Heuristic Planning in Feature-based Inspection for Coordinate Measuring Machines

Robert B. Delvalle*††, Kavous Roumina*, Steve Ruegsegger†, Francis Merat*

*Case Western Reserve University, Cleveland, Ohio 44106-7122

†Materials Laboratory, WPAFB, Ohio

††Wisdom Systems, Inc., Pepper Pike, Ohio

Abstract

This system is a practical application of intelligent reasoning about the planning and sequencing of inspection tasks performed by a coordinate measuring machine (CMM). This research is part of an Air Force supported effort to develop a feature-based concurrent engineering system using an object-oriented development platform. Manipulation of design, manufacturing, and inspection features (in this system) is unique in that features are transformed into domain specific objects rather than degraded into elementary geometry. The object-oriented platform provides a means for transformation through built-in qualities such as inheritance, tree-search, and a cohesive part/sub-part organization. Inspection features represent not only geometry, but references to a world coordinate system, proper probe approach, and preferred evaluation method. Reasoning through inspection rules forms a single data object which can then be transformed into CMM executable code.

1. Introduction

The Rapid Design System (RDS)[1] is a feature-based concurrent engineering expert system for intelligent design, manufacturing, and inspection of machine parts, in which "memory" serves as a complement to "rules" in productive guidance of the system. The representation of information as features has been developed for geometric tolerances and inspection process planning information on an object-oriented development platform. The part geometry and tolerances are transformed into inspection features that subsequently build a data object describing the inspectability of the part-model. Inspection features represent methods and detailed procedures for evaluating tolerances in conformity with industrial practices. More details of the approach to representing geometry as features can be found in [LeClair,91].

The inspection planning subsystem of the RDS is called the Inspection Planning and Evaluation Module, or IPEM. Through a cohesive object-oriented system based on inheritance through a part/sub-part hierachy - in this case consisting of a specification of the part geometry as well as dimensioning and tolerance information--to produce an inspection plan. Specifically, the IPEM reasons about the geometry and tolerances of the designed part (just as an inspector does) and creates detailed instructions to properly inspect the part using a computer controlled Coordinated Measuring Machine (CMM). Considerations which are important to inspection planning for CMMs such as the generation of collision free inspection probe paths are included. Once an inspection process plan is generated it is translated into executable code or instructions for a computer controlled CMM.

The heuristic constraints in this inspection system transform design and tolerance specifications into a logical ordering of steps based on Geometric Dimensioning and Tolerancing (GD&T), the ANSI tolerancing standard. This "logic" might be considered a set of rules from which the inspection plan, conforming to GD&T, can be developed. Since tolerancing is already performed by the designer, a process plan can be formed using the heuristics described in Section 4. Intelligent inspection planning systems of comparable scope attempt to use classical expert or rule-based system and data-base managers to choose the appropriate tolerances for a part, inspection station or CMM, accessibility of measurement surfaces, and devise the final plan. Three worth mentioning are by H.A. ElMaraghy and P.H. Gu at McMaster Univerity/Canada[ElMagarthy,87], Curtis W. Brown at Allied Signal, Inc.[brown,91] and C.H. Menq et al. at Ohio State University[Menq,92].

ElMaraghy and Gu, one of the earliest efforts in the development of a feature-based intelligent integrated manufacturing system. The inspection module included an expert system as a consultant in the tolerancing of turned parts, semantic pattern recognition to group parts by similar inspection method, and a knowledge-based system to create an reasonable order of tolerance inspection. Knowledge-based reasoning is applied to feature accessibility as well. The intent was to reason about whether the inspection probe could access a target surface based on combinations of factors such as the probe approach vector, the dimension of possibly obstructing adjacent features, and the normal of the surface at the inspection point. It seems difficult to imagine a finite set of rules and combinations thereof that would handle the accessibility problem adequately except in systems producing high volume single lots.

---

1. This research was supported under U.S. Air Force Contract No. F33615-87-5250 through Wright Laboratory, WPAFB, Dayton, OH

Brown's system deals with prismatic parts. The system conception is explicitly laid out in nine modules, each with sub-modules. The overall expectation is that a knowledge-based inference engine, together with a data base of facts pertinent to each module, or sub-module, would process any part design within the domain of the data base. The final output would produce the inspection plan in CMM control code form. Again, a solely generative system reliant on only those facts known to the data base.

Menq, et al. developed CAD/CAM integrated system to deal with inspection of casted or otherwise sculpted parts. They present interesting methods of dealing with the problem of evaluating measurement data on complex surfaces. There is also recognition of the need to include inspection "attributes" associated with the feature geometry, such as design intent or manufacturing statistical process control, which would contribute to the inspection plan formulation. The automated inspection system they propose employs knowledge-based reasoning drawing from a set of "catch-all" rules that apply GD&T, and other ANSI or ISO standards. Unfortunately, it is very unclear how the intelligent system is organized or how the "inspection attributes" are integrated or accessed in the planning scheme.

Of the systems mentioned none are developed on an object-oriented platform. They must rely on data-base storage of dimensional, tolerance and all other pertinent data. There is also the implication that an ever-expanding rules set will be required. Knowledge-based system have an advantage over expert systems in that rules can be added within to the applicable domain. But this requires an expert to know where the new rule is applicable. By contrast, the advantage of a memory-driven system is that rules can be extracted from the cache of past design or inspection episodes. A memory system called the Episodal Associative Memory (EAM) is integrated into the RDS and explained in [Pao,91]. Although a simple set of rules and heuristics are included, there is not the burden on the user to continually add to the rule set. The intention is to provide a hybrid intelligent system where the output of a rule-based algorithm can either be tuned, or tweaked by the expert operator, or else created interactively where the system becomes aware of the methods, preferences, and/or personal touch of the operator and modifies the rule-set accordingly.

