

# Compilation of Step Ap-214 Data to Generate Computer Aided Process Planning for Automotive Parts



Vinod V Rampur, Sachhidanand Reur

**Abstract:** *The current manufacturing scenario feature technology which supports the integration of various systems of CAD with CAM then CAM with CAPP or any other computer aided systems. The most important function supported by feature technology the transfer of product and process data. Product model data is the standard for exchange of product information from one CAD system to different systems for seamless integration with CAM. Most of the CAD systems are developing the software's are using STEP AP-214 product model data. AP-214 is a systematic neutral data format for developing a feature-based process planning. In this paper, collection of STEP AP-214 based product model data as stated by ISO 10303 to identify the features of the product for manufacturing of prismatic part and generating a standard process plan. Further, this process planning is used to generate computer numerical control (NC) codes for CAM system to complete the manufacturing of part.*

**Keywords:** CAD, CAPP, CAM, CNC, STEP, AP-214.

## I. INTRODUCTION

Systematic planning of complete activities to be performed by which product can be manufactured at low cost and converting from raw material in to finished product is known as Process planning. Process planning is used to govern the specific sequence of operations to be performed based on the geometrical features, dimensional accuracy required, material and machineries are available for manufacturing the final product. [1,2,3]

Now a days the integration between design and manufacturing is significant challenge for seamless flow of data between CAD and CAM within the manufacturing system. The CAPP is the system developed to fill the gap between CAD and CAM. Under such situation now, the key to achieve the success is being recognized as a seamless data flow resulting in efficient CAD/CAPP/CAM integration. However, in the current manufacturing framework, solutions for CAD/CAPP/CAM and then CNC activities are being obtained from different vendors, that is resulting in a manufacturing environment of diverse applications.

**Revised Manuscript Received on November 30, 2019.**

\* Correspondence Author

**Vinod V Rampur\***, Assistant Professor, Department of Mechanical Engineering, PESITM, Shivamogga, India (Email: [vinod.rampur@pestrust.edu.in](mailto:vinod.rampur@pestrust.edu.in))

**Dr. Sachhidanand Reur**, Professor, Department of Industrial and Production Engineering, PDACE, Kalburgi, India. Email: (sr.chidu@gmail.com)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Even though good number attempts for integration approaches during the last decade have been tried, software integration and efficient exchange of product related data are yet to be achieved and the concerned issues are yet to be addressed effectively for the realization of CIM.

STEP standard file accommodates a detailed information of a product that to be manufacture. STEP standard categorizes various application protocols (AP) to exchange the product model data to integrate the different design software's which is related to development of the product [4]. Each AP is a file describing the life cycle of a product i. e application activity model (AAM). These standards include product and process data through formats designated as STEP, AP214 enables to overcome the CAD/CAPP integration problem. More specifically STEP-NC file can be employed to represent product related data in the form of a specific file known as neutral file and with help of this file exchange and sharing of data can be achieved in an integrated manufacturing system

### A. Integration of CAD/CAPP/CAM

As mentioned above CAD, CAPP and CAM are been used in the several streams of the manufacturing industries. But these three systems have some limitations sharing the product information among them which minimize productivity of the system. Further exchange of product model data between these three systems are required in the manufacturing environment. Practically various industries have separate perspective of integrating CAD/CAPP/CAM, but these three systems have some common features to integrate.

- To extract the part information from CAD system and transmit to CAPP system to generate process plan.
- The CAPP system provide method for the user to obtain, share, utilize and create a process plans automatically.
- The product information generated by the CAPP system by integrating CAPP/CAM in the process planning in converted in to NC codes to control the NC machines,

In simplest words basically, CAPP reduces the preparation time and related effort required to develop process plans and provides more dependable process plans. Mainly, it provides automated interface from CAD to CAM, thereby facilitating integration of various functions within the manufacturing system. Computer-aided process planning is a decision-making process. It governs the activities which are required to manufacture a part. For designing the CAPP system there are four main concerns are considered such as input, output, database and decision-making rules.

To manufacture each part, a necessary information's are entered to make proper process plan. The computer-aided process plan is preparing on the basis of machining operations, machine tools are using, method of manufacturing process, resources are required and so on. The CAPP system is developed basically by two approaches are retrieval or variant approach and generative approach.[5].

