From Design to Inspection: The Use of GDT for Predictable Assembly

By

Bill Tandler, Multi Metrics, Inc.
Mark Denman, Metrologic USA

- Bill -

Introduction:

We are makers of things, ever so many things, different things, amazingly complex and capable things, and somehow we have to get from the idea thing to the actual thing. In the good old days I thought it up, I sketched it, I made it, I inspected it, I assembled it, it didn't work, I sketched it again, I made it again and . . . . it worked !

In the better new - no longer good old - days, I either think it up, or I sketch it, or I make it, or I inspect it, or I assemble it, but I only do one of the five. In these better new days therefore, we need to communicate our ideas clearly or clearly understand those communicated by others, if we are to succeed in getting from idea to thing. And because of the complexity of our new things and the cost of getting from idea to thing, the people who come up with ideas need to be sure they're worth being turned into things before they release them.

As **G**enerally **D**epressing **& T**rying as it might be, GD&T is an empowering resource at the heart of all this. Namely it is the only tool we have with which to not only say so precisely what we want that others can do precisely what we say, but with which to discover that what we think we want is not what we actually need, and with which to research and develop alternatives which guarantee success.

But if we have good reason to dive into the GD&T pond, we must also know how to lead the swim team. Far too much GD&T is pure "decoration" requiring "interpretation". But interpretation is simply unacceptable, because the object must be to generate unique and unambiguos instructions which guarantee achievement of our goals. The gift of GD&T is the ability to help us discover and uniquely specify fault tolerant limits of imperfection which enable cost effective manufacturing, absolutely reliable metrology and guaranteed assembly and operation of the devices we invent, prior to drawing release. Rule based GD&T permits reliable "encoding" and "decoding" of part and feature functions, and relegates the concept of "interpretation" to the dust bin where it belongs.

As the title of the paper states, functional, rule based GD&T and Inspection are important partners which can make huge contributions to the process of turning ideas into functional objects. Rule based GD&T enables generation of absolutely precise statements about geometric requirements, which coordinate metrology can then turn into absolutely precise statements about how well said requirements are being met, and absolutely precise statements about what modifications if any, need to be made to improve manufacturing processes. The capabilities of GD&T and its interactions with coordinate metrology pose many opportunities and many problems on which this paper attempts to cast some light.

First some definitions:

- GD&T is a symbolic language for researching, refining and encoding the functions of each feature of a part by specifying permissible limits of imperfection which guarantee assembly and operation.

- Coordinate Metrology is a point cloud collection, manipulation and geometry processing tool with which to assess and improve manufacturing process performance, but also with which to research ways to correct manufacturing shortcomings in individual parts of the expensive kind.

Next, some GD&T Insights:

1. GD&T is complex, but no more complex than the world of real, imperfect machine parts.
2. GD&T must be encoded by human beings and is therefore subject to human error.
3. If the syntax of a GD&T encoded statement is improper, the statement is useless.
4. If the syntax of a GD&T encoded statement is proper, but the code does not represent the actual functions of a part, the code is also useless.
5. In the best of all possible worlds, the GD&T code is not only syntactically correct and function based, but also represents a highly fault tolerant design.
6. Whereas GD&T is often understood to specify manufacturing processes, it merely guides those processes by specifying the assembly and operational objectives of a part.
7. Syntactically legitimate and functionally valid GD&T specifies coordinate metrology requirements down to the last detail, is not open to “interpretation”, and must be followed to the letter.

Next, some Coordinate Metrology Insights:

1. Coordinate metrology software is complex due to the need to live up to the rule based requirements of GD&T on the one hand, and due to the need to deal effectively with the messy world of the imperfect, incomplete point clouds collected on the features of imperfect real parts.
2. Coordinate metrology software will never be “perfect”.
3. Coordinate metrology software which provides fully automatic processing is highly desirable, and although dangerously subject to software errors, can be thoroughly and reliably tested because it is totally independent of operator involvement.
4. Coordinate metrology software which relies on programmers and operators for implementation is marvelously flexible and after careful investigation, can be “misused correctly” to produce proper GD&T specified results, however is dangerously subject to being “used incorrectly”, and shall forever remain subject to potential software errors which can never be fully documented due to the ever present human element.
5. Coordinate metrology is always beholden to measuring uncertainty, and it better be small compared to the specified tolerances.
6. Coordinate metrology is widely understood to serve primarily to separate the wheat from the chaff with the ability to say exactly why. But its main objective is to help manufacturing reduce the chaff by refining processes, and in the case of expensive parts, even convert chaff into wheat.

Next, some Examples of Bad, Good and Better GD&T:

Here follow four examples of GD&T running from “purely decorative” and therefore totally useless, to “fault tolerant and functional”, therefore highly useful:

1. Purely “decorative” GD&T containing significant syntax errors and non-function based controls,
2. Largely “decorative” GD&T suffering no syntax errors and guaranteeing assembly, but not operation,
3. Merely “functional” GD&T suffering no syntax errors and guaranteeing both assembly and operation,

4. “Fault tolerant & functional” GD&T suffering no syntax errors, guaranteeing assembly and operation, but doing so more reliably and at lower cost than alternative 3.

Application and analysis of GD&T is, as usual, based on the assembly and operational functions of the considered part, which is to mutually locate two additional parts. Of these, part 1 is a motor with an axis-concentric cylindrical mount, and part 2. is part of an assembly which contains a drive train with a spindle which M10 bolts in four threaded bores in part 2. are responsible for making coaxial with the motor’s axis during assembly. The bottom face and central bore of the considered part interact with the face and cylindrical mount of part 1, whereas its top face and bore pattern interact with the face and four fixed fasteners in mating part 2.

<u>Approach #1 – Syntax Error Laden “Decorative” GD&T.</u>:

In Figure 1. below, almost every GD&T call-out is syntactically or functionally inappropriate. The Parallelism tool applied to the planar surface opposite Datum Feature A is incapable of controlling its location, and must be replaced with the Surface Profile tool. The all around Surface Profile control references Datum Features which it controls, which is nonsensical, and fails to reference Datum Feature A which is essential. It also includes a Tolerance Zone Size modifier, which is nonsensical and illegal. Finally, both Position Feature Control Frames are missing the tolerance zone shape modifier “Ø” to designate cylindrical tolerance zones, and reference only Datum Features B and C, which must be preceded by Datum Feature A in order to be functional. In addition, the fact that Datum Feature C consists of two coplanar surfaces is merely hinted at by the connecting extension line, instead of being stated explicitly as shown in Figure 2. below.

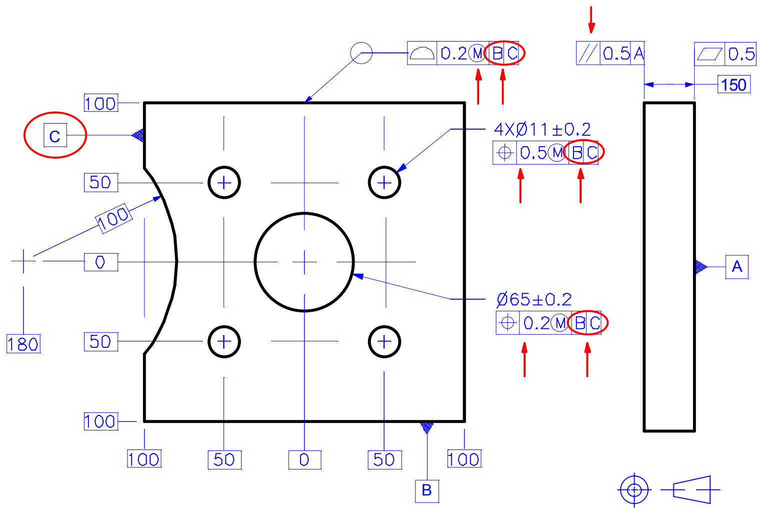


Figure 1. Purely "Decorative" GD&T

<u>Approach #2 – Syntax Correct but Largely “Decorative” GD&T</u>:

In Figure 2. below, all the syntax errors in approach 1 have been corrected. Datum Features B and C have been properly controlled as part of the all around Surface Profile control which now properly references only Datum Feature A. Furthermore, the Position Feature Control Frames for all five bores now reference A as the primary Datum Feature, fully encoding its critical function, and the Parallelism tool has been replaced by the Surface Profile tool, actually enabling the intended location constraint. In addition, the tolerance zone shape modifier “Ø” has been added to both Position Feature Control Frames to specify the necessary cylindrical tolerance zone, and finally, the fact that Datum Feature C consists of two coplanar instead of just a single planar surface is stated explicitly. However this approach still merits the designation “decorative” because the location of the central bore is controlled relative to functionally irrelevant Datum Features B and C, which the four bore pattern will actually determine during the assembly process. In addition, both Position Feature Control Frames include the Tolerance Zone Size modifier (M) which encodes the clearance function, when in fact all the features serve centering functions.