It can be quite time-consuming for an inspector to understand a drawing and determine how to inspect a part using the automated inspection systems present in competitive industry today. First of all, the tolerance standard is moving away from plus/minus tolerances and toward a geometry-based standard, ANSI Y14.5, a.k.a. Geometric Dimensioning and Tolerancing (GD&T)[ANSI,87]. Secondly, the CMM programming and tolerance evaluation is based on the GD&T paradigm[12]. Programming CMMs is time-consuming, tedious, and highly susceptible to human error. This overhead is particularly critical when it's spread over a number of low-volume parts. An integrated automated inspection system can reduce the turnaround from part design to final inspection, cutting costs and allowing for better response.

The creation of an inspection process planning system integrated within a CIM system is welcome by inspectors whose work is concerned with design intent, the tedious calculation of inspection points, and the complexity in planning efficient spacial movements. Often the inspector must query the designer's rationale over tolerances specified with neglect to inspectability. A fully integrated system provides easy access to design and/or manufacturing information if needed. Once the inspection strategy is formulated, the inspector must calculate points on the part as well as create a computer program in CMM control language to instruct the instrument to move about those points. This program, in the cryptic control code, specifies the path of the probe, the evaluation of part geometries, and allowable tolerances. Any error in the code specifying the CMM task path may result in a costly collision with the part or other objects. Hence, the ultimate goal of the IPEM is to develop an inspection process plan using data from all areas of the manufacturing system and to automatically generate accurate CMM code, that can simply be downloaded to the inspection device.

2. Representation

Tolerances are specified according to the Geometric Dimensioning and Tolerancing(GD&T) standard [ANSI,87] and can be either coordinate tolerances (i.e. plus/minus) or geometric tolerances. Special emphasis has been placed on geometric tolerances because of their application to CMM inspection. An example of a prismatic part toleranced in GD&T is shown in Figure 1. This part is essentially a single block with a slot feature removed from the top, and a through hole drilled in the bottom of the slot. The slot and the hole can be thought of as negative features attached to the surfaces they intersect....

Inspection of a given part can be parsed into sub-tasks called *inspection plan fragments* (IPF) describing the measurement and evaluation of tolerance geometries. The physical measurements that create each tolerance geometry, or *measurement requests* (MR), are grouped in *setup* objects that specifies the part orientation and a spacial frame of reference called a *datum reference frame* (DRF) within which tolerance data is evaluated. Specific information for CMM control such as destination points and collision avoidance methods are represented in the object *CMM-metacode*. This section defines these terms, lists the makeup of the inspection objects, and describes the formulation and flow of information toward a complete inspection plan. **explain figure one and a part figure Frnk is writing this part**

2.1 The RDS Object Structure

The development package used to build the entire RDS, including the IPEM is the Concept Modeller(TM) by Wisdom Systems, Inc.. Built upon CLOS (Common Lisp Object-Oriented System), the Concept Modeler is a higher level language that provides a part/sub-part hierachy, an inheritance relationship within the hierarchy, a part/property relationship, as well as the facility to relate parts geometrically. These tools provide the basis for highly integrated, functional system development. There has been a great deal of success in the case of the RDS, where design, manufacturing and inspection interrelate. For example, from the set of design primitive geometries (e.g. block or cylinder) a substantial set of more sophisticated design features (e.g. hole, blind hole, pocket, rib, skewed rib, etc.) are created which inherit the primitive properties. They, in turn, can then be organized geo-

Figure 1. A sample part designed using GD&T.

metrically through attachments, dimensions, and offsets to form a variety of the common parts relevant to the users area of industry. Associated with each part or feature are methods of manufacturing and a set of appropriate tolerance assignments. These methods and assignments can themselves be represented as parts that inherit the properties of the design features they are assigned to. A further expansion is the representation of the manufacturing and inspection process plans in the part/subpart scheme. This paradigm has been fully exploited in all areas of the RDS system, further discussion of which can be found in Radack (Radack et al.,90], Keotter(Keotter et al.,90] and Merat(Merat and Radack, 92].

