Automated Inspection Process Planning: Algorithmic Inspection Feature Recognition, and Inspection Case Representation for CBR

F. S.Y. Wong* , K.B. Chuah and Patri K.Venuvinod

Department of Manufacturing Engineering and Engineering Management,

City University of Hong Kong

*** Author for correspondence**

Mr. Francis Seung-yin Wong

Department of Manufacturing Engineering and Engineering Management,

City University of Hong Kong,

Tat Chee Avenue,

Kowloon,

Hong Kong

E-mail:sf8088@netvigator.com

Revised : September, 2004

Automated Inspection Process Planning: Algorithmic Inspection Feature Recognition, and Inspection Case Representation for CBR

S.Y. Wong , K.B. Chuah and P.K.Venuvinod

Department of Manufacturing Engineering and Engineering Management,

City University of Hong Kong

Abstract

General metrological inspection planning is among the least explored CAPP domains. This paper explores certain basic issues involved in inspection planning using case-based reasoning in an environment of a Generic CAPP Support System. Firstly, algorithmic methods for characterizing and extracting inspection features are proposed and discussed. A sequential knowledge based filtering method is developed to reduce the number of inspection features typical encountered in metrological inspection planning. Finally, a formalized approach for case representation of relevant inspection domain knowledge using a newly developed parametric-list technological feature graph (PLTFG) is presented.

Key words: Computer-aided process planning, Computer-aided inspection process planning, Inspection Feature Recognition, Case-based reasoning

1. Introduction

Concurrent engineering has emerged as a commonly accepted solution to the sharply decreasing time-to-market problem in discrete product industry. In practice, the effectiveness of concurrent engineering depends critically on the availability of highly automated and reliable computer-aided process planning (CAPP) tools. As a result, a variety of algorithmic and artificial intelligence-based (AI-based) methods have been developed in the last two decades to address different problems arising in CAPP [1, 2, 3, 4, 5, 6, 7, 8, 9].

The manufacture of any part/product requires the execution of several manufacturing processes: machining, casting, injection moulding, inspection, etc. Under each of these process classes, there are several sub-processes each with very distinct characteristics. For instance, machining may involve a variety of operations such as turning, milling, and drilling carried out on separate machines or on a single machining centre. Likewise, inspection could be carried out through a sequence of operations each carried out with the aid of a different metrological instrument or on a single coordinate measuring machine (CMM). Further, one needs to evaluate several competing manufacturing processes at each stage in process planning. All this suggests that there is a strong need for a *comprehensive* CAPP system that encompasses most of the commonly found manufacturing processes in a seamless and integrated fashion. However, it appears that most that, in general, the currently available CAPP solutions do not meet this criterion. Firstly, the systems are very sparse, i.e., they address very few of the commonly used manufacturing processes. In particular, while machining and assembly have been extensively explored, many other manufacturing processes and inspection (with the exception of inspection using a CMM) have received very little attention. Secondly, since the planning of a manufacturing process requires reasoning over a great deal of process-specific technological knowledge, CAPP has generally become highly fragmented with each process being addressed by a totally independent module. As a result, the system modules/elements become highly redundant.

The issue of fragmentation of CAPP was brought into focus recently by the writers' team in [10]. It was suggested that, while each CAPP domain might be distinct in terms of the technological knowledge, the competing manufacturing processes at each stage of CAPP have one thing in common - they have to perform in with almost the same part/product objective. It should therefore be possible to aggregate the reasoning processes related to diverse manufacturing processes into a common platform called the Generic CAPP Support System (GCAPPSS)—see Figure 1— that precedes the individual process specific modules. It was also suggested in [10] that the downstream process-specific modules could be integrated by adopting a common technological reasoning strategy and that case-based reasoning is a good candidate in this regard.

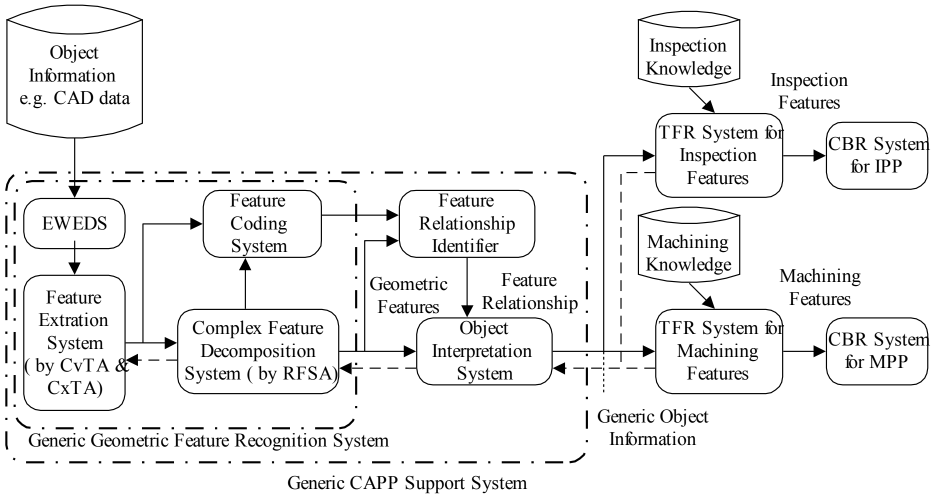

Figure 1 The role of GCAPPSS in comprehensive CAPP [10]

However, the implementation of the strategy outlined in Figure 1 requires that we already have a fair understanding of the product-based reasoning strategies required in various CAPP domains. Unfortunately, the present state of CAPP is such that the degree of understanding of is quite uneven across different CAPP domains. Amongst the most studied seems to be the domain of machining. Among the least studied seems to be the domain of inspection.

This paper aims to provide a deeper understanding of the product-based reasoning strategies required in implementing a computer-aided inspection process planning

(CAIPP) system based on the notion of GCAPPSS. We will focus mainly on dimensional inspection of parts containing polyhedral and cylindrical features that need to be inspected by using a variety of commonly found metrological equipment. The next section will present an overview of the literature related to CAIPP. This will be followed by a brief review of dimensional inspection. The intention is to arrive at a reasonably complete and generalized set of observations regarding dimensional inspection that will lead to the characterization of inspection features in a logical manner.

2. Inspection Process Planning

An inspection process planning system in the area of "Inspection" for CAPP applicable in a CIM environment needs to include automated or semi-automated modules or systems capable of performing the following tasks:

i. Identifying and recognizing the inspection features.

ii. Identifying and recognizing the associated inspection constraints for the inspection features.

iii. Recommending an appropriate inspection method for each inspection feature.

iv. Integrating the various individual inspection operations into an effective and efficient overall inspection plan.

Although much of the inspection carried out in industry continues to be conducted using conventional metrological equipment, most previous work on CAIPP has been directed towards inspection operations performed on coordinate measuring machines (CMM). For instance, there were seven basic types of CAIPP systems reported in Juster [11]. Significantly, all the seven types were directed towards CMM-based inspection. Likewise, the majority of the subsequent CAIPP developments were also directed towards CMM-based inspection: probe accessibility and orientation for prismatic parts [12]; optimum determination of measuring points and the associated paths, pre-hit distance, and probe collision prevention [13]; quick turnaround cell (QTC) inspection planner based on a feature-based part model [14], etc.

The CAD model is essentially a specification of the geometry of the part. This model needs to be interpreted for the purpose of inspection process planning. Hence, as with any process planning domain, automated geometric feature recognition (GFR) is an essential requirement of CAIPP. The problem of GFR has attracted a great deal of attention of researchers over the last few decades [15, 16, 17, 18, 19, 20, 21, 22, 23].

