UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

IMPROVING PRODUCT DESIGN TOLERANCES USING METR-ONTOLOGY

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

CHINMAYA P. KATHE
Norman, Oklahoma
2018

IMPROVING PRODUCT DESIGN TOLERANCES USING METR-ONTOLOGY

A THESIS APPROVED FOR THE
SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING

BY

Dr. Shivakumar Raman, Chair

Dr. Janet Allen

Dr. Theodore Trafalis

This thesis work is dedicated to my family. To my dear mother, your love, support and encouragement has propelled me to reach the completion stage of my thesis. To my loving dad, you hard-work has always inspired me to put maximum efforts in completing this research. To my twin brother Omkar, you will always be the most reliable person in my life.

Acknowledgements

I would like to express my heartfelt thanks to my advisor, Dr. Shivakumar Raman, who has supported me in completing this research work. It has been a journey of almost two years since the day I stepped into your office to express my desire to do research under you. Your lectures and previous research work in metrology helped me a lot while writing this thesis.

I would also like to thank Dr. Theodore Trafalis and Dr. Janet K. Allen for being part of my thesis committee. Additionally, I would owe them thanks for strengthening my fundamentals in systems engineering through their graduate courses. I would also like to express my gratitude to ISE faculty and staff for their contributions to my academic success. Special thanks to Dr. Kim Wolfinbarger, who never let work-pressure overwhelm me and always encouraged me to prioritize my research over my office work.

Finally, I express my gratitude to my friends and family who have supported me through highs and lows of my graduate life. Special mention to Bhagyashree, Prithviraj, Tejas, Tejaswi, Vardhan and Vishvaja. You’ve each impacted me to become the person that I am, and I would always strive hard to live up to your expectations.

Table of Contents

List of Tables

List of Figures

Abstract

It is important to know the dimensions, material composition, manufacturing processes, product performance history, and inspection methodology while designing products and selecting manufacturing processes. This information could be encapsulated into product lifecycle management tools. Use of ontologies as knowledge base for product information have been on a rise to overcome the drawbacks inherent to many knowledge-based approaches. The current ontologies that exist lack sufficient information regarding product inspection and tolerance, that is important from the perspective of the field of product and process metrology. This research discusses the development of an engineering ontology for product metrology and is termed as "metrontology" herein. The focus of this developed metrontology is to aid in the understanding of product tolerances for future products being designed. The methodology is demonstrated through an example of single-cylinder engine. This research could lead to the creation of a metrology information enclave that will host the knowledge base of metrology information of several products.

Chapter 1: Introduction

A proficiently designed product is essential for it to be successful in a competitive market. In order to design this product several factors such as functionality, specifications, materials, manufacturing process, end use, past performance, etc. are often taken into consideration. These factors, either individually or in combination, help in making design decisions that ultimately determine the success of the product. When engineering content is created and applied during the product life cycle, it is often stored and forgotten. The existing information retrieval approaches are often based on statistical methods and keyword matching. But, these are not effective in understanding the context of engineering content. They are not designed to be directly applicable to the engineering domain. Therefore, engineers have very limited means to harness and reuse past designs (Li, Yang, and Ramani 2009).

"In recent times there has been a lot of development in tracking product performance at every stage of its life from its cradle to grave. This development is encapsulated into product lifecycle management (PLM) tools that are used for managing the entire life cycle of a product" (Kurkin and Januška 2010). The concept of PLM has gained prominence in recent years due the increasing complexity of the product as well as due to increases in out-sourcing and designing and manufacturing considerations. Therefore, for a successful implementation of PLM it is necessary to properly represent and manage product information (Barbau et al. 2012).

Another concept integral to development of PLM is Model Based Definition (MBD). MBD of a product comprises of 3D CAD models of the product instead of the 2D drawings that were traditionally used by companies in storing their product information (Wikipedia 2018). Although MBDs encompass geometric and tolerancing information regarding the product design, its use would require the redesigning of the manufacturing and inspection processes that are currently in use (Thilmany 2015).

The field of metrology deals with the measurement and verification of the dimensional quality of products. It is essential to capture the metrology features of various parts to determine if the product as a whole is functioning correctly. A wide variety of sensors are deployed to capture this metrology and tolerance data. This information is not just captured during the post manufacturing stage of the product, but the sensors are also used to capture the changes in product dimensions after each maintenance. In this process, huge amount of product information is generated which needs to be systematically stored for later use.

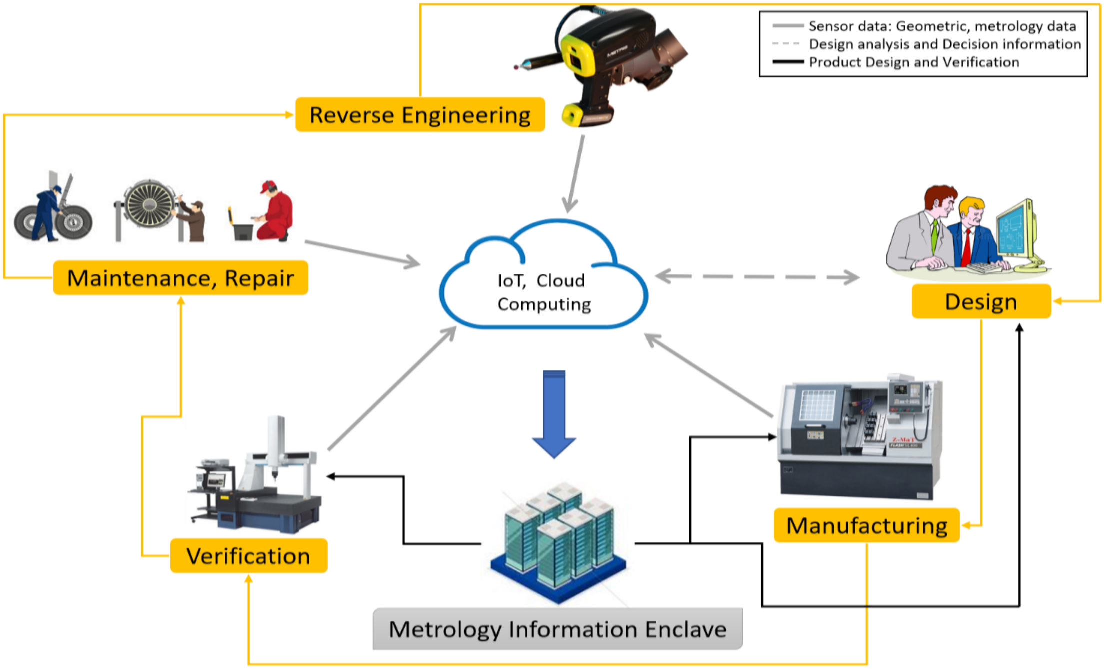


Figure 1. Sensor driven storage and retrieval of Metrology data

Therefore, there is a need for a system that can capture metrology information of a wide range of product environments, through their life-cycles, which can later be used for designing entire products or parts of them. This could aid in making design modifications and suggestions, even for products designed for the very first time based purely on past experiences with similar features, materials, and geometries.

In order to achieve this goal, it is necessary to develop a formal metrology markup language (MML) and integrate time series metrology data with it. For formulation of a MML that will help in incorporating metrology data into a prototype PLM software enclave, a first step is developing ontology that will capture the post manufactured data about the product. This research deals with creation of engineering ontology (EO) for metrology that captures the geometric and beyond geometry information pertaining to inspection of products over their entire life. Also, the focus of this research is to demonstrate how integration of this information will aide in improving the dimensional design of the product.

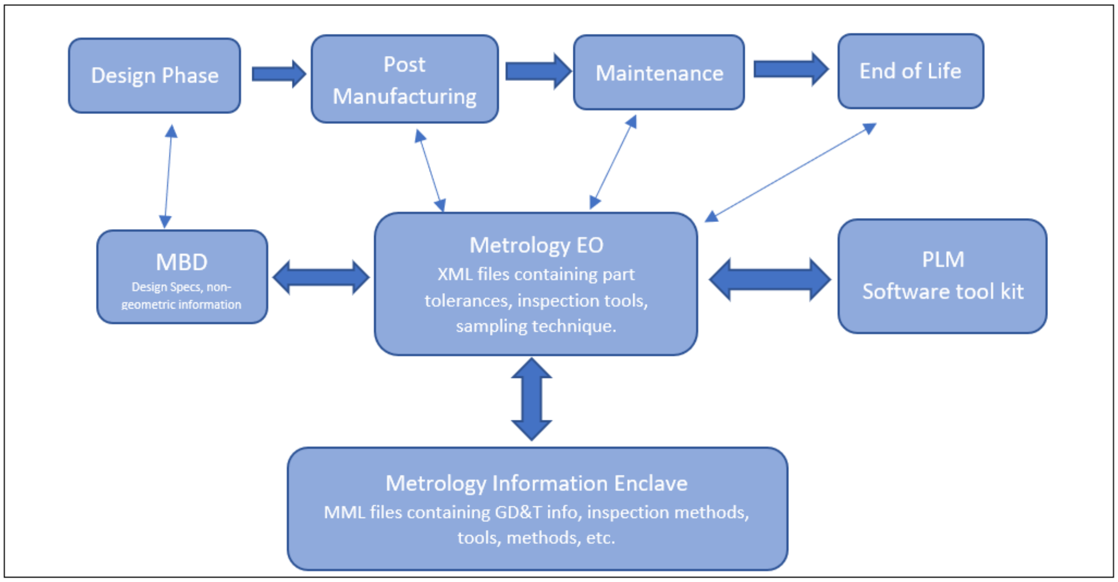


Figure 2. Conceptual description of Metrology EO

In chapter 1 model-based design and how it plays a role in product development is introduced. Also, a need of developing ontologies for metrology is expressed along with a reason for its necessity. In chapter 2 we discuss in depth about product life cycle management as well as the role of model-based definition in the development of a product. Also, key metrology concepts of measurements, tolerances and coordinate measuring machines (CMMs) are discussed.

Chapter 3 discusses previous work done in the field of product lifecycle management with focus on methods and concepts used for integrating engineering knowledge. Also, previous work related to inclusion of metrology information into product design are discussed. In chapter 4, the main focus is on the development of an ontology that can be later incorporated into a product lifecycle management system based on similar platform. A key part of developing this ontology is feature selection. The practical implementation of this ontology is discussed in chapter 5. The software that will be used for development of ontology is also discussed. The sixth chapter presents observations drawn from the preliminary research carried out in the development of the metrontology and includes suggestions for future work on improving and further enriching the ontology.

Chapter 2: Background

2.1 Product information storage practices

Since the dawn of industrial revolution, the design process of a product has evolved a lot. Initially the designs were hand drafted over large sheets of papers, with assemblies and sub-assemblies of a single product consuming ample amount of paper. As technology has progressed, the product designs began to be drafted on large Mylar sheets which were also called as blue prints. Later, the blue prints were replaced by diazo prints which have blue lines over white background. In all the above product information storage procedures, a Bill of Material would be included in a corner to briefly describe non-geometrical information of the product and its sub-parts (Drouillard 2001). With the advent of computers, the product design process went through a drastic change. The traditional drafting was replaced by CAD software and thus reduced the dependence of engineers on physical copies of the product design under consideration (Narayan 2008).

While designing of the product were evolving on one hand, prototyping and product testing also went through lot of changes. From creating physical prototypes made from wood and wax to recent methods of 3D printing, verification of product design in its physical form has come a long way. So are the ways in which the product testing is done has evolved from experimenting on physical product replicas to performing computer-based simulations on the models generated using CAD software. Along with prototype testing, it is also necessary that the manufactured product comply with certain quality standards, laid down by institutions like ANSI and ISO, that are essential for the success of the product and promote standardization of some components of the product. For this purpose, there is again a need for a quality control system. These systems too have evolved from physical inspection of products for design verification to computed automated inspections involving coordinate measuring machines and laser-based scanners.

All these developments point us to the issue concerning storage and retrieval of vast amount of data that is generated in the process of design-analyze-prototype-verify. As the aforementioned process is getting more and more computer reliant, so is the amount of data generated. Although there have been efforts to organize this data in a systematic way, majority of approach has relied on traditional database management whose implementation has not been that helpful in retrieval of past information concerning the designing of the product (Li, Yang, and Ramani 2009). Also, there exists a limitation to the extent up to which the traditional database management systems can store the information regarding the product design and verification process. It was thus thought important to investigate into new avenues for designing a single repository that can be used as a one stop destination for retrieval of engineering information pertaining to the product lifecycle.

2.2 Evolution of Product Life Cycle Management (PLM)

Product Lifecycle Management is an approach that is being widely adopted recently due to overcome the challenge of managing multiple designers, manufacturers and suppliers in an interconnected economy (Barbau et al. 2012). The concept of PLM was first conceived at American Motors Corporation in 1975 (Wong 2009). It was done in order to speed up development of CAD modelling, quicker resolution of design conflicts and centralizing the storage product data there in generated. Later on, PLM was adopted by Chrysler and the idea spread to other industries in America.

At international level, to standardize the exchange of product lifecycle data between various organization, ISO formulated the Standard for Exchange of Product model data (STEP) (ISO 10303) (Ryan Mayes 1994). STEP addresses product data for many industries ranging from automotive, aerospace, building construction, oil & gas, process plants and ship building (Brunnermeier and Martin 1999). The wide adoption of this standard can be credited to its platform independent nature, i.e., STEP is not tied to a specific set of software for implementing the concept of product lifecycle management.