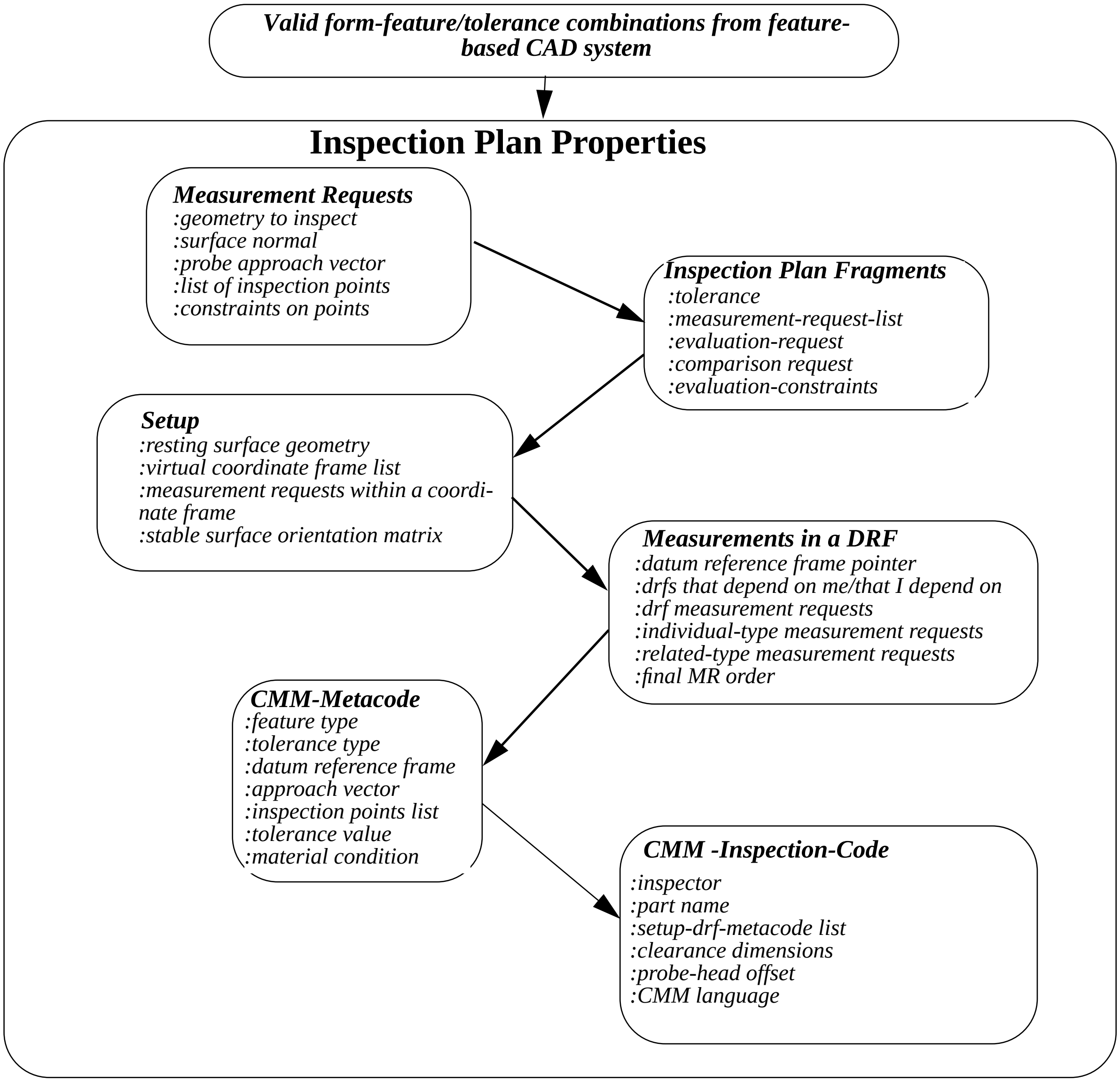
2.2 The IPEM Part-Model and Properties

The object definitions in the IPEM built upon the notion of part/subpart, and part/property. The main part-model in the IPEM is called the “inspection-planner-class”. The main thematic breakdown of the inspection part-model is that of part/property. Figure 1 shows the major properties as headings, under which are attributes of each. The attributes are objects or lists of objects, each created by methods on their type. The figure is meant to demonstrate the flow, of transformation from one domain to the next. The part-model also has subparts that play an auxiliary role.

2.2.1. Inspection Plan Fragments (IPF)

The process of creating the overall inspection plan consists of identifying viable sub-plans for each tolerance in the design. That is, determining all the possible ways each tolerance geometry can be accessed and successfully measured. A single tolerance geometry can often be measured in multiple ways resulting in the generation of redundant sub-plans. For example, with the part in one orientation, the target geometry would be approached from one direction - in another part orientation the same geometry is approached from a different direction. At this stage an intractable problem of finding an optimum plan from a set equally valid sub-plans is uncovered. The descriptive name given to the sub-plans are inspection plan fragments.

The composite of information that makes up an IPF is as follows:



igure 2. Inspection planner objects and their attributes. Arrows indicate the order in which knowledge is transformed. The final result is a listing of CMM code.

- the geometry of the tolerance
- how the part is to be oriented to permit inspection of that geometry,
- how the part surface is to be measured,
- how the obtained data points are to be evaluated, and
- how the measured geometry is to be compared with the tolerance specification.

IPFs are generated on a one-to-one basis from the tolerance features but, many-to-one with respect to tolerance geometries. Specifically, each IPF is created by linking each instance of a tolerance object to an IPF object [11]. Figure 2 shows the relationships between the tolerance features and the IPFs for a part with four diametral tolerances, three position tolerances, and two datum reference frames (a.k.a. DRFs). The positional tolerances are linked to the GD&T datum reference frames through their specification. This relationship is shown as horizontal links in Figure 2.

The IPFs generated from the tolerance features are valid for non-intersecting form features. Interacting form features are defined to be features which intersect geometrically in such a way that not all of a feature's surface is present in its entirety. Work is in progress to detect interacting form features and modify the IPFs as appropriate.

For each tolerance geometry, separate IPFs can be generated for different CMM probes and probe orientations. Although many IPFs can be generated for each tolerance geometry, only one IPF need be executed. Redundant IPFs are eliminated from the inspection process plan during the optimization phase of the planning system.

An IPF logically contains the following specific data objects, corresponding to the last three bullets listed above:

- measurement request: a link to the surface of a feature to be measured, the number of points to be measured, and any constraints on those points, e.g., that they be non-collinear. Specifically this describes a tolerance geometry.
- evaluation request: a high-level representation of the type of evaluation to be performed.
- comparison request: a high-level representation of the method of comparison to be performed.

2.2.1.1 Measurement Requests

Measurement requests are assigned to the significant surfaces of a feature on which a tolerance is placed. For example, the significant surface of all tolerances of a thru-hole is the cylindrical inner wall. Tolerances on a thru-hole require one MR. The significant surfaces for most tolerances on a thru-slot such as position, parallelism, or perpendicularity are the parallel inner walls. These tolerances require two MRs. There is a one-to-one correspondence between measurement requests and the physical surfaces of the part or a feature (e.g. thru-hole, slot,etc.) involved with the tolerance. A measurement request is a concise prescription of the action to be taken on the surface(s) that defines a tolerance geometry. The following information is provided:

- Geometry I.D. of the surface to inspect
- Type of the surface, e.g. planar, or cylindrical
- Total number of inspection points needed for this surface
- List of xyz-coordinates of the inspection points
- List of constraints placed on inspection points e.g. random, co-planar,...
- Direction of approach for CMM probe
- Move vector for any of the sample points
- A pointer to corresponding IPF(s)

A surface will have only one measurement request per tolerance, though a surface may be multi-toleranced. During optimization, redundant measurement requests pointing to the same surfaces will be eliminated, leaving those that satisfy the minimum point requirement of all tolerances involved.