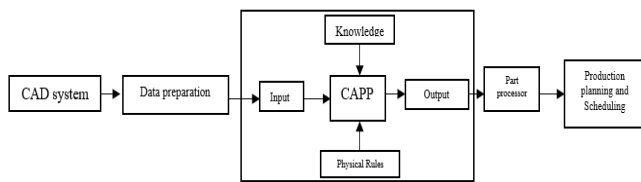


Fig 1: Basic structure of CAPP [5]

II. EXTRACTION AND FORMATTING OF DATA FROM STEP AP-214 NEUTRAL FILE

Machines specifically rely on product data to manufacture the product in CAD / CAM applications. Extraction is therefore the key principle for generating the correct data on the material. The goal is to make the pre-process function slowly accurate. Nevertheless, CAD information needs to be extracted to increase the accuracy. For feature recognition, machining planning, planning process, time scheduling, automatic NC part program generation etc., the information extracted from the CAD models will be used. In essence, the goal of automated extraction is to reduce human errors while understanding CAD data, which has inspired many CAD / CAM software system developments. In manufacturing, the developer was inspired to improve the manufacturing system to increase production efficiency and to reduce the manufacturing cost. Extraction is important to extract CAD data from the CAD model and present it for the manufacturing process as product information. Extraction is used to extract the object's lines, arcs, circles, vertex point, cartesian point, text, and boundaries within the model. The information collected will be listed and stored in a centralized database. Current application problem related to this research is the extraction of CAD component model data from its reorganized STEP neutral archive to produce product information and this information can be used for downstream CAPP activities mainly for feature identification, machining preparation and automated generation of NC part software. The method or procedure in the module is explained below.

The STEP neutral file is an ASCII file, it is interpreted as a string of characters when processing each line of the file, and data from this string is extracted using the Visual Basic language's string processing functions. In a formal database, these derived data are stored. Either curved or planar, each face of the component template is known as Advanced Face. If the advanced face is a plane surface, then the data grouped consists of, face Id, face bound Id, face bound Id (if present), plane Id, edge loop Id, number of outer bound edges, edge Id, edge form (line, circle etc.), edge dimensions, edge length, edge orientation and position of the regular vector to the plane Advanced Face. In the case of curved surface, the grouped data are, face Id, face outer bound Id, edge loop Id, number of outer bound edges, edge Id, form of edge (circle, cylindrical surface), distance, point center coordinate, normal direction of vector. The above group of data represent the complete description of each Advanced

Face, in turn it represents information about the part or product. This helps in understanding the model in terms of extracted data and makes easy for identifying features present in the model.

The computer program reads the STEP file line by line and looks for a line containing, 'CONFIGURATION CONTROLLED DESIGN' to confirm the file belongs to Application Protocol AP-214 of STEP-10303 Part-21 file and the data stored are according to ADVANCED\_BREP\_SHAPE\_REPRESENTATION. Once it is confirmed that the file is STEP AP-214, next it searches for line containing string "CLOSED\_SHELL" which is an entity contains id or pointers to ADVANCED\_FACE entity of the model. These id or pointers are recorded and counted to know the number of Advanced Face present in the CAD model by processing the current string. The program can search each Advanced Face by using the recorded id of Advanced Face or by using the word ADVANCED\_FACE as search parameter this can be applied to all other entity which can be searched by using their recorded id or by using entity name as search parameter.

When an ADVANCED\_FACE entity is searched, which normally consists of id or pointers of FACE\_OUTER\_BOUND and PLANE entity, if the advance face contains internal feature edges, then there will be an additional id which refers to entity FACE\_BOUND. Record these ids related to FACE\_OUTER\_BOUND, FACE\_BOUND and PLANE. Next these entities will be searched in sequence. Search the entity FACE\_OUTER\_BOUND, which normally has one id which refers to EDGE\_LOOP entity and record this id. Next search for EDGE\_LOOP entity by using recorded id, count and record the number of ids of the entity. This will be an important data, the number of id represent number edges which bound the Advance Face and all these id refer to entities ORIENTED\_EDGE. Now search for entities ORIENTED\_EDGE which referred by recorded id. The ORIENTED\_EDGE entity normally has single id and it refers to EDGE\_CURVE entity, record this id. Apart from id the entity ORIENTED\_EDGE also has a Boolean Flag, which will be used to reorient the edges if required so record these flags. Now search for entity EDGE\_CURVE with the recorded id, this entity has normally three id, either they will refer to LINE or CIRCLE entity. The first id refers to first vertex point, second id refers to second vertex point and third id refers to LINE entity, when these ids refer to CIRCLE, the first and second id refer to vertex point and third id refers to CIRCLE entity. Record these three ids. Search for first VERTEXT\_POINT entity with the first recorded id, the VERTEXT\_POINT has one id and record this id. This id refers to entity CARTESIAN\_POINT, record this id. Search the entity CARTESIAN\_POINT with the recorded id. The CARTESIAN\_POINT entity has no id, but has three parameters which represent X, Y and Z coordinates of VERTEXT\_POINT, record these coordinates. Search another CARTESIAN\_POINT with next id and record its X, Y and Z coordinates. Next with the third id search for entity which may be LINE or CIRCLE, if it is LINE then it has two id, first id refers to CARTESIAN\_POINT entity and the second id refers to entity VECTOR. Record these ids.

