JT-IF Implementation Guidelines for PMI in JT

Version 1.0, 12 December 2023

Status: Release

Contacts:

Jochen Boy
PROSTEP AG
Dolivostraße 11
64293 Darmstadt / Germany
jochen.boy@prostep.com

Sarah Giese
PROSTEP AG
Dolivostraße 11
64293 Darmstadt / Germany
sarah.giese@prostep.com

Contents

Table of Contents

Figures

Tables

Document History

Version	Date	Author	Description
0.1	2020-09-16	S. Giese	Initial Creation
0.2	2020-10-28	J. Boy	General update
0.3	2020-11-17	R. Beckers	Updated section 4.3; added Figure 5
0.4	2021-03-25	S. Giese	Inserted chapter for callout dimensions
0.5	2021-08-	S. Giese	Added section on counting PMI
0.6	2022-12-02	J. Boy	Added definition to Dimensioning Standard and references to Validation Properties
1.0	2023-12-12	S. Giese, J. Boy	Initial public release after validation in testing

1 Introduction

1.1 About the JT Implementor Forum

The JT Implementor Forum was set up in 2010 and is a joint project organized by the prostep ivip Association and the German Association of the Automotive Industry (VDA). The aim of the project group is to provide the participating vendors with a platform where they can test the translators that they are developing “behind closed doors”. The JT Implementor Forum is a neutral forum for software vendors, where they can perform testing in an atmosphere of mutual trust and exchange information on experience already gained.

The work performed in the JT Implementor Forum is carried out in cooperation with the JT Workflow Forum and is enriched by the experience gained from the JT Translator Benchmarks.

New functionality is tested and optimized until it is declared “ready for market”. This also gives rise to requirements relating to implementation guidelines and pre-release testing.

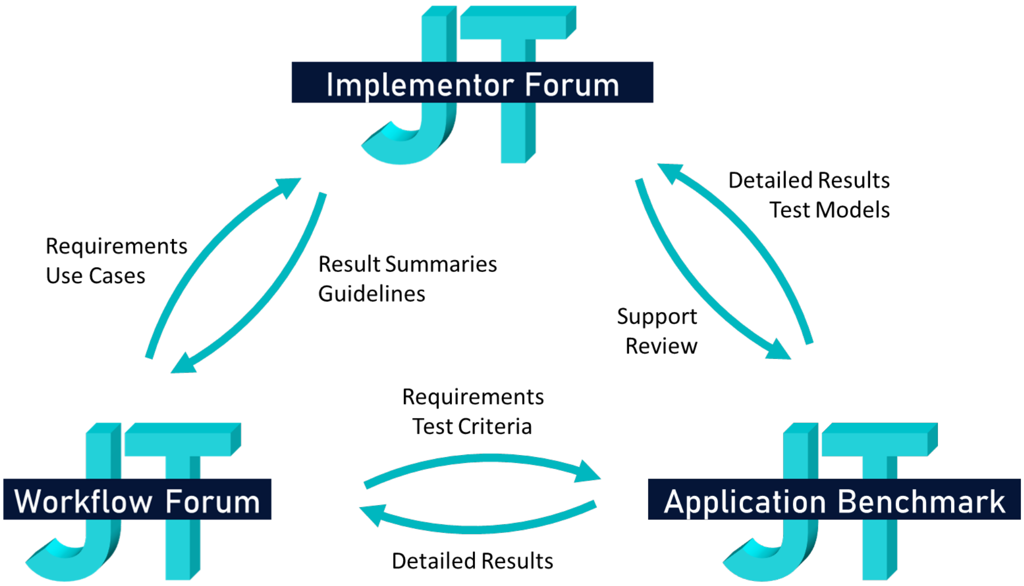

Figure 1: Interaction of the VDA / prostep ivip JT initiatives

1.2 PMI in JT – Need for Action

Product & Manufacturing information (PMI) is essential engineering or manufacturing information embedded within a 3D CAD file. It is governed by international and industry standards, such as ISO 1101, ISO 16792, ASME Y14.41, ASME Y14.5 and others.