Application Protocols (APs) are used to implement data specifications of STEP. Since, 1994-95 ISO has published several APs which include AP 202, AP 209, AP 210, AP 212, AP 214, AP 224, AP 225, AP 227, AP 232 (Standardization 1998). Among them, the most commonly used APs are AP203:Configuration controlled 3D design of mechanical parts and assemblies (Standardization. 1994), AP214:Core data for automotive mechanical design processes (Standardization. 2003) and AP239:Product life cycle support (Standardization 2005). ISO released AP 242 in 2014 that combined and replaced AP 201, AP 202, AP 203, AP 204 and AP 214. A majority of these APs focus on product data management and geometric information.

Like other ISO standards, STEP is a copyright of ISO and is therefore not freely available. However, schemas created in EXPRESS are opensource implementation of STEP that are freely available and are preferred way of exchanging product information. EXPRESS is a standard data modelling language for product data that was formalized in ISO standards as ISO 10303-21 (Standardization. 2004-11). The EXPRESS files that follow this standard are usually referred to as Part21 or p21-File. Schemas (or Data models) that are formulated in EXPRESS are network of concepts that are used to describe product attributes and relationships amongst them. Although EXPRESS can be used to develop syntactically correct product data models, it cannot represent the explicit semantics of product data (Sarigecili, Roy, and Rachuri 2014). Therefore, STEP has great difficulty in interpreting the semantics of product data in different phases of product life cycle and implementing the exchange of such semantics among heterogeneous product development systems (Lu et al. 2015). It is therefore necessary to carry out further study to be able to further enrich the product data.

2.3 Metrology concepts

For creating ontology for metrology that can be inserted to PLM knowledge base containing geometric and beyond geometry information, it is essential to get acquainted with key metrology concepts.

The International Bureau of Weights and Measures (BIPM) defines metrology as "the science of measurement, embracing both experimental and theoretical determinations at any level of uncertainty in any field of science and technology" (Goldsmith 2010). From PLM perspective, it is necessary to capture the changes in the dimensions of the parts of the products in order better understand the working of the product as well as a precursor in taking design decisions while designing the next iteration of the product. While a large dimensional change in the product won’t occur under normal circumstances, small changes in the dimensions of the product can be captured in form of tolerances.

Both ASME as well ISO have their own set of definitions for various geometric and tolerance keywords. While ISO has several dimensional specifications under various standards like ISO 1101, ISO 3040, ISO 5458, etc. (Goldsmith 2010); a majority of

ASME standard conventions have been covered in ASME Y14.5 (2009). This document “establishes uniform practices for stating and interpreting dimensioning, tolerancing, and related requirements for use on engineering drawings as in related documents.” (Engineers 2009). The metrology ontology that will be formulated in later sections will be based on the tolerancing definitions from the American Society of Mechanical Engineers(ASME) Y14.5, Dimensional and Tolerancing (2009).

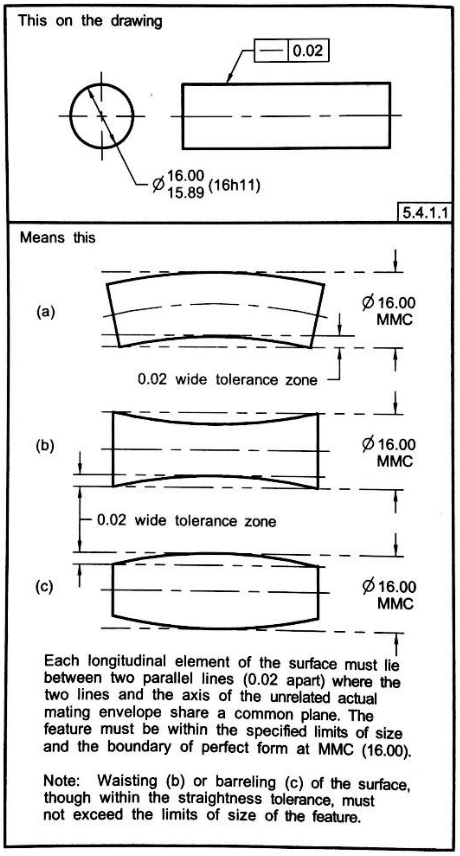
From PLM perspective, an engineer would have to deal with five fundamental forms of tolerances, namely; size tolerance, form tolerance, orientation tolerance, location tolerance, tolerances of profile and run-out tolerance. Size tolerance is the most common and describes the zone of variability of size, unilaterally or bilaterally, from the target dimension.

2.3.1 Form Tolerance

It states how far a real surface, or a feature is allowed to vary from its ideal form. Form tolerance is further classified into straightness, flatness, circularity, and cylindricity.

Straightness:

It is used to define a control line element to a flat surface in a single direction. As shown in figure below, it can also be used to apply in two directions. Straightness needs to be specified with respect to a datum feature.



Each longitudinal element of the surface must lie between two parallel lines (0.02 apart) where the two lines and the axis of the unrelated actual mating envelope share a common plane. The feature must be within the specified limits of size and the boundary of perfect form at MMC (16.00).

Note: Waisting (b) or barreling (c) of the surface, though within the straightness tolerance, must not exceed the limits of size of the feature.

Figure 3. Straightness (ASME.Y14 2009)

Flatness:

It is a condition of a real surface or a derived median plane having all the elements in one plane. The tolerance zone is defined by two parallel planes within which all the surface elements of the real plane must reside. Figure 3 shows the flatness encapsulates the definition of flatness.

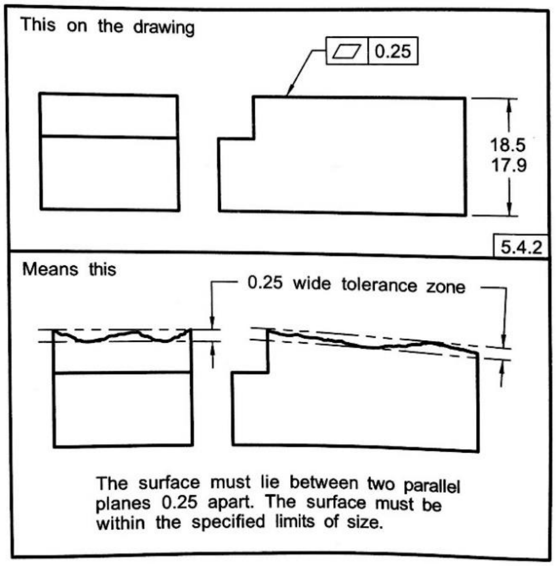


Figure 4. Flatness (ASME.Y14 2009)

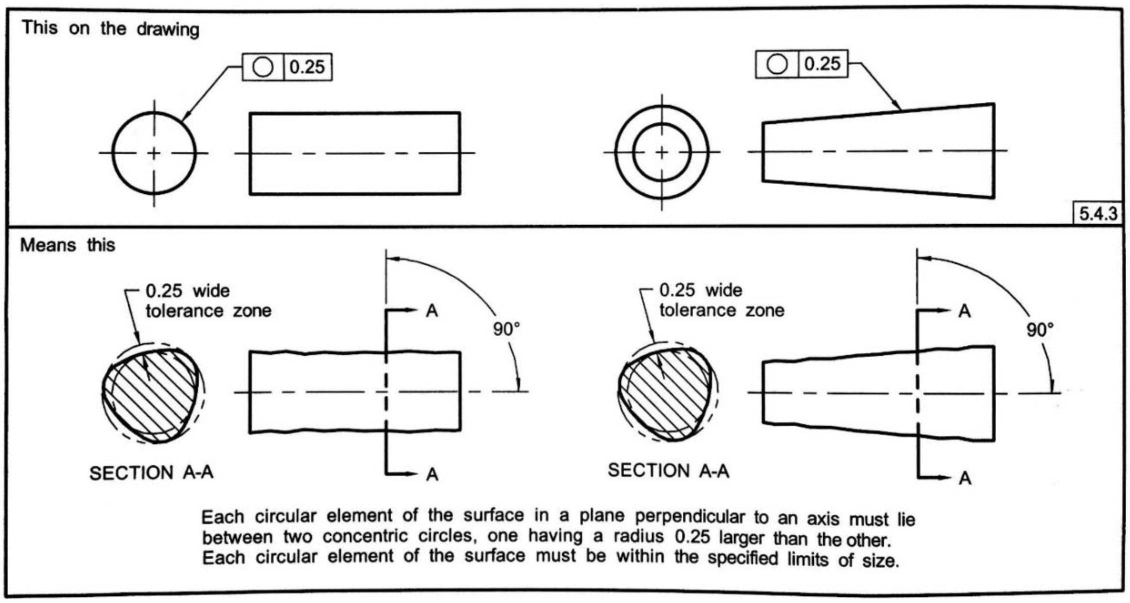
Circularity:

Circularity tolerance is defined in following ways:-

(a) in case of a non-spherical surface, it is a measure of deviation from an axis of all the points in a surface that is perpendicular to this axis.

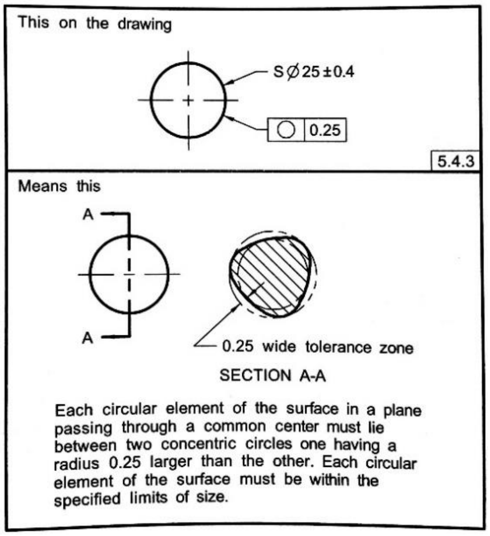
(b) in case of a spherical surface, it is a measure of deviation of all the points from a common center point of the sphere. It is also referred to as sphericity.

The tolerance zone is bounded by two concentric circles in case of circle while it is bounded two concentric spheres in case of spherical surfaces.



Each circular element of the surface in a plane perpendicular to an axis must lie between two concentric circles, one having a radius 0.25 larger than the other. Each circular element of the surface must be within the specified limits of size.

Figure 5. Circularity in case of a non-spherical surface (ASME.Y14 2009)



Each circular element of the surface in a plane passing through a common center must lie between two concentric circles one having a radius 0.25 larger than the other. Each circular element of the surface must be within the specified limits of size.

Figure 6. Circularity in case of a spherical surface (ASME.Y14 2009)

Cylindricity:

It is condition of surface of revolution wherein all the points on a common surface, curved or otherwise, are equidistant from a common axis. The tolerance zone in this condition are bounded by two concentric cylinders within which the surface must lie.

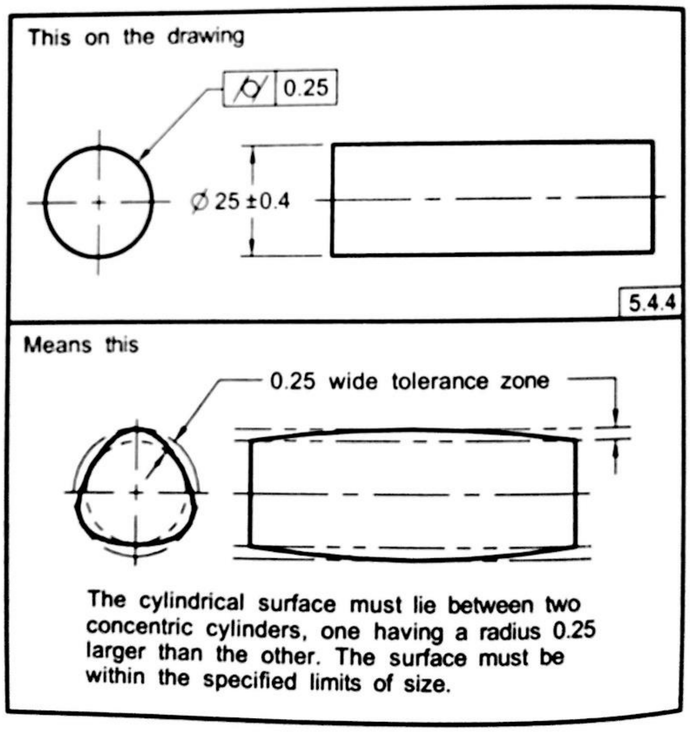


Figure 7. Cylindricity (ASME.Y14 2009)

2.3.2 Orientation Tolerance

As the name suggests, an orientation tolerance states how much a feature or a form is allowed deviate in angle or orientation. Orientation tolerances are comprised of parallelism, perpendicularity, and angularity.

Parallelism:

It is the condition where all the points of the surface feature's or form's center plane are equidistant from a datum plane. The tolerance zone is determined by two parallel lines that encapsulate the real orientation of the axis or plane with respect to datum axis or plane.

