International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Conception and Representation of GD&T Module in an Expert CAPP System for Rotational Parts

Rafik Elaakil[1], Rechia Ahmed[1], Abdelilah El Mesbahi[2], Oussama Jaider[3]

[1, 2, 3]Department of Mechanical Engineering, Faculty of Sciences and Technics of Tangier
Old Airport Road, Km 10, Ziaten.BP: 416. Tangier – Morocco

Abstract: ***Computer-aided process planning (CAPP) is an essential interface for linking design and manufacturing processes. The purpose of CAPP is to transform a part design specification obtained from CAD system into a sequence of machining operations which can be used by CAM system involving the selection of necessary manufacturing processes, in order to transform a designed part into a final part economically and competitively. This paper focuseson one hand, inthe extraction of tolerances (GD&T), and on the other hand,in the association between (GD&T), and the related features, embedded into an expert CAPP system.***

Keywords: CAD/CAM, CAPP, machining operation sequencing, Expert system, knowledge bases, Artificial intelligence.

1. Introduction

Computer Aided Process Planning is an important activity in an intelligent manufacturing system. Several techniques have been proposed, implemented and tested in the context of generative CAPP; it consists of several activities including machining feature recognition, selection of machining operation type, selection of manufacturing resource, setup planning and operation sequencing.

The objective of precedence-constrained sequencing problem is to locate the optimal sequence dealing with time and costs, so that the resulting operation sequence could satisfy those constraints established by geometric, technological, economical requirements, and then generate the optimal or near optimal operation sequences.

The extraction of tolerances data is an essential key that helps to identify the basic constraints of the related input part.

To achieve a complete representation for a process-planning model, data tolerances (GD&T) needs to be completely extracted and represented through the input data file.

The development of a robust methodology for extracting this data for the use of a knowledge-based expert system, incorporating an artificial intelligence algorithm is the main thrust of the work reported in this paper.

2. Related Researches Work and Review

Form features are not sufficient so that the part will be manufactured at a reasonable cost and will perform its intended function. Most researchers have concentrated on geometric data extraction, Antonio Armillotta [1] describes a method for the generation of tolerance specifications from product data, for each feature one or more geometric controls will have to be identified. Mehmet I. Sarigecili [2] presents a way to interpret Geometric Dimensioning and Tolerancing (GD&T) specifications in STEP for tolerance analysis by utilizing OntoSTEP plugin [3], which consists of translating the APs defined in the EXPRESS language [4], and the actual instance files represented using the ISO 10303-21 (or simply part21) format [5].

Pei Lei [6], presents an on-machine measurement data model based on STEP-NC. This model represents tasks in process plan in an object-oriented structure, containing information on geometry and tolerances.

3. ICAPP-TURN Architecture

A main objective in modern design and advanced machining process is to achieve a significant reduction in product lead time through seamless integration between the various design and process planning activities. The concept of using component features to integrate a design system and a manufacturing system has been a major research direction in recent years. Hakki Eskicioglu [7], describes the characteristics of an existing expert systems building tools and discusses the use and the importance of these basic characteristics from a perspective of computer aided process planning (CAPP) requirements in order to automate the process planning functions.

The framework of the system proposed in this paper is decomposed into several modules. Figure 1 shows the conceptual model that defines the structure of ICAPP-TURN.

The process start from the knowledge acquisition step that consists of structuring and organizing knowledge from one source, usually human experts, so that it can be used and represented into the knowledge representation module.

The knowledge database represents a centralized repository to store constraints and rules provided by the knowledge representation module.In other side, the technical data information is retrieved through the integration module that extracts the technical information of such part, which can be represented to the system across the data representation module that maps this information into technical objects.