PMI may include the following information:

1. GD&T (Geometric dimensions & tolerances).
2. Bill of materials (BOM).
3. Surface finish.
4. Weld symbols
5. Material specifications
6. Metadata & notes
7. History of engineering change orders
8. Legal/proprietary/export control notices
9. Other definitive digital data

3D PMI is a significant step to digital transformation, cross-domain collaboration and quality control for any company or department.

The current push towards Model-Based Development (MBD) or Model-Based Engineering (MBE) results in the need to supplement 3D CAD models with PMI, and to convey this enriched data across applications, domains, and companies.

This requires that the information be conveyed in an unambiguous way. For a neutral format such as JT, this means the resulting data in the JT file shall be equivalent regardless of the source application, so that any consuming IT tool can deal with the information consistently.

Challenges are posed by the fact that different applications have different internal data models, varying ranges of capabilities and disparate design philosophies, so that a 1:1 mapping to JT is not always possible. Also, as JT has evolved over time, there are cases were a certain piece of information can be mapped in more than one way.

This guideline aims at providing guidelines how to map PMI to and from JT in an unambiguous way, as well as to point out cases where a desired result may not be fully achievable due to system restrictions.

2 Scope

2.1 In Scope

The following are within scope of this document:

- Definition of the different concepts for storing PMI in JT
- Providing guidelines for mapping specific constructs and structures
- Outlining cases where data cannot be fully mapped

2.2 Out of Scope

The following are out of scope of this document as they are covered elsewhere:

- Description of the JT file format or PMI data model in JT. For this, refer to the JT File Specifications listed in Annex A.
- Definition of Validation Properties for PMI in JT. Please refer to the separate Implementation Guideline for JT Validation Properties.

3 Fundamentals & Concepts

3.1 Levels of PMI data in JT

PMI can be transferred in JT in various ways, which can be linked to different use cases.

Graphic PMI is intended for human consumption only. For this, the 3D annotations created in the source CAD system are converted to (tessellated) lines, arcs and faces. These can be efficiently displayed by any viewing application. While any engineer looking at the screen will be able to understand the information being displayed, the geometric elements presenting the PMI cannot be further interpreted by any software tool. Since any text (numbers, letters, symbols) also gets converted to geometry, it is also not modifiable.

Semantic PMI conveys the data in a computer-interpretable way. It follows a pre-defined structure, consisting of a set of parameters (key-value pairs) and links to the underlying model elements. The intention is for the data to be modifiable or directly usable in any target application (multi-CAD or downstream use cases). This is at the core of MBD/MBE and of utmost importance to user companies.

In JT, semantic PMI is always an extension of graphic PMI. In addition, a varying extent of styling information is given, including fonts, line styles, colors, widths etc., which allow a target application to recreate and modify the contents of the graphic PMI when changing the definition of the model, while maintaining the appearance.

3.2 Underlying work done in the JT-IF

PMI has been long-running a topic in the JT-IF. Over the past years, there have been various activities to examine issues encountered when exchanging PMI data and to improve data quality.

The exchange of graphic PMI has been working quite reliably for a long time. However, the exchange of semantic PMI can still be improved.

One of the questions is, which attributes are required in the JT file to correctly describe a PMI. To simplify the search for the right attributes, a subset of all PMI for closer examination was identified at the beginning of the analysis. A survey was carried out for this purpose, where the users ranked PMI types by importance, and where vendors and software manufacturers provided feedback on which types of PMI they support on export and import.

The identified subset of PMI was used to build a model that contains this subset of PMI.

Subsequently, the attributes written per PMI were analyzed in detail. It was found that the number of attributes populated for a specific type of PMI varies greatly depending on the source system and the JT translator being used, sometimes by as much as factor of ten. However, no common mandatory sub-set of attributes could be identified this way. The links between the PMI and to the geometry probably also play a role.

This document summarized the finding from these analyzes, as well as agreements based on technical discussions in the JT-IF.

4 Recommendations for PMI in JT

4.1 Linear vs. Radial dimensions

In JT, four types of dimensions are defined: angular, curve length, linear, and radial. While the first two do not pose greater challenges during data exchange, there is a lot of ambiguity in the use of the two latter ones, linear and radial.

