representation" Proceedings of National Conference on Computer Graphics and Multimedia, university Technology Malaysia, 2002.
[14].Sunil V. B., Agarwal R., and Pande S.S., "An Approach to Recognize Interacting Features from B-Rep CAD Models of Prismatic Machined Parts Using a Hybrid Technique" Computer in Industry, 61, 686-701, Elsevier B.V., 2010.

(Appendix)
STEP AP203 File for (3D CAD Models of case Study Designed by Solid Works)


```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (( 'STEP AP203' ),
'1' );
FILE_NAME ('Case Study.STEP',
  '2012-07-06T07:56:57',
  ( '' ),
  ( '' ),
  'SwSTEP 2.0',
  'SolidWorks 2011',
  '' );
FILE_SCHEMA (( 'CONFIG_CONTROL_DESIGN' ));
ENDSEC;
DATA;
………………..
………………..
#29=APPLICATION_CONTEXT('CONFIGURATION CONTROLLED 3D DESIGNS OF MECHANICAL PARTS AND ASSEMBLIES');
#30=APPLICATION_PROTOCOL_DEFINITION('INTERNATIONAL STANDARD','config_control_design',1995,#29);
#31=MECHANICAL_CONTEXT('3D Mechanical Parts',#29,'mechanical');
#32=PRODUCT('CASE STUDY','CASE STUDY',' ',(#31));
#33=PRODUCT_RELATED_PRODUCT_CATEGORY('detail',$,(#32));
#34=PRODUCT_CATEGORY('part',$);
#35=PRODUCT_CATEGORY_RELATIONSHIP('None','None',#34,#33);
………………..
………………..
#160=AXIS2_PLACEMENT_3D('',#157,#158,#159);
#161=CIRCLE('',#160,3.0);
#162=EDGE_CURVE('',#133,#150,#161,.T.);
#163=ORIENTED_EDGE('',*,*,#162,.F.);
#164=EDGE_LOOP('',(#139,#148,#156,#163));
#165=FACE_OUTER_BOUND('',#164,.T.);
#166=ADVANCED_FACE('',(#165),#129,.T.);
#167=CARTESIAN_POINT('',(60.0,65.0,170.0));
#168=DIRECTION('',(0.0,0.0,1.0));
#169=DIRECTION('',(1.0,0.0,0.0));
#170=AXIS2_PLACEMENT_3D('',#167,#168,#169);
#171=CYLINDRICAL_SURFACE('',#170,3.0);
#172=ORIENTED_EDGE('',*,*,#138,.T.);
#173=CARTESIAN_POINT('',(60.0,65.0,160.0));
```

**#174=DIRECTION('',(0.0,0.0,-1.0));
#175=DIRECTION('',(1.0,0.0,0.0));
#176=AXIS2_PLACEMENT_3D('',#173,#174,#175);
#177=CIRCLE('',#176,3.0);
#178=EDGE_CURVE('',#150,#133,#177,.T.);
#179=ORIENTED_EDGE('',*,*,#178,.F.);
#180=ORIENTED_EDGE('',*,*,#155,.F.);
#181=CARTESIAN_POINT('',(60.0,65.0,180.0));
#182=DIRECTION('',(0.0,0.0,1.0));
#183=DIRECTION('',(1.0,0.0,0.0));
#184=AXIS2_PLACEMENT_3D('',#181,#182,#183);
#185=CIRCLE('',#184,3.0);
#186=EDGE_CURVE('',#131,#141,#185,.T.);
#187=ORIENTED_EDGE('',*,*,#186,.F.);
#188=EDGE_LOOP('',(#172,#179,#180,#187));
#189=FACE_OUTER_BOUND('',#188,.T.);
#190=ADVANCED_FACE('',(#189),#171,.T.);
#191=CARTESIAN_POINT('',(60.0,65.0,160.0));
#192=DIRECTION('',(0.0,0.0,-1.0));
#193=DIRECTION('',(-1.0,0.0,0.0));
#194=AXIS2_PLACEMENT_3D('',#191,#192,#193);
#195=PLANE('',#194);
#196=ORIENTED_EDGE('',*,*,#162,.T.);
#197=ORIENTED_EDGE('',*,*,#178,.T.);
#198=EDGE_LOOP('',(#196,#197));
#199=FACE_OUTER_BOUND('',#198,.T.);
#200=ADVANCED_FACE('',(#199),#195,.T.);
#201=CARTESIAN_POINT('',(60.0,65.0,180.0));
#202=DIRECTION('',(0.0,0.0,1.0));
#203=DIRECTION('',(1.0,0.0,0.0));
#204=AXIS2_PLACEMENT_3D('',#201,#202,#203);
#205=PLANE('',#204);
#206=ORIENTED_EDGE('',*,*,#186,.T.);
#207=ORIENTED_EDGE('',*,*,#147,.T.);
#208=EDGE_LOOP('',(#206,#207));
#209=FACE_OUTER_BOUND('',#208,.T.);