Search for CARTESIAN\_POINT with first id and record coordinates X, Y and Z as explained earlier.

Search for entity VECTOR with second id. This entity has two parameters, the first one is an id which refers to entity DIRECTION and the second parameter is magnitude of the line. Record these two parameters. Search for entity DIRECTION, this entity has three parameters represented by 0 and 1. These parameters indicate the directions X, Y and Z. The direction in which the line move is shown by 1 and the others as 0. Record these parameters. If the third id of EDGE\_CURVE entity refers to CIRCLE entity then this entity has two parameters, the first one is id which refers to entity AXIS2\_PLACEMENT\_3D and the second parameter is radius, record these parameters. Search for entity AXIS2\_PLACEMENT\_3D with recorded id, the AXIS2\_PLACEMENT\_3D has three id, the first id refers to entity CARTESIAN\_POINT which represent the point through which the plane passes having circular edge and the other two id refer to entity DIRECTION. The DIRECTION entity of the first id provides the direction of normal vector to the plane of the circular edge and the second id DIRECTION entity provides direction of an axis perpendicular to the normal vector direction. The entities CARTESIAN\_POINT and DIRECTION are processed in the same manner as explained previously and their parameters are recorded. Repeat the above process for other ORIENTED\_EDGE to complete the data extraction related to FACE\_OUTER\_BOUND.

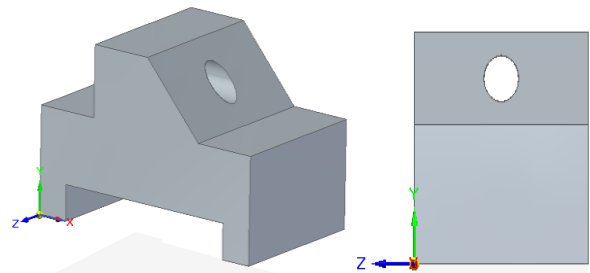
Assuming the ADVANCED\_FACE entity previously considered has an id which refers to FACE\_BOUND entity, this is possible only when an Advance Face has an internal feature edge. The processing of FACE\_BOUND entity is similar to that of processing FACE\_OUTER\_BOUND entity, FACE\_OUTER\_BOUND entity deals with the outer bound edges of ADVANCED\_FACE whereas the entity FACE\_BOUND deals with inner bound edges of ADVANCED\_FACE which refer to internal feature edges.

The last id in ADVANCED\_FACE entity refers to PLANE entity. Search this entity with the recorded id, the PLANE entity has one id which refers to entity AXIS2\_PLACEMENT\_3D, record this id. Search the entity AXIS2\_PLACEMENT\_3D with the recorded id and record its id. The processing of entity AXIS2\_PLACEMENT\_3D is made in same way as explained previously, only the first DIRECTION entity gives the direction of normal vector to the plane of ADVANCED\_FACE, this is an important data helps in identifying features. Record all data related to entity CARTESIAN\_POINT and two DIRECTION entities.

The above data extraction procedure is for first id of CLOSED\_SHELL entity which refer to ADVANCED\_FACE. The same procedure is applied for remaining all id of CLOSED\_SHELL entity to extract the data of all ADVANCED\_FACE entity to get the complete product information and store this information in the structured database.

### A. Reorganization of STEP File of an advanced face

The modules and their options are operated through menu driven system. The methods or procedures which are implemented in the file module which is concerned with reorganization and compilation of STEP file of the given part are explained in the following sections. For explanation the part shown in Fig 2 and its STEP file is used.



**Fig 2: Part model showing Through-Step, Through-Slot, Through Hole Features**

The reorganization of STEP file is made according to Advanced Face of the part model. This is done to group the data related to an Advanced Face which makes convenient for data extraction from STEP file. Initially this data is scattered throughout the file and it is done by tracing the id or pointers of the entity. The File module of the developed program is used for reorganization. The method of reorganization is explained with reference to the STEP file of part model shown in Fig 2. This method is implemented in the program.

- Search for entity CLOSED\_SHELL in STEP file, which has pointers of Advanced Face and is shown below,
- #300 = CLOSED\_SHELL  
(",(#285,#286,#287,#288,#289,#290,#291,#292,#293,#294,#295,#296,#297,#298,#299));

The above entity has pointers or id from #285 to #299, each one is pointing to an ADVANCED\_FACE entity. Each ADF entity represents one face of the model whether it is planar or curved surface.