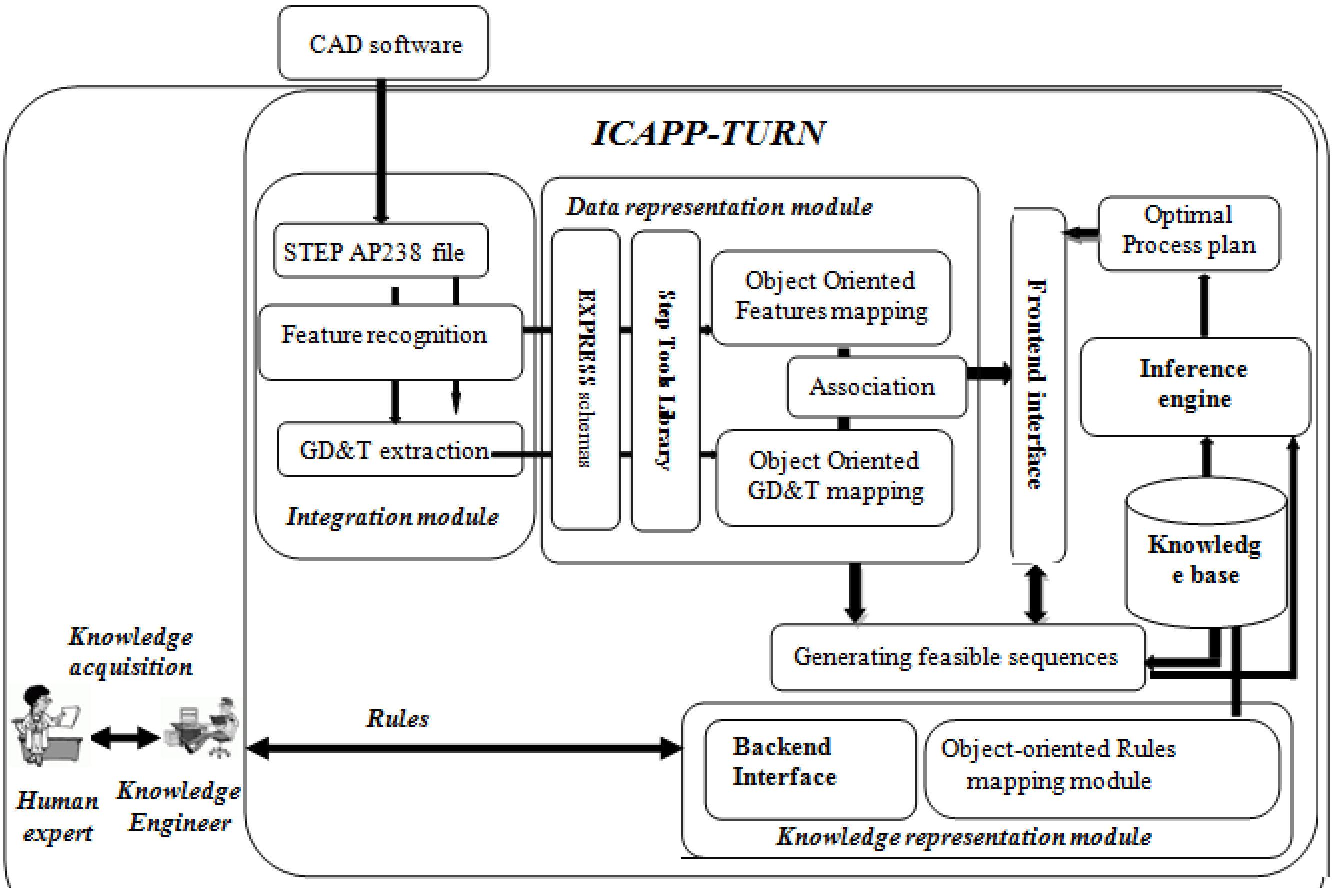

Figure1: Architecture of ICAPP-TURN

The core system is represented as an inference engine which helps to determine the optimal sequences plan using an artificial intelligence method, this module interacts with the Knowledge database.

The main function of each module is described below:

- **Knowledge acquisition:**this module consists of extracting, structuring and organizing knowledge from one source, usually human experts, so it can be used and represented in the next module of ICAPP-TURN System.
- **Knowledge representation module**: this module represents the knowledge acquisition from the previous module in a form that ICAPP-TURN system can utilize to provide the knowledge to the database so it can be used to solve the optimization problem.

1. **Backend Interface**: In software architecture, there may be many layers between the hardware and end user. Each can be spoken of as having a front end and a back end.The separation of software systems into front and back ends simplifies development and separates maintenance.This module represents the interface from which the knowledge engineer can organize those facts, rules and situations; it can also be used to serves in support of the expert system.

2. **Oriented Object Rules Mapping module**: used to map the technical data information about rules from the knowledge acquisition module to oriented objects code, which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.

- **Data Representation module:**
  1) **Express Schema module**: This module uses Express data modelling language for the product data (extracted features and GD&T) based on ISO Standard.
  2) **Oriented object Features/GD&T Mapping module**: used to map the technical data information aboutFeatures/GD&T from the input file to oriented objects code.
- **Knowledge Base**: This Module represents a centralized repository for information structured as a database, used to store the precedence constraints and rules provided by the knowledge representation module.
- **The feasible sequences module**: This Module represents the generated feasible sequences of such part respecting the precedence constraints.The concerned information about the constraints can be retrieved from the knowledge database.
- **Inference Engine based on Artificial Intelligence**: This Module represents the core component of the system which helps to determine the optimal sequences plan using an artificial intelligence method.This module interacts with the knowledge database.
- **Integration module:**
  1) **Features recognition module**: This module extracts features from the AP 238 file which contains all the concerned information of such part.
  2) **GD&T extraction module:** This module extracts tolerances from the AP 238 file which contains all the concerned information of such part.
- **The optimal process plan generated:** This module represents an output of the optimal process plan generated.

- **Front End Interface:** A user interface by which the expert system interacts with an end user. These can be through dialog boxes, command prompts, forms, or other input methods

4. Technical Data Integration and Representation Modules

The integration module consists of recognizing manufacturing features and tolerances from a STEP AP238 file, then associate each tolerance to the related feature through the data representation module.

Jaider Oussama [8] developed an automatic feature recognition system for rotational parts which adopted a STEP AP203 file as input to the system.

STEP AP203 is an application protocol of the Standard for the Exchange of Product model data (STEP), defined as the international standard ISO 10303 [9]. For the fact that the definitions of geometric and topological entities of the STEP AP203 are included in the AP238, we have used this feature recognition system to detect features from a STEP AP238, generated by STEP-NC Machine [10].By applying this recognition, the system can recognize multiple features types.

Figure 2 show an input part to our system.

The feature recognition system recognizes in this case:
- One outer diameter, which is composed by two cylindrical surfaces.

#303=ADVANCED_FACE('Corps principal',(#149),#174,.T.);

#307=ADVANCED_FACE('Corps principal',(#179),#174,.T.);

- One shoulder, which is composed by 3 advanced faces, one planar surface and two cylindrical surfaces.

