This mms. is published in *Proc. ASPE Summer Topical Meeting on Tolerance Modeling and Analysis*, University of North Carolina at Charlotte; ASPE Press, Raleigh, NC, July 2002.

DATUM SYNTAX AND SEMANTICS IN GEOMETRIC TOLERANCING

by

Herbert B. Voelcker
CORNELL UNIVERSITY
Ithaca, NY

May 2002

Keywords: geometric tolerance, GD&T, datum, datum reference frame

Summary

Datums are used in geometric dimensioning and tolerancing (GD&T) to establish spatial reference systems called Datum Reference Frames (DRFs). GD&T languages, such as the Y14.5 American Standard, provide a 'callout block' syntax for declaring datums and DRFs; the associated semantics is defined through increasingly complicated sets of rules and procedures. This paper proposes a new approach to datum and DRF declaration that seeks simplicity and clarity without disruptive changes to the overall structure of GD&T. The essential principle: dispense with complicated rules by stating, either explicitly or by reference, every constraint governing the induction of every datum in a DRF. The paper concludes with a summary of issues and extensions.[1]

---

1. INTRODUCTION

The distinguishing characteristic of geometric tolerancing is the use of *containment zones* for controlling the spatial variability of 'features' of mechanical parts. Containment zones are spatial solids, i.e. regions of space with homogeneous interiors and well defined boundaries. Zones that control a feature's intrinsic properties – mainly size and form – 'float' in space, and may be positioned in any manner that enables the containment criterion to be met. Zones that control extrinsic properties – typically a feature's orientation or location relative to other features – must be positioned relative to those other features, which are termed *datum features.* Thus an extrinsic tolerance specification must designate, in addition to the tolerance *per se*,

1) the datum (reference) features,
2) the 'material condition' of each datum feature when that term is relevant[2],
3) the order in which the datums are to be induced to establish a Datum Reference Frame (DRF) on an actual part, and
4) all inter-datum relations (generally called *constraints* in this paper) needed to establish the DRF.

Current GD&T practice, as codified in the American Standard Y14.5 [ASME 94a] and its ISO counterparts, honors these requirements, but not always effectively. We illustrate below some current problems and propose more effective mechanisms.

* *

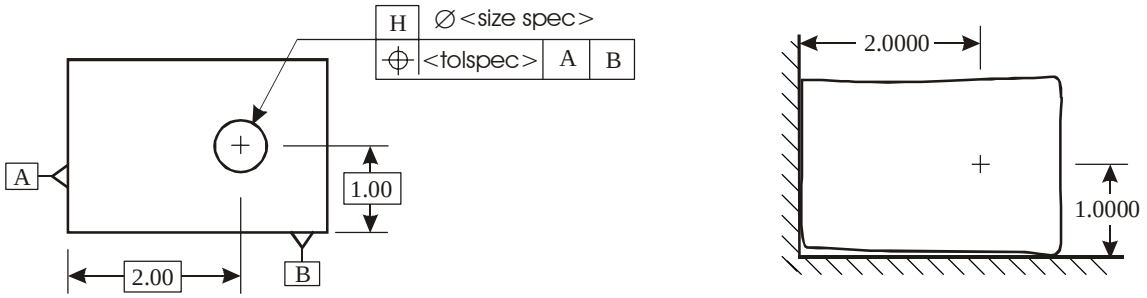
[1] A popular version of some of this material was published recently in the trade press; see [Voel 02].

[2] In current practice, 'material condition' applies mainly to 'features of size', which are subsets of spherical surfaces, cylindrical surfaces, or pairs of parallel planes with opposed 'material sides'. Three material conditions are recognized: Maximum ('MMC'), Least ('LMC'), and 'actual' ('RFS', for Regardless of Feature Size). We focus on 'actual' (RFS) conditions, and often denote them with symbol Ⓢ for clarity, because they are usually the most stringent.

This mms. is published in *Proc. ASPE Summer Topical Meeting on Tolerance Modeling and Analysis*, University of North Carolina at Charlotte; ASPE Press, Raleigh, NC, July 2002.