When examining JT files with PMI, there are notable differences. While the total number of dimensions often doesn't change that much, there is a tendency that there are more radial dimensions than expected, but less linear dimension than expected.

A possible cause for this confusion is the diverging nomenclature used in different CAD systems, as shown in Figure 2 below.

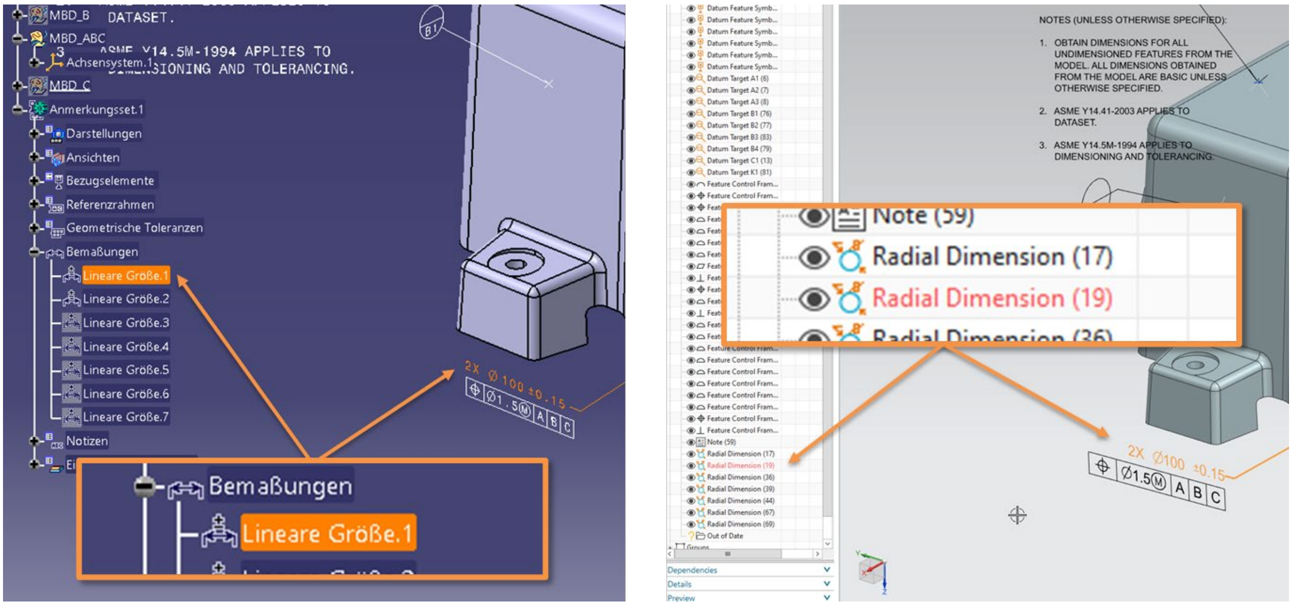

Figure 2: Different designations (linear vs. radial) for the same PMI element in different CAD systems

Depending on how a consuming application interprets the data, this might lead to issues, as a downstream application may interpret and extract semantic information based on the dimension type. Inspection, CAM, and Variation Analysis applications do query the type and may have some logic around it. Example, if type is linear, an application may try to compute/figure out measurement direction, but for radial, it wouldn't attempt to do so.

In order to ensure the dimension type is stored consistently in JT, the following recommendation is given:

<u>**Recommendations**</u>

- Any dimensions related to circles or circular curves, which are defined either with a diameter symbol (Ø) or a radius symbol (R), shall be stored in JT as radial dimension (Dimension.type = 3)
- Any other straight dimension shall be stored as linear dimensions (Dimension.type = 1)

4.2 Callout Dimensions

For JT version 10 files and forward an additional dimension type is available for definition. The type is a Callout Dimension. Callout Dimension values are derived from a hole or threaded hole features parameters. Callout Dimensions will have the name “Linear Hole and Thread Callout” or “Radial Hole and Thread Callout.”