- Search the file for first Advanced Face pointed by id #285 of entity ADVANCED\_FACE and is shown below,  
#285=ADVANCED\_FACE(",(#253),#271,.F.);
- The above entity has two pointers #253 pointing to entity FACE\_BOUND and #271 pointing to entity PLANE.
- Search the file for FACE\_BOUND entity pointed by id #253 and is shown below,  
#253=FACE\_BOUND(",#235,.T.);
- The above entity has one pointer #235 pointing to entity EDGE\_LOOP.
- Search the file for EDGE\_LOOP entity pointed by id #235 of FACE\_BOUND entity and is shown below,  
#235=EDGE\_LOOP  
(",(#23,#24,#25,#26,#27,#28,#29,#30,#31,#32,#33,#34));
- The above entity has twelve pointers #23 to #34 and all are pointing to entity ORIENTED\_EDGE entity of respective id.
- Search the file for ORIENTED\_EDGE entity pointed by id #23 of EDL entity and is shown below,  
#23=ORIENTED\_EDGE(",\*,\*,#99,.F.);
- The above entity has one pointer #99 pointing to entity EDGE\_CURVE.
- Search the file for EDGE\_CURVE entity pointed by id #99 of ORIENTED\_EDGE entity and is shown below,  
#99=EDGE\_CURVE(",#137,#138,#163,.T.);
- The above entity has three pointers #137, #138 and #163. The pointer #137 points to entity VERTEX\_POINT, the pointer #138 points to another VERTEX\_POINT entity and the pointer #163 points entity LINE.



- Search the file for VERTEX\_POINT entity pointed by id #137 of EDGE\_CURVE entity and is shown below, #137=VERTEX\_POINT (",#405);
- The above entity has one pointer #405 pointing to entity CARTESIAN\_POINT.
- Search the file for CARTESIAN\_POINT entity pointed by id #405 of VERTEX\_POINT entity and is shown below, #405=CARTESIAN\_POINT (" (60.,30., -50.));
- The above entity has no pointers or id but has three coordinate (X, Y, Z) values. Search for entities through pointers or id ends here, but for other pointers or id which are left while following one id. This will be repeated for all id and traced till id exhaust.
- Next search for entity pointed by id #138 of entity EDGE\_CURVE pointed by id #99 and is shown below, #138=VERTEX\_POINT (" , #406);
- The above entity has one pointer #406 pointing to entity CARTESIAN\_POINT.
- Search the file for CARTESIAN\_POINT entity pointed by id #406 of VERTEX\_POINT entity and is shown below, #406=CARTESIAN\_POINT (" ,(40.,50.,-50.));
- The above entity has no pointers or id but has three coordinate (X, Y, Z) values. Search for other pointers or id which is left while following this id.
- Next search for entity pointed by id #163 of entity EDGE\_CURVE pointed by id #99 and is shown below, #163=LINE (" ,#404,#199);
- The above entity has two id #404 which points to entity CARTESIAN\_POINT and id #199 which points entity VECTOR.
- Search the file for CP entity pointed by id #404 of above LINE entity and is shown below, #404=CARTESIAN\_POINT (" ,(45.,45.,-50.));
- The above entity has no pointers or id but has three coordinate (X, Y, Z) values. Search for other pointers or id which are left while following this id.
- Search the file for VECTOR entity pointed by id #199 of above LINE entity and is shown below, #199=VECTOR (" ,#334,1000.);
- The above entity has one pointer #334, which points to entity DIRECTION.
- Search the file for DIRECTION entity pointed by id #334 of above VECTOR entity and is shown below, #334=DIRECTION (" ,(-0.707106781186548,0.707106781186548,0.));
- The above entity has no pointers or id but has three parameters used to identify direction of LINE entity. Search for first ORIENTED\_EDGE #24 of entity EDGE\_LOOP having id #235 finishes here, the search for other id #25,#26,#27,#28,#29,#30,#31,#32,#33 and #34 should be made in the same way as made for id #23 following steps from 5 to 14. The only id left for searching is #271 of entity ADVANCED\_FACE, trace the id followed by #271,
- Search the file for PLANE entity pointed by id #271 of entity ADVANCED\_FACE and is shown below, #271=PLANE (" ,#313);
- The above entity has one pointer #313 pointing to entity AXIS2\_PLACEMENT\_3D.
- Search the file for AXIS2\_PLACEMENT\_3D entity pointed by id #313 of PLANE entity and is shown below, #313=AXIS2\_PLACEMENT\_3D (" ,#403,#332,#333);
- The above entity has three pointers #403, #332 and #333. The pointer #403 points to entity CARTESIAN\_POINT, the pointer #332 points to DIRECTION entity and the pointer #333 points to another entity DIRECTION.
  - Search the file for CARTESIAN\_POINT entity pointed by id #140 of above AXIS2\_PLACEMENT\_3D entity and is shown below, #403=CARTESIAN\_POINT (" ,(0.,0.,-50.));
  - The above entity has no pointers or id but has three coordinate (X, Y, Z) values. Search for other pointers or id which are left while following this id.
  - Search the file for DIRECTION entity pointed by id #332 of above AXIS2\_PLACEMENT\_3D entity and is shown below, #332=DIRECTION (" ,(0.,0.,1.));
  - The above entity has no pointers or id but has three parameters used to identify direction of normal vector to the plane.
  - Search the file for DIRECTION entity pointed by id #333 of above AXIS2\_PLACEMENT\_3D entity and is shown below, #333=DIRECTION (" ,(1.,0.,0.));