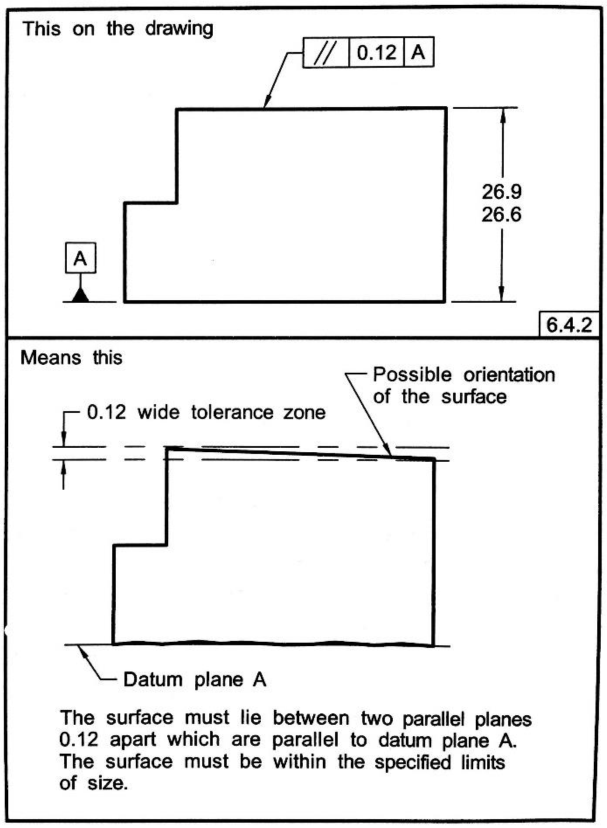


Figure 8. Parallelism (ASME.Y14 2009)

Perpendicularity:

It’s the condition of the surface feature’s or form’s center plane or axis is at right angle to datum plane or datum axis. The tolerance zone is determined by two parallel lines that encapsulates the real orientation of the axis or plane with respect to datum axis or plane.

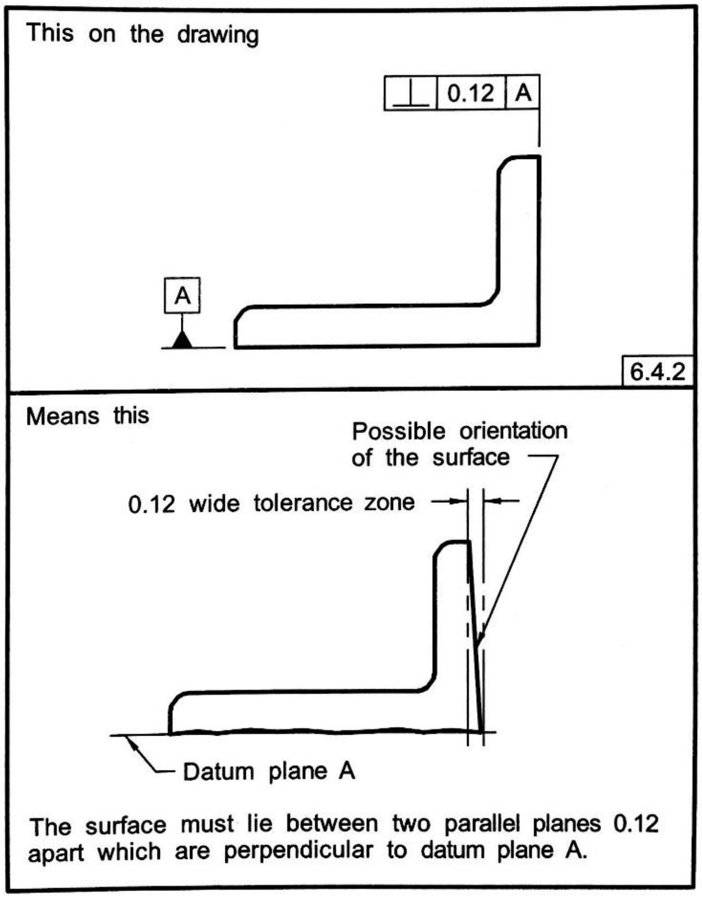


Figure 9. Perpendicularity (ASME.Y14 2009)

Angularity:

It's the condition of the surface feature's or form's center plane or axis is at angle specified to datum plane or datum axis. The tolerance zone is determined by two parallel lines that encapsulates the real orientation of the axis or plane with respect to datum axis or plane.

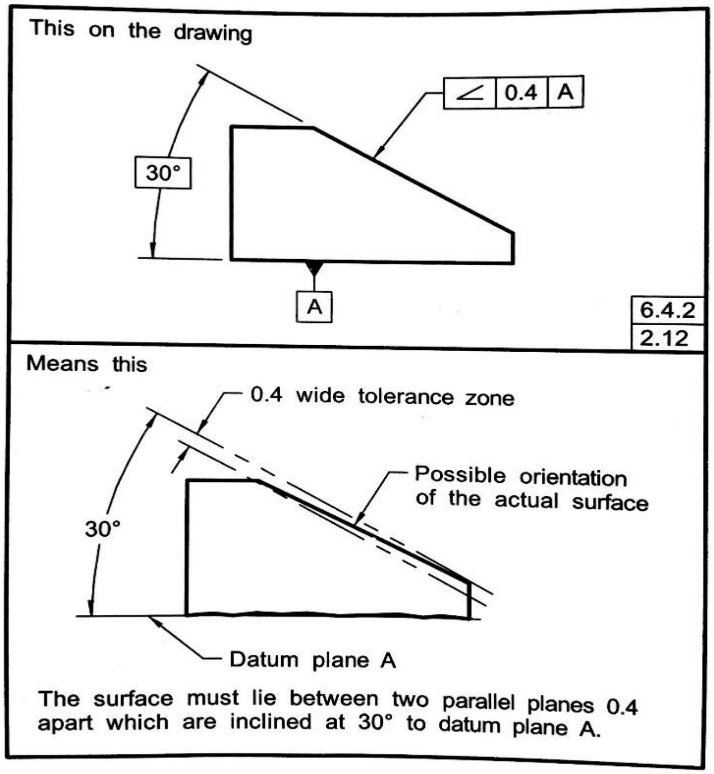


Figure 10. Angularity (ASME.Y14 2009)

2.3.3 Location Tolerance

A location tolerance defines how much a feature is allowed to vary from its ideal position as described by the datum, datums or other features given in a drawing. Location tolerances are further classified into position, concentricity, and symmetry.

Position tolerance:

It can be defined as the location of one or more features with respect to more or more datum feature or relative to another feature, usually placed on the same surface. Figure10 shows how positional tolerance is applied to features located on the same surface.

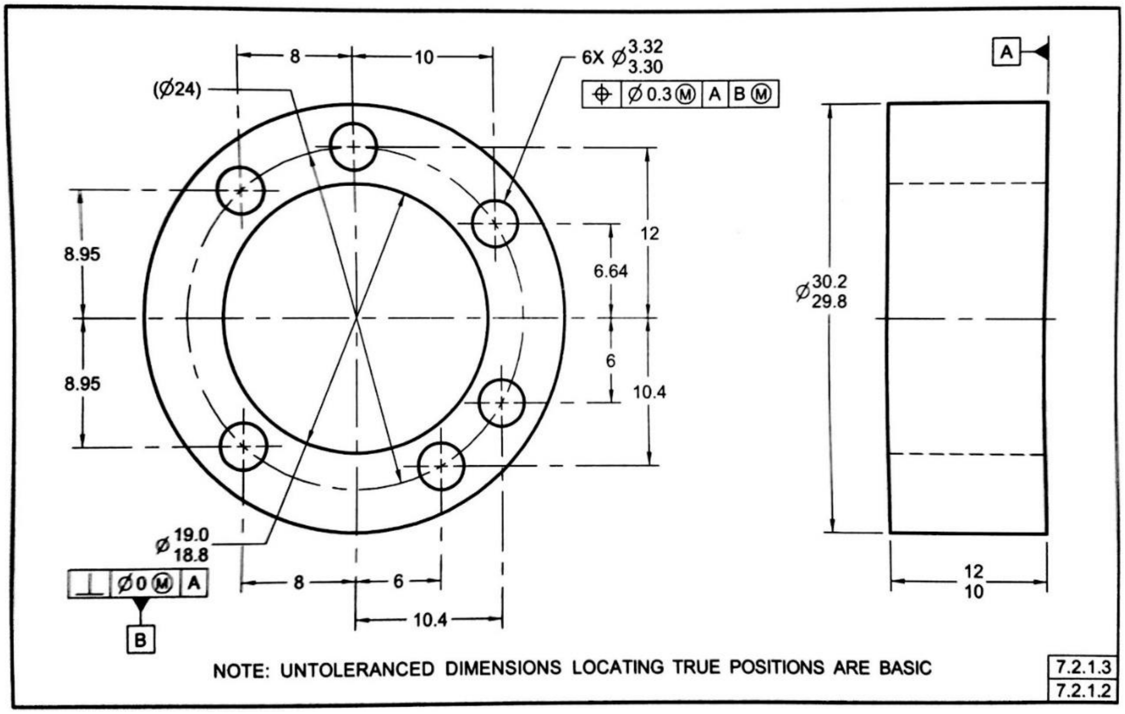
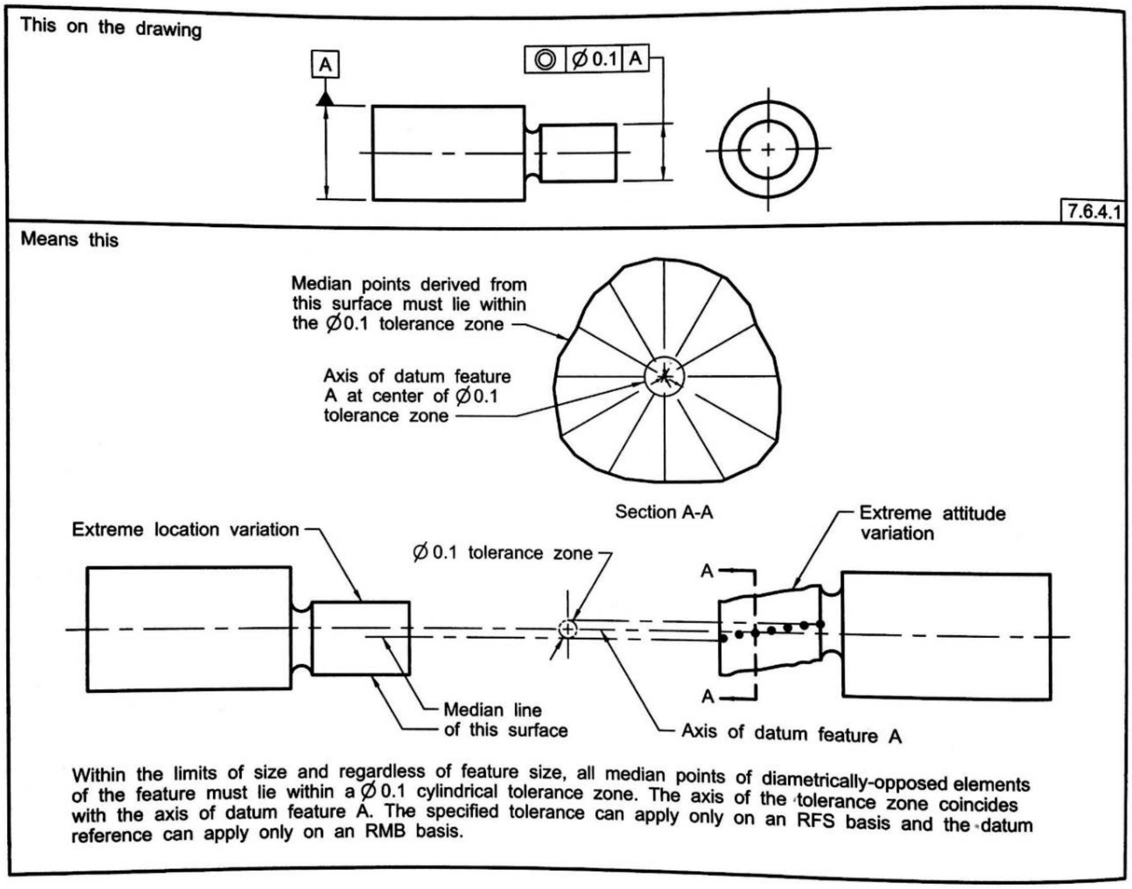


Figure 11. Position Tolerance (ASME.Y14 2009)

Concentricity:

It is a condition of location tolerance where the median of points of all the diametrically opposite elements of a curved surface are congruent with datum axis. The tolerance zone is specified as a cylindrical (or spherical) in shape whose axis (or center point) coincides with the datum feature(s).



Within the limits of size and regardless of feature size, all median points of diametrically-opposed elements of the feature must lie within a Ø0.1 cylindrical tolerance zone. The axis of the tolerance zone coincides with the axis of datum feature A. The specified tolerance can apply only on an RFS basis and the datum reference can apply only on an RMB basis.

Figure 12. Cocentricity (ASME.Y14 2009)

2.3.4 Tolerances of Profile

Profile tolerance is applied to the entire outline of the surfaces under consideration. It can be applied to entire part, multiple features on the part, individual surfaces, or to individual profiles taken at various cross sections through a part. Profile tolerance is further classified into profile of a surface and profile of a line.

Profile of a surface:

The profile of a surface is applied parts of any shape which may include parts having constant cross-section, parts having surface of revolution, etc. The tolerances zone thus established is three dimensional, extending along the length and width of the considered feature or features.