#35=ADVANCED_FACE('Corps principal',(#36),#70,.T.);
#75=ADVANCED_FACE('Corps principal',(#76),#70,.T.);
#242=ADVANCED_FACE('Corps principal',(#199,#203),#207,.T.);

- One face which is composed by one planar surface:

#312=ADVANCED_FACE('Corps principal',(#190),#194,.T.);

- And one chamfer, which is composed by joining two conical surfaces:

#132=ADVANCED_FACE('Corps principal',(#133),#127,.T.);
#92=ADVANCED_FACE('Corps principal',(#93),#127,.T.);

In the following section, we will explain how a type of tolerance such as a linear distance (named as DIRECTED_LINEAR_DISTANCE_DIMENSION) is extracted from STEP AP238 file, and then associated to the corresponding feature.
As shown in figure 2, a distance between the faces F242 and F312 represent a linear dimension tolerance of 30 mm as a nominal value.This tolerance is embedded in the STEP AP238 data file of the example part as shown in figure 3.
In figure 3, the CALLOUT ( ) entity defines a callout identifying a particular face (ADVANCED_FACE) on the part that can have some properties (e.g. Linear distance dimension).
The DIRECTED_LINEAR_DISTANCE_DIMENSION(#248) function (figure 4) defines a distance between two faces, it has an attribute denoted as DIMENSION_VALUE_TOLERANCE: #248, #249, #250, #251, where #251is a pointer to the value of the distance between F312 and F242, which is given by the following record:

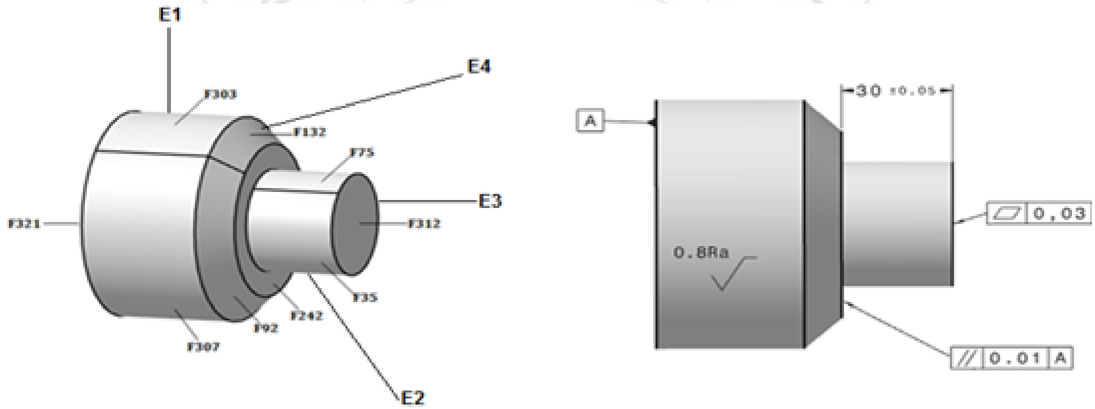

Figure 2: An input example part that contains toleranced features

#251=(
LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(30.),#256
)
REPRESENTATION_ITEM('linear distance'));

The DIRECTED_LINEAR_DISTANCE_DIMENSION (#248) function has another attribute that defines the target face, denoted as TARGET: #248, #313, where

```
***********************************************
* Application object: DIRECTED_LINEAR_DISTANCE_DIMENSION (#248)
* DESCRIPTION: #248,['']
* DIMENSION_VALUE_TOLERANCE: #248,#249,#250,#251
* TARGET: #248,#313
* ID: #248,['linear distance']
* ORIGIN: #248,#308
* PLUS_MINUS_LIMITATION: #248,#252,#276
*/
#248=DIRECTED_DIMENSIONAL_LOCATION('linear distance','',#308,#313);
#249=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#248,#250);
#250=SHAPE_DIMENSION_REPRESENTATION('',(#251),#32);
#251=(
LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(30.),#256)
REPRESENTATION_ITEM('linear distance')
);
#252=PLUS_MINUS_TOLERANCE(#276,#248);
/***********************************************
* Application object: CALLOUT (#308)
* ITS_WORKPIECE: #308,#19,#18
* EXPLICIT_REPRESENTATION [*]: #308,#309,#310,#311,#312
* ITS_ID: #308,['']
*/
#308=SHAPE_ASPECT('','',#19,.F.);
#309=PROPERTY_DEFINITION('','',#308);
#310=PROPERTY_DEFINITION_REPRESENTATION(#309,#311);
#311=REPRESENTATION('',(#312),#32);
#312=ADVANCED_FACE('Corps principal',(#190),#194,.T.);
/***********************************************
* Application object: CALLOUT (#313)
* ITS_WORKPIECE: #313,#19,#18
* EXPLICIT_REPRESENTATION [*]: #313,#314,#315,#316,#242
* ITS_ID: #313,['']
*/
#313=SHAPE_ASPECT('','',#19,.F.);
#314=PROPERTY_DEFINITION('','',#313);
#315=PROPERTY_DEFINITION_REPRESENTATION(#314,#316);
#316=REPRESENTATION('',(#242),#32);
/***********************************************
```

Figure 3: A partial STEP AP238 text file of the definition of directed linear distance of the part of figure2

#313is a pointer to the callout of the target face F242. Another attribute that defines the origin face, is denoted as ORIGIN: #248, #308, where #308 is pointer to the callout of the origin face F312.