2. CURRENT PRACTICE

Figure 1 provides a simple 2-D example wherein a position tolerance (symbol ⌖ in Fig. 1a) is imposed on hole H. The exact ('true') position of H is defined in Fig. 1a by two Basic (boxed, signifying 'exact') dimensions that are anchored to part features A and B; the labels on these features designate them as datums. The left-to-right ordering of A, B in the ⌖ 'callout block' sets the datum precedence: A is primary and B is secondary. The position tolerance ('<tolspec>' in Fig. 1a) is a small disk-shaped zone defining the allowable variability of H's position, but we shall ignore the tolerance here, and in later examples, because our focus is on datums and datum systems (DRFs).



(a) Partial specification *(b) Establishing the true position of hole H.*

Figure 1: A simple two-dimensional example (Example 1).

Datum simulators are used to establish H's true position (and hence the location of the tolerance zone) on the imperfect actual part shown in Fig. 1b. Datum simulators are relatively precise approximations to the ideal datum features tagged in the design geometry; surface plates typically would be used as simulators for the linear A and B faces in Fig. 1a. Because the A and B simulators are required (by rule) to be orthogonal, the 90° angle plate shown in Fig. 1b is an effective simulator for the (A,B) system. (CMM procedures that simulate the angle plate could also be used.) The part is 'immobilized' in the fixture by procedures that reflect the datum precedence declared in Fig. 1a: 2-point (in 2-D) contact is established with the A-simulator, and then 1-point contact is established with the B-simulator.[3] When the (A,B) DRF has been established for the actual part by immobilizating the part in the fixture, the true position of the hole is established by simulating the Basic dimensions to 'gagemakers' precision'.

The DRF in Example 1 is easy to establish, as are DRFs in many (perhaps most) GD&T applications. But 'problem cases' continue to appear with disturbing frequency, and usually these are resolved by adding new rules to an already complicated DRF semantics.[4] Figure 2 illustrates a 'tertiary datum problem' that has vexed standards committees for a decade. The task in Fig. 2 is to establish the true position of Hole D, using the Basic dimension and the ordered datums A (planar), B (cylindrical), and C (slot), with B and C at material condition RFS.

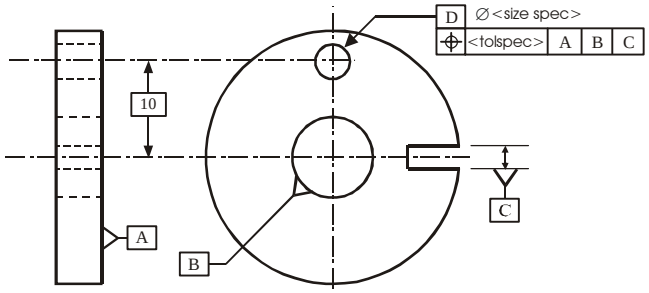


Figure 2: Example 2 – a 'tertiary datum problem'.

[3] 'Immobilization' presumes an orthogonal-plane DRF in which the part initially 'floats'. The datum declarations are used to reduce the initial six degrees of freedom (in E^3; three in E^2) of the floating part to zero degrees of freedom.

[4] 'Syntax' refers to symbol structures, e.g. the ⌖ callout block in Fig. 1a. 'Semantics' associates meanings with symbol structures, e.g. via the rules and procedures for establishing the DRF in Fig. 1b.

This mms. is published in *Proc. ASPE Summer Topical Meeting on Tolerance Modeling and Analysis*, University of North Carolina at Charlotte; ASPE Press, Raleigh, NC, July 2002.

RFS simulators for the B and C datums are, respectively, the largest perfect cylinder that will fit into the B-hole *and* be ⊥ to the A simulator, and the largest perfect 'slab' that will fit into the C-slot *and* be ⊥ to the A simulator. Because the B and C simulators are ⊥ to A, we can represent their geometry by the 2-D drawings in Figure 3. In Fig. 3a, the midplane of the C-simulator is forced to contain the B-simulator axis (and be ⊥ to A and of maximal size), whereas in Fig. 3b the midplane of C and the axis of B are independent. Observe also that the 3b case admits the two dimensioning patterns shown in Fig's 3b-1 and 3b-2. Which of the three cases – 3a, 3b-1, or 3b-2 – is correct? Close reading of both the Y14.5 and Y14.5.1 [ASME94b] Standards is needed to discover that Fig. 3b-1 is the proper interpretation. A designer might want to specify any one of the three cases, but has no mechanism for doing so.