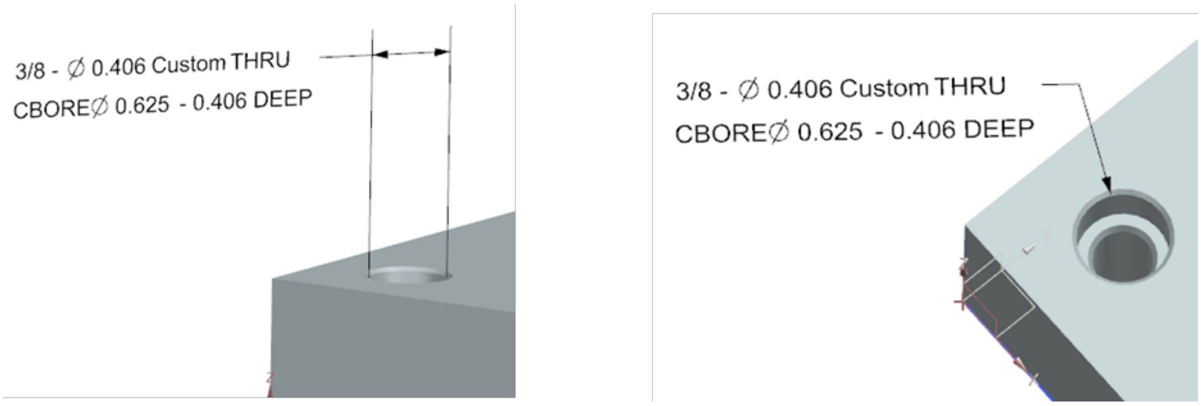

Figure 3: Left: Linear Hole and Thread Callout, Right: Radial Hole and Thread Callout

The data assigned to a Hole or Threaded Hole Callout will be identified as TypeID: JtkPMICalloutDimension. The properties for this TypeID are defined as ParameterDimension[#].<property name> in the JT file.

A Parameter Dimension array represents the data associated with each parameter for a hole or threaded hole feature. Each parameter of the feature will have its own ParameterDimension list of properties:

Figure 4: Parameter Dimension Property

All TypeID: JtkPMIDimension properties can also exist as TypeID: JtkPMICalloutDimension properties.

The Linear and Radial Hole and Thread Callout dimension property for “type” is defined using the same property for “type” definition as all dimensions in JT. Because of this JT Dimensions created as “*Radial* Hole and Thread Callout” must set the type parameter to 3

(ParameterDimension[#].type = 3)

and JT Dimensions created as “*Linear* Hole and Thread Callout” must set the type parameter to 1

(ParameterDimension[#].type = 1)

This use of “type” should not create confusion with regard to the Linear vs. Radial dimensions recommendation as the ParameterDimension[#].name property identifies the parameters as being related to a feature not a linear dimension.

prop: key = ParameterDimension[#].name *value = Radial Hole and Thread Callout*

Validation properties

CalloutDimension is an annotation value that is included in total count for annotation values in a JT file.

As stated by Siemens, hole callouts are preferred for PMIs with hole patterns. For single holes, Radial Dimension PMI is preferred (also see 4.1).

A "Callout Dimension" in JT is a separate annotation class in addition to the plain Dimension class.

A Callout Dimension represents the dimensioning based on one or more parameters in a feature tree, each of which is represented by a Parameter Dimension.

Callout Dimensions are represented by the JtkPMICalloutDimension class, which is derived from the JtkPMIDimension class and has the following properties:

Property	Type	Description
parameterDimension	property[]	The Parameter Dimensions contained in the Callout Dimension.
parameterSpaceFactor	double	The space between the parameters of the Callout Dimension.
parameterLineFactor	double	The space between the lines of the parameters of a Callout Dimension.

Table 1: Properties of the JtkPMICalloutDimension class

When evaluating the results with the PMI Print Tool, it became apparent that these callout dimensions are often not counted as the radial dimensions.

Within the JT-IF, Callout Dimension so far were found solely in JT files created by Siemens out of NX.