This function has another attribute denoted as PLUS_MINUS_LIMITATION: #248, #252, #276, where #276 is an identifier to the

TOLERANCE_PLUS_MINUS_VALUE (#276) entity. This function has two attributes, denoted as LOWER_LIMIT: #276, #277 and UPPER_LIMIT: #276, #278, where #277 and #278are respectively the pointers to the lower bound value (-0.05) and upper bound value (0.05) of this dimensional distance.The definition of the set of functions that define a diameter in the STEP AP238 file is shown in the flowchart of figure 4.

The DATUM_FEATURE_FACE ( ) function definesone or more faces that are a reference point for atolerance.Depending on the types of tolerance, datuminformation is required. The linear distance tolerance must have datum information (A). The DATUM_FEATURE_FACE (#238) function and thepointer to the face F242 are put in the same block,where #238 is an identifier to the face F242.

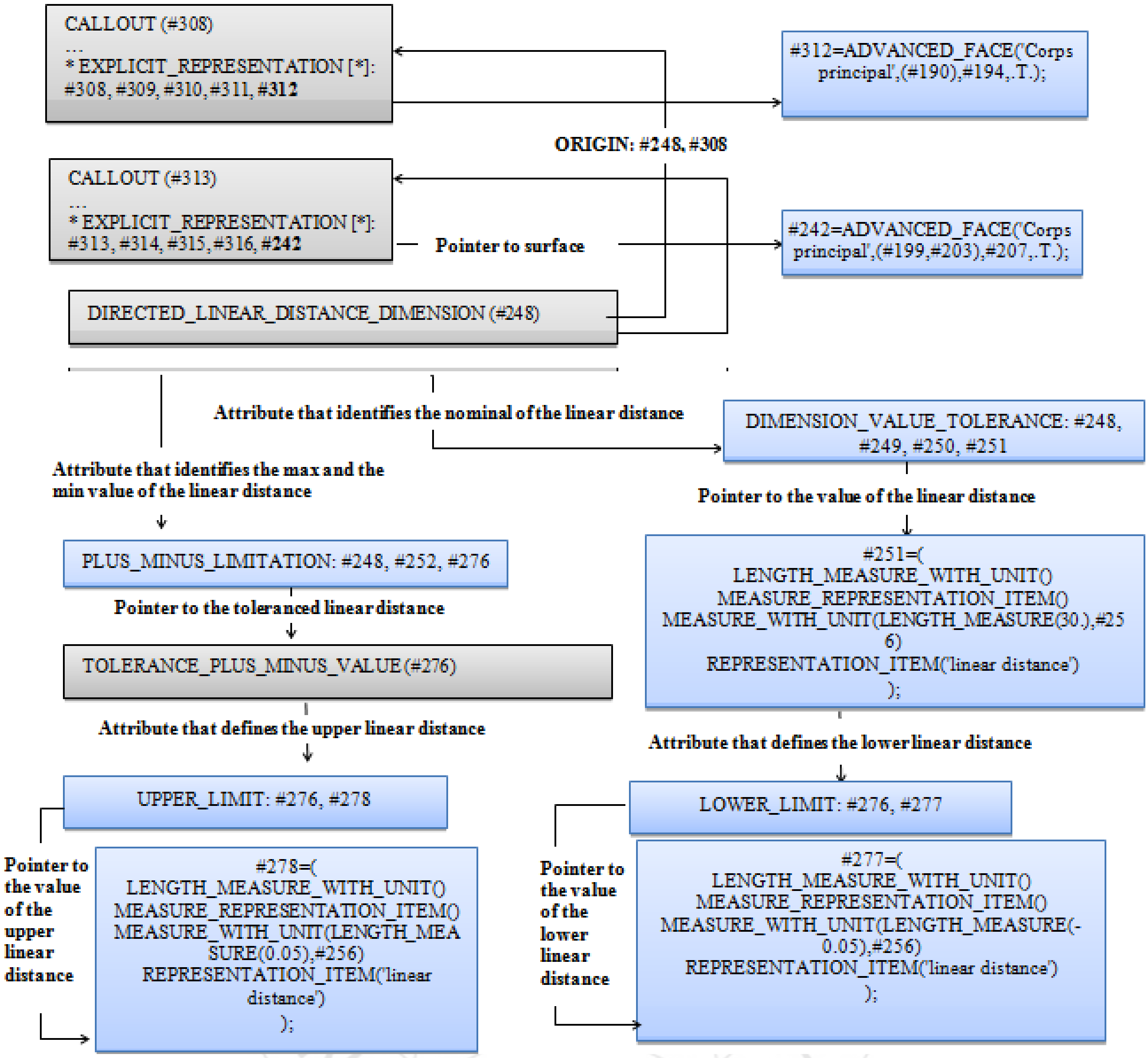

Figure 4: Definition of DIRECTED_LINEAR_DISTANCE_DIMENSION properties in STEP AP238

The DATUM_DEFINED_BY_FEATURE (#236) function defines implicitly the name of the datum andthe face for which the reference datum A is applied to, which is F242.It has an attribute denoted as DATUM_NAME: #236,['A'] that defines the name of the datum, and a secondattribute denoted as DEFINED_BY: #236, #237, #238. where #236 is a pointer to the datum A, denoted as #236=DATUM('','',#19,.F.,'A');and #238 is a pointer tothe identifier of the reference face F242 .

Thus, following these indicators brings us allocate the linear distance dimension, its nominal value and the concerned advanced faces using defined functions. The feature recognition module recognizes each feature and represents it as an object, for which the ADVANCED_FACE (that compose the feature) are known. This leads us to specify the feature for which the linear dimensional distance has been applied to.However, there is no association between the feature and the tolerance in term of objects. In fact, STEP Tools Library used in this module [11], generates an oriented object classes based on EXPRESS schema; the **outer_diameter_to_shoulder**class doesn't have an attribute representing the tolerance applied to the concerned feature.