It should be clear from the above that, while the process of recognizing a given geometric feature may be largely technology independent, the process of deciding *what* specific features need to be recognized is essentially technology-dependent. It is also clear that, with the exception of inspection processes performed on a CMM, the issue of recognizing geometric features of interest in the context of inspection using common metrological equipment has so far received very little attention. This paper aims to fill this gap. The specific scope includes the characterization of inspection features and the development of a method for automated extraction of inspection feature from the viewpoint of dimensional inspection of prismatic parts with polyhedral and cylindrical features. The inspection feature recognition process is carried out in an environment of GCAPPSS as illustrated in Figure 1.

3. Case-based Reasoning

Rule-based expert systems have once been recognised as the most successful AI technique for provision of AI researches problem-solving knowledge as models which could be processed computationally. However the rule-based system is found to be suffering from a serious problem described as "Knowledge-elicitation Bottleneck" by Waston [24]. Watson summarized the four causes of "Knowledge-elicitation Bottleneck" as i) availability of the rule-based system, ii) availability of the knowledge engineer of the system, iii) ability of the knowledge engineer to the problem and iv) ability in knowledge representation scheme.

Of the various AI techniques, case-based reasoning (CBR) seems to be particularly suited in the present context. CBR is a problem-solving paradigm that uses previous episodes of solving problems similar to the problem at hand (the new problem) as the basis for solving the new problem. CBR has gained prominence in many fields (including the field of design) in recent years. Schank and Jona [25] describes the nature of CBR as follows: *"Most people prefer not to have to think hard if they can help it. They will try to get by with whatever worked before, even if it is less than optimal. We believe that, roughly speaking, people's everyday cognition consists of abut 90% retrieving of past solutions and only about 10% or less of actual novel problem solving. Because of our belief about the relative importance of retrieval, it follows that if one wants to understand what it makes to model human intelligence, one should focus on the type of processing that contributes the most to people's everyday behavior, namely retrieval and adaptations of old solutions."*

The implementation of any CBR system requires one to define the basis on which the cases in the case library would be unambiguously delimited. Like the expert system approach, CBR relies on explicit symbolic representation of knowledge bases derived from past experience. However, the expert system uses past knowledge mainly to derive a set of general heuristics that are stored as production rules for solving new problems and stored the past knowledge in some specific schemes involving clauses, frames, semantics, and/or objects. In contrast, CBR uses representations of specific episodes of past problem solving exercises to learn to solve a new problem. In short, a CBR system stores past experiences in individual problem solving episodes whereas expert systems store past experience as generalized rules and/or objects. Five further additional advantages of CBR can be identified:

i) Heuristic analogy: When problem is complicated and difficult, using the successful solution to a previously addressed similar problem could allow one to by-pass the resolution process [26,27].

ii) Last resort analogy: When a complete theory or a model is not available for a solution of a problem, a proven solution of a similar problem is the last resort [28, 29].

iii) Avoidance of repetition of past mistakes: Failure records as well as success records are present in the case. A failure record will be taken as a caution for the repeat of a past mistake.

iv) Self-learning: New case with proven solution is added to the case library. The new case provide learning through additional knowledge to the system.

v) By-pass of knowledge acquisition: In CBR system the knowledge is not necessary in the knowledge-based expert system.

The result of the current problem is stored in the system's memory as a new case that can be reused in future problems. CBR has been applied to many different domains, such as design, planning, cooking, medicine and law. CBR is a knowledge-based technique that has gained prominence in many different domains and fields , such as design, planning, cooking, medicine and law in recent years. Of these, none has utilized case-based reasoning in inspection process planning. This research study appears to be the first ever attempt to adopt the CBR approach to develop a generative inspection process planning system.

4. Dimensional Inspection Features and their Extraction

Dimensional inspection features can be of four basic types: *external*, *internal*, *offset,* or features requiring coordinate (or profile) measurement of a curved surface. An *external inspection feature* is a pair of faces whose face normals directed away from the material side at the probe points are parallel but are directed away from each other. For example, the face-pair f12/f4 of a setting gauge for illustration in Figure 2 forms an external inspection feature.

An *internal inspection feature* is a pair of faces whose normal vectors directed away from material side at the probe points are parallel, but are directed towards each other. For example, the face-pair f12/f14 forms an internal inspection feature.

An *offset inspection feature* is a pair of faces whose normal vectors directed away from material side at the probe points are directed similarly. For example, the face-pair f1/f7 in Figure 2. forms an offset inspection feature.

The fourth class of inspection features subsumes inspection features consisting of curved surfaces (of the standard or freeform type) that need to be (i) probed at several points or (ii) compared with tailor-made special template/profile gauge section by section or (iii) measured by special profile tracing instrument such as Form Talysurf of Taylor Hobson so as to evaluate the dimensional conformance of the surface with the form intended by the “Design Specifications”. Figure 2 includes one such feature, f17.

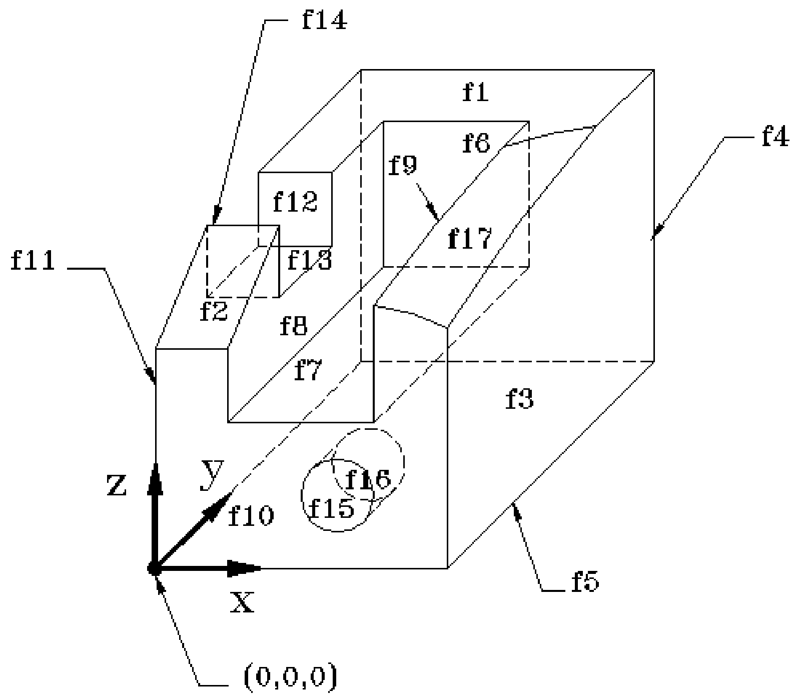

Figure 2 Test Piece-A Setting Block

The following syntax is developed in the present work for specifying an inspection feature:

Inspection_feature (Inspection_feature_ref_No,

Inspection_feature_class, FaceList-of-the-Two-Measuring-Faces,

Face_List_of_the_Boundary_Faces_of_the_First_Measuring_Face,

Face_List_of_the_Boundary_Faces_of_the_Second_Measuring_Face).

The following explanations should be useful. The "*first measuring face*" may serve as the datum face during probing, setting and alignment. For instance, if a measurement is to be performed by means of a comparator, the first face may be used for the 'zero' setting. Alternatively, it may be used for seating the anvil of a depth gauge. The *"second measuring face"* can then be taken as the *target* face during probing. A *"Face_List_of_the_Boundary_Faces_of_the_First_Measuring_Face"* consists of a list of boundary faces of "*the_First_Measuring_Face".* The boundary faces determine the location system, the supporting method, alignment method, other constraints, .etc. A "*Face_List_of_the_Boundary_Faces_of_the_Second_Measuring_Face"* consists of a list of boundary faces of "*the_Second_Measuring_Face".* The boundary faces determine the accessibility of the probe or measuring head, its path, fixturing, alignment method, other constraints, etc.