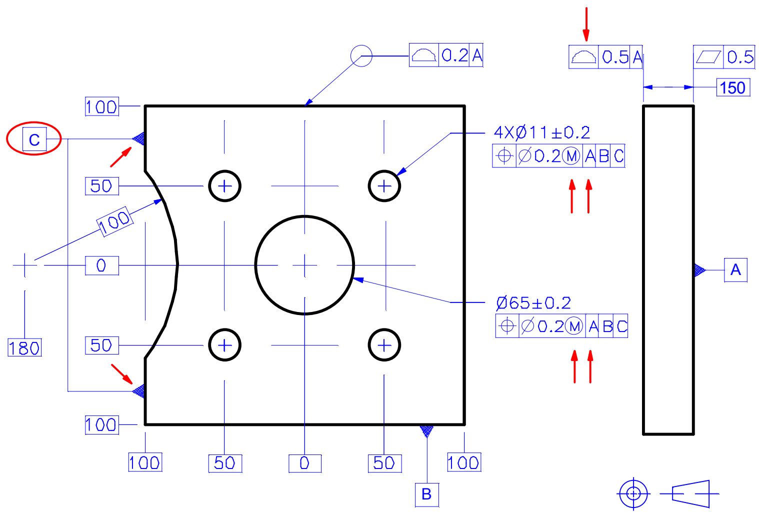


Figure 2. Still Largely "Decorative" GD&T

Approach #3 – Syntax Correct and Functional GD&T:

In Figure 3. below, syntactically correct GD&T has been replaced by syntactically correct, functional GD&T. The four bore pattern, which will actually locate the plate relative to mating part #2, has been made Datum Feature B, and the Tolerance Zone Size modifier has been made (S) to encode its "aiming" or "centering" function. The tolerance on the periphery has been loosened substantially to represent its relative unimportance, and the central bore has been controlled relative to the bore pattern using Tolerance Zone Size and Mobility modifiers (S) to represent its "centering" function. However, we could do better. Namely we could create a much more fault tolerant, capable design by making just two of the locating bores take on the aligning and locating responsibility, allowing us to tighten their tolerances since they will no longer compete with the remaining two, and loosen the tolerances on the remaining two, which assume purely a clearance function. See approach #4 for details.

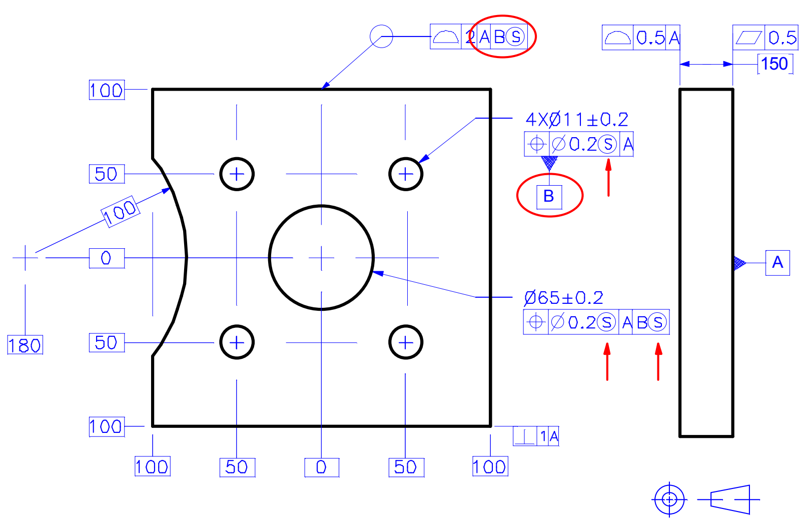


Figure 3. Merely "Functional" GD&T

Approach #4 - Syntax Correct, Fault Tolerant and Functional GD&T:

In Figure 4. below, we see that two of the four holes in the bore pattern have been reduced in size and the other two enlarged. The two smaller bores also have tighter size and Position tolerances, include a Tolerance Zone Size modifier (S) to encode the centering function, and are identified as Datum Feature B officially assigning them the aligning and location constraining role they actually have. The remaining two bores, with significantly looser tolerances, now serve only a clearance function which is clearly encoded by their larger nominal size and the Tolerance Zone Size modifier (M). We also see the true function of the central bore explicitly encoded by using the Tolerance Zone Size and Mobility modifiers (S) in its Position Feature Control Frame. Finally we see the composite nature of Datum Feature C much more clearly stated by attaching Datum Feature "suction cups" to both of the applicable surfaces, and can instantly understand the combined impact of Datum Features A and B with the help of the Datum Reference Frame axis labels X[A,B] and Y[A,B]. This code is not only good enough to dump into a coordinate metrology software system capable of automatically converting it into an inspection process and completing the analysis in milliseconds with no operator / programmer involvement, but also guarantees that if the parts meet their requirements, they will also be eminently functional, two of the most important objectives of GD&T.

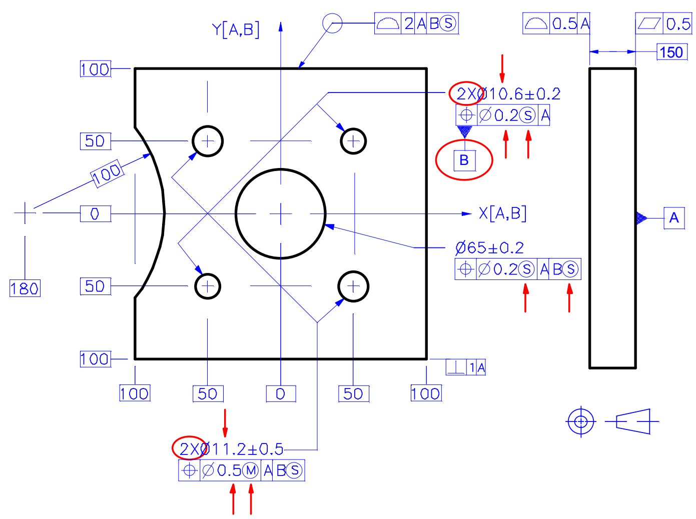


Figure 4. "Fault Tolerant & Functional" GD&T

- Mark -

Next, some Examples of the Impact of Bad and Good GD&T and of Bad and Good Coordinate Metrology Software on Inspection Process Outcomes:

Example 1.

I once asked a group of inspectors, "Why do we inspect the parts we manufacture?" They all answered, "To make sure they are good parts."

I argue that, although this may, indeed, be the *reason*, it is only part of the *job*. If the inspection is completed and the part deemed to be nonconforming, I believe it is our responsibility to help decide if the part can be salvaged and to help create the salvage plan. If the part cannot be salvaged, we, the quality control community, should be working together with manufacturing to glean the information required to ensure the nonconforming condition can be avoided in the future.

So what is a “Good Part”? While the majority of those I asked stated it was one that conforms to the drawing, a few (apparently used to working with bad drafting), maintained that a good part is one that assembles well and functions as intended. In fact, the latter alternative is correct, and if the drawing leads to its rejection, the drawing needs to be revised. Simply put, the functional objectives of parts do not always get translated into drawings or inspection processes. This is largely a result of archaic tolerancing techniques and incorrect Datum Feature selection.

If we ask ourselves, “Why do we perform a part alignment when inspecting a component?” shouldn’t the answer be, “To establish a Datum Reference Frame that most closely simulates the assembly conditions of the part being inspected”? But most Datum Features are selected for ease of manufacturing or inspection, and have absolutely nothing to do with assembly. If the assembly function of the part is not represented in the drawing, what then is the point of the inspection? Furthermore, in cases where datum features have been chosen correctly, what is the value of an inspection effort if those datum features are improperly used due to a lack of capability in the inspection equipment (software)?

A very commonly used, simple DRF is one based on a plane and two holes, where the plane is the primary and the holes are the secondary and tertiary datum features respectively. This datum feature scheme is familiar to everyone. Why then are there still softwares in use that cannot construct this DRF correctly? Many software systems force us to establish this DRF using a plane, a line and a point. The problem here is that neither a line nor a point is a feature of size and, therefore, we fail to simulate the actual assembly conditions of the part and prohibit taking advantage of a material condition bonus if authorized by the modifier (M).

So let’s ask why we make holes in things. Sometimes it’s just for fun, but mostly it’s because we want to put a pin in the hole. The pin in question can be anything from the bearing surface of a camshaft to a threaded stud. But the interaction between the hole and the pin is still the same. If you put any round pin into any round hole two conditions arise just because of the physical attributes of the two features. First, any movement of the pin in the hole is contained within a circular zone, which, strangely enough, resembles a true position tolerance zone. Second, if you reduce the size of the pin and/or increase the size of the hole, the amount of relative movement available increases. This is exactly what is described by Maximum Material Condition.