2.2.1.2 Evaluation Requests

Evaluation requests are data objects which contain information about how measurement data should be evaluated. Based on specification from the designer, the touch probe sample point data can produce geometries based on the following possible methods:

- Least Square Fit [13]: the most commonly used method; fitting a geometric object to sampled points.
- Fitting points to mathematical equations for creation of various geometries: Line, Circle, Plane, and Sphere.
- Maximum Inscribed or Minimum Circumscribed for circular surfaces only. Maximum Circumscribed: fitting the largest circle to set of points from without (outside); Minimum Inscribed: fitting the smallest circle to a set of points from within (inside).

These methods range from general to specific. The output of the evaluation process is a geometric object.The IPEM offers the designer or inspector a choice of the evaluation method. The present IPEM implementation only offers Least Square Fitting due to the specific CMM currently being used for inspection (LK Microvector 80).

2.2.1.3 Comparison Requests

Comparison request data objects provide the methods to compare the evaluated geometries with the tolerance constraint geometries. The constraint geometries are those which create the allowable tolerance zones within which the evaluated geometries should fit.

The comparison task is a Boolean operation. That is, either the geometry passes inspection or it does not. This process could be expanded to determine the degree of failure, informing the inspector the part might be reworked to meet specified tolerances. For instance, a hole with a smaller diameter than specified might be re-bored to meet the specification.

2.2.2. Setup

A setup object is established once the IPEM has analyzed the spacial accessibility of all the tolerance geometries specified for each stable orientation of the part (it is assumed, for prismatic parts, that one or more external surfaces will provide free-standing support on the inspection surface). Within a setup, one or more three dimensional coordinate frames, or virtual coordinate frames, must be established in order for inspection points to be located in space, based on the given orientation. Therefore, a list of accessible measurements within the context of this setup is included. A summary of information available within a setup is:

- Geom of Resting Surface: Surface geometry i.d. of the stable surface
- MR List: List of measurement requests inspectable in this orientation
- Measurements in a DRF: Information to create a virtual coordinate frame in which tolerances are associated with the datum reference frame that establishes the origin they reference
- Stable Surface Orientation Matrix: Transformation matrix needed to transform the part so that the resting surface of it is lies on the x-z plane

- Virtual Coordinate Frame List: List of virtual coordinate frames (i e. a set of three mutually perpendicular geometries, for example, a datum reference frame) accessible in this orientation.

2.2.3. Measurements in a Datum Reference Frame

A CMM requires a “world” coordinate system within which points on the physical part are specified. However, GD&T tolerance and geometric features create local coordinate frames. A coordinate frame object is used to identify the coordinate frame measurements referred to.

The software used to command the commercial CMM used in this work (CMES[7]) requires that a part origin be manually established by intersecting three ortho-normal vectors evaluated from the measurement data. The required intersecting vectors are created using the 3-2-1 isostasies principle from the part geometries; plane, axis, or vertex. According to this principle, any object can be completely fixed in space by simultaneous contact on three, two, and one point(s), respectively, on three mutually perpendicular geometries. Essentially, CMES requires the measurement of these points with the part fixed to the inspection table. Typical geometric combinations are:

- three planes,
- a plane, an axis, and a plane,
- a plane, and two axes.

The “measurements-in-DRF” data structure provides an attribute for the reference geometries, known as datums, and the coordinate frame created, called a datum reference frame (DRF). Within each coordinate frame object is an attribute for members of the “related” tolerance category (e.g. position, concentricity). These tolerances must be measured within a specified DRF and are therefore grouped accordingly. Form tolerances, categorized more generally as “individual” tolerances (e.g. flatness, straightness) are not required to be measured in a specified DRF. An attribute “final-MR-order” contains a list both MR types, ordered according to minimum physical separation. A complete listing and explanation of the attributes follows:

- DRF-pointer: Memory location pointer
- DRF-name: External design name; i.e.ABC,ADE,etc.
- DRF-type: Abbreviation of datum geometries; PPP, PAP,etc. (P= planar, A= axis)
- Datum-Resting-Surface: Geometry of the part resting surface
- DRF-I-Depend-On: DRF that must be measured and evaluated before this DRF can be established
- DRFs-Depend-On-Me: DRFs that require this DRF to be measured and evaluated before they can be established
- DRF-MR-List: Measurement requests to establish this datum reference co-ordinate frame
- Individual-Type-MRs: Measurement requests for “individual” type tolerances
- Related-Type-MRs: Measurement requests for “related” type tolerances
- Final-MR-order: Measurement requests sequenced in a minimum distance probe path within a datum reference coordinate frame

2.2.4. CMM Metacode

Because there are many different CMM manufacturers, many with their own proprietary language, the IPEM generates the inspection plan in a generic form which is later translated into the desired CMM language by an appropriate language generator. This intermediate form of the inspection plan is called the “metacode” because it is a code that describes another code, the desired CMM code.

The metacode is created by direct translation of the MRs in the SETUP-DRF-MR structure produced by the process planner. The key to this feature translation is the inheritance and tree search capabilities built into the object-oriented platform. Each MR object is recognized as its own type. The appropriate method then operates on the information held by the MR and its parents to produce the metacode object.

This translation to metacode is not one to one. Metacode generation employs different functions within the inspection plan: one may measure and evaluate; another measure, save the points and evaluate; and yet another uses saved points from other measurements to only evaluate. This process variation arises from the removal of redundant MRs, described above, where two tolerances on the same surfaces can use the same measurement points, but evaluated differently based upon their tolerance. Once created, the metacode can be saved to disk for future retrieval.