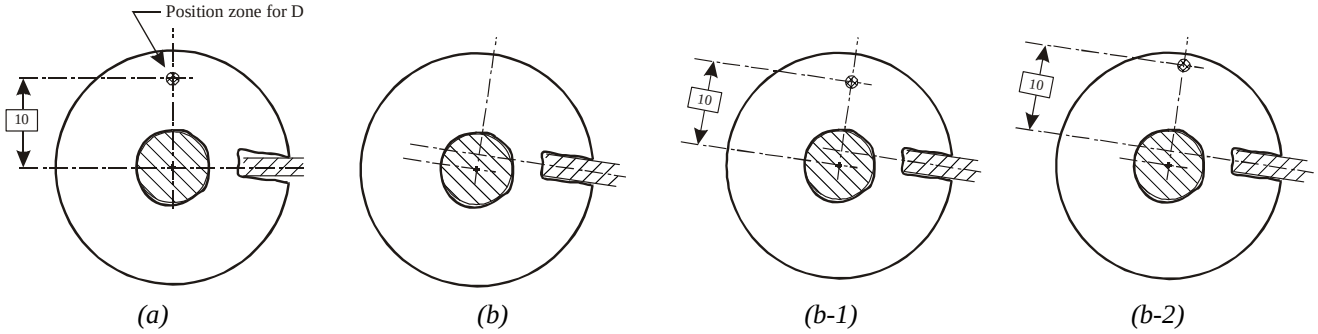


Figure 3: Datum simulator geometry for Example 2 (Fig. 2).

DRF declarations that admit several plausible interpretations, as in Fig. 3, are easy to construct. Usually – but not always – they can be resolved to a single 'right answer' by increasingly complex rules. The primary problem lies in the 40-year-old Y14.5 (and ISO) syntax: it accommodates Requirements 1-3 in our initial list (see Section 1), but fails badly on the fourth – inter-datum constraints. The 'immobilization' procedure is a secondary complicating factor; it can be traced to the mechanical fixturing concepts that dominated the early development of GD&T.

The remaining pages of this paper summarize a different approach to DRFs that is simple, powerful, and does not require disruptive changes to the overall structure of GD&T.

* *

3. DRF Construction with Explicit Constraints

This approach retains the notions of datum precedence and material condition (Requirements 1-3), but replaces all of the interpretive rules currently used to handle inter-datum constraints (Requirement 4) with a single new requirement: all constraints must be stated explicitly. It also replaces immobilization as the DRF instantiation paradigm with construction: DRFs are built datum-by-datum on the part, rather than by tying-down a part floating in a preconceived DRF.

The procedure is very simple: establish the primary datum simulator as at present, and then establish secondary and tertiary simulators using only constraints stated explicitly in the datum declarations. Figure 4 provides an introductory example. For the design geometry in Fig. 4a, the declarations in 4b dictate independent establishment of the A and B simulators. The orthogonal DRF in Fig. 4c requires the explicit statement 'B ⊥ A' in the declaration.