Another DRF, arguably the simplest, is that established using three planar surfaces. This also causes issues for some inspection softwares, which we will discuss shortly.

Recently, while working at a large manufacturing company, I was in a discussion with the manager of the drafting group about why he should change the drawings for a set of parts the company manufactured, to improve the GD&T. The current drawings had incomplete DRFs and basic dimensions which were all defined from a single point, which was not part of the DRF. Even though we were working with a family of parts that all served the same function and were all built the same way, each drawing was markedly different from the next. One particularly heated argument stemmed from these statement,: “GD&T is an interpretive standard, and I choose to interpret it this way”, “I’m not required to adhere to ASME Y14”, and my favorite, “You keep talking about Degrees of Freedom, I don’t know what you mean by Degrees of Freedom”. Faced with this kind of resistance (and ignorance), many members of the quality community have walked away and, “done the best they can with what they have”.

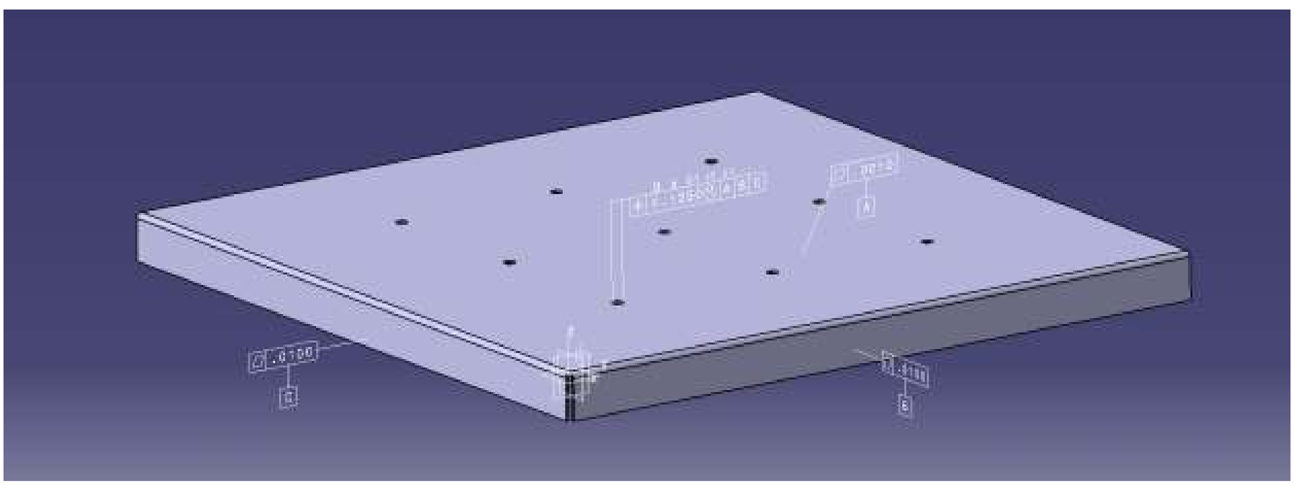
The problem with this is that each individual can have a different idea of what “best” is. This conversation occurred AFTER I had dealt with the manager of a parallel group and had the drawings for another part family standardized. GD&T was applied thoughtfully and correctly, using the assembly features for datum features and the functionality of the other features being manufactured. This improvement meant that CMM programs were standardized, allowing new programs to grow from old ones and dramatically reducing the time taken to generate them. Most important, however, was the impact on the layout inspection process. Since the drawings now defined the part specifically and succinctly, the layout inspectors also used consistent and controlled methods of inspection, which eliminated the arguments that had occurred between CMM data and layout inspection data.

One of my colleagues asked me why I was so dead set on using GD&T when a simple +/- tolerance had always been good enough. I fetched a plate with a perpendicular pin, and a block with a hole perpendicular to the bottom face. I asked him to put the block on the pin and said, “Can you move that block in the square pattern described by a +/- tolerance? Furthermore, does a +/- tolerance properly control the orientation of a hole? Finally, do you realize that by putting the pin in the hole you have encountered Material bonus?” He scratched his head and said, “You win, let’s use GD&T.” This was an extremely simplistic argument but it showed for just one feature/tolerance combination the functional relationship GD&T has to assembly. The GEOMETRIC control imposed by the form and function of manufactured features can only be properly ENCODED using GD&T.

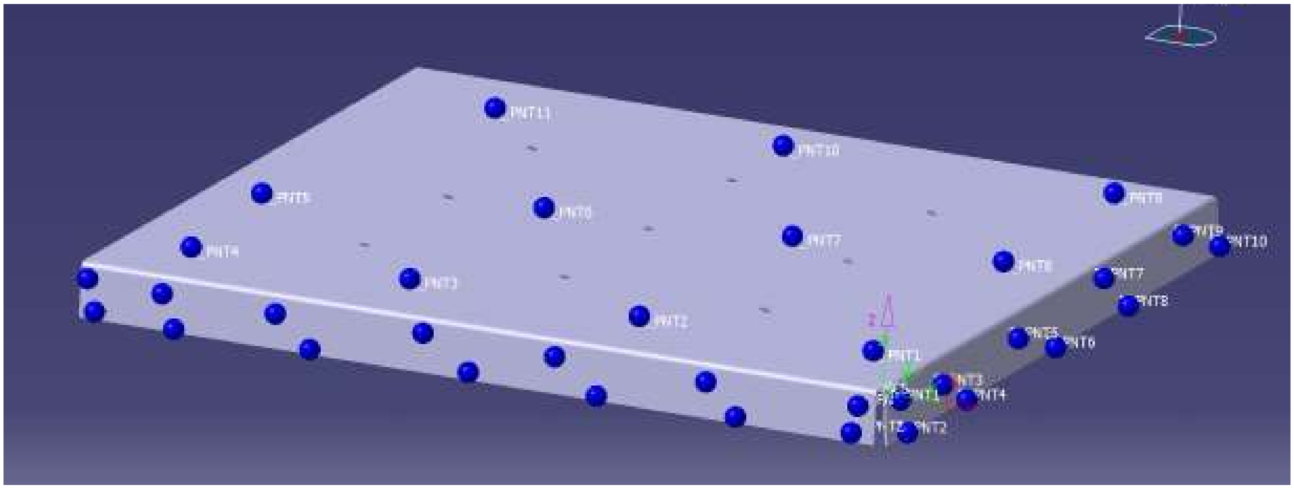
I am a devout disciple of GD&T. I have seen its value since I first learned it as BS308 while in high school in England. While working at many manufacturing facilities during my career I have seen both ends of the spectrum in its use. I take it as a personal challenge to try and improve its application whenever I see the need. I believe that is one of the responsibilities that we in the quality control community have to our industry.

The following example is simple and should not have posed any problems in its inspection. The data has been changed slightly to protect the innocent. As you can see this is a simple part. It will be placed in an assembly, locating on it top face and the two sides shown in this view. It is a fabrication in that it was made from a single sheet of metal whose sides were bent to nominal 90 degree angles to the primary face. A pattern of holes were drilled, and threaded studs would pass through these holes and into corresponding holes in the mating part.

The Engineer took pains to put in what he thought was, “good GD&T”

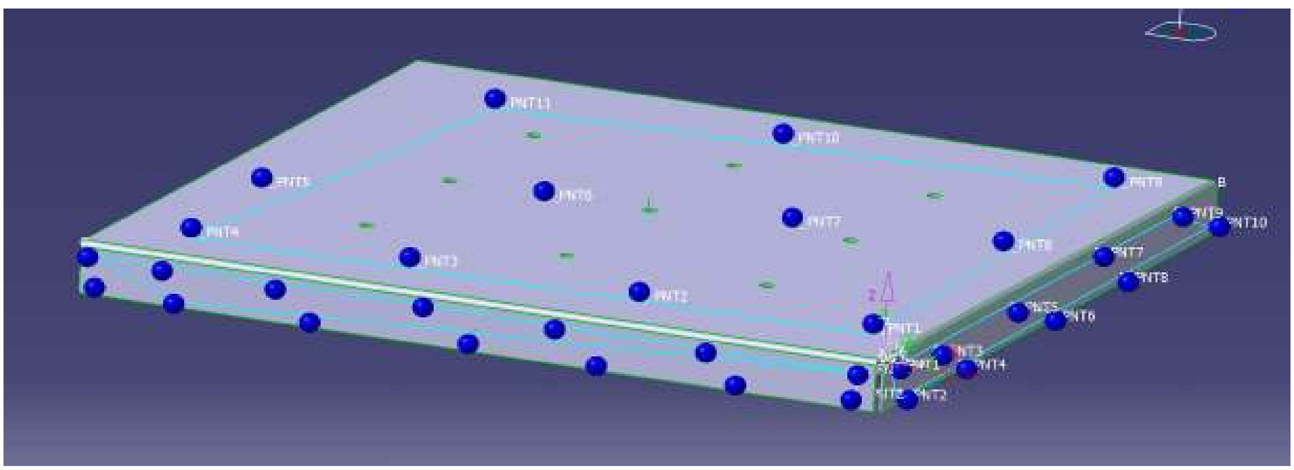


He defined the datum features using the mating assembly features. He controlled the form of those datum features but he neglected to control their mutual orientation. . He also authorized material condition bonuses on the holes. The Inspector collected data on all the datum features and established the alignment