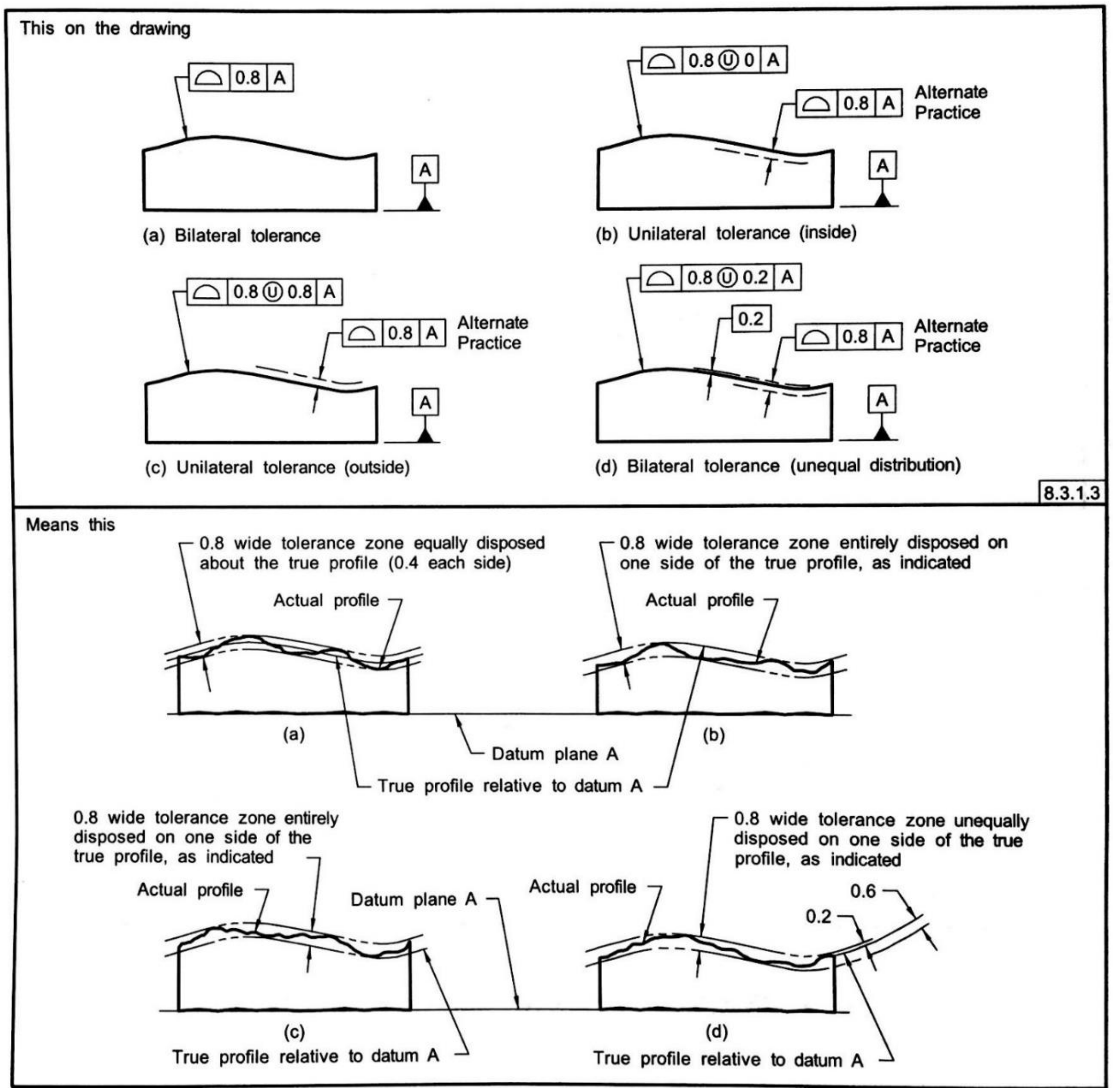


Figure 13. Profile of a Surface Tolerance Application (ASME.Y14 2009)

Profile of a Line:

Profile of a line is applied to surfaces with varying cross section line like a tapered wing of an aircraft or a constant cross-section, such as an extrusion, where it is not desired to have a tolerance zone included for entire surface of the feature. The tolerance established by the profile of a line encompasses two-dimensional area and the tolerance zone is normal to the true profile of the feature at each line element.

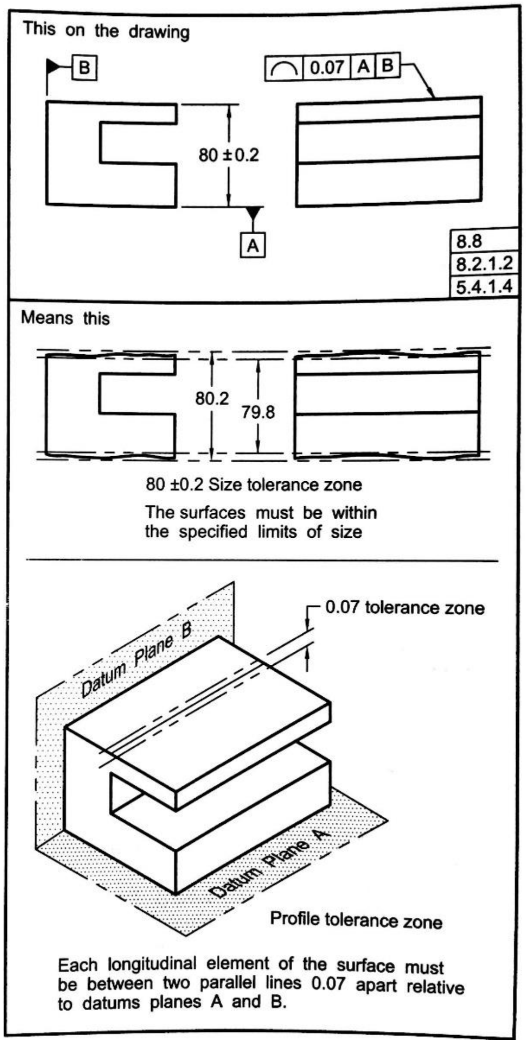


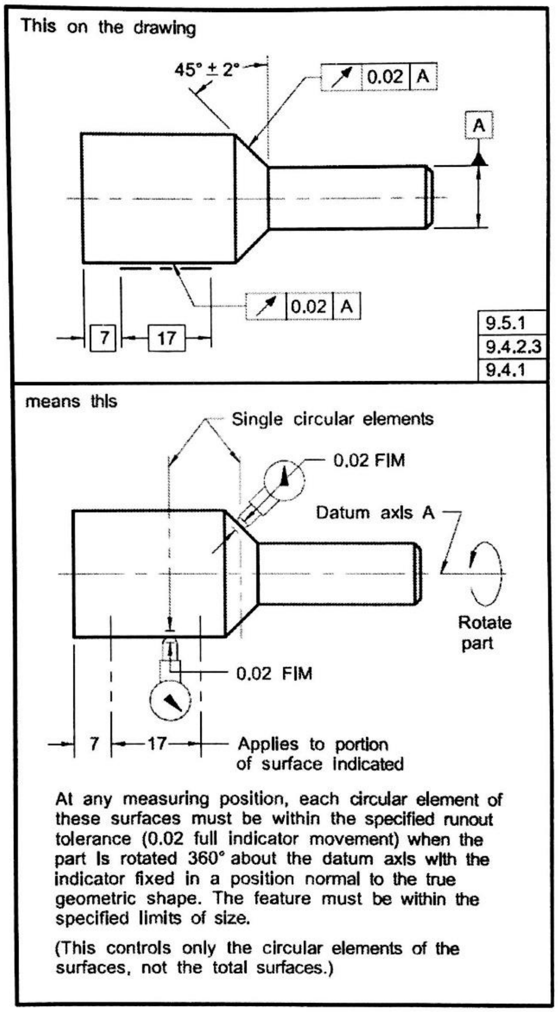
Figure 14. Profile of a line Tolerance Application (ASME.Y14 2009)

2.3.5 Runout Tolerances

A run out tolerance defines how far an actual surface or feature is permitted to vary from the desired from implied by the drawing during one full rotation (360^o) of the part on a datum axis. Runout tolerances are further categorized into circular runout and total runout.

Circular runout:

It defines the tolerance control that is applied to the circular elements of a surface. The tolerance is applied independently at each circular measuring point at the part is rotated about its axis to the angular extent up to which its possible. Figure 15 shows how the measurement of circular runout is carried out.



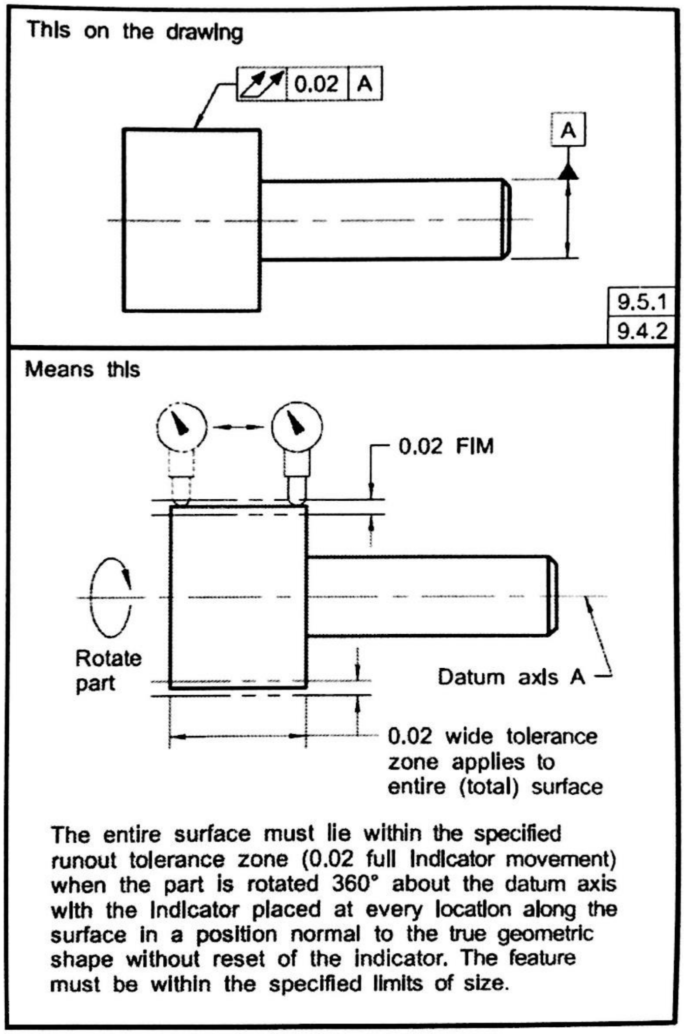
At any measuring position, each circular element of these surfaces must be within the specified runout tolerance (0.02 full indicator movement) when the part is rotated 360° about the datum axis with the indicator fixed in a position normal to the true geometric shape. The feature must be within the specified limits of size.

(This controls only the circular elements of the surfaces, not the total surfaces.)

Figure 15. Circular runout (ASME.Y14 2009)

Total runout:

It is used to define the tolerance control of all the surface elements. The tolerance is applied at the same time to all the circular and profile measuring positions as the part is rotated 360^{o} about the datum axis. Figure16 shows how the measurement of total runout is carried out.



The entire surface must lie within the specified runout tolerance zone (0.02 full indicator movement) when the part is rotated 360° about the datum axis with the indicator placed at every location along the surface in a position normal to the true geometric shape without reset of the indicator. The feature must be within the specified limits of size.

Figure 16. Total run out (ASME.Y14 2009)

Chapter 3: Literature review

3.1 Enriching Product lifecycle knowledge

An ontology can be defined as a formal way of naming and defining the types, properties, and interrelationships of the entities that exist in the particular domain of discourse (Patil, Dutta, and Sriram 2005). In engineering, the term is primarily used for knowledge representation and for the reuse of existing available knowledge (Stojadinovic and Majstorovic 2012). The rich syntactic and strong logical reasoning mechanism of an ontology enables integration and sharing product data throughout its lifecycle. Besides the need of representing and reusing the knowledge, it is also necessary to share the correct knowledge of a product data in a collaborative development of a complex product. This sharing of product data is not just restricted to initial design stage but also extends to the product's entire lifecycle. As discussed in the previous chapter, ISO formulated STEP as a standard format for exchanging the product data. But STEP has difficulty in capturing beyond geometry data of a product.

In order to overcome this drawback, many studies were carried out to enrich the semantics of the product data. Initial studies revolved around developing a method for translating the STEP product data written in EXPRESS to ontological format OWL (Ontology Web Language) (Institute of Systems Engineering 2014). However, this implementation only supported translation of modular STEP application protocols. Therefore, there were certain limitations on the translation of product data (Krima et al. 2009) . These limitations were concerned with the complexity of a mechanical product arising due to variety of information elements (like function, behavior, structure, geometry and material, assembly features, tolerances, etc.) needed to be represented in the ontology. In another paper, (Barbau et al. 2012) proposed a translation method to translate STEP schema and its instances to OWL. This translation was formulated into a Protégé plugin by name OntoSTEP plugin. The aforementioned plugin is capable of translating EXPRESS schema and CAD files to OWL so that a semantically enriched product data model is generated. But the major part of product data pertains to the geometric dimension of the product and very less beyond geometric data.

(Subrahmanian et al. 2005; Fenves et al. 2008) combined OntoSTEP model with core product model and open assembly models that were previously developed at NIST. This combination of knowledge enabled further enrichment of the semantics of the product lifecycle data in terms of product analysis, product function, etc. (Lu et al. 2015). (Huang et al. 2015) proposed an ontology that would improve the assembly process knowledge of the concerned product while taking into consideration assembly requirement, spatial information, assembly operation and assembly resource. The ultimate goal of this ontology is to aide in improving assembly process planning for a complex assembly product.