Recommendation

- It is meaningful to interpret the ParameterDimension[#].<property values> in a TypeID: JtkPMICalloutDimension definition individually as they represent the properties for the PMI displayed for hole and threaded hole features?

Autodesk also offers something similar for dimensioning holes, called “Hole Information”. This is often mapped as "notes" in JT but does not appear as semantic PMI dimensions.

4.3 PMI in Relation to Model Hierarchy

In CAD models, PMI is usually stored at assembly level or at part level. Hereby, PMI stored at part level only apply to the part they are stored in (and every instance of it), whereas PMI stored at assembly level may reference several parts or individual instances.

When a CAD model is converted to JT, the structure may change. While an assembly node will always be kept as an assembly node, CAD parts may be converted to JT assemblies containing one or more JT parts tagged with the subnode property. This reflects the sub-part structure of CAD models, where a part may contain multiple bodies or geometrical sets. Bodies and geometrical sets in CAD files are usually constructed from multiple construction elements. These elements should not be converted to JT nodes. Figure 5 shows two possible options for mapping the CAD structure to JT structures.

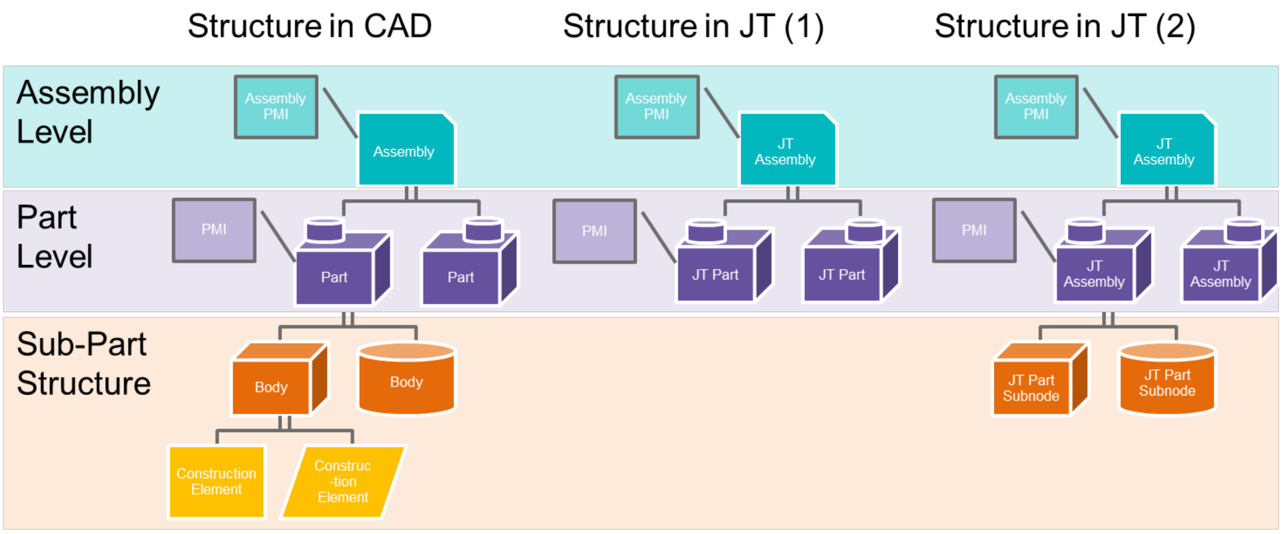

Figure 5: Model structure mapping

Recommendations

- PMI data shall always be stored at the level the PMI applies to.
- In JT, part PMI can be stored on a JT part node, or on a JT assembly node representing a CAD part node in the source model.
- Subnodes shall not be used to store PMI.

However, when reading a JT file, all nodes should be checked for PMI to avoid data loss.

4.4 Counting PMI

If the number of PMI elements is compared when replacing PMI, there are often discrepancies in the number of elements counted. This mostly results from the respective counting method.

Recommendation

When counting PMI, only the actual PMI should be counted.