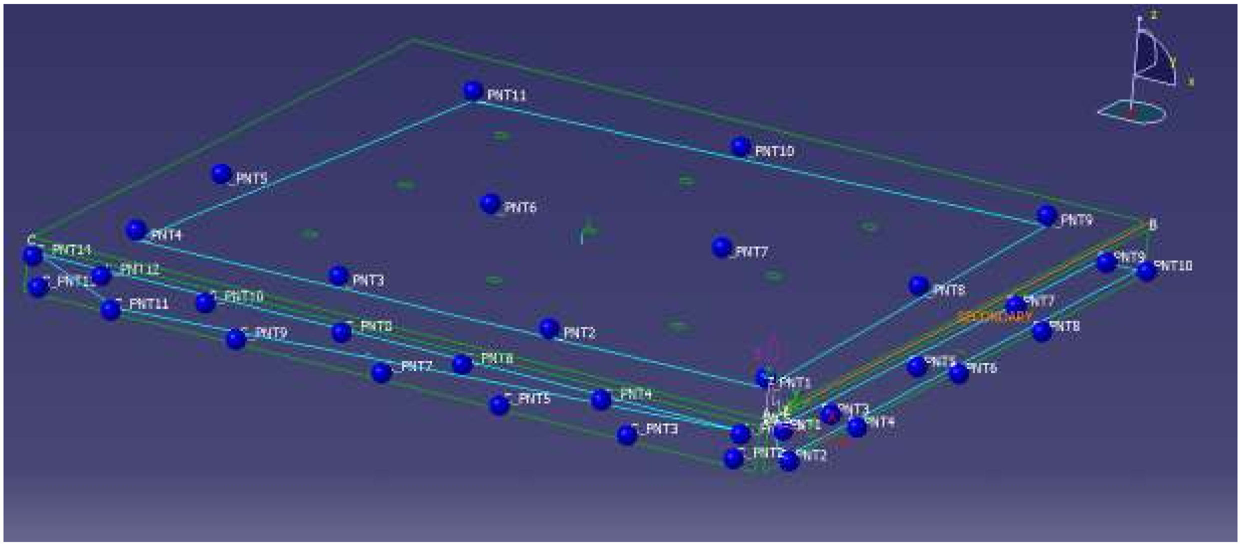


As you can see he made sure to collect enough data points to make a good alignment.

The software used by the Inspector did not process datum features as they should have been. In fact the software cannot make a DRF using three planar features. It promotes a PLANE – LINE – POINT alignment method. The Inspector was very capable in the use of his software, and knowing he was forced to use a plane, a line and a point, he very deliberately made sure the software got the most accurate example of each he could make. He measured all three datum features as planes.



This next step is a common operation in the use of some measurement softwares. A line was required as secondary alignment feature and a line was constructed by intersecting the primary plane A with the secondary plane B.



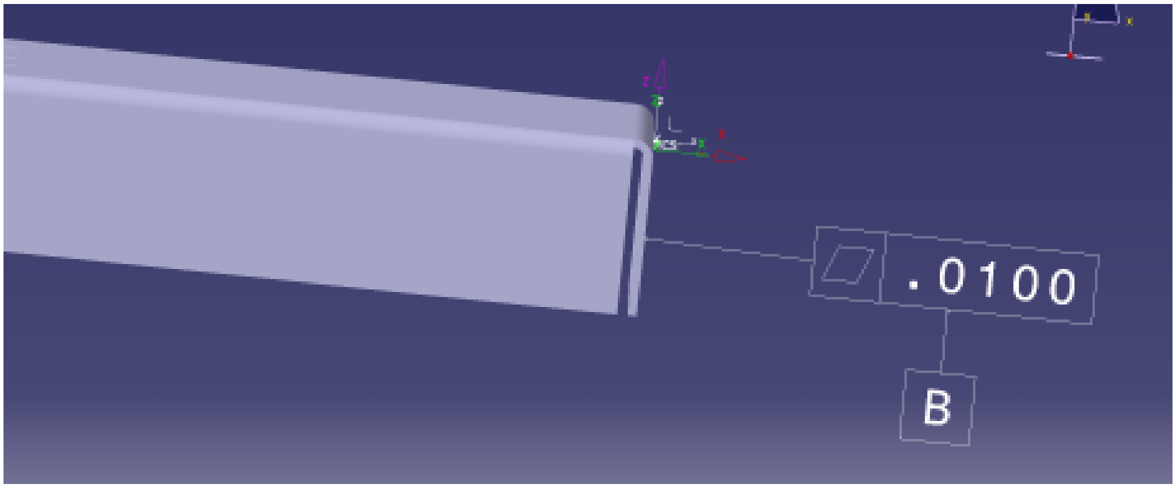
The next step was to construct a point; this was done by intersecting this previously constructed line with the tertiary datum feature C. Then by using the **measured** Primary plane A and the **constructed** secondary line along with the **constructed** tertiary point an alignment could be established. However, although it located the part wonderfully; unfortunately it did NOT represent the specified DRF.

Per ASME Y14.5-2009

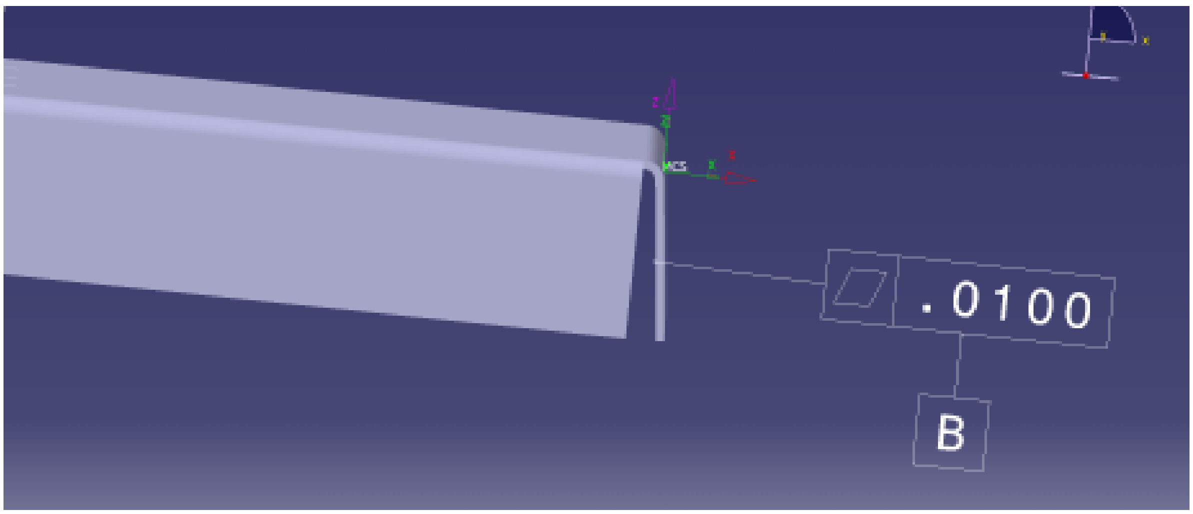
> **4.8 DATUM FEATURES**
>
> A datum feature is selected on the basis of its functional relationship to the toleranced feature and the requirements of the design. See Figs. 4-5, 4-6, 4-36, 4-37, and 4-38. To ensure proper assembly, corresponding interfacing features of mating parts should be selected as datum features. However, a datum feature should be accessible on the part and of sufficient size to permit its use. Datum features must be readily discernible on the part. Therefore, in the case of symmetrical parts or parts with identical features, physical identification of the datum feature on the part may be necessary.

The line that was created using Datum Feature B did not represent the impact of a “Corresponding interfacing feature” nor did the point created using Datum Feature B.
If the software had been able to analyze and deal properly with the imperfections of the datum features it could have established a functional DRF and produced legitimate results. Unfortunately the part was NOT perfect, which is normal, and the software WAS inadequate, which is all too common.