#210=ADVANCED_FACE('',(#209),#205,.T.);
#211=CLOSED_SHELL('',(#166,#190,#200,#210));
#212=MANIFOLD_SOLID_BREP('4BC',#211);
………………
#566=ORIENTED_EDGE('',*,*,#538,.F.);
#567=ORIENTED_EDGE('',*,*,#466,.T.);
#568=ORIENTED_EDGE('',*,*,#553,.T.);
#569=ORIENTED_EDGE('',*,*,#505,.T.);
#570=EDGE_LOOP('',(#566,#567,#568,#569));
#571=FACE_OUTER_BOUND('',#570,.T.);
#572=ADVANCED_FACE('',(#571),#565,.F.);
#573=CLOSED_SHELL('',(#254,#278,#288,#328,#368,#390,#421,#452,#491,#526,#543,#560,#572));
#574=MANIFOLD_SOLID_BREP('4AE',#573);
#575=CARTESIAN_POINT('',(60.0,-5.0,170.0));
#576=DIRECTION('',(0.0,0.0,1.0));
#577=DIRECTION('',(1.0,0.0,0.0));
#578=AXIS2_PLACEMENT_3D('',#575,#576,#577);
#579=CYLINDRICAL_SURFACE('',#578,3.0);
#580=CARTESIAN_POINT('',(63.0,-5.0,180.0));
#581=VERTEX_POINT('',#580);
#582=CARTESIAN_POINT('',(63.0,-5.0,160.0));
#583=VERTEX_POINT('',#582);
#584=CARTESIAN_POINT('',(63.0,-5.0,180.0));
#585=DIRECTION('',(0.0,0.0,-1.0));
#586=VECTOR('',#585,20.0);
#587=LINE('',#584,#586);
#588=EDGE_CURVE('',#581,#583,#587,.T.);
#589=ORIENTED_EDGE('',*,*,#588,.F.);
………………..
………………..**

```
#610=AXIS2_PLACEMENT_3D('',#607,#608,#609);
#611=CIRCLE('',#610,3.0);
#612=EDGE_CURVE('',#583,#600,#611,.T.);
#613=ORIENTED_EDGE('',*,*,#612,.F.);
#614=EDGE_LOOP('',(#589,#598,#606,#613));
#615=FACE_OUTER_BOUND('',#614,.T.);
#616=ADVANCED_FACE('',(#615),#579,.T.);
#617=CARTESIAN_POINT('',(60.0,-5.0,170.0));
#618=DIRECTION('',(0.0,0.0,1.0));
#619=DIRECTION('',(1.0,0.0,0.0));
#620=AXIS2_PLACEMENT_3D('',#617,#618,#619);
#621=CYLINDRICAL_SURFACE('',#620,3.0);
#622=ORIENTED_EDGE('',*,*,#588,.T.);
#623=CARTESIAN_POINT('',(60.0,-5.0,160.0));
#624=DIRECTION('',(0.0,0.0,-1.0));
#625=DIRECTION('',(1.0,0.0,0.0));
………………..
………………..
#661=CLOSED_SHELL('',(#616,#640,#650,#660));
#662=MANIFOLD_SOLID_BREP('4AA',#661);
#663=CARTESIAN_POINT('',(60.0,-5.0,130.0));
#664=DIRECTION('',(0.0,-1.224647E-016,-1.0));
#665=DIRECTION('',(1.0,0.0,0.0));
#666=AXIS2_PLACEMENT_3D('',#663,#664,#665);
#667=CYLINDRICAL_SURFACE('',#666,3.0);
#668=CARTESIAN_POINT('',(63.0,-4.999999999999999,140.0));
#669=VERTEX_POINT('',#668);
#670=CARTESIAN_POINT('',(63.0,-5.000000000000001,120.0));
#671=VERTEX_POINT('',#670);
………………..
………………..
#1016=ORIENTED_EDGE('',*,*,#861,.F.);
#1017=ORIENTED_EDGE('',*,*,#929,.T.);
#1018=ORIENTED_EDGE('',*,*,#1003,.T.);
#1019=ORIENTED_EDGE('',*,*,#954,.T.);
#1020=EDGE_LOOP('',(#1016,#1017,#1018,#1019));
#1021=FACE_OUTER_BOUND('',#1020,.T.);
#1022=ADVANCED_FACE('',(#1021),#1015,.F.);
#1023=CLOSED_SHELL('',(#704,#728,#738,#778,#818,#840,#871,#902,#941,#976,#993,#1010,#1022));
#1024=MANIFOLD_SOLID_BREP('498',#1023);
#1025=CARTESIAN_POINT('',(120.000000000000030,-10.000000000000057,290.0));
#1026=CARTESIAN_POINT('',(120.000000000000040,70.0,290.0));
#1027=CARTESIAN_POINT('',(0.0,-10.000000000000030,290.0));
#1028=CARTESIAN_POINT('',(1.421085E-014,70.000000000000028,290.0));
………………..
………………..
#1315=CLOSED_SHELL('',(#1064,#1095,#1119,#1143,#1160,#1200,#1246,#1268,#1285,#1302,#1314));
#1316=MANIFOLD_SOLID_BREP('450',#1315);
………………..
………………..