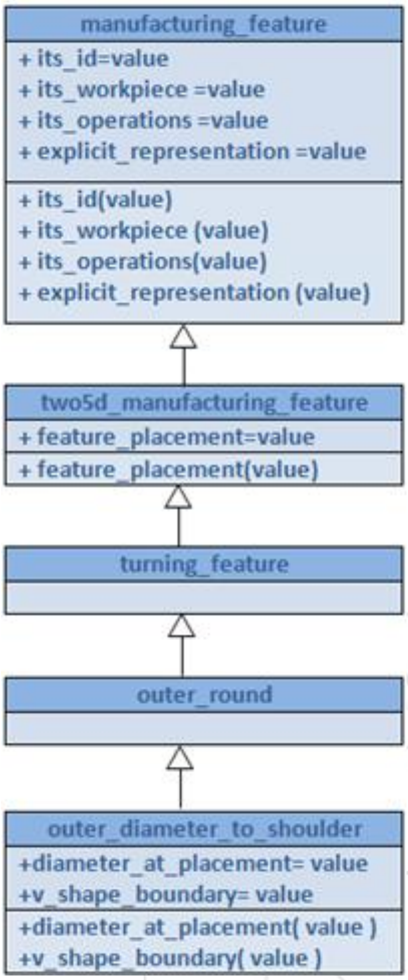

Figure 5: The UML diagram of the class **outer_diameter_to_shoulder**

Figure 5 shows the class representing the **outer_diameter_to_shoulder**feature according to **ISO 14649**, and the upstream classes from which the**outer_diameter_to_shoulder**class inherits attributes.
As shown in figure 5, the **outer_diameter_to_shoulder**class doesn't have an attribute that represents the tolerance applied to the concerned feature.

The solution proposed in this paper concerns the creation of an association between the parent class of the recognized features and the classes representing the tolerance, which means that an instance of the class **outer_diameter_to_shoulder**for example is composed of zero or a list of instances (zero-to-Many Mapping) for each type of tolerance, whether it is a Linear Distance Dimension or another tolerance type defined in STEP AP238.

Figure 6 shows the new association between turning_feature parent class and the new classes that can represent any type of tolerance. Taking the fact that tolerances classes have become part of the turning_feature class, each manufacturing feature (for example the outer_diameter_to_shoulder class) for which the turning_feature class is parent will inherits tolerances as objects.

After describing how surface roughness is embedded in the STEP AP238 data file and how tolerances can be associated to the related manufacturing features, the parameters representing tolerances will be searched and located using defined methods.
Figure 7 show the class **outer_diameter_to_shoulder**containing the concerned tolerances inherited from the **turning_feature** parent class.

It must be noticed that **DIRECTED_LINEAR_DISTANCE_DIMENSION**has been taken just as an example; the new inherited classes can also handle other types of tolerances defined in STEP AP238 such as, roughness, geometric tolerances and Datum. The input file that has been taken in this example will just consider the geometric dimension tolerance (Linear Distance Dimension).

Therefore, the main tasks in GD&T extraction and features/tolerances association are to extract, and instantiate then associate between each manufacturing feature and its related tolerance as an attribute.