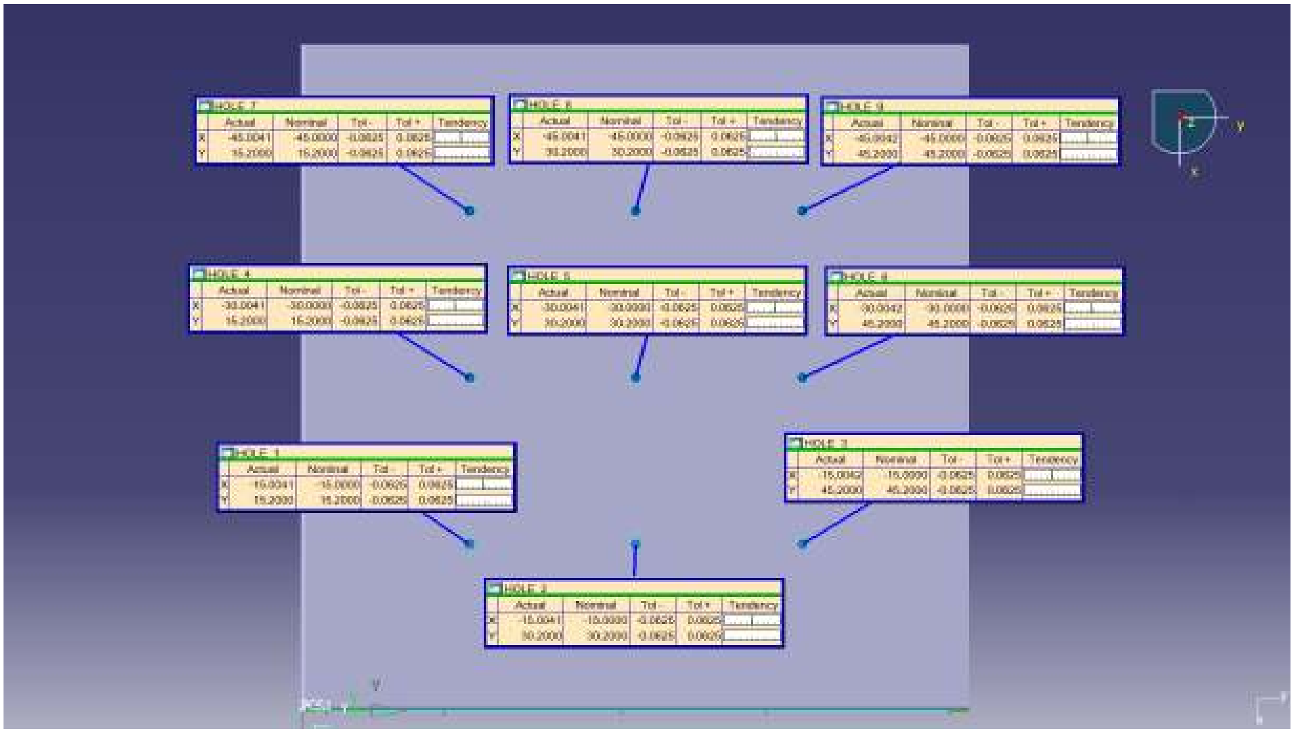
In fact, the part that was expected to look like this:..



actually looked like this:…



So when the alignment was made using the intersections based line and point it was generated at the position shown above. And when the holes were examined for position (something else the software did poorly as it output all data in the current alignment regardless of the DRF specified in the FCF), this is what was presented.

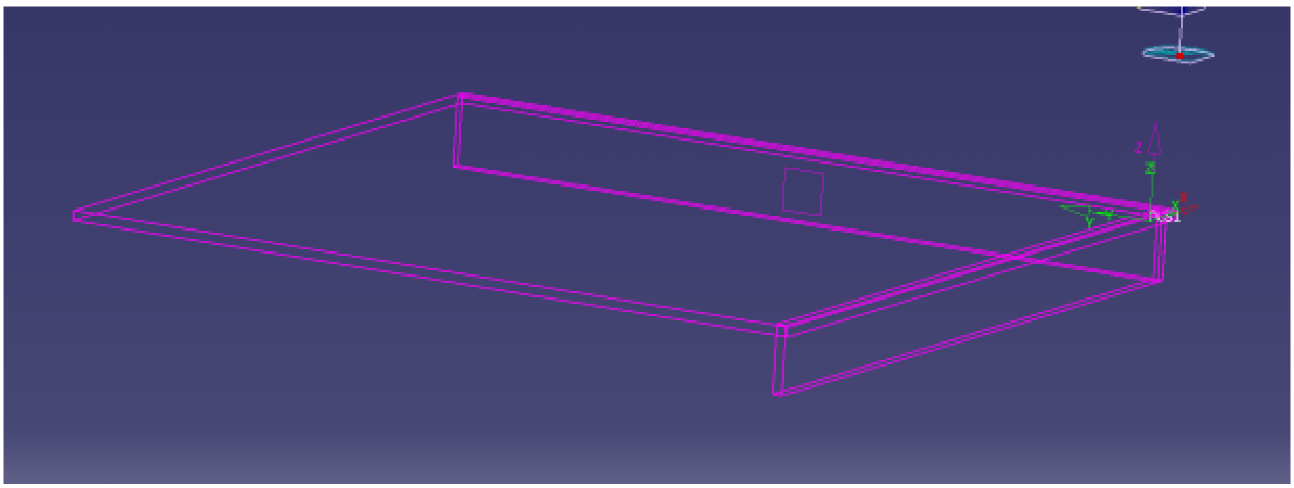


There were very small deviations and everything was well within tolerance. The part was accepted and sent on to assembly. If the software had established the Datum Reference Frame per the ASME standard it would have simulated the normally required physical datum feature simulators, and made Datum A the tangent plane on datum feature A

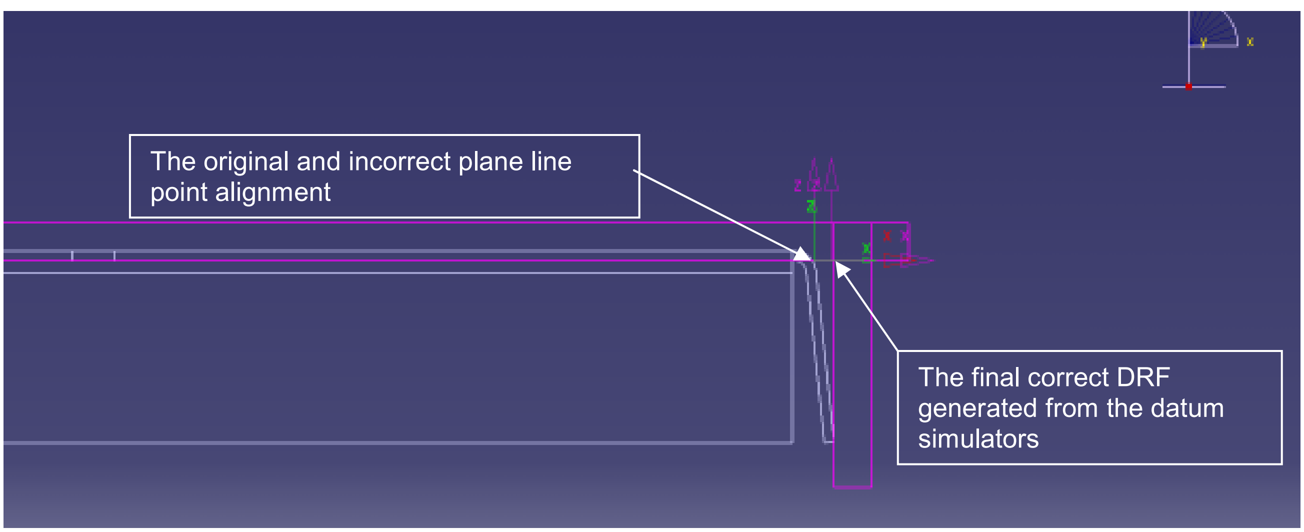
Datum B the singly orientation constrained (perpendicular to A) tangent plane on datum feature B