**III. DATABASE FOR STORING EXTRACTED DATA FROM STEP FILE-PART DATABASE**

The CAD model data extracted from the STEP neutral file with the above procedure is stored in a structured database called Part Database, designed specifically for this purpose. This database is used as a backend to the computer program developed. The schema of the database is shown below,

**Table 1: Field names**

ADF	FOT B	F B	P L	ED L	EL I	GEI D	GE	RADI	CPX	CPY	CPZ	X1	Y1	Z1	X2	Y2	Z2	L L	LD	SC	FP
-----	----------	--------	--------	---------	---------	----------	----	------	-----	-----	-----	----	----	----	----	----	----	--------	----	----	----

The above database schema has the following field names and the data which they store is shown below,



**Table 2: Application of Part Database for Feature Recognition**

Field Name	Entity Name / Meaning	Description
ADF	ADVANCED_FACE	Stores id
FOTB	FACE_OUTER_BOUND	Stores id
FB	FACE_BOUND	Stores id
PL	PLANE	Stores id
EDL	EDGE_LOOP	Stores id
EDC	EDGE_CURVE	Stores id
ELI	EDGE_LOOP Id	Stores number of ids in the EDGE_LOOP entity
GEID	Geometric_Entity_ID	Stores id of geometrical entity like Line, Circle, Cylindrical Surface
GOE	Geometric Entity	Stores type of geometrical entity like Line, Circle, Cylindrical Surface
RADI	Radius	Stores radius of curved surface
CPX	Centre point X coordinate	
CPY	Centre point Y coordinate	
CPZ	Centre point Z coordinate	
X1	Line starting X coordinate	Stores X coordinate of starting point of LINE.
Y1	Line starting Y coordinate	Stores Y coordinate of starting point of LINE.
Z1	Line starting Z coordinate	Stores Z coordinate of starting point of LINE.
X2	Line ending X coordinate	Stores X coordinate of ending point of LINE.
Y2	Line ending Y coordinate	Stores Y coordinate of ending point of LINE.
Z2	Line ending Z coordinate	Stores Z coordinate of ending point of LINE.
LL	Line Length	Stores length of line
LDI	Line Direction	Stores direction of line
SC	Shape Code	Stores code according syntactic pattern
FP	Face Perpendicular to	Stores direction of normal vector to the plane

The features of part model are recognized by interpreting the data from part database. Procedures are developed to recognize the features using these data. As mentioned earlier

the records in the database are grouped together according to Advanced Face of part model. The interpretation of data for recognizing features of part model is explained below for different types of Advanced Face.

**A. Advanced face representing perfect quadrilateral face with no internal feature edges**

As mentioned earlier, any type of face of a part model considered as Advanced Face in the STEP file whether it is a planar or curved or cylindrical surface. The Perfect Quadrilateral Face shown in Table 3. Every topological information of the Advanced Face has an identifier of the form #N. The structure of the data stored in the part database related to this Advanced Face is explained below field by field and record by record starting from the first record of the Advanced Face.

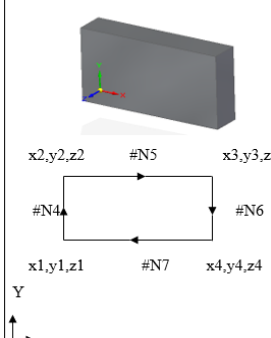
Every Advanced Face database record begins with the id (#N1) of the face which will be stored in the first field ADF of the entity ADVANCED\_FACE which represents entire face. The second field FOTB stores the id (#N2) of entity FACE\_OUTER\_BOUND which represents the entire bounding edges of the face. The third field FB stores the id of entity FACE\_BOUND, but is null. This field will remain null for all the records of Advanced Faces which are not having any internal feature edges, this will be an important hint to know whether a face has internal feature or not. The fourth field PL stores the id of PLANE entity, but for the current record it is null. The fifth field EL stores id (#N3) of entity EDGE\_LOOP, this is the loop of edges which bounds the Advanced Face. The sixth field ELI stores the number of edges present in the edge loop, for Perfect Quadrilateral Face the number of edges in the EDGE\_LOOP are four. The seventh field GEID stores the id (#N4) of the geometric entity of EDGE\_LOOP. The eighth field stores the type (Line) of geometric entity present in the EDGE\_LOOP. The ninth field Rad stores radius of arc or circle or cylindrical entity, but it is null for this face. The tenth, eleventh and twelfth field CPX, CPY and CPZ store the Centre point coordinates of the arc or circle or cylindrical entity. The thirteenth, fourteenth and fifteenth field X1, Y1 and Z1 store the starting coordinates (x1, y1, z1) of geometric entity, in this case it is a line, its starting point coordinates are stored. Similarly, the sixteenth, seventeenth and eighteenth fields X2, Y2 and Z2 store the ending coordinates (x2, y2, z2) of the line. The nineteenth field LL stores the length of the LINE entity and the twentieth field stores the direction (X, Y, Z) of the line. Where ever required the direction of the LINE entity is reoriented to suit the syntactic pattern by changing the Boolean flags of the ORIENTED\_EDGE entity. The order of the geometric entity is set in such a way that, the lower left corner of the face is taken as initial or start point and moving in clockwise direction, then returning back to initial point forming a continuous and closed loop. The twenty-first field SC stores the syntactic pattern code (A, B, C, D) assigned to the geometric entity depending on the direction. The last field FP stores the direction of normal vector to the Advanced Face, for this record it is null.