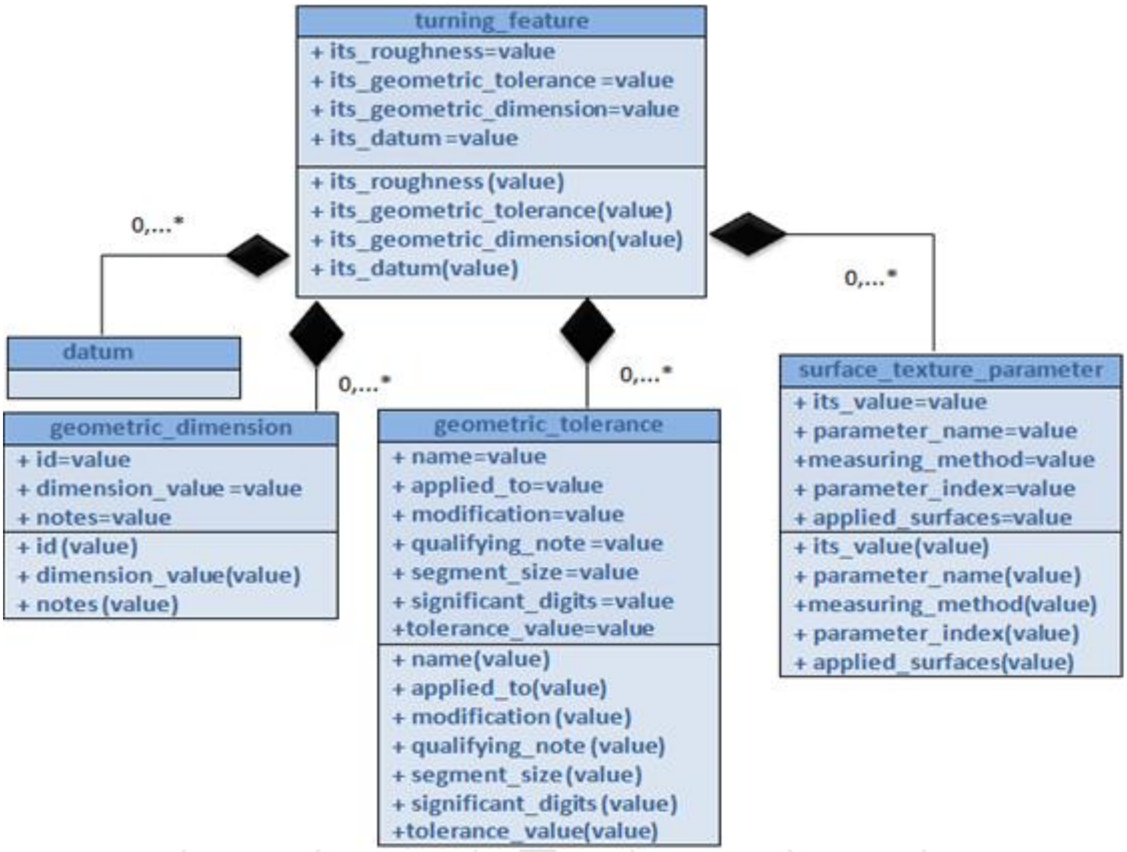

Figure 6: New association between **turning_feature** and related tolerances

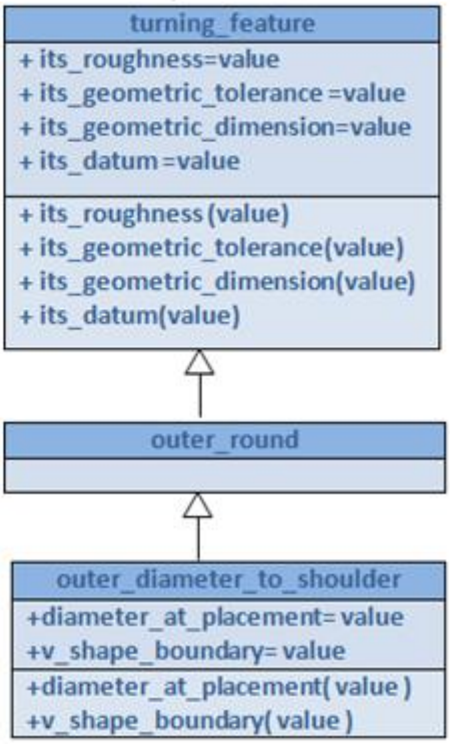

Figure 7: The new association representing the **outer_diameter_to_shoulder** class that inherits the tolerances type

Figure 8 illustrates the flowchart of the algorithm established for the **DIRECTED_LINEAR_DISTANCE_DIMENSION** extraction from STEP AP238, and its association with the corresponding feature. For the positional directed linear distance dimension, the algorithm is divided into a three-stage process.

In the first stage, the positional **linear_distance_dimension, the callout target** and **the callout origin**are located.

The second stage consists in instantiating a new object of type **linear_distance_dimension**. The last stage consists in associating the **linear_distance_dimension**to the corresponding manufacturing feature.

Taking the fact that the ADVANCED_FACEs that compose each feature are known, the new tolerances objects will be then associated to the corresponding feature by applying the new association method explained in this work.

This solution will lead us to get objects inherited from **turning_feature**, and representing tolerances related to the corresponding manufacturing features.

4. Machining Applications

As shown in figure 2, the machining process transforms the feature from a raw state to a finite state.The transition between these two states takes place through the intermediate states generated by the machining applications (roughing, semi-finishing, finishing).

Depending on the dimensional, geometric and surface condition requirements, the machining applications can be determined for each entity.

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

The Dimensional tolerances are imposed on the surfaces of a part to ensure functional and assembly requirements. The smaller the number of dimensional tolerance, the tighter the tolerance, and more likely a finishing pass is required.

Table 1 provides the indicative information between the tolerance interval, the surface condition and the number of machining applications.

Table 1: Relationship between tolerances and number of operations

	Tolerance interval			*Quality*			*Roughness*		
Characteristics	TI ≥ 0.3	TI > 0.05 and TI <0.3	TI ≤ 0.05	> 12	8, 9, 10, 11	6.7	Ra ≥ 6.3	Ra > 0.8 et Ra < 6.3	Ra ≤ 0.8
Applications Machining	F	R F	R 1/2F F	F	R F	R 1/2F F	F	R F	R 1/2F F

The number of operations for each feature can be retrieved from the input file and according to Table 1 as shown in Table 2.

Table 2: Number of operations related to each feature recognized

Features	***Applications***
E1(Cylindrical surface)	**R** (Roughing)
	½F(semi-finishing)
	F(finishing)
E2(Outer Shoulder to diameter)	**R** (Roughing)
	½F(semi-finishing)
	F(finishing)
E3(Face)	**R** (Roughing)
	½F(semi-finishing)
	F(finishing)
E4(Chamfer)	**F**(finishing)

Feasible applications generator determines automatically and firstly all machining applications for the concerned recognized features. The generation process is based on information provided by Table 1 which can be structured into the knowledge database that represents a centralized repository to store all necessary information including precedence constraints, rules and tolerance interval