A technique based on a new concept called the Multi-Attributed Spatial Graph (MASG) and algorithm was developed by the authors [30] for the extraction and recognition of an inspection feature.

5. Knowledge-based '*Filters*' in Inspection Process Planning

Suppose that the above algorithm has recognized *n* dimensional features for inspection. Let n_c be the total number of cylindrical features (full cylinders, not partial cylinders which are treated as curved surfaces) and free form surface out of the n features. If every one of the n dimensional features were to be inspected, the complete inspection plan for the part would consist of n inspection processes. More importantly, there are $(n- n_c)$ polyhedral inspection features each of which will have an image of its own. As a result, the n inspection processes can be organized in $n\ [_{(n-nc\ )}P_{(n-nc)/2}]$ ways.

To appreciate the magnitude of the problem at hand, consider the results obtained from the application of our inspection feature recognition algorithm [30] to the Testing piece shown in Figure 2. In this case, our algorithm (coded in PROLOG) automatically identifies 64 inspection features in the test part of which one is cylindrical, so n=64 and n_c=2. This means that, although the part has just 17 faces, there are 64 * ($_{62}P_{31}$) = 2.44937×10^{53} different ways of organizing the inspection process plan! In practice, it is not uncommon to encounter parts with hundreds of faces. Clearly, the problem is intractable if a purely algorithmic approach were to be pursued. Of the enormous number of possible inspection plans, we need to select the most desirable single process plan based on a range of technological and practical considerations. This would require us to draw upon much technological knowledge and human expertise.

6. Filters for General Engineering Application

We will now recommend a preliminary set of eleven potentially useful filters for general engineering purposes. We call it "Filter Set of Eleven" which is capable of handling general engineering application.

(i) Product specifications filter:

The specifications of a part provide the information necessary for a process planner for decision of inspection of an inspection feature. International Standard ISO406 [31] specifies standard tolerance symbols and also sets out methods for the indication of tolerances on drawings of assembled parts.

(ii) Domain filter

The domain of application of a part demands special attention to certain inspection features. The outside diameter, for example, is an important inspection feature of a dowel pin.

(iii) Application filter

The application could be a critical factor sometimes. For example, the outside diameter of the spool of a direction valve is likely to be invariably inspected. In contrast, the diameters of the reservoir grooves are unlikely to be important.

(iv) General practice filter

Some inspection features need not be inspected once the tooling has been approved, e.g., the size of a name plate, and the diameter of a boss of a pre-approved plastic part, but some features are important and must therefore be inspected.

(v) Trade practice filter

Sometimes there are certain tolerance or accuracy requirements of a part specific to a particular trade domain. For instance, ISO2768 [32] specifies the permissible machining variations in dimensions without tolerance indication.

(vi) Process capability filter

If it is known in advance that the manufacturing processes leading to particular class of critical dimensions are not well controlled, one naturally needs to inspect that class of inspection features.

(viii) Role/task filter

Some inspection features that are critical for the life, matching, assembly, performance and safety will require special attention and inspection. For example the size of a shaft might need to be produced to a desired diameter in order to match with the bore of a journal bearing.

(ix) Special attention filter

Engineers learn from experience. Some inspection features which have bad failure records, etc. need special attention and inspection. Sometimes, the customer may demand that special attention is given to some inspection feature.

(x) Customer filter

Different customers may have different requirements for the same product in view of the specific usage requirements, the location of the customers, the government regulations at the customer's location, safety requirements, etc.

(xi) User (manual) filter

For some products, the dimensional tolerance and quality inspection constraints could be varied to suit different customer segments, e.g., those for Germans versus the Chinese.

The above set of eleven filters is not exhaustive but thought to be sufficiently extensive to cover general trading, engineering and real life application. This list is a useful reference of the end-user, who can build a set of similar filters based on his own expert knowledge and experience.

6.1 Development of Filters

Filters for the "Inspection of the Setting Gauge" as an illustration were described in this section. The set of filters is a sub-set of knowledge based system. Production rules are the main cores of the system.

The rules are developed from gained from past experience and first expressed in day to day language. A typical example is the set of rules for “Product Specification Filters” described as follows:

Rule for Product Specification Filter:

An inspection feature is necessary to inspect,

if its tolerance zone is less than or equal to ± 0.2mm.

After a rule in human language has been verified, it can be then coded, say in PROLOG. The Prolog coding of the above production rule for the “Product Specification Filter” is as follows:

```
/*Filter ("Product_Specifications")*/
/* Tolerance <=0.2mm*/
n_if(IF_N,filter("Product_Specifications")):-
inspection_feature(IF_N,_,[Face1,Face2],_,_),
dim([Face1,Face2],nominal(_),tolerance(T)),
T<=0.2.
```

The input conditions can be extracted from the design blue print by process planners in the case of a semi-automatic system. In the case of a fully automatic system, the input conditions are extracted automatically by an intelligent feature recognition system.

Seven filters were developed and applied to the setting gauge as shown in Figure 2.

The filters are translated to production rules and coded in Prolog.

The seven rules written in day to day language are as follows:

Filter No. 1 Product Specification Filter

An inspection feature is necessary to inspect,

if its tolerance zone is less than or equal to 0.2mm.

Filter No.2 Domain Filter

An inspection feature is necessary to inspect,

if it is a hole.

Filter No.3 Application Filter

An inspection feature is necessary to inspect,

if it is an external feature and

upper limit=+0.1mm

lower limit=0.

Filter No.4 General Practice Filter

An inspection feature is necessary to inspect,

if its tolerance is less than or equal to 0.05mm.

Filter No.5 Trade practice Filter

An inspection feature is necessary to inspect,

if its tolerance is less than or equal to 0.02mm.

Filter No.6 Process Capability Filter

All parts with tolerance over 0.3 mm are under control.

An inspection feature is necessary to inspect,

if its tolerance is less than 0.3mm.

Filter No.7 Special attention Filter

All parts with nominal sizes over 30 mm need special attention.

An inspection feature is necessary to inspect,

if its nominal size is over 30 mm.

Full listing of the source code of above seven rules can be found in Appendix 2.

When applied to the part in Figure 2, this algorithm is able to filter off all the other features and extracts only 18 necessary inspection features and corresponding inspection feature images. Of these, 8 are external, 3 internal, 6 offset, and one free form surface (f17).

7. Representation of Inspection Process Plan Cases

The goal of this research work is to develop a case representation scheme as a base for the development of a CBR CAIPP system for inspection process planning. Currently we are not aware of any such representation scheme. This representation scheme should serve the following purposes:

i. To provide precise, adequate and accurate representation of previous cases in the case memory for development of a CBR CAIPP system.

ii. To enable explicit definition of the new inspection problem.

iii. To make the whole process suitable for CAPP.

iv. To able readability and processability by a computer.

v. To provision of an effective basis for indexing, retrieving, adapting and machine learning.

vi. To facilitate presentation of inspection measurement cases to the process planner as well as to computer.

vii To support the identification of the parametric elements of a case.