This includes:

- Dimensions
- Feature Control Frames
- Datum Targets
- Datum Feature Symbols
- Notes

It does not include:

- Model Views
- Reference Coordinate Systems
- Reference Axes

4.5 PMI Validation Properties

JT supports Validation Properties for PMI. It is highly recommended to include these with every JT file that does include PMI, as they are an efficient and effective mechanism to validate the correctness and completeness of the data after import.

The properties provide values obtained by counting or measuring PMI-related information in the originating systems and storing the results of these calculations in in the JT file. An importing system can perform the same calculation and then compare its results to those read from the JT file to determine the success of the import.

Refer to the "JT-IF Implementation Guidelines for JT Validation Properties" (Version 1.0, 21 June 2021) for details.

4.6 Defining the Dimensioning Standard

Different dimensioning standards define the visual representation and interpretation of the symbology of the tolerances. For the receiving system to create the correct visual representation of the tolerance, the dimensioning standard must be known.

In JT, the Dimension Standard is given for each individual PMI element.

Attribute Key	Applicable PMI Types		Attribute Value
standard	0x0081	Feature Control Frame	String representing enumeration such that:
	0x0082	Dimension	"0"=asme_y145m_1994,
	0x0084	Datum Feature Symbol	"1"=jis,
	0x0088	Datum Target	"2"=iso,
	0x0100	Note	"3"=bs,
	0x0235	Balloon Note	"4"=ansi_y145m_1982,
	0x0239	Coordinate Note	"5"=asme_y1441m_2003,
	0x0240	AttributeNote	"6"=din,
	0x0241	Bundle or Dressing Note	"7"=gm_addendum_1994,
	0x0244	E Marking (Note)	"8"=asme_y145_2009

Table 2: Dimensioning Standards for PMI Elements in JT

For more details, please refer to the JT File Format Specification (see JT IAP in Annex A).

Note: If an ASME standard is specified, the tolerance principle is assumed to be the envelope requirement unless otherwise specified. Likewise, if an ISO standard is specified, the tolerance principle of independence is assumed.

5 Known Limitations for PMI in JT

5.1 Datum Target Types

JT provides a list of datum target types: ‘point’, ‘line’, ‘circular’, ‘rectangular’, ‘cylindrical’, ‘area’, ‘annular’, and ‘arbitrary’. The intention of these types is to classify the restricted area a datum applies to. Since CAD systems support datum targets in different ways, there are resulting differences in how this information can be mapped to JT:

- In CAD systems such as NX and Creo, datum targets can be created as different types, or a certain type is explicitly selected before creating a datum target. The applicable type from the list above is mapped to JT.
- In CATIA, this is not the case. Here, the datum target type is determined implicitly by the selected target geometry. Thus, datum targets out of CATIA are typically mapped to JT as ‘arbitrary’.
- Inventor does not support datum targets at all, hence they are not mapped to JT.

Annex A Bibliography

JT Implementor Forum Implementation Guidelines:

Version 3.0 – December 2018

https://www.prostep.org/fileadmin/downloads/JT_Implementation_Guidelines_v3.0.pdf

JT Implementor Forum Implementation Guidelines for JT Validation Properties

Version 1.0 – 21 June 2021

https://www.prostep.org/fileadmin/downloads/PSI_JT-IF_ImplGuide_Validation_Properties_v1_0.pdf

ISO JT File Specification:

ISO 14306:2017

Industrial automation systems and integration — JT file format specification for 3D visualization

https://www.iso.org/standard/62770.html

JT Industrial Application Package (JT IAP)

Version 3 – June 2021

https://www.prostep.org/fileadmin/downloads/PSI14_1_V3_JTIAP_Format_Description_and_annexs.pdf

DIN SPEC 91383:2021

JT Industrial Application Package (JT IAP)

https://www.beuth.de/de/technische-regel/din-spec-91383/341903011

Siemens JT File Specification:

In accordance with industry requirements, and following the recommendations given in DIN SPEC 91383, the preferred JT file format to be used in the JT-IF is JT version 10. While the prostep ivip JT IAP explicitly calls out version 10.0., the target version in focus for most user companies is Version 10.5. The JT file format specification for version 10.5 is publicly available at:

https://www.plm.automation.siemens.com/global/en/products/plm-components/jt-open-program.html