passing through the two high points on the secondary datum feature,

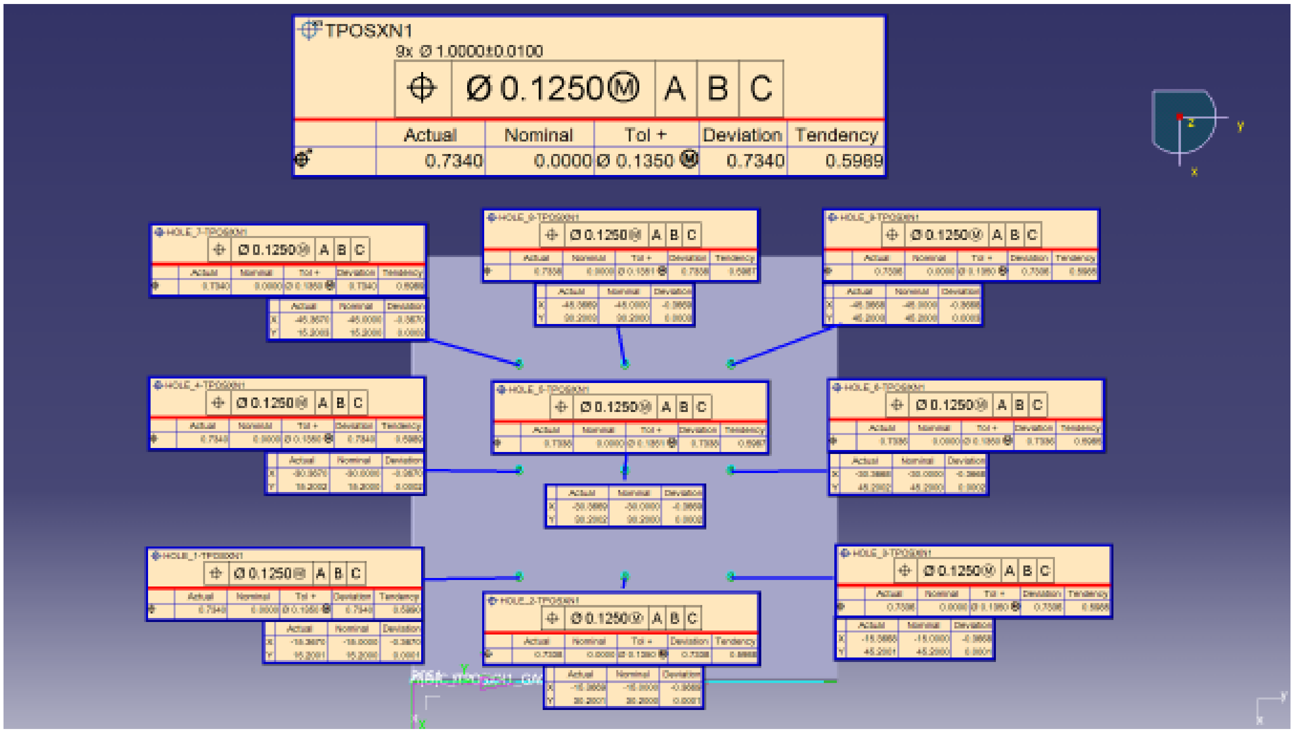
and datum C the doubly orientation constrained (perpendicular to both A and B) tangent plane on datum feature C passing through the highest point on C as shown in the associated figures.



In accordance with the ASME Y14.5 2009 Standard, the origin of DRF[A,B,C] is located at the junction of the three simulators.



Using software that correctly applied the GD&T as specified on the drawing, in spite of its failure to constrain the perpendicularity of B to A and C to A and B, generated this report.



It was more complete, Material condition was applied to the tolerance, and a positional value was calculated This analysis of the data found the part to be significantly out of tolerance due to the unfortunate tilt of datum feature B relative to A.

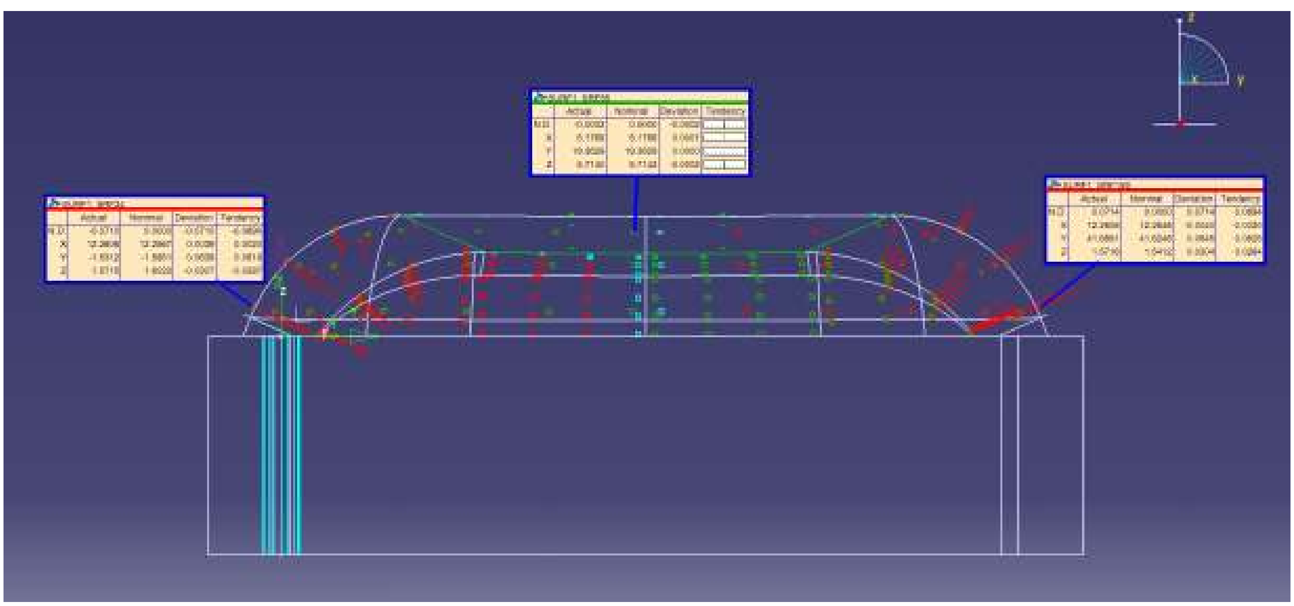
So there are two factors that could have improved this situation. First, the inspection software should have been capable of analyzing the inspection data per the ASME standard, which would have caught the nonconforming part. Second, the GD&T applied on the drawing should have been better. In fact the drawing should have been rejected as inadequate to warrant an inspection effort. The drafter thought he was applying good technique because he controlled the form of the datum features, and he did. He had a flatness requirement on the primary datum A and on the secondary datum B. This was inadequate. If the secondary datum feature B had a perpendicularity requirement both the form AND the attitude could have been controlled in one tolerance. This would have made it an inspection item which would have rejected the part right from the outset; not for all the right reasons but it would have caught the underlying problem.

Example 2.

This next example involved a very simple tool. An automotive die, used for stamping a car body panel, the design had reasonable GDT. Datum features were the top mating face of the die and two locator holes which would position the male and female halves of the die during its use. A surface profile tolerance was placed on the functional surface of the die, which should control the shape of the surface and also its location relative to the locating holes. This is an important feature as both the male and female halves would be physically located in the press by the holes and any mismatch of the profiles would result in the stamped products possibly being the wrong shape or too thin in a localized area. Once again the inspection software promoted a plane, line, point method of alignment, Material condition was ignored. The inspection software also could not analyze the surface feature as an entity; it presented the inspector with an individual error for each and every point collected on the surface feature.

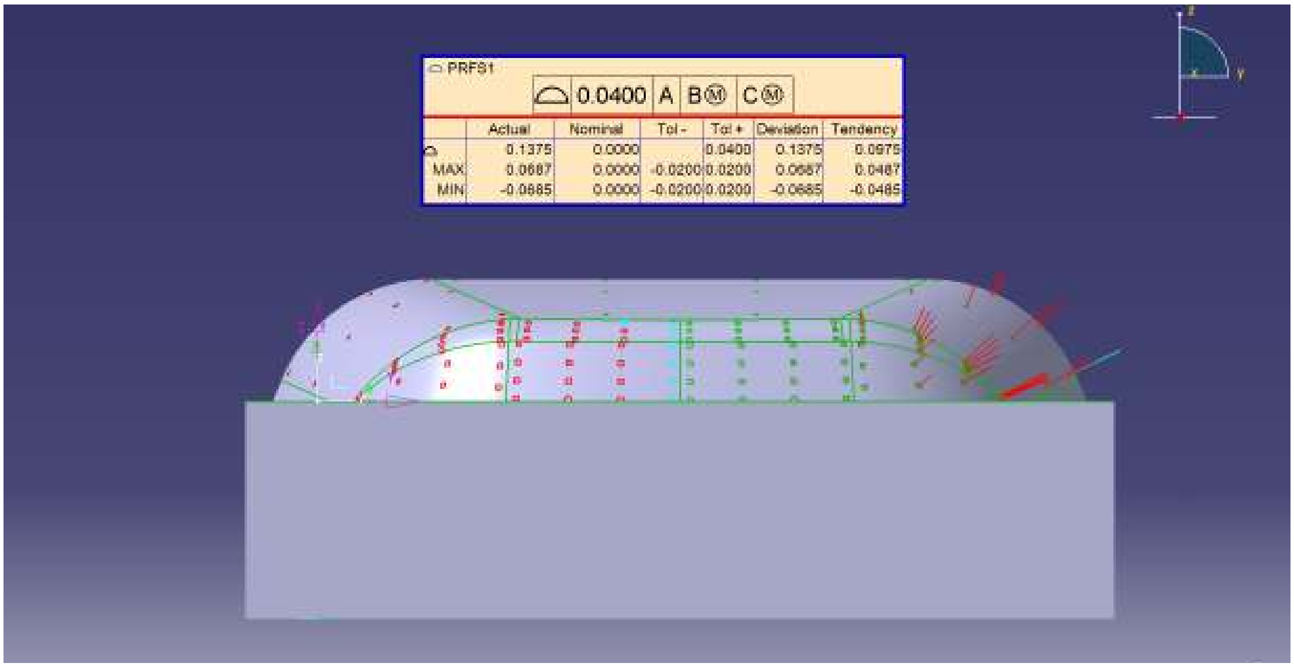
In the case of this die, the inspection was completed on the die and the data analyzed. The die failed the inspection.