Our method is a two-level hierarchical case representation scheme. The base level of the case memory is the "Primitive Inspection Case" which represents the inspection process plan of one individual measurand. A graph of parametric attributes will be employed. Each attribute is parameterized as a variable. The aim of this level is to represent a whole operation of the inspection process of a single inspection feature. The top level of an inspection case is the sub-case level that represents the flow of inspection sequence, the relationship between the primitive inspection case and the routing of the inspection plan.

The scope of the present paper is limited dealing with prismatic parts with polyhedral, cylindrical features and free form surface features. The following issues will be addressed: (i) a representation scheme for a primitive inspection case, (ii) a methodology to represent a primitive inspection case in computer, (iii) a representation scheme for a sub case, (iv) a methodology to represent a sub case in computer, (v) a methodology to sub-divide the inspection case of a whole part into sub-cases, and (vi) a methodology to build the case memory in the proposed two level approach.

7.1 An Inspection Measurement Model

The goal of an inspection measurement of a given part is to evaluate the degree of conformance of the part to the specifications contained either explicitly or implicitly in the computer model(s) or drawing(s) supplied by the individual or team designing the part. Inspection necessarily involves a set of measurement processes where each process is directed towards an individual measurand (in the present case, a linear or angular dimension). Hence, if a part consists of n inspection features, $\{IF_1$ to $IF_n\}$, the part/product will be said to be fully conformed if every one of these *IF*s has met the corresponding specification. It is of course assumed that each measurand is measurable using the available equipment.

Clearly, an inspection plan is composed of an ordered sequence of individual measurement processes each of which is directed towards a specific and possibly distinct inspection feature. Inspection measurement of a single inspection feature in turn involves the evaluation of the compliance of the inspection feature against its specification. The specification of an inspection feature consists of a nominal value and a tolerance value (explicit or implicit). The measurement process itself needs to take account of the measurement conditions such as temperature, measuring force, etc. Further, each measurement process consists of an ordered sequence of sub-processes. Figure 3 illustrates one way of modelling an inspection process.

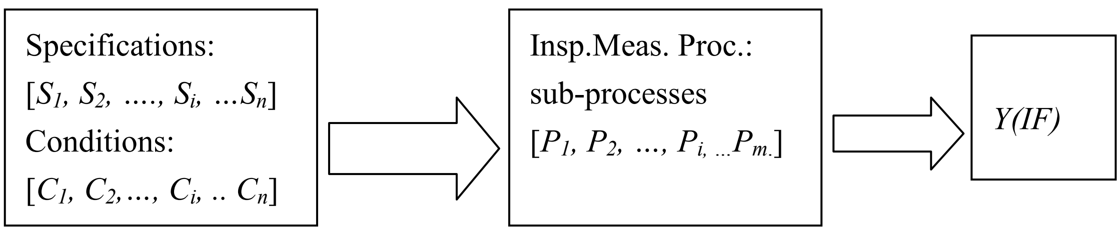

Figure 3 Model of an individual inspection process

Consider for example the offset inspection process described earlier for the setting block shown in Figure 2: *inspection_feature* (*3, ''offset", [1,7], [3,4,11,12,10,9,6,8], [10,9,6,8]).*

For this feature, we might have

- S(1) Measurement type: Depth, S(2) Nominal size: 30, S(3). Tolerance: + 0.2, -0.2.
- C(1) Temperature: 20°C, C(2) Temperature Limits: +1°C, - 1°C, C(3) Stabilzation Time: 8 Hours, C(4). Meaurement equipment: calibrated.
- P(1) Clean the part with alcohol and cloth, P(2) Stabilize the part at a temperature of 20°C, P(3) Temperature Limit during measurement is ± 1°C, P(4). Measurement equipment is a depth gauge, P(5) The resolution of the depth gauge is 0.01 mm, P(6). Use face 1 as datum face, P(7) Use face 7 as target face, P(8). Measure the depth of face 7 below face 1.

8. Representation of an Inspection Case at the Basic Level

An inspection process plan of an individual single measurand is defined as a Primitive Inspection process plan. A case referring to a specific inspection process plan directed towards a single measurand is defined as a "Primitive Inspection Case".

A parametric-list technological feature graph (PLTFG) is now proposed to facilitate systematic representation of a primitive inspection case. A PLTFG of a primitive inspection case is a graph $G = (N, A, PL)$ where

- N is a set of nodes representing the N technological features of a primitive inspection case. Here, the notion of technological features encompasses the technological data, information, equipment, etc associate with the primitive inspection case. Key words, action words or feature terms will be used to describe the technological features.

- *A* is a set of arcs representing the relationships between the technological features in the case.
- *PL* is a set of the parametric values of these relationships. The data structure of this parametric list is designed to properly fit the requirements and properties of the corresponding technological feature.

Table 1 shows the node and the associated technological features of the nodes for the primitive inspection case presently under discussion. Figure 4 shows the PLTFG for the primitive inspection case being discussed. Consider node “IF(3)” for example. Node “P(1)” is the technological feature “cleaning”. There is an arc between IF(3) ”inspection feature” and P(1) “cleaning” with the self-explanatory parametric list [alcohol, cloth]. The node “IF(3)” is a technological feature “inspection feature 3”. The node “P(1)” is a technological feature “cleaning”.

There is an arc between IF(3) ”inspection feature” and P(1) “cleaning” with parametric list of the arc is [alcohol, cloth]. It represents a cleaning process on the inspection feature by alcohol and cloth. Definitions of nodes and technological features associated with respective nodes of this primitive inspection case being discussed is shown in table 1.

Table1 Node and Technological Feature

Node	Technological Feature
IF(3)	inspection feature 3
C(1)	stabilisation_environment
C(2)	Stabilisation temperature limit
C(3)	stabilisation time
C(4)	calibration of equipment
S(1)	measurement type
S(2)	nominal size
S(3)	tolerance
P(1)	cleaning
P(2)	stabilisation
P(3)	measuring temperature limit
P(4)	measuring equipment
P(5)	resolution
P(6)	datum
P(7)	target face
P(8)	method

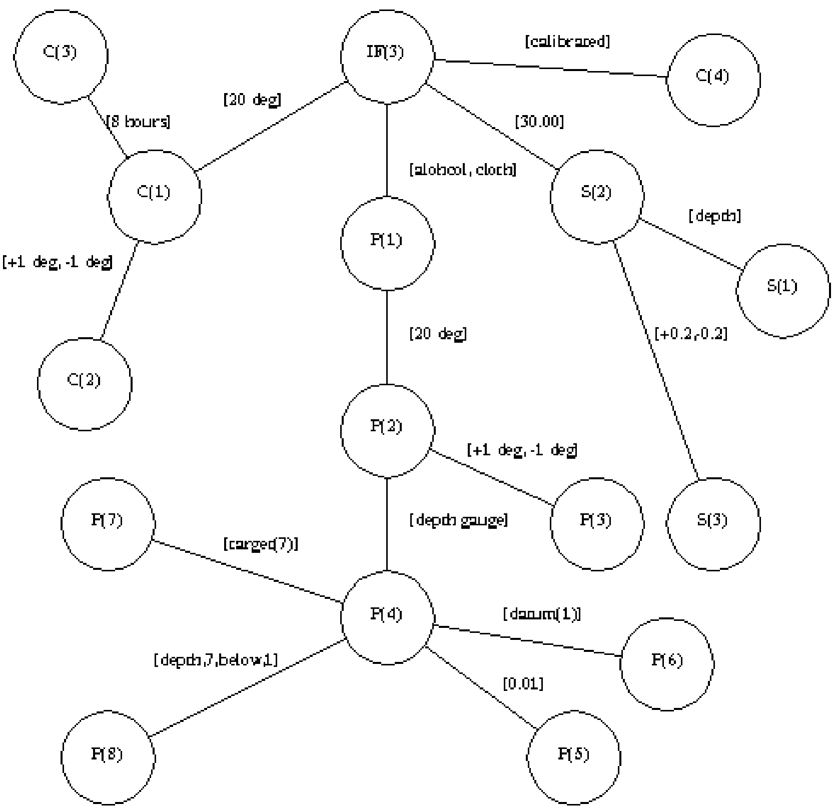

Figure 4 PLTFG of a primitive inspection case

The "Primitive Inspection Case" is coded in form of a predicate. The syntax is: *primitive_inspection_case(Part_reference, Inspection_feature_reference).*