- Feature-type: Encrypted form-feature type (e.g. TH = through-hole)
- Tolerance-type: Encrypted tolerance-feature (e.g. POS = position)
- DRF: Name of the datum reference frame
- Approach-vector: Direction (X,Y,Z) points should be taken in.
- POS-info: List of information about where the feature is located. It differs for each type of feature to be inspected.
- TOL-info: List of information about the tolerance values placed on the feature.
- Num-points: Number of points to be inspected.
- Feature-name: String with the feature's name from the FBDE. Used for comments in the CMES file.
- Save-meas?: Whether or not the surface to be inspected has been inspected before and is therefore redundant (not presently used).

3. Inspection Planning Algorithm

The algorithm used to generate a part inspection plan is graphically illustrated in Figure 3 and described in greater detail in the following sections.

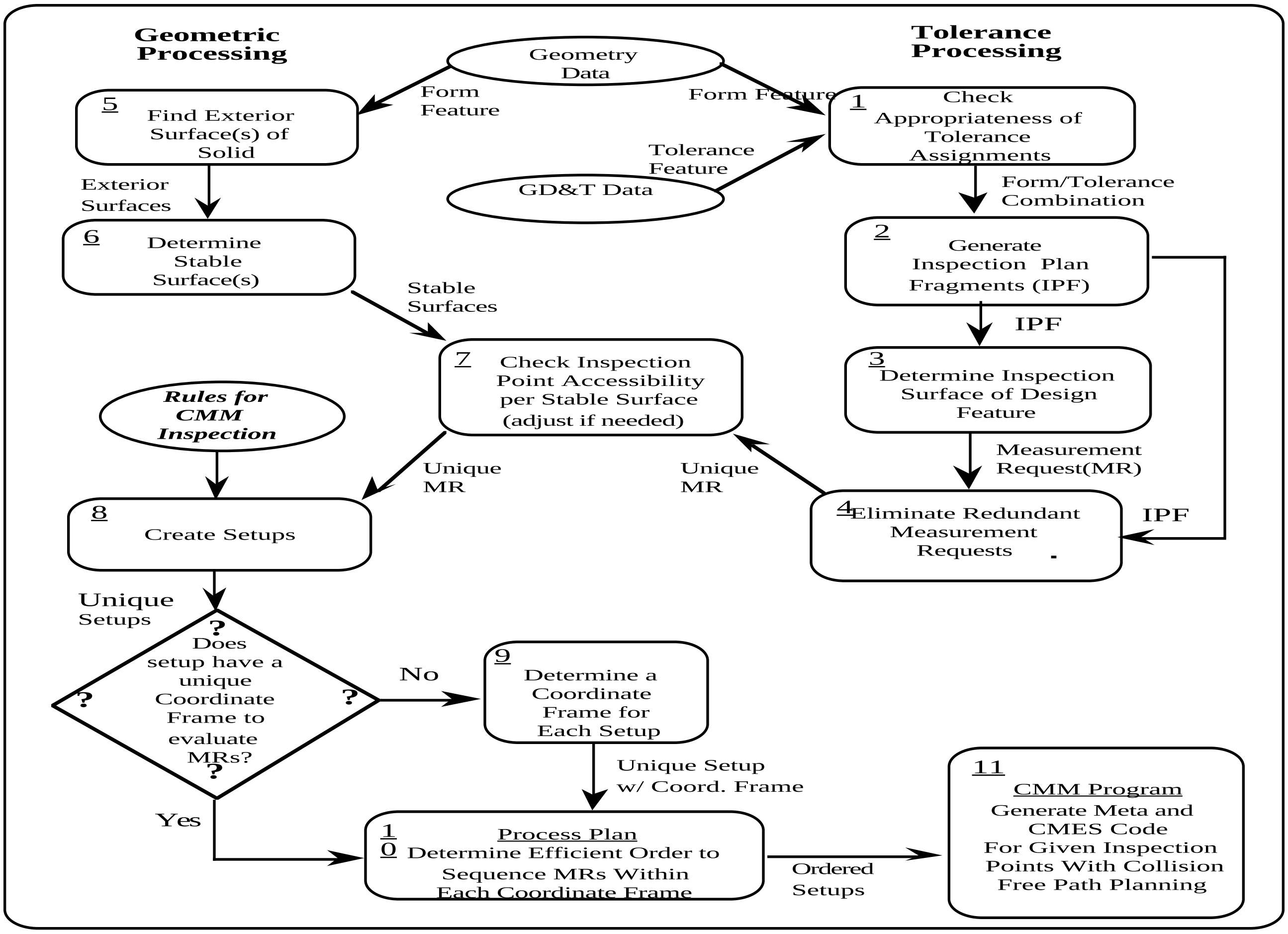


Figure 3. Inspection planning algorithm

3.1 Check Appropriateness of Tolerance Assignments

Since it is possible in principle to assign any GD&T tolerance to any form feature there needs to be a mechanism for avoiding erroneous situations. For example, placing a cylindrical tolerance on a pocket makes no sense. Therefore, the “Allowable Specification” table, (Fig. 4) is referenced each time a tolerance feature is to be created. The entries of this table are “Y”: an appropriate combination, “NS”: non-standard GD&T but allowed by the system, and “NA”: not allowed.

form features→ tolerance features ↓	*Blind / Through Hole*	*Boss*	*Pocket*	*Through Slot*	*Open Step*	*Step-to-Shoulder*	*Edge Cut*	*Rib*
Flatness	NA	NA	NA	NA	NA	NA	NA	NA
Straightness	NA	NA	NA	NA	NA	NA	NA	NA
Circularity	Y	Y	NA	NA	NA	NA	NA	NA
Cylindricity	Y	Y	NA	NA	NA	NA	NA	NA
Perpendicularity	Y	Y	NS	Y	Y	Y	Y	NS
Angularity	Y	Y	NS	Y	Y	Y	Y	NS
Parallelism	Y	Y	NS	Y	Y	Y	Y	NS
Position	Y	Y	Y	Y	NA	NA	NA	NS
Concentricity	Y	Y	NA	NA	NA	NA	NA	NA

Table 1. Table of allowable specifications

3.2 Generate Inspection Plan Fragment for Tolerance Features

Each tolerance feature initiates an Inspection Plan Fragment which is a set of specifications about how a tolerance is to be inspected/evaluated. An IPF is of the inspection feature class with attributes as described in Figure 1.