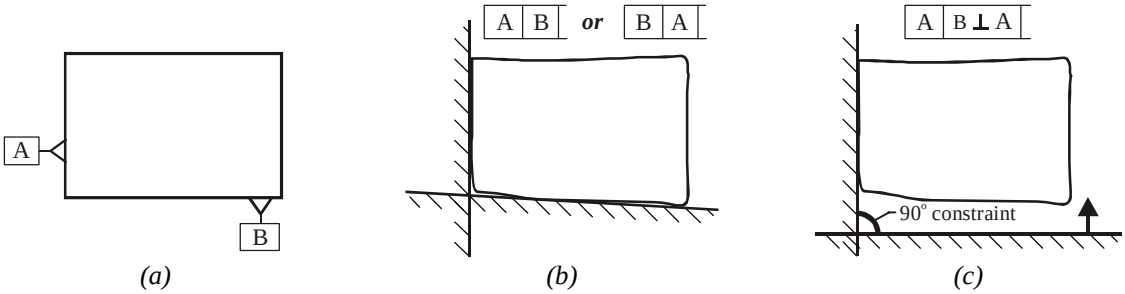
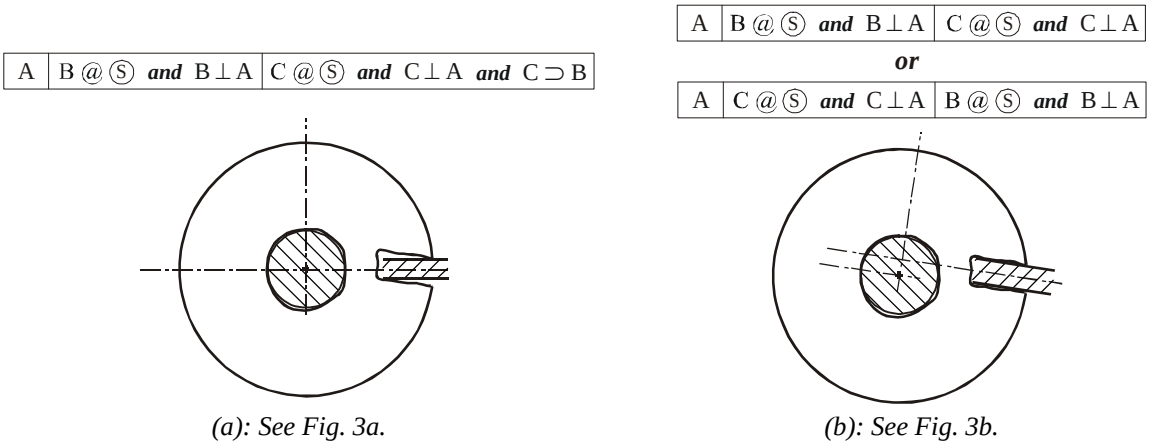


Figure 4: DRF construction with explicit constraints (Example 3).

This mms. is published in *Proc. ASPE Summer Topical Meeting on Tolerance Modeling and Analysis*, University of North Carolina at Charlotte; ASPE Press, Raleigh, NC, July 2002.

Figure 5 provides more substantial examples in the form of explicit-constraint declarations for the plausible simulator geometries in Fig. 3. (In Fig. 5a, the statement 'C ⊃ B' is shorthand for 'Centerplane(C) ⊃ Axis(B)'.)



(a): See Fig. 3a. *(b): See Fig. 3b.*

Figure 5: DRF declarations with explicit constraints for Example 2.

It should be evident from this brief introduction that the explicit-constraints approach is powerful, because constraints are handled through a *language* rather than a fixed set of rules. It is also simple, because there are no rules other than 'honor datum precedence'. Unfortunately the approach is *verbose,* and perhaps 'too fancy' for some detailers and shop-floor people.

There is a way to finesse the verbosity problem: provide a catalog of *predefined constraints* that can be used selectively, so that designers don't have to write full constraint sets. Where can we find such a catalog? … in the part geometry! Specifically, almost all of the constraints we are likely to need – relations like ⊥ and ⊃, distances in the form of Basic dimensions, and a few more – are embedded in the design geometry of the part; all we need is a procedure – provided in Section 4 below – that enables the constraints to be used selectively.

* *

4. DRF CONSTRUCTION USING DESIGN-GEOMETRY CONSTRAINTS

This procedure is very similar to the explicit-constraints method described above. The main difference is that constraints embodied in the part geometry are always used unless cancelled explicitly or replaced with different explicit constraints. Specifically …

- Establish the primary datum simulator as at present, i.e. without inter-datum constraints.
- Establish the secondary simulator using all part-geometry constraints between the primary and secondary datums that are not over-ridden by the designer.
- Establish the tertiary simulator similarly, using all part-geometry constraints between the primary, secondary, and tertiary datums not over-ridden by the designer.
- Two types of designer over-rides are admitted:
  – *constraint cancellations,* which are frequent, and are implemented below with a cross-out notation, and
  – *constraint impositions,* which are rare, and are implemented by inserting explicit constraints.

Figure 6 provide an introductory example. The design geometry in Fig. 6a carries the implicit constraint 'A ⊥ B'. This constraint is used in 6b and 6c because it is not over-ridden in the datum declarations. In Fig. 6d, however, the constraint is cancelled – note the cross-outs in the secondary datum slots – and the primary and secondary simulators are established independently. Specifically: when A is primary in Fig. 6d, the secondary-datum declaration '$\mathrm{B}\ \not{\mathrm{A}}$' means 'cancel (ignore) A-based constraints on B when establishing the B simulator'.