The first argument of the predicate is the reference number of the part and the second is the predicate consisting of the reference number of the inspection feature so as to provide an index and reference mechanism for global indexing among the inspection cases of different parts in the case memory, e.g., *primitive_inspection_case (“A001”,if(3))*. The predicate works as a pointer for the whole primitive inspection case. The primitive inspection case is represented by PLTFG and coded in the semantic net which is readable by computer. The semantic net can either be a part of the computer programme itself or be a separate independent stand_alone data file. The primitive case will make the programme bulky in the first way of approach while the second approach provides flexibility in storage and will not increase programme size. The primitive case can be called up or loaded via interface procedure with the predicate ““Primitive Inspection Case”. The semantic nets of information of this primitive inspection case “*primitive_inspection_case (“A001”,if(3))”* consist of four parts : area (i) the inspection feature, area (ii) the structure of the primitive inspection case, i.e. the specifications, conditions and processes, area (iii) the definitions of the nodes in term of technological features, and area (iv) information related to the arcs. Thus, the syntax of the database of this primitive_inspection_case will be as follows:

/* Database of *primitive_inspection_case(Part_ref, Inspection_feature_reference)*/*

/* Database of “*primitive_inspection_case (“A001”,if(3))******************/*

/ Part 1: the inspection feature */*

inspection_feature (Inspection_feature_ref_No, Inspection_feature_class, *Face_list_of_the_Two_Measuring_Faces,*

Face_list_of_the_Boundary_Faces_of_the_First_Measuring_Face,

Face_list_of_the_Boundary_Faces_of_the_Second_Measuring_Face).

/ Part 2: Structure of the case representation */*

/ structure_of_case(Type_of_inf_1, Type_of_inf_2, Type_of_inf_3)*

structure_of_case(specifications, condition, process).

/ Part 3: Definition of the nodes in term of technological features */*

node(Node_ref, Technological_feature).

/ Part 4: information of the arcs */*

arc(Node1, Node2, Paramteric_list_of_relationship_between_Node1_and_Node2).

The database of "*primitive_inspection_case ("A001",if(3))   IF3* is as:

/ Database of*

*Primitive Inspection Case of inspection feature if(3) of Part No. A001" */*

inspection_feature(3, "offset", [1,7], [3,4,11,12,10,9,6,8], [10,9,6,8]).

structure_of_case(specifications, condition, process).

node(if(3), inspection_feature).

node(c(1), stabilisation_environment).

node(c(2), temperature_limit).

node(c(3),stabilisation_time).

node(c(4), calibration_of_equipment).

node(s(1), measurement_type).

node(s(2), nominal_size).

node(s(3), tolerance).

node(p(1),cleaning).

node(p(2), stabilisation).

node(p(3),measuring_temperature_limit).

node(p(4), measuring_equipment).

node(p(5), resolution).

node(p(6), datum).

node(p(7), target face).

node(p(8), method).

arc (if(3),s(2), [30.00]).

arc (s(2),s(1),[depth]).

arc (s(2),s(3),[+0.2,-0.2])

arc(if(3),c(4), [calibrated]).

arc(if(3),c(1), [20 deg]).

arc(c(1),c(3),[8 hours]).

arc (c(3),c(2),[+1 deg, -1 deg]).

arc (if(3),p(1), [alohcol, cloth]).

arc (p(1),p(2), [20 deg]).

arc (p(2),p(3),[+1 deg, -1 deg]).

arc (p(2),p(4), [depth_gauge]).

arc (p(4),p(7),[target(7)]).

arc (p(4),p(5),[0.01]).

arc (p(4),p(6),[datum(1)]).

arc (p(4),p(8),[depth,7,below,1]).

The above representation method has been found to be adequate for the 64 inspection features contained in the Test part in Figure 2. The above approach is capable of representing all the 18 inspection features found to be needing inspection by the filtering process described in Section 5 and 6. These primitive inspection cases may now be stored in the inspection primitive case library.

9. Representation of An Inspection Case at Sub-case Level

It is important to note the following heuristic criteria usually adopted in manual inspection process planning:

i. Minimize setup requirements by maximizing the number of measurements carried out with each setup.

ii. Wherever possible, perform measurement between the datum and target faces directly, i.e., unless there exist specific reasons related to access or setup, do not obtain inspections result through a dimensional chain (obtaining a measurand, M, from a series of measurements $\{m_1, m_2,,, m_n\}$ as $M = m_1 \pm m_2 \pm .... \pm m_n$).

iii. Use the same datum face as far as possible.

iv. Minimize the variety of measuring equipment needed by maximizing the number of measurements of inspection features performable with each type of measuring equipment.

When a part model is captured in an engineering drawing, one usually adopts one or more of the six orthographical principal elevations: front, plan, left-end, right-end, rear, and bottom. In the case of features on an inclined plane, an auxiliary plane is sometimes created in the orientation of the inclined plane. When a CAD model is used, it is not a difficult to algorithmically associate each linear dimension with a certain principal plane. Recognizing the principal elevations is useful during process-planning since this enables one to align with the coordinate system adopted by the designer. In particular, each elevation points to a certain desirable setup orientation and access direction for the measuring instrument. Figure 5 shows the six orthographical principal elevations , namely, plan, right-end, front, left-end, rear and bottom of the Testing part shown in Figure 2. The viewing direction of individual principal elevation in third angle projection is shown in the table 2 as follows:

Table 2 Viewing direction of individual principal elevation in third angle projection

Principal elevation	**Viewing Direction**
PLAN	[0,0,-1]
RIGHT-END	[1,0,0]
FRONT	[0,1,0]
LEFT-END	[-1,0,0]
REAR	[0,-1,0]
BOTTOM	[0,0,1]