This is an important data helps in identifying features of part model.

The next records belong to the remaining edges of the EDGE\_LOOP entity. The EDGE\_LOOP has four edges, the first record includes the first edge of the loop, so the next three record pertains to remaining edges (#N5, #N6, #N7). The initial fields ADF, FOTB, FB, PL, EDL and ELI are null for all three records and data are assigned for remaining fields of the records I.e. for fields GEID, GE, Radi, CX, CY, CZ, X1, Y1, Z1, X2, Y2, Z2, LL, LDI, SC and FP. The last record related to the Advanced Face belongs to PLANE entity.

The initial fields of the record ADF, FOTB, FB are null, the next field PL is assigned id (#N8) of the PLANE entity, again the fields EDL, ELI, GEID are null. The next field GOE is assigned the value "Plane", indicating the Advanced Face is a planar surface. The next fields Radi, CPX, CPY and CPZ are assigned null values. The next fields, X1, Y1 and Z1 assigned the coordinate (x, y, z) values through which the Plane passes. This is called Plane Position Values (PPV). The fields X2, Y2, Z2, LL and LDI are assigned null values. The next field SC is assigned a string (DABC) which is formed by the concatenation of all codes assigned to each geometric entity of edge loop in the previous records, which also represents the shape of the Plane. The last field FP is assigned value (X, Y, Z) which is the direction of normal vector to the plane of the Advanced Face. This data structure can be used to represent all the Perfect Quadrilateral Face of the part model.

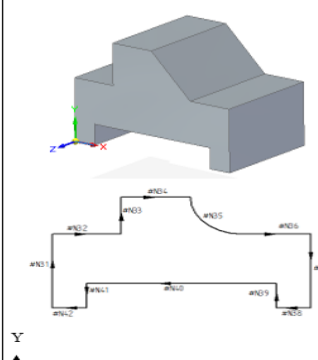
Table 3. Perfect Quadrilateral Face

Sl. No	ADF	PLANE	PPV	FP
01	#N1		x, y, z	Z

**B. Advanced Face Representing Complex Polygon Face with no internal Feature Edges**

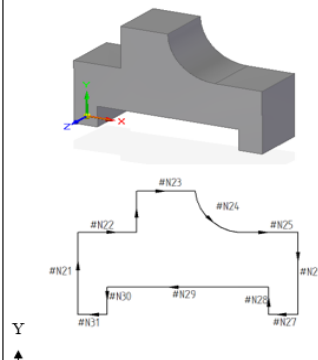
A Complex Polygon Face is formed by the intersection of external features with the bounding edges of Perfect Quadrilateral Face. The intersection modifies the bounding edges and converts a Perfect Quadrilateral Face (PQF) to Complex Polygon Face (CPF) is shown in Table 4.2. The data structure for a CPF made by lines is similar to that of data structure of PQF explained above, the only difference is that, the number of edges in the EDGE\_LOOP is four in PQF where as in CPF these edges are more than four. These additional edges records are added to the existing records of the PQF structure to make the structure of CPF in database.

Table 4. Complex Polygon Face

Sl. No	AF	PLANE	FPV	FP
02	#N9		x, y, z	Y

The data structure of a CPF made by the combination of lines and arcs is shown in Table 3.3, is similar to that of CPF made by lines, the only difference is that, the records related to edges formed by arc are added to the structure of CPF made by lines. The structure is shown in Table 4&5.

Table 5. Complex Polygon Face with Curved Edge

Sl. No	ADF	PLANE	FPV	FP
03	#N23		x, y, z	Y

**C. Advanced Face with internal Feature Edges**

The internal feature edges in a face are formed by its intersection with the face. The face is shown in Table 6. In the STEP file these edges are considered as closed edge loops and are identified by entity FACE\_BOUND. The presence of FACE\_BOUND entity gives hint about the presence of internal feature in the face. The data structure for the face whether it is PQF or CPF with internal feature edges is same as that of PQF or CPF, but the only difference is, the records related to edge loop of the FACE\_BOUND entity of internal feature will be added.