The Table1 depicts a comparison between the models/ methodologies proposed in different research articles discussed in the above section. The criteria used for the comparison are; C1: Inter-platform compatibility, C2: Covers all phases of product lifecycle, C3: Captures geometric as well as beyond geometric data of the product. C4: Captures tolerance and metrology data of the product.

Table 1. Comparison between research works on Enriching Product lifecycle knowledge

Research Article	C1	C2	C3	C4
S-TEN—intelligent self-describing technical and environmental networks **(Institute of Systems Engineering 2014)**	X		X	
OntoSTEP: Enriching product model data using ontologies **(Barbau et al. 2012)**	X	X		
A revised core model for representing design information. **(Fenves et al. 2005)**	X	X	X	
CPM2: A core model for product data. **(Fenves et al. 2008)**	X	X		X
Ontology model for assembly process planning knowledge. **(Huang et al. 2015)**		X	X	X

3.2 Enriching GD&T knowledge

It is worth mentioning that none of the previously mentioned research works has any tolerance or metrology related information enveloped into their product lifecycle model. In order to incorporate tolerance related information into the product ontology, (Sarigecili, Roy, and Rachuri 2014), combined the OntoSTEP model developed in previous research work with a tolerance analysis oriented model. The integrated ontology thereby obtained was proposed to help the designers help in interpreting the semantic information pertaining to tolerance allocation of the product during the different phases of its lifecycle. However, this information is not enough to reflect the real life dimensional changes of the product and its parts

(Zhong et al. 2013) decided to integrate variational geometric constraint (VGC) model along with the OntoSTEP model so as to enrich the product data extracted from CAD systems. The benefits of the ontology thus developed includes consistency checking, knowledge reasoning and performing automatic semantic queries. However, this approach helps in capturing the tolerance data of the product at its design stage only. Also, it does not capture the metrology data pertaining to tolerance information after the product has been manufactured. In their later work, (Lu et al. 2015) developed an ontology-based approach for automatic generation of assembly tolerance type. The authors claimed that this approach would reduce the uncertainty associated with assembly tolerance specification design.

Meanwhile, (Qin et al. 2014) developed an ontology based model for tolerance representation for spatial relationships. But in this model, every instance of the class had to be created manually; which is a time-consuming process. Also, the assembly tolerance representation only deals with product lifecycle in its design and post manufacturing phase. In their later work, (Qin et al. 2017) proposed a descriptive logic ontology based approach for representing composite positional tolerance for a pattern of holes. To develop this ontology, the authors also used semantic web rule language [SWRL]. Similar to their previous work, the authors focused on product tolerance in the design phase of the product lifecycle. (Zhong et al. 2013) used OWL to establish the ontology model for a product model and an assembly tolerance. SWRL rules were used to represent the assembly tolerance knowledge and define the relationship between assembly features and assembly tolerances. On the basis of their work, automatic marking of assembly tolerance is achieved.

(Stojadinović and Majstorović 2014) worked on designing a metrology-based ontology that can capture such information. The methodology implemented involves the development of an ontology for the construction of knowledge base as a part of an intelligent system for the inspection of prismatic parts. In this case, the model developed is restricted to the inspection of post manufactured prismatic products, i.e. products with non-cylindrical components only. Also, the example cited in the research work includes an ontology developed for a single component product where as in real life scenario, a single product consists of several components and sub-components assembled together. As in the previous cases, the instances of the class had to be created individually.

The Table 2 depicts a comparison between the models/ methodologies proposed in different research articles discussed in this section. The criteria used for the comparison are; C1: Inter-platform compatibility, C2: Covers all phases of product lifecycle, C3: Captures geometric as well as beyond geometric data of the product. C4: Captures tolerance and metrology data of the product.

Table 2. Comparison between research works on Enriching GD&T knowledge

Research Article	C1	C2	C3	C4
Interpreting the semantics of GD&T specifications of a product for tolerance analysis	X	X		X
Enriching the semantics of variational geometric constraint data with ontology	X		X	
Explicitly representing the semantics of composite positional tolerance for patterns of holes	X		X	X
An assembly tolerance representation model based on spatial relations for generating assembly tolerance types	X			X
Automatically generating assembly tolerance types with an ontology-based approach	X		X	X
Towards the development of feature-based ontology for Inspection Planning System on CMM	X		X	X

Chapter 4: Methodology

This chapter discusses the methodology developed for creating the ontology that captures the metrology information of a product during its entire life cycle. The process begins with comprehending and consolidating the steps used in the creation of engineering ontologies from previous research works. (Li, Yang, and Ramani 2009; Gruber 1995) are a few attempts at creating and validating ontologies for product lifecycle representation. The main contribution to product ontology, according to those authors, is the creation of a new, systematic, and more structured ontology development method assisted by a semiautomatic acquisition tool. The researchers laid down six steps to be followed for designing an engineering ontology (EO). For creation of an ontology that would aide in capturing the metrology product data an additional step of feature selection was added. Figure 16 depicts the process of development of EO for metrology.

While there have been very few attempts in developing engineering ontologies in the field of coordinate metrology (Stojadinović and Majstorović 2014), all of them have been focused on ontologies for prismatic parts. Also, the ontologies were designed only for post manufacturing inspection phase of a product’s lifecycle. A common theme while designing these ontologies was the emphasis on clarity in class definition, uniformity among different classes and extendibility of ontology to different domains.

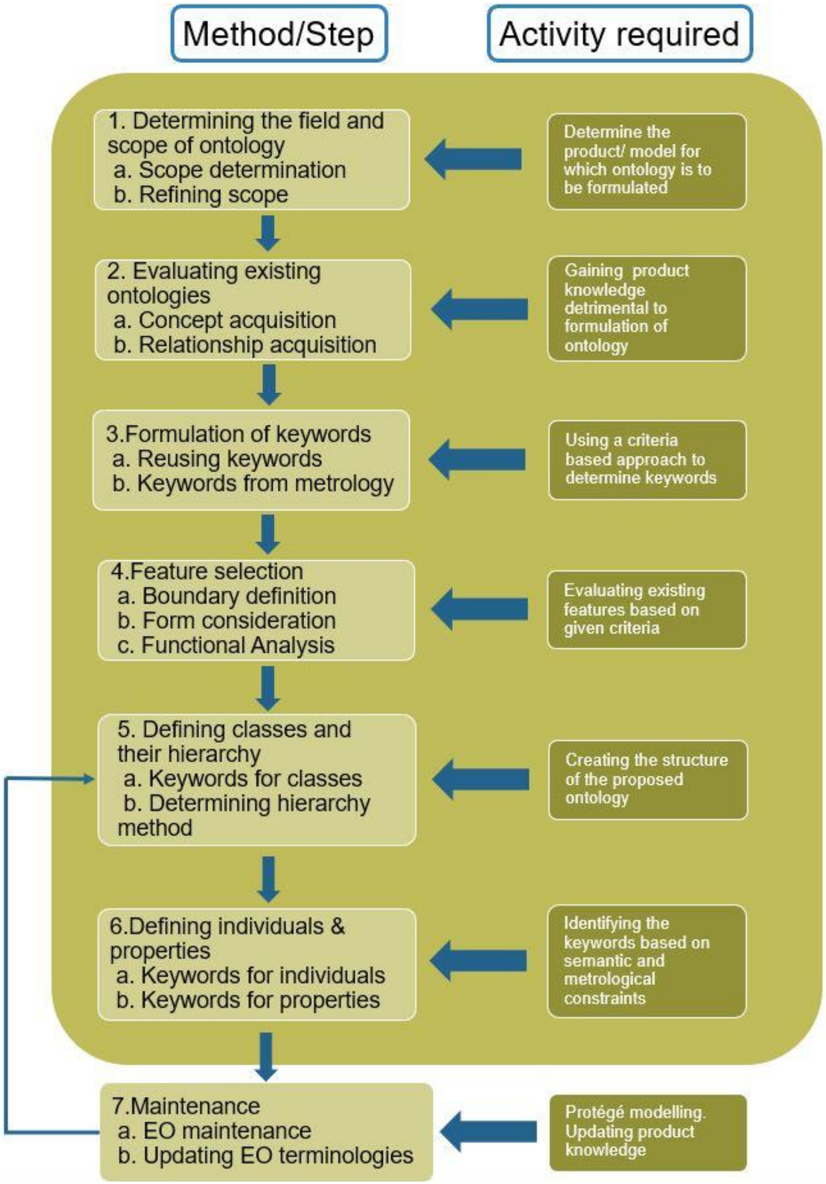


Figure 17. EO development process

Steps for designing Engineering ontology

In case of developing an EO of a product, it is necessary to gather all the relevant product information in a compatible format for developing the ontology. The first step for any product information starts from its design documents, that includes in large part the CAD files, which are usually stored in IGES or STEP format. Also, other files can contain non-geometric data about the product which can be relevant in creating the knowledge base for developing EO. This information will help us in formulating the metrontology template that can be later adapted on case by case basis, depending on the product under consideration.

The steps for the proposed methodology are as follows:

Step1. Determining the field and scope of ontology

It includes domain of ontology, purpose of ontology, and maintenance of ontology. One of the ways to determine the scope of ontology that was discussed by (Grüninger and Fox 1995) is by making a list of questions that the ontology must be able to answer. These questions and answers to them will help in improving the ontology in its nascent form and also forming a boundary for the scope of the ontology.

For developing a metrontology, the scope lies within the domain of metrology concepts such as geometric dimensions, tolerance properties and types of tolerances applied to the parts, inspection techniques used for ascertaining part dimensions, and manufacturing process used for the part. The purpose of the ontology can be ascertained by keeping the end goal in mind, that is capturing the metrology data of the part over its entire life-cycle. Table 3 enlists the scope, purpose and questions considered for designing metrontology.

Table 3. Determining Scope and Purpose of metrontology

Scope	Geometric dimensions, tolerance properties, types of tolerances, part inspection techniques, material properties.
Purpose	Capturing time series metrology data of existing part, aide in improvement of product tolerances, knowledge reuse.
Questions	What are the components of the ontology? How will the information be captured and reused? What is the degree of complexity of the product and its components? Which existing ontology can be reused in creating new ontology? How can the ontology be integrated with metrology information enclave and/or PLM software tools?

The maintenance of ontology would involve updating the concepts and keywords used in the ontology from time to time as and when the product design is updated. To understand this better, consider an example of a simple cylindrical rod. The design features that would have to be considered are enlisted as follows: dimension, tolerances, surface finish, material etc.

Step2. Evaluating existing ontologies

It is necessary to understand existing ontologies and scope of those ontologies. Also, it would give an idea about adaptation of certain portions from existing ontologies to the new ontology that is being developed. Thorough analysis of existing ontologies need to be done in terms of hierarchy & organization of classes, purpose of the classes, similarity between the domain of existing ontologies and the one being developed.

Libraries with existing ontologies were accessed at (ProtegeWiki 2018). Several ontologies were accessed for understanding keyword selection, class-hierarchy formulation and instances creation. One of the EO developed by (Stojadinović, 2014) considered geometric primitives (circle, cylinder, line, sphere, etc.) as classes and numbered the occurrence of those primitives in certain order. Another number was used to denote the instances of the class. For example, to denote the instance of a point on the surface of component, (Stojadinović, 2014) used notation K_1_3; where K_1 denotes the label used to describe the point in the EO while '_3' denoted the third occurrence of the point on the surface of component.

Step3. Formulation of keywords for EO

This is the most important step in developing the ontology, and a step that needs to be repeated more often than other steps during the initial development of EO. Based on the information collected in previous two steps, evaluation of possible terms and key words is carried out.

Some of these terms will then be used for creating classes, some will be used for creating class properties, few others will be used for creating object properties, and remaining terms will be unused as they are not important to define optimal scope of the EO for the time being. Many of these keywords can be adopted from existing ontologies that have similar scope as the ontology being developed. There are three ways in which an EO ontology can be developed; top-down approach, bottom-up approach, and combined approach.

In case of metrontology, the keywords are formulated or adopted based on following criteria:

a) *CAD model*: The keyword information already exists in form of component and sub-components names of the product assembly. The CAD model can be a useful source of keywords.

b) *Metrology concepts*: these include geometric and tolerance values, types of tolerances, sampling techniques used to capture the tolerance data, sample size, and name/type of the machine used to capture this data.

c) *Time-series instances*: As the intention of this metrontology is to capture the metrology data of the product at different time intervals throughout its lifecycle, keywords pertaining to various instances also needs to be formulated.

Table 4 enlists the keywords based on the criteria described above:

Table 4. Criteria based approach for formulating keywords for metrontology

Criteria	Keywords
CAD-model	*assembly_name, component_name, FI_i_feature_name, material_name*
Metrology concepts	Sample size (*ss16, ss32, ss64*), tolerance types (*Flatness, Circularity, Cylindricity, Parallelism, Perpendicularity, Angularity, Position tolerance, Concentricity, Profile of a surface, Profile of a Line, Circular runout, Total runout*), sampling techniques (*Hammersley_method, aligned_systematic_method, random_sampling_method, Hammerspi_method, spiral method*)
Time-series instances	*I_i_feature_name, InspI_i, post_manufacturing, maintenance, end_of_life*

Here, FI_i denotes feature number and i=1,2,3…n

I_i denotes instances at which the data is captured and i=1,2,3...n

Later, these keywords will be to represent classes, subclasses, class properties, object(instance) properties, and instances of the EO.