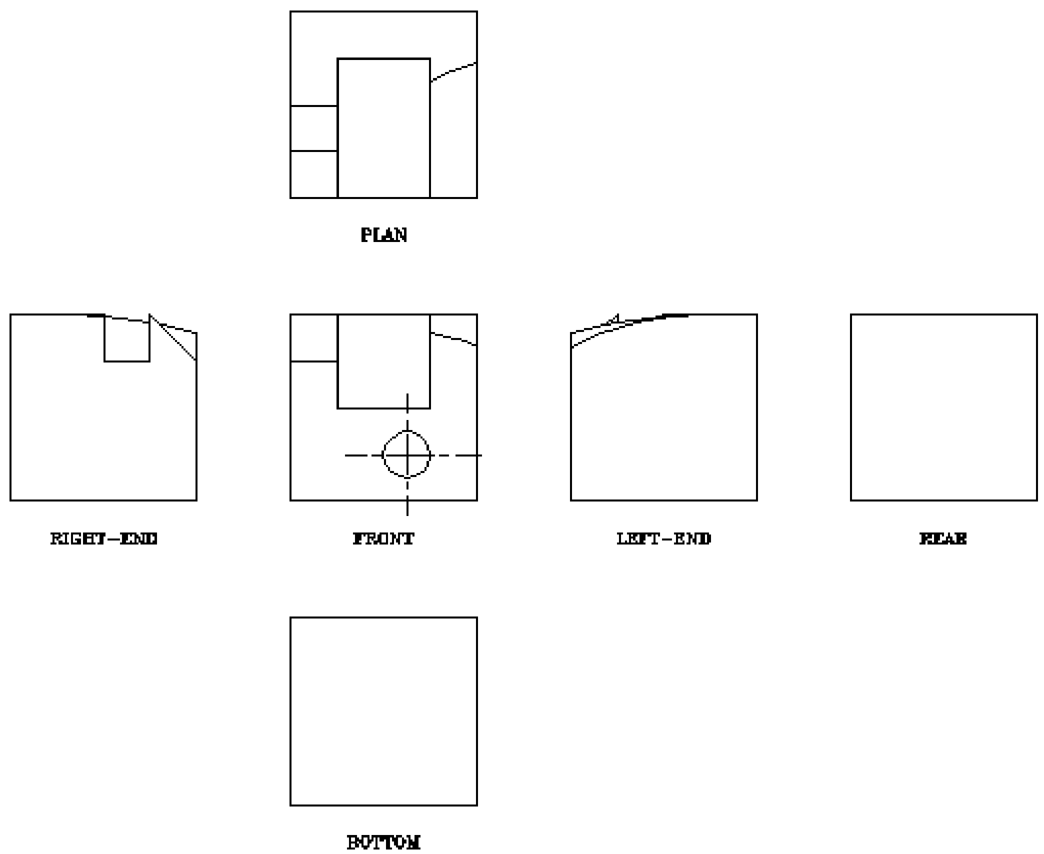

Figure 5 The six principal elevations of the Testing piece shown in Figure 2

Take the front elevation in Figure 5 of the Testing piece in Figure 2 (with Part Number :No.A001) as an example, the part is aligned in a view point of (0,-1,0). In viewing the front elevation in a direction [0,1,0], any feature which is visible in this direction is assessable from this direction [0,1,0]. In other word all the visible inspection features in the front elevation can be assessable from direction [0,1,0].

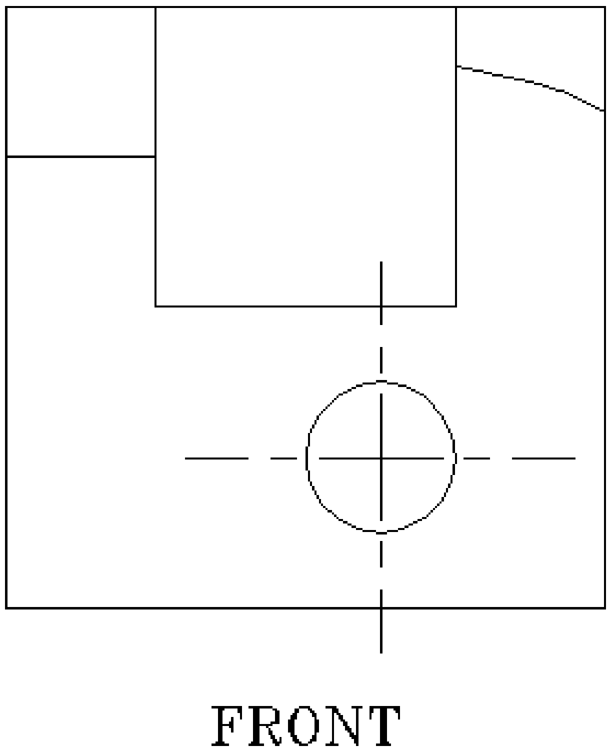

Figure 6 Front elevation of the Testing part shown in Figure 2

If we now search through the inspection features visible in the front elevation, we find the following eleven visible inspection features: *IF1, IF3, IF6, IF7, IF8, IF9, IF10, IF11, IF12, IF14* and *IF18.*

Details of these eleven inspection features visible in the front elevation are as follows:

inspection_feature (1, ”external”, [1,5], [3,4,6,7,8,9,10,11,12], [3,4,10,11]).

inspection_feature (3, ”offset”, [1,7], [3,4,6,7,8,9,10,11,12], [6,8,9,10]).

inspection_feature (6, ”external”, [5,15], [3,4,10,11], [16,10]).

inspection_feature (7, ”internal”, [8,9], [1,2,6,7,10,12,13,14], [1,6,7,10]).

inspection_feature (8, ”offset”, [10,6], [1,2,3,7,8,9,11], [1,7,8,9]).

inspection_feature (9, ”offset”, [10,16], [1,2,3,7,8,9,11], [15]).

inspection_feature (10, ”external”, [11,3], [1,2,4,5,10,12,13,14],[1, 4, 5,10]).

inspection_feature (11, ”external”, [11,8], [1,2,4,5,10,12,13,14] [1,2,6,7,10,12,13,14]).

inspection_feature (12, ”external”, [11,15], [1,2,4,5,10,12,13,14],[16,10]).

inspection_feature (14, ”internal”, [15,15], [16,10], [16,10]).

inspection_feature (18, ”free_form_surface”, [17,17], [1,3,10,9], [1,3,10,9]).

The corresponding ten primitive inspection cases can then be located easily in the inspection primitive case library, as *primitive_inspection_case (“A001”, IF(1*)), *primitive_inspection_case* (*“A001”, IF*(*3*))*, ,,etc.* where the part in Figure 4 is labeled as *A001*.

Armed with such information, the sub-case may be represented in a predicate “sub_case” using the syntax : sub_case(Part_ref, Sub_case_ref)

The predicate works as a pointer for the sub-case. The sub-case is represented in semantic nets of information consisting of six parts as follows:

Part (1) view_ref:

The view reference is with respect to the product model database. Figure 5 shows the six orthographical principal elevations of the Testing piece shown in Figure 2. There would be some auxiliary views or user-defined views other than those six principal elevations.

Part (2) orientation_parametric_list:

This list sets out the accessible direction of the visible inspection features in this sub-case. In this example, the accessible direction of the visible inspection features of this sub-case is [0,1,0].

Part (3) external_measurement_case_list:

The section details the primitive inspection case of external measurement in this sub case.

Part (4) internal_measurement_case_list:

The section details the primitive inspection case of internal measurement in this sub case.

Part (5) Offset_measurement_case_list:

The section details the primitive inspection case of offset measurement in this sub case.

Part (6) Free_form_surface_measurement_case_list:

The section details the primitive inspection case of Free form surface measurement in this sub case.

This results in the database of sub-case (*“A001”, 1*) being coded as:

```
/* **Database of    Sub_case Syntax*********************************

view_ref(View_name).

access_direction(Orientation_parametric_list).

external(External_measurement_case_list).

internal(Internal_measurement_case_list ).

offset(offset_measurement_case_list ).

free_form_surface(Free_form_surface_measurement_case_list ).

*********************************************************/

/*********Database of sub_case(“A001”,1)*********************/

view_ref(front).

access_direction([0,1,0]).

external([1,6,10,11,12]).

internal([7,14]).

offset([3,8,9]).

free_form_surface([17]).
```

Some times, an inspection feature might be visible in more than one elevation. For example, *IF7* is an internal inspection feature with face-pair [f8, f9] as its root feature and this feature is visible in both the plan and front elevations of the part. In such a case, the particular elevation with more visible inspection features is selected so as to minimize the number of setups.