3.3 Determine Inspection Surface(s) of Design Feature

Before the measurement request attribute of an IPF can be instantiated, the surface of the form feature that will actually be inspected (measured) must be determined. For example, when a position tolerance is placed on a blind hole, only the cylindrical side surface is to be inspected. A position tolerance does not require any measurements of the hole's bottom surface. Each valid combination of form and tolerance feature specifies a certain subset of the form feature’s surface(s) to be inspected. A measurement request (MR) instance is generated for each specified surface and linked to the IPF which was generated for the tolerance feature instance.

3.4 Eliminate Redundant Measurement Requests

In the process of generating measurement requests for all datums and tolerances, multiple requests may have been generated for inspection of the same surface. For example, a blind or through hole to be inspected for perpendicularly and cylindricity will initially produce two IPFs with identical measurement requests. Since the information needed to evaluate both tolerances for the hole can be obtained from the same inspection points, only one set of inspection points is required. Hence, these two MRs can be merged into one MR, provided that enough points are sampled to satisfy the requirements of both tolerances.

3.5 Find Exterior Surface(s) of Solid

At some point in the physical inspection process part needs to be placed on the CMM table for inspection. For this to happen one needs to determine the flat exterior surfaces upon which the model can be placed. Some of these exterior surfaces may not be able to support the design model without the use of fixtures; therefore it is necessary to determine which of these surfaces are stable. Hence, an exterior surface is either intrinsically stable or is considered unstable. An unstable surface may be designated as a resting surfaces if fixturing is to be used. Currently no provision for fixturing is implemented.

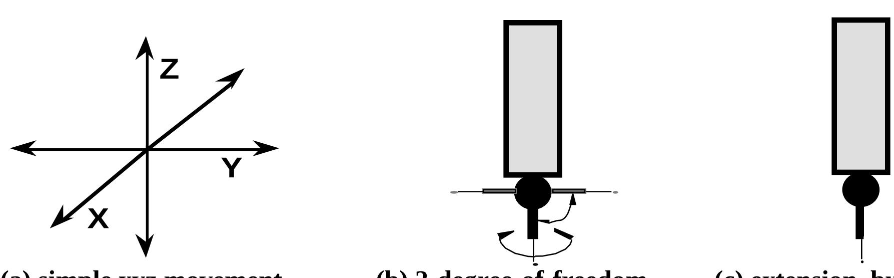
3.6 Determine Stable Surface(s)

The stability test involves the projection of the candidate resting surface to the X-Z plane followed by the creation of a two dimensional convex hull area for the surface in question[Preparata,77]. If the projection of the center of the mass of the solid model onto the X-Z plane lies within the convex hull the surface is stable.

3.7 Create Setups

Each stable surface establishes a possible part orientation for inspecting the part. Each surface to be inspected is assigned to a particular, stable part orientation by orienting the part such that it is resting on a stable surface. The surfaces to be inspected which can be accessed by the CMM in this part orientation are then identified. The setup surface and list of accessible MRs are stored in the setup class object. There exists one setup per stable surface. As explained in 3.5.4, it is possible that a surface is inspectable in more than one setup. This redundancy is removed by the process planner. Simple heuristics are used to determine which surfaces are inspectable in a particular orientation. For instance, only those inspection surfaces visible from a vantage point directly above the surface are accessible.

The CMM accessibility test assumes to a probe holder with zero degrees of freedom (d-o-f) (Figure 5c). That is, the CMM model is capable of movement in 3 directions, as in a **xyz** coordinate frame (Figure 5a).Specialized probe holders add one or two additional rotational degrees of freedom (Figure 5b). Our model does not add degrees-of-freedom beyond the basic **xyz** movement and treats the probe as an extension of the robot arm in the z-direction.



(a) simple xyz movement. **(b) 2-degree-of-freedom probe holder.** **(c) extension, but no additional d-o-f**

Figure 4. CMM movement configurations.

A heuristic is used to test the space directly above each inspection point (i.e. a half-line) for potential probe collisions with the part. If an obstruction is detected in that space, the surface on which the inspection point is located will not be considered inspectable.

To avoid erroneous affirmation from the “z-axis” accessibility test, a geometric simulation of the CMM probe is created. For each point deemed accessible by the “z-axis” test, the geometric probe simulation is positioned at the point as it would be if the actual CMM was taking the measurement. If an intersection between the part-model solid and the model probe geometry is detected, the point’s accessibility is rejected.

This access testing will be expanded, in future versions of the IPEM, to include probe holders which add additional degrees-of-freedom (as shown in Figure 5b). The integration of the configuration space search proposed by

3.8 Associating Measurement Requests with Datum Reference Frames

A valid setup is one in which all datums (of a DRF) can be accessed for measurement by the CMM. At least one DRF (typically the resting surface) is identified for manual probing. The tolerances and other DRFs that are linked to the manually established DRF (and are accessible), are grouped in the "measurement-in-DRF-list" property within the setup object. The measurement-in-DRF feature contains specific details about each DRF accessible in the orientation represented by the setup. The critical information being:

- must the DRF datums be located by a manual manipulation of the CMM probe?
- Otherwise, is one (or more) of the DRF datums located in the course of evaluating a tolerance (such as position, perpendicularity, etc.) placed on it? and
- which tolerances, accessible in a given part orientation, are related tolerances and which are individual?

This information is essential to determining the order in which part surfaces are measured. First, the DRF used to initially locate the part on the CMM table must be probed manually. Therefore, the first action taken in the implementation of any inspection probe path plan is to locate the appropriate datum reference frame for this task.