In this first set of data you'll see that the points to the left of the screen have a negative deviation and the points on the right have a positive deviation while the points around the center of the part are close to perfect. The failure of this part caused a problem with the schedule, putting even more pressure on the manufacturing group to produce a good part. When the second part was made and inspected it also failed the inspection. When the inspector used the inspection software to examine the data he was dismayed and confused to see that the data very closely resembled the data from the first die.



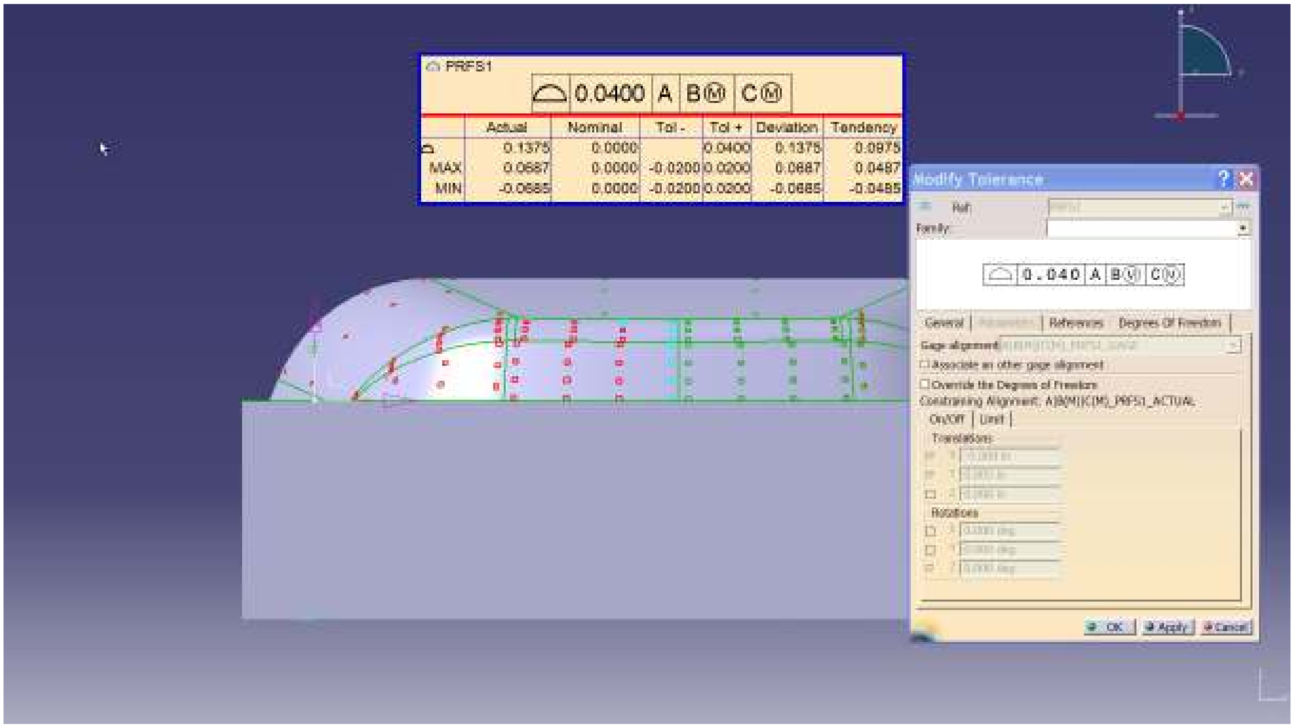
Not only were the magnitudes of the deviations very similar but the pattern of the deviations was also similar. This had the manufacturing group scratching their heads as to what the problem could be. How could they have made two parts incorrectly? They had examined the process on the first die and found the tool used to machine the surface had been the wrong tool, damaged, blunt and did not fit the tool holder correctly. With the second die they had used new sharp tools, confirmed they were correctly located in the tool holder and to be sure of making a good finish had used a slower cutting speed. So all possible means of ensuring a good part had been implemented and it looked like they had produced the same bad part again. WHAT WAS GOING ON????

At this point I was not involved in this process and was unaware of the issues. One of the engineers on the project knew I was in the plant, that I had experience with the inspection software in use and had heard I had some software that could “do magic” with analyzing data. I read in the data collected from the first inspection and applied the GDT tolerance per the standard. The DRF was made using the plane and two cylinders Material Condition was applied, and the part failed the analysis with very similar results to the original inspection. Any differences were due to the material condition of the datum features B and C.

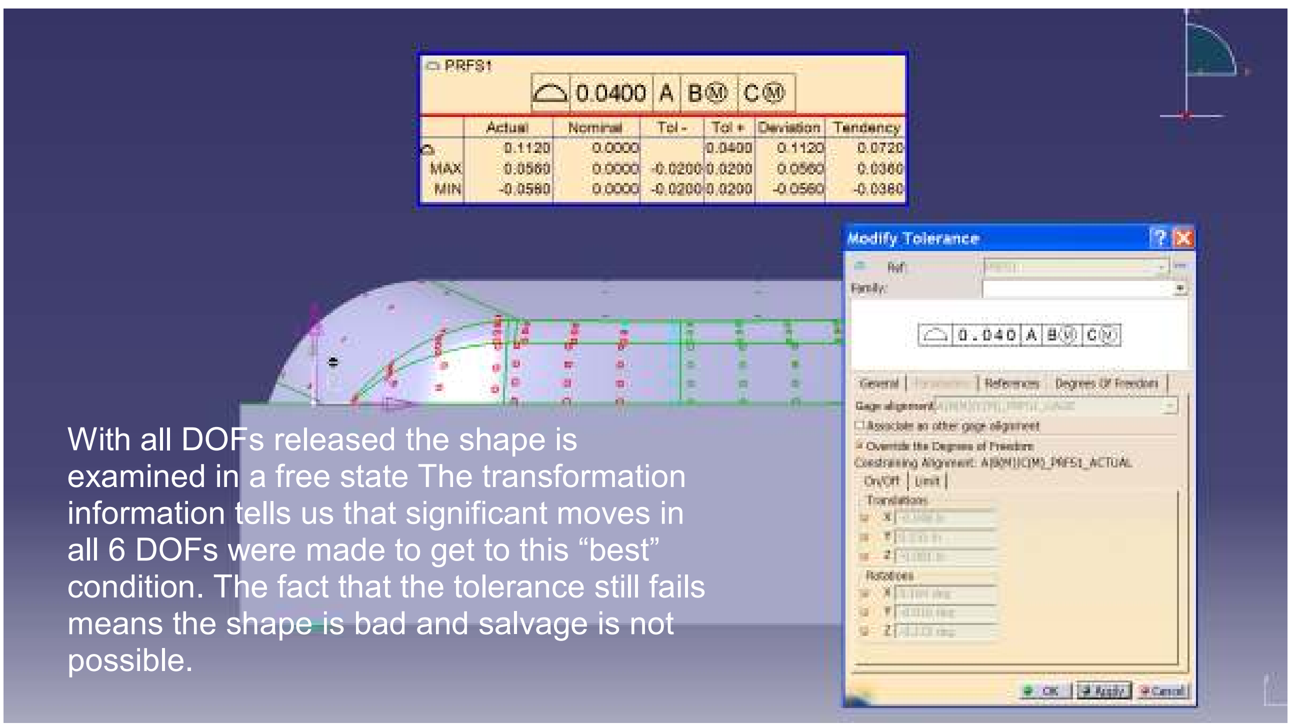


Since a profile tolerance consists of two elements, the shape, and the location of the shape. Each element should be individually available in the analysis. This is where the software, and indeed the process, can rise above the mere tollgate function of a pass/fail inspection system. The profile can be analyzed with the **D**egrees **O**f **F**reedom released. This will allow the shape alone to be considered regardless of its location