By repeating the above procedure to cover all possible sub-cases and aggregating the results, one arrives at the inspection case for a complete part in a hierarchical manner.

10. Conclusion

Amongst the various CAPP domains, notwithstanding its enormous importance in industry, non-CMM-based inspection process planning has attracted very little research effort so far. This paper has tried to fill this gap by addressing two basic issues: inspection feature recognition and inspection case representation. This contribution underpins the future works directed towards inspection process planning utilizing CBR in CAPP environment (such as the Generic CAPP Support System proposed in [10]). A series of domain-specific and knowledge-based filters have been proposed to contain the problem of inspection feature explosion.

A 2-level hierarchical method for decomposing an inspection case into primitive elements has been suggested. By taking advantage of six or more orthographic projections of the part model, the approach enables information regarding setup and access to be included while describing cases. However, there would be occasions when the engineering drawing does not include all the necessary elevations so that some of the inspection features would not be visible and are presented in hidden lines. The hidden lines implies three possible situations: 1) the hidden features which are visible in the opposite direction, 2) the hidden features which are visible in the side direction, i.e. external/ internal undercut features, 3) the hidden feature which are not visible from all directions , i.e. closed cavity. In case of hidden features of case 1 and 2, orientation and setting of the part or re-design of the inspection tool path in the inspection process plan to make the hidden feature visible could solve the problem. Yin etal. [33] applied visibility theory and developed algorithms for set-up workpiece and mould parting. Kweon etal. [34] utilizes the visibility map to solve part orientations for CMM inspection. For the case (3) of hidden feature, either redesign of part or application of non-touching measuring equipment such as X-ray, ultrasonic measurement could help. In further exploring these issues, it should be remembered that the reference frame should, in general, be the same as that used by the designer. Often the designer would take the functional direction of the part as the reference direction. Other times, he might adopt the manufacturing direction. Whatever is the designer's intention should be evident if one has an engineering drawing containing orthographic elevations or a solid model with a specified global coordinate system. Further work is needed to arrive at a robust solution. This paper provides a framework of the methods and algorithms in the **"Represent"** sub-system of an automated CBR system for Inspection Process Planning.

Acknowledgments

The authors thank the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong for the facilities to conduct this study.

Appendix 1 Abbreviations

AI-based	Artificial intelligence-based
CAIPP	Computer-aided inspection process planning
CAPP	Computer-aided process planning
CBR	Case based reasoning
CMM	Coordinate measuring machine
CvTA	Concave triggering algorithm
CxTA	Convex triggering algorithm
EWEDS	Enhanced winged edge data structure
GCAPPSS	Generic CAPP support system
GFR	Geometric feature recognition
IPP	Inspection process planning
MPP	Machining process planning
PLTFG	Parametric-list technological feature graph
QTC	Quick turnaround cell
RFSA	Root face segmentation algorithm
TFR	Technological feature recognition

Appendix 2 Source Code of the Filters Applied

```
/* Filter */

/* 1) Product Specification Filter */

/*    An inspection feature is necessary to inspect,

        if its tolerance zone is less than 0.2mm*/

/*Filter ("Product_Specifications")*/

/* Tolerance <0.2mm*/

n_if(IF_N,filter("Product_Specifications")):-

inspection_feature(IF_N,_,[Face1,Face2],_,_),

dim([Face1,Face2],nominal(_),tolerance(T)),

T<0.2.

/*Domain Filter */

/* 2) Domain Filter*/

/* An inspection feature is necessary to inspect,

      if it is a hole. */

/*Filter("Domain")*/

/* All diameter of Holes */
```

```
n_if(IF_N,filter("Domain")):-

inspection_feature(IF_N,"internal",[Face1,Face2],_,_),

Face1=Face2,

face(face_no(Face1),_,"int_cylindrical",_).

/* 3) Application Filter*/

/* An inspection feature is necessary to inspect,

    if it is an external feature and

    upper limit=+0.1mm

    lower limit=0 */

/*Filter("Application")*/

/* +0.1 -0*/

n_if(IF_N,filter("Application")):-

inspection_feature(IF_N,"external",[Face1,Face2],_,_),

iifg([selected(Face1),selected(Face2),nominal(_),u_limit(0.01),l_limit(0)]).

/* 4) General Practice Filter */

/* An inspection feature is necessary to inspect,

    if its tolerance is less than 0.05mm.*/
```

```
/*Filter("General_Practice")*/

/* All tolerances <0.05 */

n_if(IF_N,filter("General_Practice")):-

inspection_feature(IF_N,_,[Face1,Face2],_,_),

dim([Face1,Face2],nominal(_),tolerance(T)),

T<0.05.

/* 5) Trade practice Filter */

/* An inspection feature is necessary to inspect,

    if its tolerance is less than 0.02mm.*/

/*Filter("Trade practice") */

/* All tolerances <0.02 */

n_if(IF_N,filter("Trade_practice")):-

inspection_feature(IF_N,_,[Face1,Face2],_,_),

dim([Face1,Face2],nominal(_),tolerance(T)),

T<0.02.

/* 6) Process_Capability Filter */

/*    All part with tolerance over 0.3 mm are under control.

      An inspection feature is necessary to inspect,
```

```
        if its tolerance is less than 0.3mm.*/

/*Filter("Process_Capability") */

/* All tolerances >0.3 are under control */

/* Only inspection tolerance <= 0.3*/

n_if(IF_N,filter("Process_capability")):-

inspection_feature(IF_N,_,[Face1,Face2],_,_),

dim([Face1,Face2],nominal(_),tolerance(T)),

T<=0.3.

/* 7) Special_attention Filter */

/*    All part with nominal sizes over 30 mm need special attention.

        An inspection feature is necessary to inspect,

        if its nominal size is over 30 mm.*/

/*Filter("special_attention")*/

n_if(IF_N,filter("special_attention")):-

inspection_feature(IF_N,_,[Face1,Face2],_,_),

dim([Face1,Face2],nominal(N),tolerance(_)),

N>30.
```

References

[1] Chang, T.C. Expert Process Planning for Manufacturing, Addison-Wesley Publishing, 1990.

[2] EIMaraghy H.A., Ham, I., et al. Evolution and Future Perspectives of CAPP. Annals of CIRP, Vol. 42/2/1993 p 1-13.

[3] Wang, K. An Integrated Intelligent Process Planning System (IIPPS) for Machining. Journal of Intelligent Manufacturing (1998) 9, 503-514.

[4] Baker,R.P. and Maropoulos,P.G. An Architecture for the Vertical Integration of Tooling Considerations from Design to Process Planning. Robotics and Computer Integrated Manufacturing 16 (2000) 121-131.

[5] Sadaiah, M., Yadav, D.R., et al. A Generative Computer-Aided Process Planning System for Prismatic Components. Advanced Manufacturing Technology (2002) 20: 709-719.

[6] Kayacan, M.C. and Celik, S.A. Process Planning System for Prismatic Parts. Integrated Manufacturing Systems 14/2 (2002) 75-86.

[7] Patil, L. and Pande, S.S. An Intelligent Feature-based Planning System for Prismatic Parts. International Journal of Production Research, 2002, Vol.40, No. 17, 4431-4447.

[8] Choi, J.C., Kim, C., et al. An Integrated Design and CAPP System for Deep Drawing or Blanking Products. Advanced Manufacyu=turing Technology (2000) 16: 803-813.

[9] Choi, J.C., Kim, C., et al. An Integrated Design and CAPP System for Deep Drawing or Blanking Products. Advanced Manufacturing Technology (2000) 16: 709-719.