Step4. Feature selection

Feature selection is a systematic approach that helps in reducing the number of components and geometric features that would otherwise have to be inspected leading to excessive time utilization as well as creation of redundant knowledge that may not be useful later. The feature selection can be better understood with the help of functional block diagram. Here B>A, and A>a, B>b.

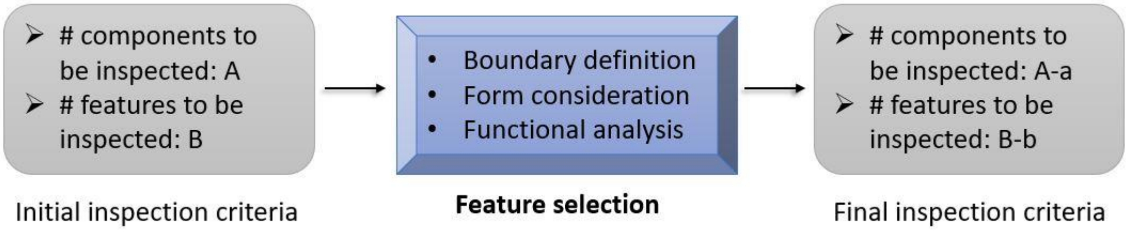


Figure 18. Feature selection

This is an additional step that needs to be considered while developing an EO focused on capturing metrology information throughout the products' lifecycles. According (Kissick 2014), in order to reduce the excessive amount of time that is being spent on evaluating all the features of a product during its maintenance, it is necessary to carry out a systematic evaluation of components to determine only few features that should be inspected. The principal criteria for selecting features of the component of a product are explained below:

a) *Boundary definition:* It involves defining a subcomponent of a product beyond which, the inspectors do not intend to divide the component further into its subcomponents. Usually a boundary is defined about an individual component or part of the assembly. A single boundary can be defined for two or more components if those perform single common task. Such components are called as Line Replaceable Units (LRUs). LRUs are usually designed to be installed and replaced as a whole unit if they fail.

b) *Form consideration:* An item's form is its external dimensions that define its boundary. In general, it is assumed that the form of the component is correct if individual parts are correct and there is no need to perform inspection on the individual parts. Knowing how this item interacts with and fits together with other subcomponents helps in determining which tolerance features can be selected for inspection. In majority of the cases, the components form is not inspected unless it has very tight tolerance zone under consideration (Kissick 2014).

c) *Functional analysis:* It is important that a component performs the function it is designed for. Thus, it is necessary to list and rank all the functionalities of the components in order of their importance, identify parts that are prone to failure while performing their function, and compile this information for narrowing down feature selection.

   Ranking the functionalities of the components in the order of their importance would not just aide in reducing the inspection time but would also make sure that designers focus on designing the components whose functionalities are core to the performance of the product. This will help in collecting just information to be included in metrontology.

Other considerations: It can include issues like performance history of the component, quality of the components, material failures, etc.

The necessity to include feature selection in developing metrontology arises from the fact that amount of data being captured and retained in a time-series instances very large. Although it’s possible to store this data in modern day servers and other storage systems, it is not possible to effectively utilize all the data while taking design decisions for new component being designed.

Step5. Defining classes and their hierarchy

From the keywords tabulated in step3, we select those which will formulate the structure of the ontology as classes and subclasses. In case of the metrontology being proposed, the selection of keywords for classes depends on product under consideration, application of feature engineering to its components, as well as usage of metrology concepts for inspection and end user i.e. product designer.

Once the keywords to be used for defining classes and subclasses have been identified, there are several ways to determine the class hierarchy for the EO being developed. (Horridge et al. 2009) represented classes as a set of individuals. The classes were organized into superclass-subclass which is also called as taxonomy. Figure 19 depicts this approach.

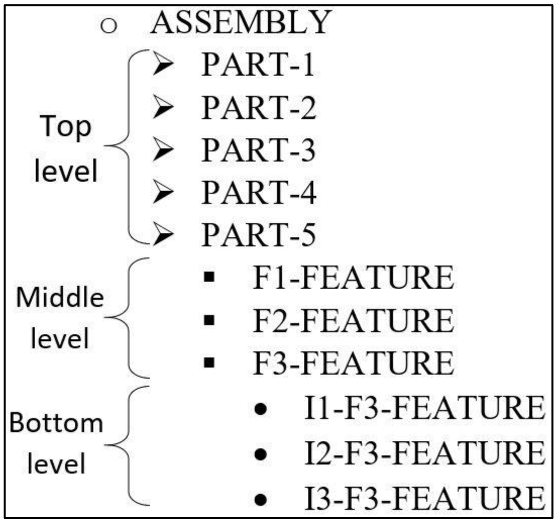


Figure 19. Taxonomy

(Uschold and Gruninger 1996), suggested that class can be defined in either of the following three ways:

Top – down: Development process of engineering ontology begins with definition of the most general concept.

Bottom – up: Development process of engineering ontology starts from the most specific classes and their hierarchy.

Combined: Development process of engineering ontology that combines the previous two ways.

Figure 20 shows all three approaches.

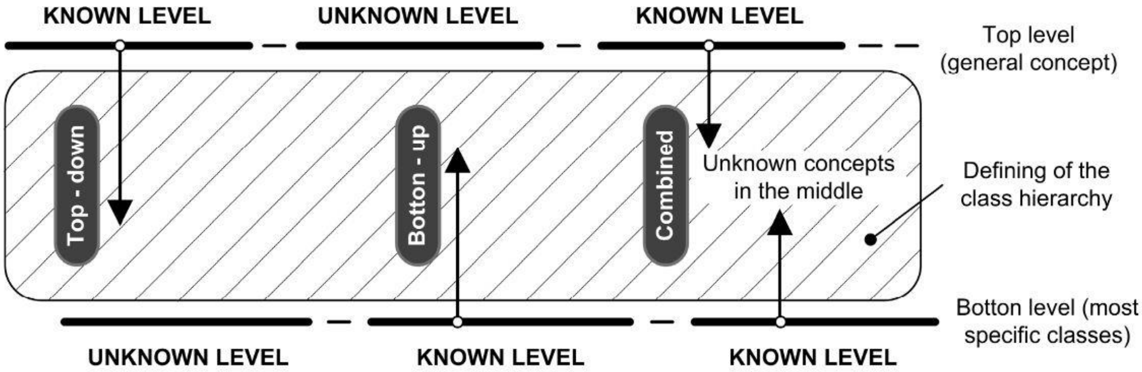


Figure 20. Various approaches to creating class hierarchy

The approach among the above three that is to be chosen depends on the level of knowledge of the specificity classes. If more knowledge is available regarding the specifics of a sub class, then a Bottom-up approach is more reliable. On other hand, if more information regarding the product as a whole is available, then it makes sense to adopt a top-down approach in designing its ontology. For the metrontology being developed, the product information dealing with assemblies and sub-assemblies is assumed to be readily available from the CAD model. This will help in determining the class hierarchy. The proposed class hierarchy has been tabulated in Table 5:

Table 5. Class and sub-class definitions of metrontology

Class name	Definition
assembly_name	Superclass of all the components of the product
component_name	Component that is subclass of the *assembly_name* imported from the CAD model
FI_i_feature_name	Defines the surface feature whose instances will be used for knowledge capture. The number of features would vary from one component to other and would also depend on implementation of feature selection on various components.

The product information (class names and basic dimension) regarding the assembly and sub-assemblies will be imported into the EO from CAD model using

OntoSTEP plugin (Barbau et al. 2012). Other information pertaining to component features and instances is imported from MS Excel to the metrontology using 'Cellfie'(Johardi 2018). It is a Protégé Desktop plugin for mapping spreadsheets to OWL ontologies.

Step6. Defining individuals and properties

In order to make the ontologies meaningful, it is necessary to define characteristics in terms of properties as well as include examples from the knowledge base to justify the creation of classes that were created in previous step.

Individuals are real world examples of the classes/ sub-classes that were created to be part of EO. They are also called as instances and represent the lowest possible level of hierarchy in the ontology. Individuals can be created manually or imported automatically with help of various java plug-ins that have been designed for various data types.

The keywords that used for defining in individuals in metrontology have been tabulated in Table 6:

Table 6. Defining Individuals of metrontology

Individual name	Definition
I_i*_feature_name*	Encapsulates the knowledge for the feature_name, Ii denotes the period at which the information is recorded.
$Insp_i$	Used to identify the time period at which the inspection was carried for the entire assembly.

Properties describe the internal structure of class or concept (Lu et al. 2015). There are two types of properties; properties that describe relationship between pair of individuals are called as object properties. In metrontology, the object properties are tolerance types that determine the nature of the inspected feature of the component. Properties that describe components of individuals are called as individual properties or data properties. In case of metrontology, the individual properties describe various aspects of the tolerance values being captured of the feature. This includes time-series instance, sampling technique used for determining the tolerance and number of sample points used in determining the tolerance. Table 7 describes object properties:

Table 7. Defining Object properties of metrontology

Object property	Definition
Flatness	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the component feature is flat.
Circularity	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the shape of component feature is circular.
Cylindricity	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the inner/ outer shape of component feature is cylindrical.
Parallelism	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the feature is determined parallel wrt another feature of the component.
Perpendicularity	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the feature is determined perpendicular wrt another feature of the component.
Angularity	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the feature is determined to be at a certain angle wrt another feature of the component.
Position tolerance	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the feature location is expressed wrt a datum feature
Concentricity	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the feature location is expressed as concentric wrt a datum feature
Profile_of_a_surface	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the shape of component surface is non-linear.
Profile_of_a_Line	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the edge of component surface is non-linear.
Circular_runout	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the cyindrical component surface non-uniform diameter at a particular length along the axis.
Total_runout	Defines the relationship between feature instances *I_i_feature_name* and *$Insp_i$* when the cyindrical component surface non-uniform diameter length throughout the length of the axis.

Table 8 describe data properties:

Table 8. Defining Data Properties of metrontology

Data property	Definition
After_manufacturing	Describes the instance when the tolerance value of feature is captured post manufacturing.
maintenance	Describes the instance when the tolerance value of feature is captured during the maintenance of the component.
End_of_life	Describes the instance when the tolerance value of feature is captured when the component is determined no longer to be useful.
Random_sampling_ method	Describes the instance when the sampling method used for inspection the feature is random sampling.
Aligned_systematic_ method	Describes the instance when the sampling method used for inspection the feature is aligned systematic sampling
Hammersly_method	Describes the instance when the sampling method used for inspection the feature is Hammersly sampling.
Hammerspi_method	Describes the instance when the sampling method used for inspection the feature is hammerspi sampling.
Spiral_method	Describes the instance when the sampling method used for inspection the feature is spiral sampling.
ss16, ss32, ss64	Describes the sample size used for the inspection at the particular instance.

After the structure of the metrontology had been developed, it is necessary to populate it with tolerance data collected at various instances over the product's lifecycle.

Step7. Populating and Maintaining of EO

After creation of ontology it necessary to maintain it UpToDate. In case of an ontology that stretches over the entire lifecycle of the product, it is updated periodically as and when new data is obtained after servicing of the product. Also, with product upgrades, it would become necessary to update few key terminologies (classes, properties, etc.) that are part of existing EO.

The data collected can be added to metrontology in two ways- (a) by manually adding as an instance of the component feature. (b) by using a 'plug-in' to import several instances at one time.

Based on the above discussion of seven steps to create an EO, a block diagram illustrating the process of development of metrontology is as follows:

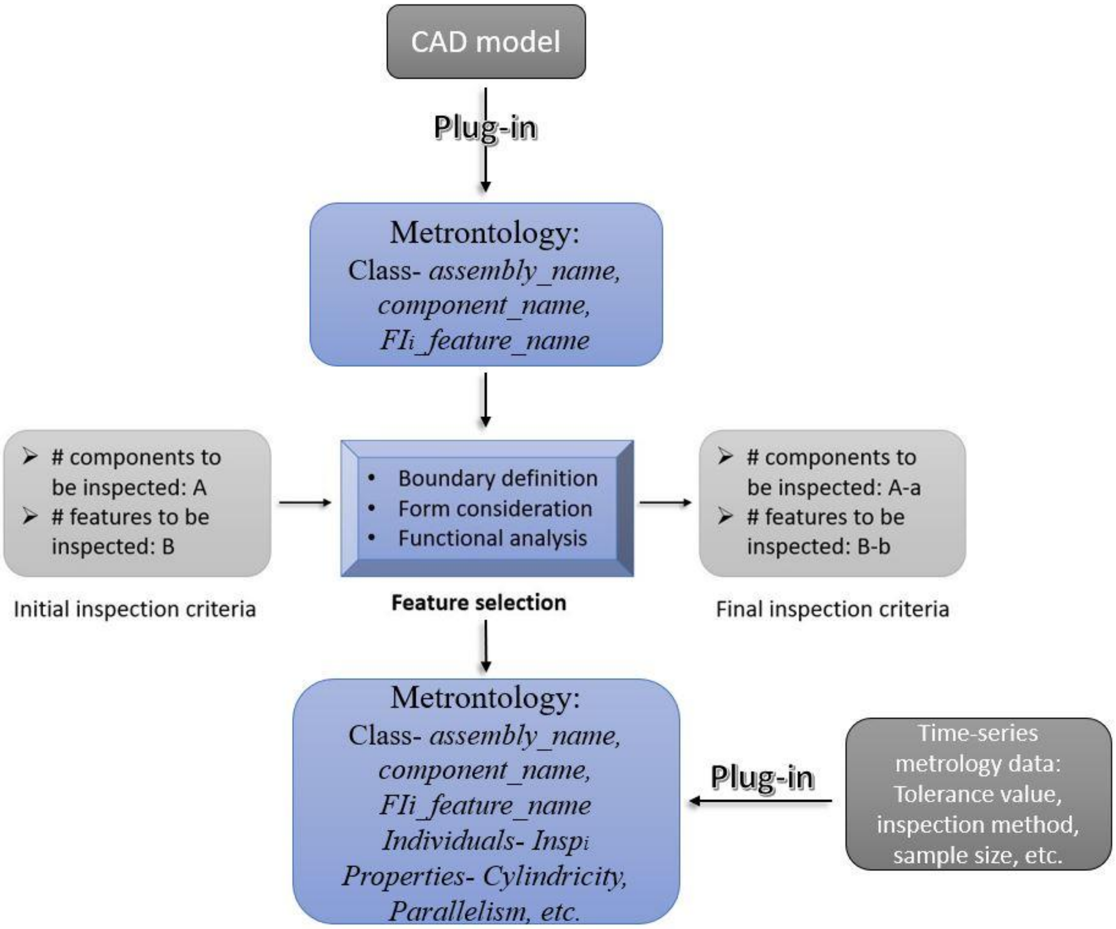


Figure 21. Formulation of Metrontology

As illustrated in the block-diagram (Figure 21), the time series metrology data is imported using a plugin. Generally, the captured data is stored in MS Excel file after the inspection of the component. Therefore, Cellfie plugin is used to transfer the data from MS Excel file to metrontology.