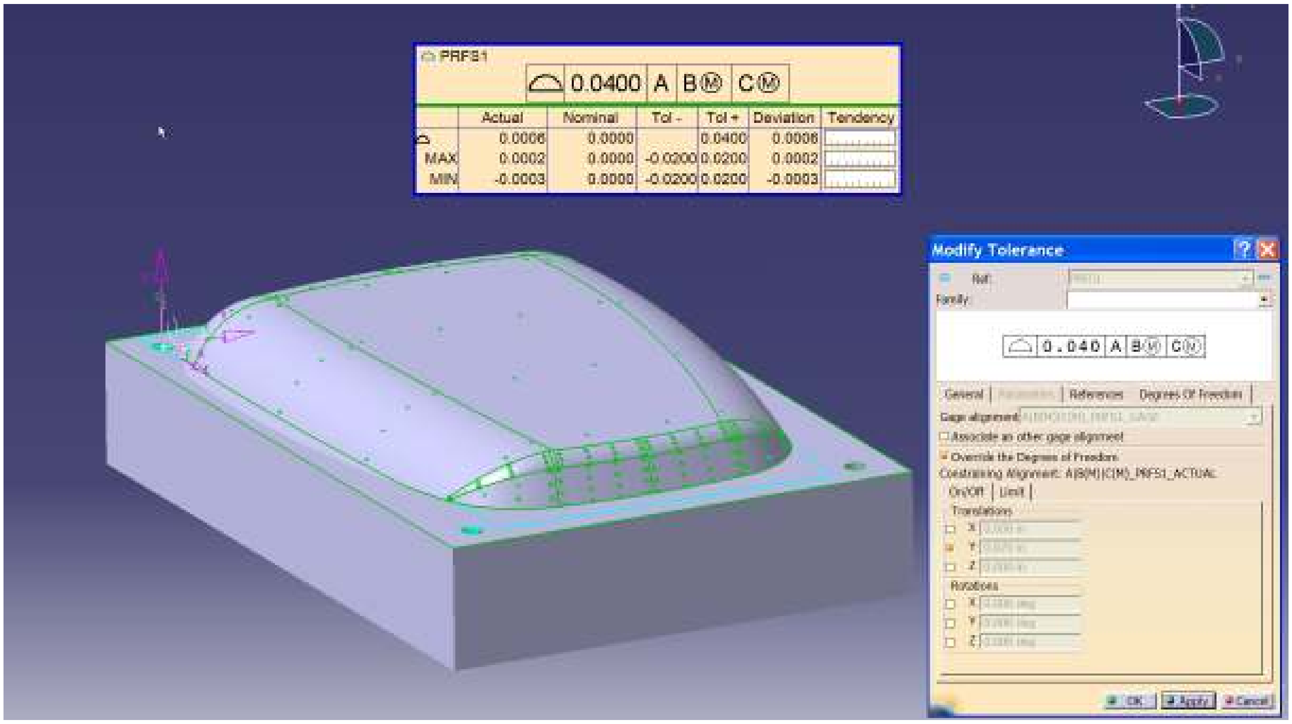
The following shows how the software allows the user to Override the Degrees of Freedom, there is also a panel showing the translations. Notice the X and Y translations and the Z rotation boxes are already selected and there is a 0.003” translation value in the Y Axis. These DOFs are allowed due to the Material bonus, the values in the fields allow us to see how much the Gage has moved from its NOMINAL position. By Overriding the DOF we can look at the feature in a controlled sequence that will give information critical to salvaging the part and improving the process.



With the first set of data we released all DOFs and looked at the profile value.



The second set of data was examined in the same way. The DOFs were released

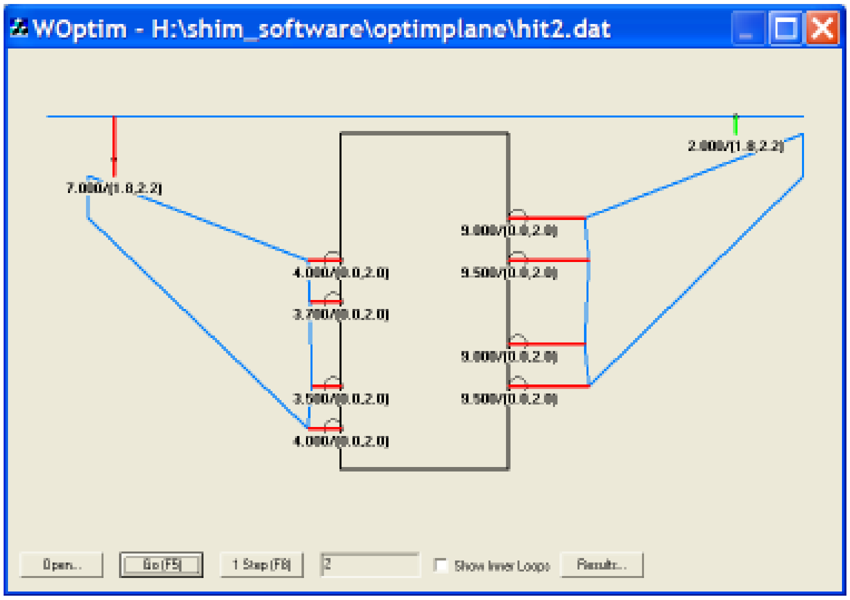


We could see that only a Y axis translation was required to make the surface data almost perfect. By using this information, we were able to make a salvage plan to machine the locator holes oversize, install a bung in each of the holes and re-machine the holes offset in Y by 0.079”. On investigating the reason for the error in the first place, we found that since this hole pattern is made in all the dies, the holes were manufactured using a separate program from the surface, and the machine was indexed off the side and end of the block. When the surface was manufactured the operator also indexed off the side and end of the block, and then entered an offset to the nominal position of the locator holes. He mistyped the offset, entering an “8” when he should have entered a “0” giving what was ultimately a 0.080” offset in the part.

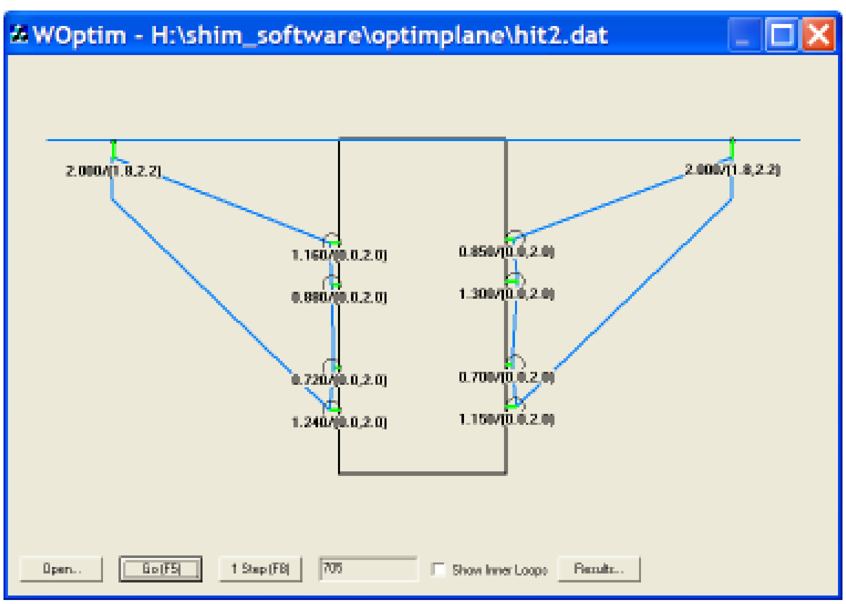
This is a very simple example of how a capable tool can be used to not only inspect a part, but also investigate and communicate the process critical information to manufacturing, allowing the quality control tools and personnel to be a manufacturing aid.

This process can be and has been advanced even more. To the point that the quality control software is used to drive a CNC machine (Milling, water jet etc.) to collect measurement data. GDT tolerances are used to analyze surfaces and patterns of holes. Since profile and position tolerances actually generate axis systems, these “GAGE COORDINATE SYSTEMS” are used to repost the CNC media through the gage transformation matrix, basically aligning the program to the part. This is not a simple 6 point alignment as provided by some CNC manufacturers but a full 6 DOF alignment based on GDT standards.

Taking this same philosophy one step further gives us tools like the shim design capability shown here. Data collected on surface profiles on parts manufactured at two separate geographical locations, can be combined due to correct and consistent application of DRFs and tolerances to allow the accurate prediction of a shim profile required, ensuring a good assembly between mating parts.



Allowing us to calculate the gaps at nominal locations and iterate through the available options until the shape and size of the applicable shim is determined, this data can then be passed onto the manufacturing group to be made into a CNC program for shim production.



I have contended that we, the quality control community can and should better support the manufacturing community, be a bigger part of the solution to manufacturing issues, and impose less of a toll gate process. I continue to hold that belief and challenge all of us to do better in this regard.