[10] Yuen, C.F., Wong, S.Y. and Venuvinod, P. K., 2003, Development of a Generic Computer-Aided Process Planning Support System, Journal of Material Processing Technology (August 2003) , 139/1-3: 394-401.

[11] Juster, N. P, Hsu, L. H. and Pennington, A.D., The Selection of Surfaces for Inspection. in Advances in Feature Based Manufacturing, edited by Shah, J. J. et al. Elsevier Science (1994) 333-362.

[12] Jackman, J. and Park, D.K. Probe Orientation for Coordinate Measuring Machine Systems using Design Models. Robotics and Computer-Integrated Manufacturing 14 (1998) 229-236.

[13] Fan, K.C. and Leu,M.C. Intelligent Planning of CAD-directed Inspected for Coordinate Measuring Machine. Computer Integrated Manufacturing System, Vol. 11, No.1-2, pp. 43-51, 1998.

[14] Albuquerque, V.A., Liou, F.W. and Mitchell, O. R. Inspection Point Placement and Path Planning Algorithms for Automatic CMM Inspection. Int. J. Computer Integrated Manufacturing, 2000, Vol. 13, No. 2, 107-120.

[15] Grayer, A.R. The Automatic Production of Machined Components Starting from A Stored Geometry Description. Advances in Computer-Aided Manufacture, Edited by McPherson, D., North-Holland Publishing Co., pp. 137-152, 1977.

[16] Woo., T.C. Feature Extraction by Volume Decomposition. Proc. Conf. CAD/CAM Technology in Mechanical Engineering, Cambridge, Massachusetts, USA, 1982.

[17] Choi, B.K., Barash, M.M, and Anderson, D.C. Automatic Recognition of Machined Surfaces from A 3D Solid Model. Computer-aided Design, Vol.16, No.2, March, pp. 81-86, 1984.

[18] Milacic, V.R. Expert System for Manufacturing Process Planning. Computer-Aided/Intelligent Process Planning presented at the Winter Annual Meeting of the American Society of Mechanical Engineers, Miami Beach, Florida, November 17-22, 1985, pp.43-53, 1985.

[19] Wong, S.Y. Automated Geometric Feature Recognition – An Essential Component for Automated Design for Assembly. M. Phil. Thesis, Department of Manufacturing Engineering, City University of Hong Kong, 1992.

[20] Venuvinod, P.K. and Wong, S.Y. A Graph-based Expert System Approach to Geometric Feature Recognition. J. Intelligent Manufacturing, Vol.6, pp. 155-162, 1995.

[21] Stefano, P. D. Automatic Extraction of Form Feature for Casting. Computer-Aided Design, Vol. 29, No.11, pp. 761-770, 1997.

[22] Fu, M.W., Fuh, J.Y.H. and Nee, A.Y.C. Undercut Feature Recognition in An Injection Mould Design System. Computer-Aided Design, Vol. 31, pp. 777-790, 1999.

[23] Yuen, C.F. Coping with the Complexity and Infinite Variety of Geometric Features: A Cooperative geometric Feature Recognition System. Ph.D. Thesis, City University of Hong Kong, 1999.

[24] Watson,I., Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann Publishers, Inc.,1997.

[25] Schank R.C. and Jona M.Y. Issue of Psychology, AI, and Education: A Review of Newell's Unified Theories of Cognition, in Contemplating Minds—A Forum for Artificial Intelligence, The MIT Press, 1994.

[26] Georgin, E, Losel, S. and McDonald, J.R. Case-based Reasoning Applied to Fault Diagnosis on Stream Turbines. First United Kingdom Workshop Salford , UK , January 1995 ; Proceedings, Lecture Notes Artificial Intelligence 1020, Springer Verlag, Berlin pp. 121 – 133.

[27] Bouchet, J. and Eichenbaum-voline, C. Case-Based Reasoning Techniques Applied to Operation Experience Feedback in Nuclear Power Plants. Third European Workshop, EWCBR-96, Lausame, Switzerland, November 1996; Proceedings, Lecture Notes Artificial Intelligence 1168, Springer Verlag, pp. 497 - 511.

[28] Malek, M., Toitgans, M., Wybo, J. and Vincent, M. An Operator Support System Based on Cased-Based Reasoning for the Plastic Moulding Injection Process. 4th European Workshop, EWCBR-98, Dublin, Ireland, September 1998; Proceedings, Lecture Notes Artificial Intelligence 1488, Springer Verlag, pp. 402 - 413.

[29] Price, C.J., Pegler , I . S , Ratcliffe, M. B. and McManus, A. From Troubleshooting to Process Design: Closing the Manufacturing Loop. Second International Conference on Case-Based Reasoning, ICCBR-97, Providence, RI, USA, July 1997; Proceedings, Lecture Notes Artificial Intelligence 1266, Springer-Verlag, pp. 114 - 121.

[24] Price , C . J . and Pegler , I . S . Deciding Parameter Values with Case-based Reasoning. First United Kingdom Workshop Salford , UK , January 1995 ; Proceedings, Lecture Notes Artificial Intelligence 1020, Springer-Verlag, Berlin pp. 121 – 133.

[25] Malek, M., Toitgans, M., Wybo, J. and Vincent, M. An Operator Support System Based on Cased-Based Reasoning for the Plastic Moulding Injection Process. 4th European Workshop, EWCBR-98, Dublin, Ireland, September 1998; Proceedings, Lecture Notes Artificial Intelligence 1488, Springer Verlag, pp. 402 - 413.

[26] Dave, B. et al. Case-Based Spatial Design Reasoning. Second European Workshop, EWCBR-94, Chantilly, France, November 1994; Selected Papers, Lecture Notes Artificial Intelligence 984, Springer Verlag, pp. 198 - 210.

[27] Wybo, J. L., Geffraye, F. and Russeil, A. PROFIL: A Decision Support Tool for Metallic Sections Design Using A CBR Approach. First International Conference, ICCBR-95, Sesimbra, Portugal, October 1995; Proceedings, Lecture Notes Artificial Intelligence 1010, Springer Verlag, pp. 33 - 42.

[28] Zhrahal, Z. and Motta, E. Case-based Problem Solving Methods for

Parametric Design Tasks. Third European Workshop, EWCBR-96, Lausanne, Switzerland, November 1996; Proceedings, Lecture Notes Artificial Intelligence 1168, Springer Verlag, pp. 497 – 511.

[29] Sanders, K.E., Kettler, B.P. and Hendler, J.A. The Case for Graph-Structured Representation. Second International Conference on case-based reasoning, ICCBR-97, Providence, RI, USA, July 25-27, 1997 ; Proceedings, Lecture Notes Artificial Intelligence 1266, Springer-Verlag, Berlin (1997) pp. 245 – 254.

[30] Wong, F.S.Y., Chuah, K.B. and Venuvinod, P.K. Automated Extraction of Dimensional Inspection Features from Part CAD Models. To be published in a international referee journal.

[31] International Standards Organisation, ISO 406: Technical Drawings – Tolerancing of Linear and Angular Dimensions (1987).

[32] International Standards Organisation, ISO2768: Permissible Machining Variations in Dimensions without Tolerance Indication (1973).

[33] Yin, Z.P., Ding, H. and Xiong, Y.L. Visibility Theory and Algorithms with Application to Manufacturing Processes. Int. J. Prod. Res., 2000, Vol.38, No. 13, 2891 –2090.

[34] Kweon, S and Medeiros, D.J. Part Orientations for CMM Inspection Using Dimensioned Visibility Maps. Computer-Aided Design Vol. 30, No.9, PP. 741 - 749, 1998.