Chapter 5: Implementation and Example Analysis

This chapter supplements the methodology described for creating metrontology with two examples; (a) simple cylindrical rod and (b) single cylinder engine. The knowledge base created for developing these EOs is primarily made of information that's relevant to tolerance inspection throughout the product's lifecycle.

The metrontologies have been created in commonly used ontology editor called *Protégé*. This application has been written in Java and uses Swing to create to create its graphic user interface. At its core, Protégé uses a set of knowledge modeling structures and actions that support the formulation, visualization, and manipulation of ontology (Fenves et al. 2008). Also, the EOs that are being created will be done so in RDF/XML (one of the most popular formats), the ontology can be incorporated into a wide variety of PLM software.

5.1 Metrontology for Cylindrical rod

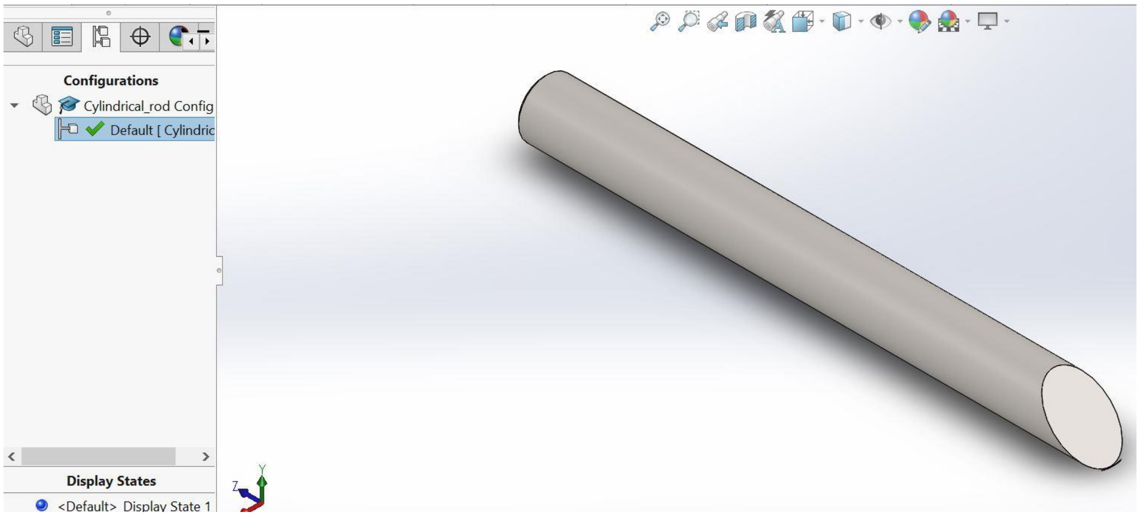


Figure 22. CAD model of Cylindrical rod

The cylindrical rod shown above had nominal geometric dimensions of 10mm radius and 100mm length. The material selected for the rod is plain carbon steel. The metrontology for cylindrical rod is developed as follows:

1. Determining the field and scope of ontology

As the ontology being developed is for capturing the metrology data of the cylindrical rod, the scope of the information is limited to the metrology concepts of geometric dimensions, tolerances, inspection technique used, etc. The dimensional and tolerance data of cylindrical rod is as follows:

Table 9. Geometric and tolerance data of Cylindrical rod

Feature	**Dimension**	**Tolerance**
Diameter	10mm	±0.5 mm
Length	100 mm	±1 mm
Inclined surface	45^o	$\pm 3^o$
Chamfer	0.5mm, 45^o	-

2. Evaluating existing ontologies

The engineering ontologies at the online ontology library (ProtegeWiki 2018), does not have ontology with focus on metrology. Also, the ontology for coordinate metrology developed by (Stojadinović and Majstorović 2014) is restricted to prismatic parts therefore cannot be used in case of cylindrical surface.

3. Formulation of keywords

As stated in previous chapter, the formulation of keywords is based on CAD-model, Metrology concepts, and Time-series instances. Table below shows the keywords based on these criteria:

Table 10. Keywords for metrontology of Cylindrical rod

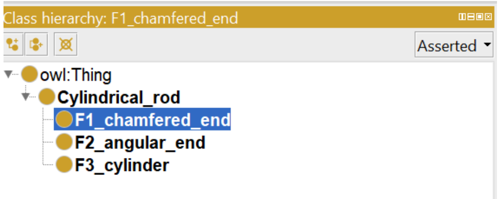
Criteria	Keywords
CAD-model	*Cylindrical_rod, cylinder, chamfered_end, angular_end*
Metrology concepts	Sample size (*ss16, ss32)* Tolerance types (*Flatness, Circularity, Cylindricity, Perpendicularity, Angularity*), sampling techniques (*random_sampling_method, Hammerspi_method, spiral method*)
Time-series instances	I_i*_cylinder,* I_i*_chamfered_end,* I_i*_angular_end, InspI1, InspI2, InspI3, post_manufacturing, maintenance, end_of_life*

4. Feature selection

As the cylindrical rod is single component with 3 features, we cannot use the concepts of feature selection in this case.

5. Defining classes and their hierarchy

The OntoSTEP plugin facilitates import of STEP file data of the CAD model into Protégé. Along with OntoSTEP plugin, Cellfie plugin is used for importing component feature keywords for the cylindrical rod. This ensures the classes are defined according to the name of components and surface features. In case of metrontology, as the information regarding component features is obtained after the information of component is known, the top-down approach of designing ontology is followed. The figure below depicts the image of class hierarchy for cylindrical rod in Protégé.



Class hierarchy: F1_chamfered_end

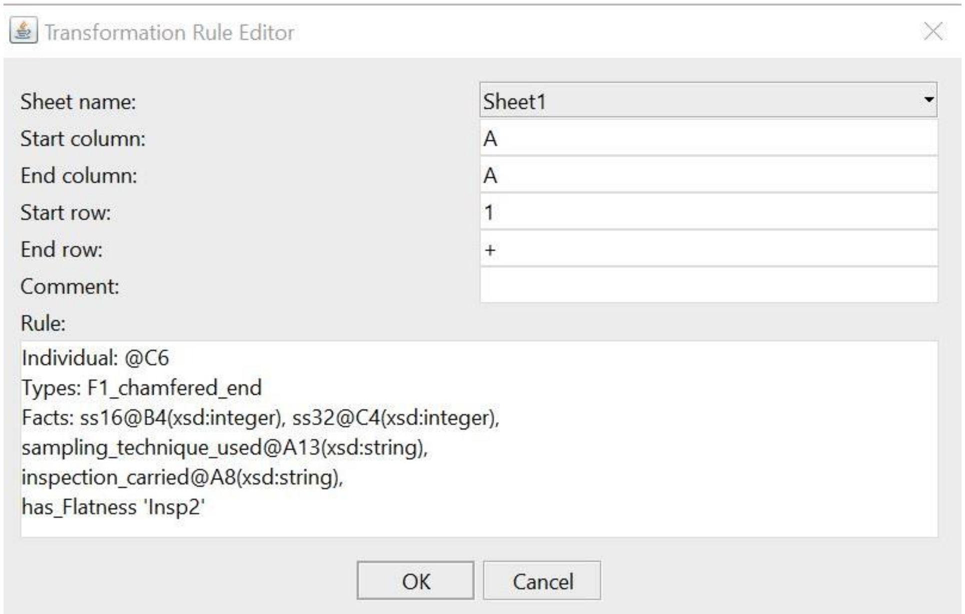
- owl:Thing
  - **Cylindrical_rod**
    - **F1_chamfered_end**
    - **F2_angular_end**
    - **F3_cylinder**

Figure 23. Class hierarchy for Cylindrical rod

It is clear from the Figure 23 that the *owl:Thing* is the super class of all the classes. The component, cylindrical rod is represented by keyword *Cylindrical_rod* as secondary level class, while the features on the cylinder are represented as its sub classes by keywords *I_cylinder*, *I_chamfered_end*,and *I_angular_end*.

6. Defining individuals & properties

Individuals and properties are selected from the keywords that are defined in the first step for creation of metrontology. The individuals are imported from an Excel file using the Cellfie plugin.



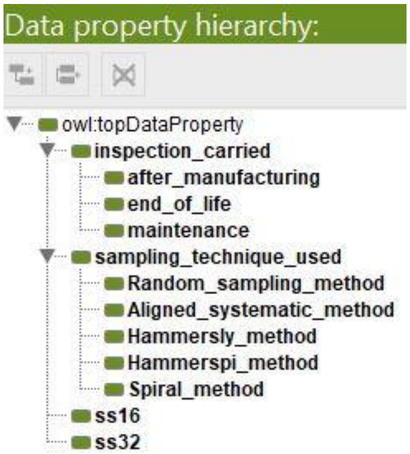
Sheet name: Sheet1
Start column: A
End column: A
Start row: 1
End row: +
Comment:
Rule:

```
Individual: @C6
Types: F1_chamfered_end
Facts: ss16@B4(xsd:integer), ss32@C4(xsd:integer),
sampling_technique_used@A13(xsd:string),
inspection_carried@A8(xsd:string),
has_Flatness 'Insp2'
```

Figure 24. Cellfie plugin denoting the creation of instances for 'Chamfered end' feature

In case of metrontology, the keywords used for defining object properties as well as data properties remain the same. Certain properties will be excluded from the metrontology of the product based on the components and features of each component. For example, the cylindrical rod has three surface features subjected to five different types of tolerances. Therefore, the object properties that will be used for developing the metrontology are *has_Flatness*, *has_Cylindricity*, *has_Circularity*, *has_Perpendicularity*,and *has_Angularity*.

While in case of data properties, as it will be impossible to extract 64 sample points from this small component, the data property *ss64* has been dropped and other properties been retained. Images below show the data properties and object properties implemented in Protégé.


Data property hierarchy:

- owl:topDataProperty
  - inspection_carried
    - after_manufacturing
    - end_of_life
    - maintenance
  - sampling_technique_used
    - Random_sampling_method
    - Aligned_systematic_method
    - Hammersly_method
    - Hammerspi_method
    - Spiral_method
  - ss16
  - ss32

Figure 25. Data properties of Cylindrical rod


Object property hierarchy:

- owl:topObjectProperty
  - has_Angularity
  - has_Perpendicularity
  - has_Circularity
  - has_Cylindricity
  - has_Flatness

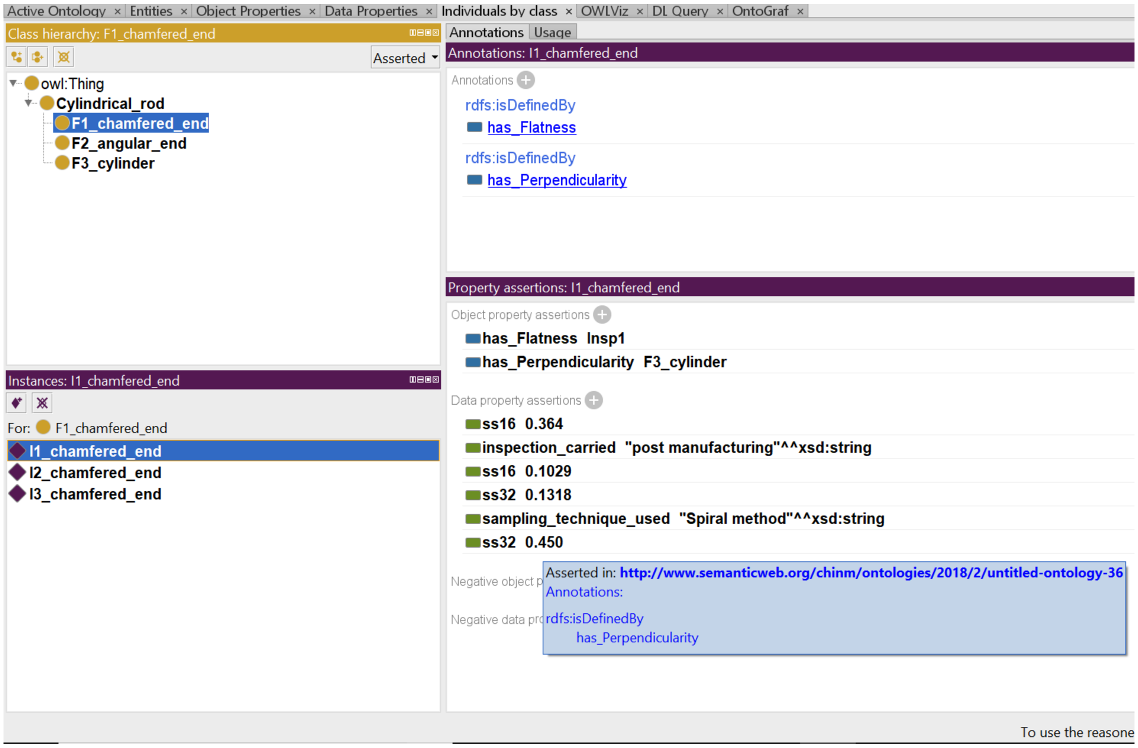
Figure 26. Object properties for Cylindrical rod

7. Populating and Maintenance of EO

The metrology data used for creating instances of cylindrical rod are stored in MS Excel file. In order to import this data as instances, Cellfie plugin is used. The advantage of Cellfie plugin lies in the fact that several instances can be created at one time in the ontology.