Second, if there are DRFs with datums that have related tolerances associated with them, the new coordinate frames cannot be established until those tolerances are evaluated. Hence, reasoning through the inspection process organizes tasks in the following order:

1) manual probing to establish the part location and a datum reference frame
2) measurement and evaluation of tolerances specifying the initial DRF
3) establishment of a DRF whose datum(s) is(are) evaluated in step 2
4) measurement and evaluation of tolerances specifying the DRF from step 3

This sequencing algorithm is recursive at two levels: through each setup, and through the number of DRFs occurring in step 3. The higher level sequencing is performed for each valid setup. Each setup requires maually measuring at least one DRF. Hence, least one measurement-in-DRF object is created per setup, but multiple DRFs are possible. The lower level planning among the measurement requests within the measurement-in-DRF.

Further actions are to select and sequence the minimum number of setups, remove the unnecessary setups, and sequence the measurement request within each setup as described in the next section.

3.9 Process Planning

The process planning algorithm consists of three sub-processes: 1) an ordering of setups, 2) an ordering of DRFs within a setup, and 3) an ordering of individual measurement requests within a DRF coordinate system. This can be seen as a global, mid-range, and a local ordering, respectively. In global ordering, the goal is to find the minimum number of setups for complete inspection of the manufactured part. This is accomplished by identifying the setup with the largest number of measurement requests (MRs), say setup *A*. Then, the MRs specified in *A* are removed from all other setups. This may lead to a situation where several setups may lose some or all of their MRs. Setups with no remaining MRs are eliminated from the process plan and the remaining setups are ordered according to the length of their MR attribute list.

Mid-range processing accomplishes a logical grouping of related MRs with the DRF they reference. If there are two (or more) DRFs in a setup, the measurement and evaluation of each DRF's datums must precede the measurement of the tolerances which reference that DRF. This reasoning process sequences the "measurement-in-DRF" objects properly.

In local ordering, MRs are spatially sequenced using a mix between reasoning about collision-free paths from MR to MR and optimizing the path between measurement points on a given surface using a Hopfield neural net algorithm[Hopfield, 1989]. MRs are grouped within the "measurement-in-DRF" object. The first MRs to be probed are those pertaining to the DRF datums. Establishing a DRF requires a surface probing order of primary, secondary, and tertiary datums in a 3-2-1 point touch sequence. From the final tertiary point, the Euclidean Distance (ED) between this point and all sample points of every other MR in the current DRF coordinate system grouping is calculated. This grouping includes MRs from the related and individual type lists.From the tertiary point the Hopfield algorithm then considers all of the possible paths between MRs and the points within the MRs at once and reaches a near-optimal shortest distance solution. This path plan, represented by an ordered list of MRs is placed into the "final-MR-list". The procedure is repeated until all MRs in each DRF coordinate system grouping have been processed. The ordering of points in this list is not final, however, without considering collision-free path planning.

3.10 Path Planning

The path planning algorithm generates a CMM inspection path according to the generated process plan which avoids collisions between the CMM touch probe and the part or associated fixtures in the vicinity of the part, such as clamps or jigs. Generating a path collision free path for the inspection probe entails creating intermediate paths that move the probe around the part, rather than trying to move through the part. These intermediate paths are created by replacing a colliding path with two or more non-colliding paths. The use of a generic, computationally expensive robot path planning algorithm has been avoided by exploiting some specific properties of the CMM path planning problem. First, the optimal (shortest) path is not needed at the expense of computational time. As a result, the travel time of the probe's path during the inspection plan around the part is typically small compared to the setup time. Second, many of the parts inspected are inherently small. Third, the path is usually easy for the inspector to visualize because the probe should simply move from one surface to another surface. This implies that the via-points are going to move the probe around the corners of the part, not necessarily through an array of obstacles.

For the above reasons, a generic AI search heuristic was not employed to find the collision free path. Instead, “common sense” algorithms are used to create the via-points. These algorithms are called in a particular order and executed if applicable. For details of the collision-free as well as an explanation of the Hopfield path minimization algorithm can be found in [Reugsegger, 1993].

3.11 CMM Code Generation

The desired output of the IPEM is the actual code to drive a CMM to automatically inspect and evaluate the tolerances on the manufactured part. Producing this code is a very tedious task for the inspector because CMM languages are cryptic, and the accuracy of specified inspection points is critical. A simple one character typographical error could result in the probe crashing into the part or the CMM table.

The currently implemented metacode-to-CMM-language software generation engine translates to CMES (Coordinate Measurement Software) language (a proprietary language of LK Tool, Inc. a division of Cincinnati Milicron). It inputs each metacode object and outputs, to a file, the CMES code to perform the desired tasks. CMES has provisions for some GD&T evaluation, and this is used to advantage. The CMES code is then downloaded or transferred by diskette to the CMM controller, usually a PC, where it is executed.

4. Results

The resulting module alleviates the time consuming tasks of developing a plan with the fewest number of setups, inspection point selection and coordinate calculation, and ultimately, creation of the CMM code. The IPEM’s evaluation and comparison functionality is not presently complete, but this is another area where it is expected to relieve the inspector of time consuming calculation tasks. It is not expected that this module, with it’s present reasoning system, produce the most suitable inspection plan on it’s own. There are proposed additions that will offer the expert inspector check-points where his interaction is appropriate. For example, an interface presently implemented, translates the process plan information to a display screen - showing each orientation of the part, the inspection points, suggested path. Presently, the user can step through each tolerance in the probe path. The goal of this interface is to allow the inspector the chance to vary or “tweak” the position of points if he chooses, or rearrange the tolerance order if need be. In advanced development stages, the system will “take note” of the adjustments he or she makes such that the machine will learn the ways of it’s operator and incorporate these concepts in future plans.