This mms. is published in *Proc. ASPE Summer Topical Meeting on Tolerance Modeling and Analysis*, University of North Carolina at Charlotte; ASPE Press, Raleigh, NC, July 2002.

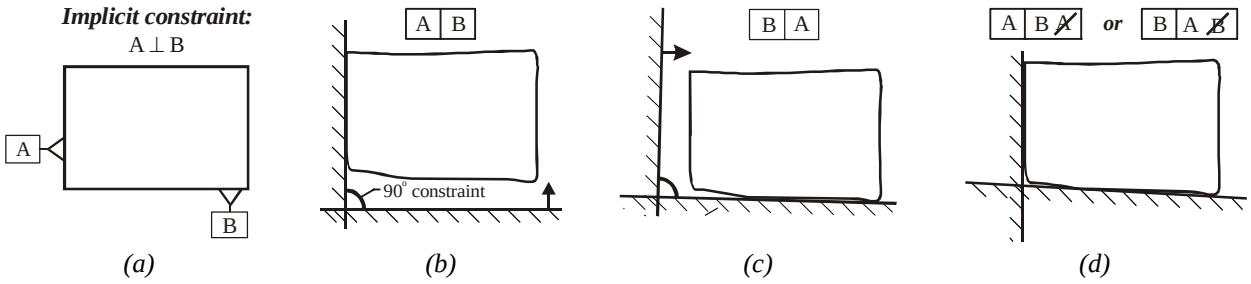


Figure 6: DRF construction with design-geometry constraints (Example 4).

Figure 7 provides more substantive examples in the form of alternative DRFs for Example 2. Fig. 7a lists the relevant constraints carried in the part geometry, Fig's 7b and 7c provide declarations for the geometries in Fig's 3a and 3b (also 5a and 5b), and Fig. 7d shows a third variant. Note that cancellation of the (B,C) constraint in Fig. 7c yields independent B and C simulators.