The records are similar to the edge loop of FACE\_OUTER\_BOUND entity. The syntactic pattern codes of the geometric entity of records of edge loops are concatenated similar to edge loop of FACE\_OUTER\_BOUND records and assigned to the SC field placed after all the records.

**Table 6. Perfect Quadrilateral Face with Internal Feature Edge**

Sl No	ADF	PLANE	FPV	FP
04	#N33		x, y, z	Z

topological information of the part in a neutral file format which leads to transfer between the computer aided system. To transfer the feature information which is collected from STEP AP-214 product data for manufacturing of prismatic part. In this work first recognizing the STEP file of a part and store extracted features of the part from the STEP file part database. Graphical representation of advance faces of the part representing the with or without internal faces. Finally reconstruct the part model as per the features identified which is stored in the database.

**D. Graphical Representation of Advanced Face**

The whole part model is a collection of Advanced Face and the data of these faces are extracted from STEP neutral file and stored in the part database with above structure of data explained in the previous section. The data stored is according to the Advanced Face of the model, to verify the reorganization of STEP file and to check the face representation by data; the faces are reconstructed from the stored data. This will provide a clear picture of each face, if any error in shape of the face can be easily detected.

The software module “Graphics” is used to show the Advanced Faces graphically in 2D from the direction X, Y and Z. The user can select any directions to view the faces, along with the shape of the face it provides information about face id, coordinates of bounding edges of face, edge id, Plane Position Values and direction of normal vector to the face. The faces are shown in above Tables.

**E. Reconstruction of Part Model**

The reconstruction of part model provides a complete information about structure, geometrical and topological information of the part model, this information helps in clear understanding of the model and this knowledge will be used to recognize the features of the part model and further will be used for, machining planning, automatic generation of part program for NC machines.

The Advanced Faces which are reconstructed in the previous section will be used for reconstruction of the part model. By connecting these faces orderly based on the Plane Position Value (PPV) and normal vectors of the face. The program module “Graphics” is used for the purpose, the module provides three options, the first option display the Advanced Faces perpendicular to X axis, the second display the faces perpendicular to Y axis and the third option display the faces perpendicular to Z axis and combines all faces from all directions and display the whole part model by using the data stored in the Part Database.

**IV. CONCLUSION**

The feature technology is supporting the integration of CAD/CAPP/CAM systems must possess geometrical and





ADF NO	FOTB	FB	PL	EDL	ELI	GEID	GE	Radi	CX	CY	CZ	X1	Y1	Z1	X2	Y2	Z2	LL	LDI	SC	FP
<b>Advanced Face Representing Perfect Quadrilateral Face with no internal Feature Edges</b>																					
#N1	#N2			#N3	4	#N4	Line					x1	y1	z1	x2	y2	z2	l11	Y	D	
						#N5	Line					x2	y2	z2	x3	y3	z3	l12	X	A	
						#N6	Line					x3	y3	z3	x4	y4	z4	l13	-Y	B	
						#N7	Line					x4	y4	z4	x1	y1	z1	l14	-X	C	
				#N8			Plane					x	y	z						DABC	Z
<b>Advanced Face Representing Complex Polygon Face with no internal Feature Edges</b>																					
#N9	#N10			#N11	10	#N12	Line					x1	y1	z1	x2	y2	z2	l11	Y	D	
						#N13	Line					x2	y2	z2	x3	y3	z3	l12	X	A	
						#N14	Line					x3	y3	z3	x4	y4	z4	l13	Y	D	
						#N15	Line					x4	y4	z4	x5	y5	z5	l14	X	A	
						#N16	Line					x5	y5	z5	x6	y6	z6	l15	(X, -Y)	A, B	
						#N17	Line					x6	y6	z6	x7	y7	z7	l16	X	A	
						#N18	Line					x7	y7	z7	x8	y8	z8	l17	-Y	B	
						#N19	Line					x8	y8	z8	x9	y9	z9	l18	-X	C	
						#N20	Line					x9	y9	z9	x10	y10	z10	l19	Y	D	
						#N21	Line					x10	y10	z10	x11	y11	z11	l10	-X	C	
						#N22	Line					x11	y11	z11	x12	y12	z12	##	-Y	B	
						#N23	Line					x12	y12	z12	x1	y1	z1	##	-X	C	
				#N24			Plane					x	y	z						DADAABABCDABC	Z
<b>Advanced Face Representing Complex Polygon Face with Curved Edge</b>																					
#N23	#N24			#N25	4	#N31	Line					x1	y1	z1	x2	y2	z2	l11	Y	D	
						#N32	Line					x2	y2	z2	x3	y3	z3	l12	X	A	
						#N33	Line					x3	y3	z3	x4	y4	z4	l13	Y	D	
						#N34	Line					x4	y4	z4	x5	y5	z5	##	X	A	
						#N35	Circle	R	cx	cy	cz	x5	y5	z5	x6	y6	z6		R	R	
						#N36	Line					x6	y6	z6	x7	y7	z7	l15	X	A	
						#N37	Line					x7	y7	z7	x8	y8	z8	l16	-Y	B	
						#N38	Line					x8	y8	z8	x9	y9	z9	l17	-X	C	
						#N39	Line					x9	y9	z9	x10	y10	z10	##	Y	D	
						#N40	Line					x10	y10	z10	x11	y11	z11	##	-X	C	
						#N41	Line					x11	y11	z11	x1	y1	z1	##	-Y	B	
						#N42	Line					x12	y12	z12	x13	y13	z13	##	-X	C	
				#N43			Plane					x	y	z						DADARABCDABC	z
<b>Advanced Face with Internal Feature Edges</b>																					
#N33	#N34			#N35	4	#N45	Line					x1	y1	z1	x2	y2	z2	l11	Y	D	
						#N46	Line					x2	y2	z2	x3	y3	z3	l12	X	A	
						#N47	Line					x3	y3	z3	x4	y4	z4	l13	-Y	B	
						#N48	Line					x4	y4	z4	x5	y5	z5	l15	-X	A	
																				DABC	
	#N40			#N41	4	#N49	Line					x1	y1	z1	x2	y2	z2	l11	Y	D	
						#N50	Line					x2	y2	z2	x3	y3	z3	l12	X	A	
						#N51	Line					x3	y3	z3	x4	y4	z4	l13	-Y	B	
						#N52	Line					x4	y4	z4	x5	y5	z5	l15	-X	C	
				#N53			Plane					x	y	z						DABC	Z