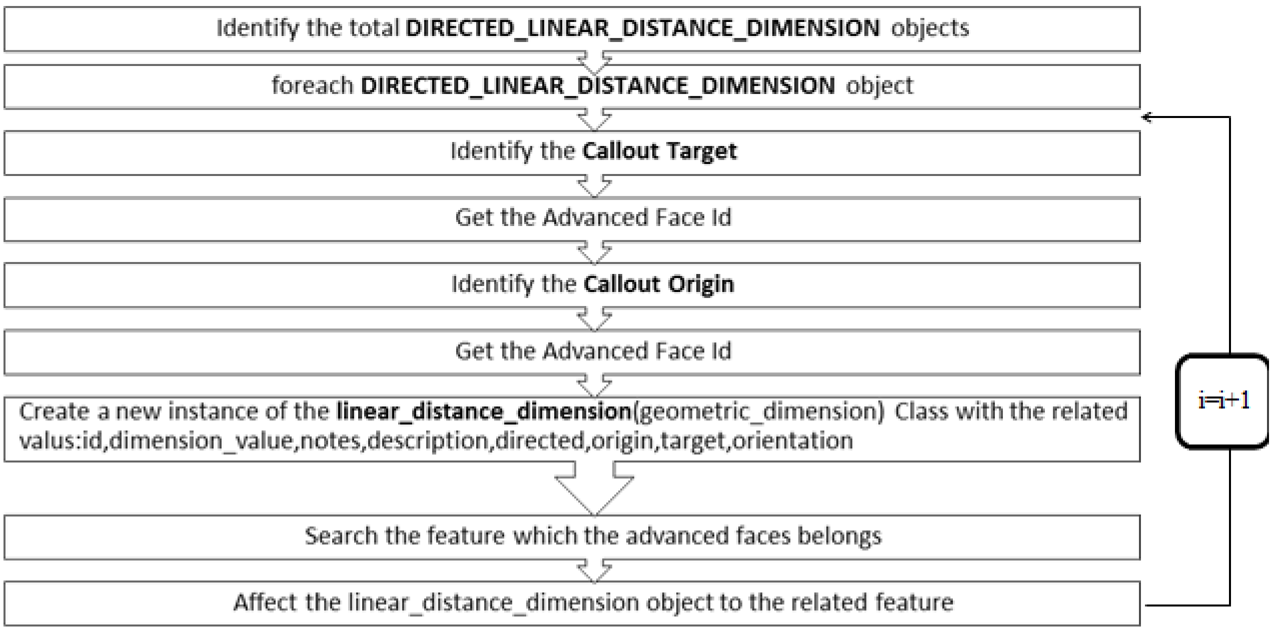

Figure 8: Flowchart of the association between the **linear_distance_dimension** object and its related manufacturing feature

information. Information about rules and constraints can be provided by the knowledge representation module which receivesall this information from a human expert and a knowledge engineer.

5. Precedence Matrix

The scheduling of the machining applications required for the manufacturing of parts is a key point in the process of automatic planning of machining processes. The determination of the scheduling consists in defining a sequencing of the machining applications of the entities while respecting the prior constraint. Extrinsic machining tolerances between feature surfaces influence sequencing of entity applications:

The parallelism geometric tolerance indicates that the feature E2 is machined taking the reference datum as setup.

For dimensional tolerance, the user has defined this tolerance of such a spell as the feature E2 is positioned with respect to the feature E3, which means that the feature E2 is earlier than the feature E3. Thus, according to Table 1, intrinsic tolerances affect the number of machining applications that can be defined for each feature. For example, the feature E3 has a roughness of 0.8, which means that the feature required three applications of machining, roughing, semi-finishing and finishing. This implies that the execution of this feature takes place in a logical sequence of machining mode: roughing, semi-finishing and finishing. The most widely used method of establishing the order of realization of the applications of the features of the part is based on the use of a precedence matrix. This matrix is of n rows and n columns, where n represents the total number of applications to be performed. By representing the priorities of the applications in this matrix, the generated results make it possible to determine the "machining levels" hierarchizing the various applications according to their order of realization, taking into account the precedence constraints.

The following method allows us to construct the matrix N of the machining levels. It is sufficient to count in each row of the matrix the number of priorities for the application in question, we report the summations of each row in the matrix N.The application which requires no priorityis executed first in the matrix Level N1.

The column of this application in the matrix is deleted, then we carry out the sum again and then we complete the matrix N.

The procedure is restarted until the columns of the matrix are completely deleted. Table 3 shows the prior matrix corresponding to the example piece of figure 2. Then, from the data in Table 3, the first level (N1) has machining applications having the same machining sequence (they have no prior constraints). Then in this case, the choice of the order of the applications in this level depends on the experience of the planer, and is done taking into account the prior constraints such as technological and economic ones.

The main strategy of the algorithm is the detection of feasible sequences, for a given set of operation.A precedence constraint comprehends the requirement that an operation j cannot be started before another operation i has been completed.
Thus, according to the result of the precedence matrix, the possible combination is concluded from the following formula:

Feasible sequences=$N_1!*N_2!*.....*N_n!$ where Ni is the number of operations in aMachining level i.
Therefore, the possible sequence number of the example provided will be: 4!*3!*2!*1! = **288.**
This shows that the input part of figure 2 has 288 machining possibilities which may be:

- 1R->2R->3R->4F->1.1/2F->2.1/2F->3.1/2F->1F->3F->2F
- 2R->1R->3R->4F->1.1/2F->2.1/2F->3.1/2F->1F->3F->2F
- 1R->3R->2R->4F->1.1/2F->2.1/2F->3.1/2F->1F->3F->2F
- 3R->2R->1R->4F->1.1/2F->2.1/2F->3.1/2F->1F->3F->2F
- 2R->3R->1R->4F->1.1/2F->2.1/2F->3.1/2F->1F->3F->2F
- 4F->3R->2R->1R->1.1/2F->2.1/2F->3.1/2F->1F->3F->2F
- 1R->4F->3R->2R->1.1/2F->2.1/2F->3.1/2F->1F->3F->2F
- …..