As proof of the functional ability of the system to date, a process has been created for the example part shown in Figure 1. As explained earlier, it has two design features, a slot and a hole, and a total of seven GD&T tolerance features. Three are datums, or reference features needed to establish a coordinate frame to locate measurement points in the CMM world. One is the datum reference frame that establishes the coordinate frame’s origin. There are two tolerances on the hole, position and cylindricity. The process planner consolidates the pair of measurement requests on that hole’s cylindrical surface to a single MR. The maximum number of required points between the two will establish the minimum number of points calculated for that surface. Finally, a flatness tolerance on the bottom surface of the pocket. Since all target surfaces are accessible to the CMM probe constrained to a z-axis approach vector, the IPEM resolves the inspection plan to a single setup. Since there is only a single DRF specified, all measurement requests will be grouped within a single “measurements-in-DRF” object. Notice there are only two MRs as expected.

Figure 5. (a) The Inspection part-model

Figure 5. (b) The setup property

Figure 5. (c) The Measurement in DRF attribute

Figures 5a, 5b and 5c show an expanded printout of the inspection planner part-model, setup property, and measurement in DRF attribute. Finally, Figure 6 shows a portion of the inspection points and path display to demonstrate to the user the suggested

CMM probe path.

Figure 6. Inspection points and path display

5. Conclusions

From a manufacturing viewpoint the IPEM produces useful CMM code. It uses a heuristic-based algorithm which incorporates reasoning about the part geometry and tolerances, as an inspector would, to produce a process plan and working CMES code. The reasoning process is represented in a format provided by an object-oriented language platform intended for the development of intelligent industrial systems. The IPEM as well as all modules of the RDS have represented their respective industrial processes in a cohesive part/subpart hierachy or part/property relationship. Because all modules can use a common platform, the high level of integration needed for feature translation is possible. Valuable tools such as property inheritance allow for efficient data integration and boundary-free access within the hierarchy using the tree search functionality. The development platform also allowed the flexibility to code directly in the base language without losing the continuity in part relationships.

The implementation of processes as parts or objects, permits data translation between modules forming new, domain specific features. Furthermore, the object-oriented paradigm provides a basis to build the single inspection part-model which imbed intelligent reasoning toward the analysis and performance of complex tasks as properties and sub-parts. Although classical AI search or inference techniques are not a main focus, much has been accomplished with domain specific heuristic rules and reasoning schemes. This aspect of the system might be thought of as the minimum required constraints or rules needed to interpret and evaluate a part toleranced in GD&T. Otherwise, specifics of the inspection “craft” cannot be reasonably entrusted to a static format such as an expert or knowledge-based system. In those systems, machines will only ever function on previously predicted situations and “pre-packaged” reasoning schemes. One of the key variations in the RDS from integrated manufacturing systems of comparable scope is the implementation of an intelligent system that learns concepts using neural-network technology[Pao and Hafez, 1992]. Though even a mild discussion of this technology has not been offered in this text, the present work in developing highly integrated data objects to represent diverse tasks, operations, or processes stands as a significant step in the implementation of the developing work in computer memory and concept formation.

6. References

1). Brown, Curtis W., *IPPEX: An Automated Planning System for Dimensional Inspection*, Proc. of the 22nd C.I.R.P International Seminar on Manufacturing Systems, June 11, 1990

2). CMES Programming Manual, LK Instruments.

3). Dimensioning and Tolerancing, ANSI Y14.5M-1982, ASME, 345 East 47th Street, New York 10017.

4). ElMaraghy, H.A. and P.H. Gu, *Expert System for Inspection Planning*, Annals of C.I.R.P., Vol. 36/1/1987, pp.85-89

5). Foster, Lowell W., Geo-Metrics II (Revised 1986 Edition), Addison-Wesley, New York, 1986.

6). Koetter, Michael W., John P. Eimermacher, Barbara A. Smith, Integration of Feature-Based Design and Manufacturing Process Planning, University of Dayton Technical Report, UDR-TR-90-135, December 7, 1990.

7). Kusiak, Andrew, Intelligent Manufacturing Systems, Prentice-Hall, 1990, p.402-409.

8). LeClair, Steve “Rapid Design System: Feature-Based Design,” Proceedings of 1991 IEEE International Systems Conference, Dayton, Ohio, August 1991.

9). Menq, C.H., H.T.Yau, and C.L.Wong, *An Intelligent Planning Environment for Automated Dimensional Inspection Using*

Coordinate Measuring Machines, Transactions of the ASME, Vol. 114, May 1992, pp. 222-230

10). Merat, F.L., G. M. Radack, K.Roumina, S.Reugsegger, *Automatic Inspection Planning within the Rapid Design System*, IEEE International Conference on Systems Engineering, Dayton, OH, Aug. 3, 1991

11). Merat, F.L. and Gerald M. Radack, “Automatic Inspection Planning within a Feature-Based CAD System,” J. Robotics and Computer-Integrated Manufacturing, Vol. 9, No.1, 1992

12). Pao, Y.H., K. Komeyli, S.R. LeClair, D. Shie, A. Winn, *The Role of the Episodal Associative Memory in Feature-Based Design*, I.F.I.P International CAD’91 Working Conference, Columbus, OH, Oct. 1991

13). Preparata, F.P. and S.J.Hong, “Convex Hulls of Finite Sets of Points in Two and Three Dimensions,” *Communications of the ACM*, Vol. 20, No.2, February 1977, p.87-93.

14). Radack,G.M., F.L.Merat, J.F. Jacobsohn, *Relating Form, Dimensioning, and Positioning Information in a Feature-Based Design Environment*, IEEE International Conference on Systems Engineering, Dayton, OH, Aug. 3, 1991

15). Radack, G.M. and F.L. Merat, “The Integration of Inspection into the CIM environment,” *Proc. of Hawaiian International Conference on System Science*, January 1990.

16). Spyridi, A.J.and A.A.G. Requicha, Accessibility Analysis for the Automatic Inspection of Mechanical Parts by Coordinate Measuring Machines, Proc. of the IEEE International Conf. on Robotics and Automation, Cincinnati, OH, 1990, pp. 1284-89.