**ACKNOWLEDGMENT**

I would like to thank my Management and Principal of PESITM, Shivamogga Karnataka, India for their support provided. My sincere thanks to guide Dr. Sachhidanand Reur for energy and motivation towards work carried out and his precious valuable guidance helped me for fulfilling my dissertation work. I express my sincere thanks to Principal and Management of PDACE, Kalburgi, Karnataka, India research center for their valuable support. We gratefully thank Visvesvaraya Technological University, Jnana Sangama, Belagavi, for support extended to this work.

**REFERENCES**

1. T. C. Chang and R. A. Wysk. An introduction to automated process planning systems. Prentice Hall Professional Technical Reference, 1984.
2. P. Gu and D. H. Norrie. *Intelligent Manufacturing Planning*. London, UK: Chapman & Hall, 1995.
3. H. C. Zhang and L. Alting. *Computerized Manufacturing Process Planning Systems*. London, UK: Chapman & Hall, 1994.
4. Suh, S. H., Cho, J. H., and Hong, H. D., "On the Architecture of Intelligent STEP-Compliant CNC," *International Journal of Computer Integrated Manufacturing*, 15(2), pp. 168-177. January 2002.

5. Li-Hong Qiao, Zhi-Bing Yang, H.-P. Ben Wang, "A computer-aided process planning methodology" *Computers in Industry*, pp. 83-94. 1994
6. Roller, D.: Design by features: an approach to high level manipulations, *Computers in Industry*, 12, 1989, 185- 191.
7. Shah JJ, Mantyla M "Parametric and feature-based CAD/ CAM: concepts, techniques, and applications". Wiley, New York, 1995.

**AUTHORS PROFILE**



**Vinod V Rampur** graduated in industrial and production engineering from PDACE, Kalburgi, Karnataka in the year 2008. He completed his Master's degree in Computer Integrated Manufacturing from GIT, Belagavi, Karnataka un the year 2011, He is currently undertaking Ph. D under Dr. Sachhidanand Reur's supervision from Belagavi He is a Life member of Indian Society for Technical Education (ISTE) since 2013. He is associated with Mechanical Department at PES institute of Technology and Management, Shivamogga and affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India. He is interested in the study of CAD/CAPP/CAM/CIM.  
E-Mail: vinod.rampur@pestrust.edu.in, vinod.rampur@gmail.com







**Dr. Sachhidanand Reur** graduated in Mechanical engineering from Gulbarga University in the year 1986. He completed Master's in Production technology from Jadavpur University, Kolkata in the year 1997. He completed his doctoral degree in CAD/CAPP/CAM/CIM in the year 2013 from VTU Belagavi, Karnataka. He is working as

Professor in the department of Industrial and Production Engineering in PDACE, Kalburgi, Karnataka. His area of interest for research CAD, CAM, CIM, Robotics, Automation, Virtual Reality, Digital Manufacturing. He is Life Member Indian Society for Technical Education (LMISTE), New Delhi. and Member Institution of Engineers, India (MIE)

E-Mail: sr.chidu1@gmail.com