Table 3: Precedence matrix of the example par of figure 2

		Matrix A										*Matrix N*			
		E1			**E2**			**E3**			**E4**	Machining level			
		Previous machining applications													
		1R	1.1/2F	1F	2R	2.1/2F	2F	3R	3.1/2F	3F	4F	**N1**	**N2**	**N3**	**N4**
E1	1R											0			
	1.1/2F	*										1	0		
	1F		*									1	1	0	
E2	2R											0			
	2.1/2F				*							1	0		
	2F					*				*		2	2	1	0
E3	3R											0			
	3.1/2F							*				1	0		
	3F								*			1	1	0	
E4	4F											0			
Machining applications to be executed in each level												1R 2R 3R 4F	1.1/2F 2.1/2F 3.1/2F	1F 3F	2F

6. Feasible Sequences Generator

The outgoing result is the result of an algorithm which allows from the input modules to generate all the possible sequences.All this is done from an algorithm explained in the flowchart of figure 9

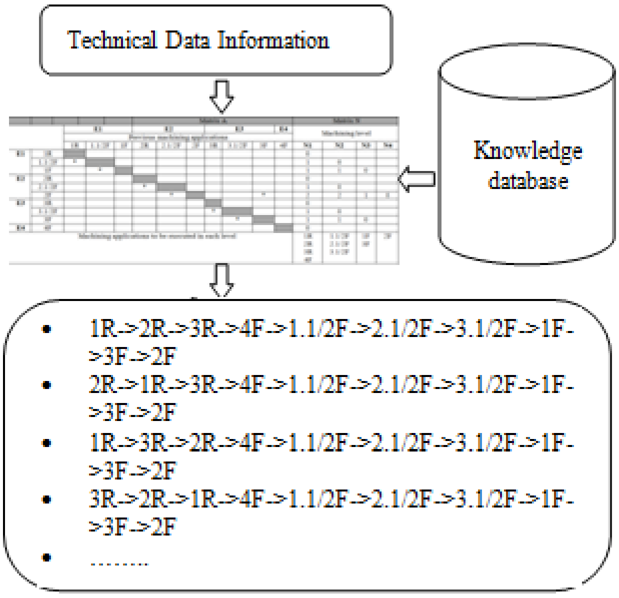

Figure 9: Flowchart describing the algorithm of the feasible sequences generator

The technical data information provided by the input moduleswhich have been described clearly above are taken by the feasible sequence generator module that interacts with the database in order to request information about tolerance intervals, number of operations, and other data that may be of interest to generate the matrix of precedence, and from this last it generates all possibilities of machining sequences.

7. Conclusion

In this paper, we have developed a new approach to extract tolerances from a step ap238 file, and then associate them with their related machining features. An algorithm was implemented and embedded into an expert system called ICAPP-TURN to extract and map tolerances data for process planning. The main task is to locate each tolerance and associates it with the related feature.

The data integration module described in this paper contains the GD&T extraction methods from a step ap238 data input file. These methods locate the linear distance dimension, its value and the advanced face to which the linear distance dimension is applied using custom defined functions.

The data representation module creates the mapping of the related machining features and tolerances to the oriented objects, then makes associations between these objects based on a new approach for conception and implementation of object oriented that improves the reusability , flexibility and the relationships between the machining features and their tolerances as properties. These modules represent an input data for ICAPP-TURN system.This data is used by the feasible sequences generator that interacts with the knowledge database in order to generate all possibilities of machining sequences.

Thus, in a future work, we will describe the knowledge representation module which provides the knowledge to the centralized database. This module receives the input knowledge about rules and constraints, and then represents this information in a form that ICAAP-Turn can utilize to store data and resolve tasks.

References

[1] Antonio Armillotta, A method for computer-aided specification of geometric tolerances, Dipartimento di Meccanica, Politecnico di Milano, Via La Masa 1, 20156 Milano, Italy, Computer-Aided Design 45 (2013) 1604–1616.
[2] Mehmet I. Sarigecili ,Utpal Roya, Interpreting the semantics of GD&T specifications of a product for tolerance analysis, Computer-Aided Design 47 (2014) 72–84.
[3] OntoSTEP Plug-in. July, 2012. http://www.nist.gov/el/msid/ontostep.cfm.
[4] ISO_10303-11, Industrial automation systems and integration — Product data representation and exchange —Part 11: Description methods: The EXPRESS language.
[5] ISO_10303-21, Industrial automation systems and integration — Product data representation and exchange —Part 21: Implementation methods: Clear text encoding of the exchange structure.
[6] Pei Lei, Lianyu Zheng, A STEP-NC compatible on-machine measurement system with automated correlation of inspection data. Applied Mechanics and Materials Vols. 397-400 (2013) pp 1772-1777
[7] HakkiEskicioglu, The Use of Expert System Building Tools in Process Planning, Middle East Technical University, Ankara, 1992, Engineering Applications of Artificial Intelligence. Vol. 5, No. 1, pp. 33-42, 1992.
[8] Jaider Oussama, al. Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 4, Issue 5(Version 1), May 2014.
[9] STEP Application Protocol (AP) 203 editions 1 & 2, Configuration Controlled 3D Designs of Mechanical Parts and Assemblies, ISO10303 1994, (2007) & (2009).
[10] http://step-nc-machine.software.informer.com/9.1/
[11] http://www.steptools.com/.
[12] Jean-Pierre Cordebois et Coli, "Fabrication par usinage", livre, édition 2, Dunod, Paris, 2003.