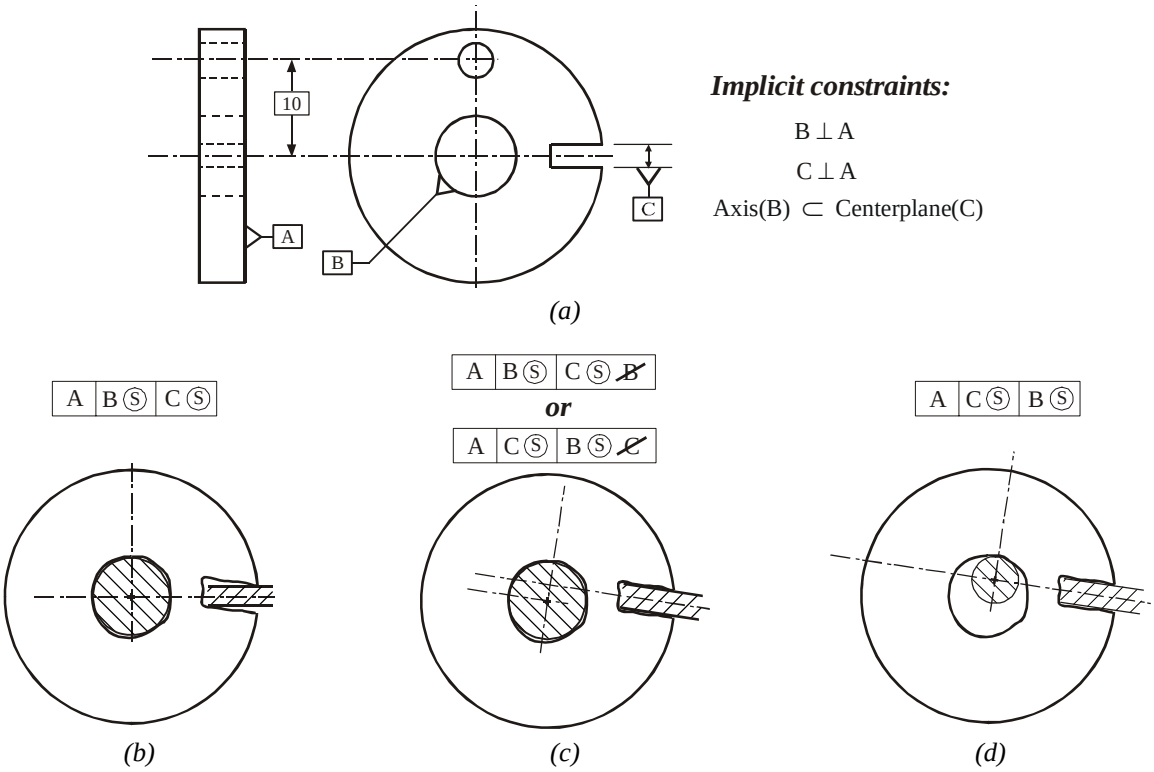


Figure 7: DRF declarations for Example 2 using design-geometry constraints.

The page limitation on this paper prevents inclusion of additional examples to illustrate constraint impositions, which are interesting albeit rare in practice. The author will provide such examples on request.

* *

5. ISSUES AND EXTENSIONS

DRF construction using design-geometry constraints (Section 4) has all of the power of the explicit-constraints approach (Section 3) if the over-ride facility is used judiciously. It is simple because it has no rules apart from 'honor datum precedence', and it is efficient because it is succinct. However, it raises an interesting question: what is 'design geometry'? The question is deeper than it may appear initially. For example: Basic dimensions must be included in 'design geometry' to make the approach work; should any tolerances be included? (Apparently not.)

This mms. is published in *Proc. ASPE Summer Topical Meeting on Tolerance Modeling and Analysis*, University of North Carolina at Charlotte; ASPE Press, Raleigh, NC, July 2002.

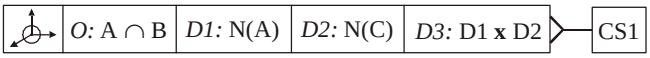
Solid modeling should provide part (perhaps all) of the answer to the question, even though solid modeling does not deal well with variational geometry.

Several extensions of this work come to mind readily. Two are summarized below.

1) Provide declaration facilities for coordinate systems (CSs).

   CSs are similar to DRFs, but 'richer' because they have well defined origins and named directions. Figure 8a provides an examplary template for CS declarations in the Y14.5/ISO 'callout block' style. Fig. 8b uses the template to declare a CS named 'CS1' that uses the datums declared in Example 2. (Notation: 'N(A)' means 'Normal vector to A', 'N(C)' is shorthand for 'N(Centerplane(C))', and so forth.) Observe that all design-geometry constraints are active because there are no cancellations, that the origin is defined as the intersection of A with the axis of B, and that the third direction is defined as the vector cross-product of the first two directions.

(a): An exemplary CS declaration template.



(b): A CS declaration using the Fig. 2 (or Fig. 7a) datums.

Figure 8: Declaring Coordinate Systems.

2) Provide means to check the *validity* of DRF and CS declarations.

   Validity checking seeks to insure that declared DRFs and CSs really do provide unique reference systems (with three independent directions, for example). Checking can be done on two levels – combinatorial and geometrical – and can be implemented either 'manually', via tables similar to 4.1 - 4.5 in Y14.5.1 [ASME 94b], or automatically, via algorithms running in CAD systems.

*

I had intended to conclude with an outline for a more formal and general approach to datum usage in GD&T, but this paper has hit its page limit.

* *

6. REFERENCES

[ASME 94a] American Society of Mechanical Engineers. Dimensioning and Tolerancing. ASME Standard Y14.5M-1994, New York NY, 1994.

{ASME 94b] American Society of Mechanical Engineers. Mathematical Definition of Dimensioning and Tolerancing Principles. ASME/ANSI Standard Y14.5.1M-94, New York NY, 1994.

[Voel 02] H. Voelcker, "Musing on datums", *mfg.* (a Brown & Sharpe magazine), vol. 9, no. 1, pp. 42-45, 2002.

* *

7. ACKNOWLEDGEMENTS

This work grew out of a 15' *ad hoc* talk I gave in 1995 to the ASME Y14.5 Committee, proposing the use of explicit constraints in datum declarations. The talk was stimulated by a full day of meandering and fruitless Committee discussion on 'the tertiary datum problem'; it was received politely, and then allowed to 'fall through the cracks'. I thank Alvin Neumann, Walter Stites, Brian Shier, Bill Tandler, and Ed Morse for helpful discussions of datum topics in subsequent years.