#1927=CLOSED_SHELL('',(#1358,#1400,#1442,#1484,#1508,#1518,#1542,#1552,#1592,#1623,#1647,#1671,#
1688,#1712,#1722,#1746,#1756,#1796,#1858,#1880,#1897,#1914,#1926));
#1928=MANIFOLD_SOLID_BREP('422',#1927);
#1929=CARTESIAN_POINT('',(95.0,40.0,192.000000000000030));
#1930=CARTESIAN_POINT('',(95.0,40.0,212.000000000000030));
#1931=CARTESIAN_POINT('',(95.0,20.0,192.000000000000030));
#1932=CARTESIAN_POINT('',(95.0,20.0,212.000000000000030));
………………..
………………..
#2077=CLOSED_SHELL('',(#1968,#1999,#2030,#2052,#2064,#2076));
#2078=MANIFOLD_SOLID_BREP('1D4',#2077);
#2079=CARTESIAN_POINT('',(24.999999999999996,20.0,192.000000000000030));
#2080=DIRECTION('',(-1.0,0.0,0.0));
#2081=DIRECTION('',(0.0,0.0,1.0));
#2082=AXIS2_PLACEMENT_3D('',#2079,#2080,#2081);
#2083=PLANE('',#2082);
```

**#2084=CARTESIAN_POINT('',(24.999999999999996,20.0,212.000000000000030));
#2085=VERTEX_POINT('',#2084);
#2086=CARTESIAN_POINT('',(25.0,40.0,212.000000000000030));
#2087=VERTEX_POINT('',#2086);
#2088=CARTESIAN_POINT('',(24.999999999999996,20.0,212.000000000000030));
#2089=DIRECTION('',(0.0,1.0,0.0));
#2090=VECTOR('',#2089,20.0);
#2091=LINE('',#2088,#2090);
#2092=EDGE_CURVE('',#2085,#2087,#2091,.T.);
#2093=ORIENTED_EDGE('',*,*,#2092,.T.);
#2094=CARTESIAN_POINT('',(25.0,40.0,192.000000000000030));
………………..
………………..
#2227=CLOSED_SHELL('',(#2118,#2149,#2180,#2202,#2214,#2226));
#2228=MANIFOLD_SOLID_BREP('1D3',#2227);
#2229=CARTESIAN_POINT('',(110.0,10.0,60.0));
#2230=DIRECTION('',(0.0,0.0,-1.0));
#2231=DIRECTION('',(1.0,0.0,0.0));
#2232=AXIS2_PLACEMENT_3D('',#2229,#2230,#2231);
#2233=CYLINDRICAL_SURFACE('',#2232,5.0);
#2234=CARTESIAN_POINT('',(115.0,10.0,60.0));
#2235=VERTEX_POINT('',#2234);
#2236=CARTESIAN_POINT('',(115.0,10.0,35.0));
#2237=VERTEX_POINT('',#2236);
#2238=CARTESIAN_POINT('',(115.0,10.0,60.0));
#2239=DIRECTION('',(0.0,0.0,-1.0));
#2240=VECTOR('',#2239,25.0);
#2241=LINE('',#2238,#2240);
#2242=EDGE_CURVE('',#2235,#2237,#2241,.T.);
#2243=ORIENTED_EDGE('',*,*,#2242,.F.);
#2244=CARTESIAN_POINT('',(105.0,10.0,60.0));
#2245=VERTEX_POINT('',#2244);
#2246=CARTESIAN_POINT('',(110.0,10.0,60.0));
#2247=DIRECTION('',(0.0,0.0,-1.0));
#2248=DIRECTION('',(1.0,0.0,0.0));
………………..
………………..
#3379=PLANE('',#3378);
#3380=ORIENTED_EDGE('',*,*,#2621,.T.);
#3381=ORIENTED_EDGE('',*,*,#3321,.F.);
#3382=ORIENTED_EDGE('',*,*,#3369,.F.);
#3383=ORIENTED_EDGE('',*,*,#2647,.F.);
#3384=EDGE_LOOP('',(#3380,#3381,#3382,#3383));
#3385=FACE_OUTER_BOUND('',#3384,.T.);
#3386=ADVANCED_FACE('',(#3385),#3379,.F.);
#3387=CLOSED_SHELL('',(#2270,#2312,#2354,#2396,#2436,#2467,#2498,#2538,#2569,#2600,#2631,#2824,#2848,#2866,#2876,#2907,#2931,#2955,#2972,#2984,#3002,#3012,#3030,#3040,#3071,#3095,#3119,#3136,#3148,#3179,#3203,#3227,#3244,#3256,#3274,#3284,#3328,#3345,#3357,#3374,#3386));
#3388=MANIFOLD_SOLID_BREP('12C',#3387);
#3389=ADVANCED_BREP_SHAPE_REPRESENTATION('ABSR1',(#212,#574,#662,#1024,#1316,#1928,#2078,#2228,#3388),#28);
#3390=SHAPE_REPRESENTATION_RELATIONSHIP('SRRPL1',' ',#3389,#122);
ENDSEC;
END-ISO-10303-21;**