Maintenance of this ontology will include updating classes and sub-classes as the designer changes the product definition of the CAD model of cylindrical rod. Also, properties would have to be updated as and when metrology knowledge like tolerance types, and sampling techniques used would change.

8. Interpretation of the developed metrontology



owl:Thing
  Cylindrical_rod
    F1_chamfered_end
    F2_angular_end
    F3_cylinder

For: F1_chamfered_end

I1_chamfered_end
I2_chamfered_end
I3_chamfered_end

Annotations

rdfs:isDefinedBy
has_Flatness

rdfs:isDefinedBy
has_Perpendicularity

Object property assertions

has_Flatness Insp1
has_Perpendicularity F3_cylinder

Data property assertions

ss16 0.364
inspection_carried "post manufacturing"^^xsd:string
ss16 0.1029
ss32 0.1318
sampling_technique_used "Spiral method"^^xsd:string
ss32 0.450

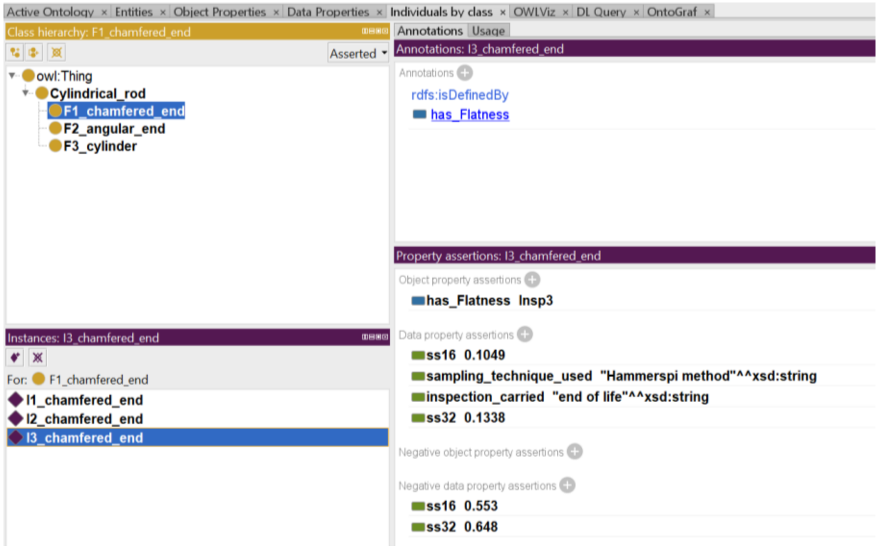
Asserted in: http://www.semanticweb.org/chinm/ontologies/2018/2/untitled-ontology-36
Annotations:
rdfs:isDefinedBy
has_Perpendicularity

Figure 27. Metrontology template view for instance *I1_chamfered_end*

The Figure 27 shows the overview of the developed metrontology for Cylindrical rod for the individual *I1_chamfered_end*. There are four windows with labels: Class Hierarchy, Instances, Annotations, Property assertions.

The *F1_chamfered_end* has been displayed as the subclass of *Cylindrical_rod*, which can be interpreted as *F1_chamfered_end* being feature of the part *Cylindrical_rod.* Annotations window displays the types of tolerances associated with the feature *F1_chamfered_end.* The Property window displays the object and data properties of the said individual. It can be interpreted that, during part inspection, the inspectors determined that the feature had *flatness* during first inspection as well as *perpendicularity* with respect to the axis of cylindrical surface.

Data property includes the information pertaining to the tolerance value obtained for checking *flatness* and *perpendicularity* of feature for the sample size of 16 and 32. The data property also include information regarding time-series instance at which the inspection was carried out as well as the sampling method used for carrying out this inspection.



Class hierarchy: F1_chamfered_end

- owl:Thing
  - Cylindrical_rod
    - F1_chamfered_end
    - F2_angular_end
    - F3_cylinder

Instances: I3_chamfered_end

For: F1_chamfered_end

- I1_chamfered_end
- I2_chamfered_end
- I3_chamfered_end

Annotations: I3_chamfered_end

Annotations

rdfs:isDefinedBy
has_Flatness

Property assertions: I3_chamfered_end

Object property assertions

has_Flatness Insp3

Data property assertions

ss16 0.1049
sampling_technique_used "Hammerspi method"^^xsd:string
inspection_carried "end of life"^^xsd:string
ss32 0.1338

Negative object property assertions

Negative data property assertions

ss16 0.553
ss32 0.648

Figure 28. Metrontology template view for instance *I3_chamfered_end*

Figure 28 shows the change in data capture in instance *I3_chamfered_end* when the feature no longer complies with one of the tolerance properties, therefore the tolerance values are classified as negative properties. As the surface *F1_chamfered_end* is no longer perpendicular to cylindrical surface *F3_cylinder*, the object property *had_Perpendicularity* is being dropped.

Figure 29 shows the block diagram for formulation of metrontology template

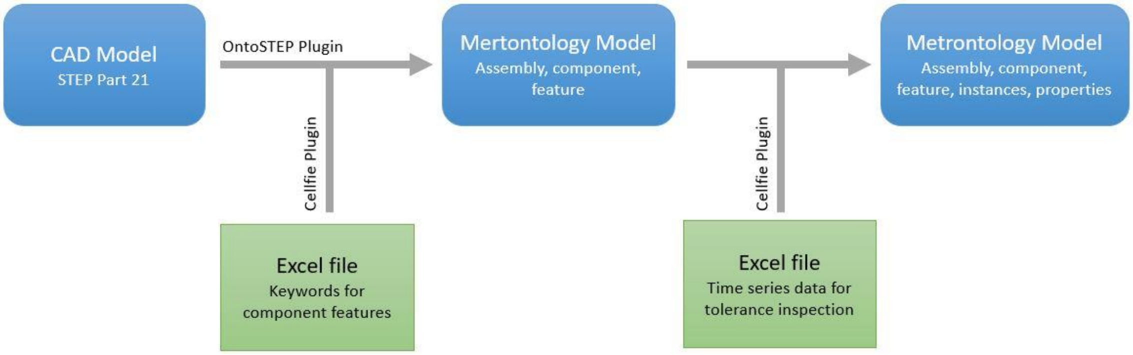


Figure 29. Metrontology template

5.2 Metrontology for Single Cylinder Engine

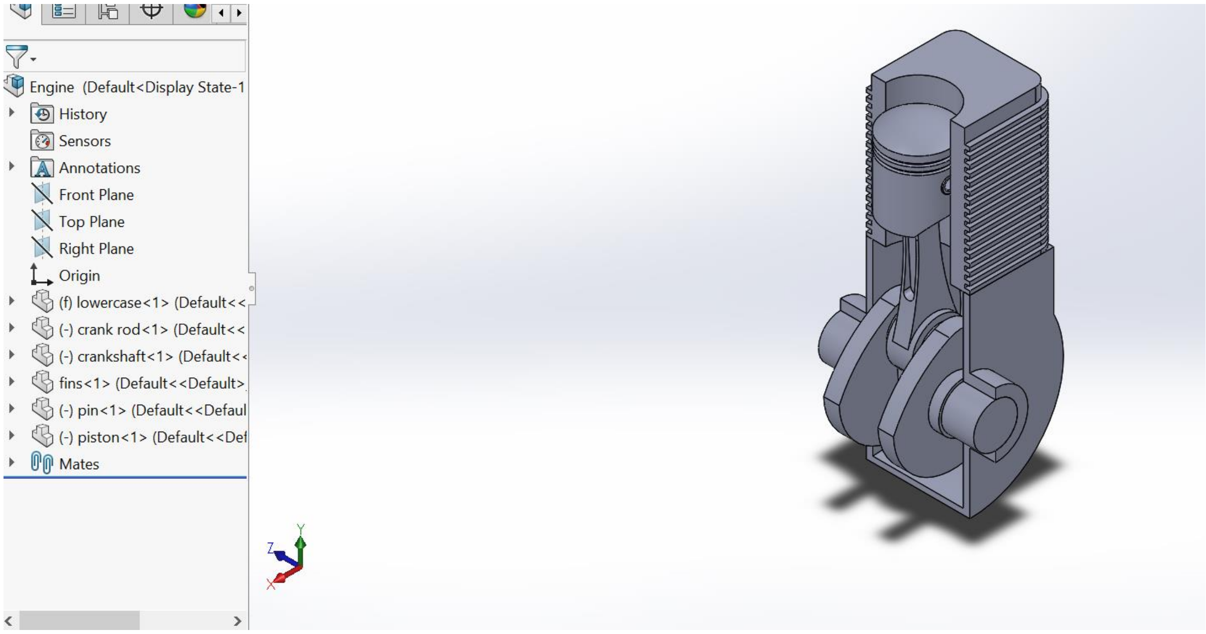


Figure 30. CAD model of Single cylinder assembly

The single cylinder engine (Lad 2016) had been modified to make it more suitable for feature extraction for creating a knowledge base for development of EO. The single cylinder engine is an assembly of crank rod, pin, piston, lowercase, crankshaft, and fins. Figure 30 shows the screenshot of the engine. The metrontology for cylindrical rod is developed as follows:

1. Determining the field and scope of ontology

In this case, the ontology is being developed for capturing the metrology data of the single cylinder engine, the scope of the information is limited to the metrology concepts of geometric dimensions, tolerances, inspection technique used, etc. The dimensional and tolerance data of single cylinder engine is tabulated in Table 11 as follows:

Table 11. Dimensional data of Single cylinder assembly

Feature	**Nominal dimension(mm)**	**Tolerance(mm)**
Inner bore	51	±0.012
Inlet valve	16.53	±0.025
Outlet valve	14.5	±0.025
Crankshaft (Bearing Journal)	34	±0.15
Crankshaft344(Connecting Rod Journal)	32	±0.15
Lower-case end-sleeve	34	±0.23
Connecting rod end 1	12.5	±0.20
Connecting rod end 2	32	±0.25
Piston head curvature	84	±0.11
Piston body	50	±0.012
Piston Pinhole diameter	12.6	±0.13
Pin diameter	12.5	±0.13

2. Evaluating existing ontologies

The engineering ontologies at the online ontology library (ProtegeWiki 2018), does not have ontology with focus on metrology. Also, the ontology for coordinate metrology developed by (Stojadinović and Majstorović 2014) is restricted to prismatic parts therefore cannot be used in case of cylindrical surface. The metrontology for cylindrical rod that was created in previous section is evaluated for keywords, class hierarchy and usage of properties and annotations for creating individuals of the subclasses of the ontology.

3. Formulation of keywords

The three criteria of CAD-Model, Metrology concepts and time-series instances for the formulation of keywords is implemented. Also, from the analysis of Cylindrical rod metrontology, it has been concluded that many of the keywords especially those concerned with properties can be reused for the development of current ontology. Following table lists the keywords to be used in the ontology:

Table 12. Keywords for metrontology of Single Cylinder Engine

Criteria	Keywords
CAD-model	*crankRod, crankshaft, fins, lowercase, pin, piston, Engine, Crankshaft_end, pinEnd, endrod1, endrod2, center_rod, Innerbore, valveopening1, valveopening2, endhole1, endhole2, mainbody, pinhole1, pinhole2, pistonBody, sphericalHead.*
Metrology concepts	Sample size (*ss16, ss32, ss64)* Tolerance types (*Flatness, Circularity, Cylindricity, Perpendicularity, Angularity, sphericity*), sampling techniques (*Hammersly_method, Hammerspi_method, spiral method*)
Time-series instances	I_i*_Crankshaft_end,* I_i*_ pinEnd,* I_i*_ endrod1,* I_i*_ endrod2,* I_i*_ center_rod,* I_i*_ Innerbore,* I_i*_ valveopening1,* I_i*_ valveopening2,* I_i*_ endhole1,* I_i*_ endhole2,* I_i*_ mainbody,* I_i*_ pinhole1,* I_i*_ pinhole2,* I_i*_ pistonBody,* I_i*_ sphericalHead, InspI1, InspI2, InspI3, maintenance, post_manufacturing, end_of_life.*

4. Feature selection

It can be realized from the assembly of the single cylinder engine that due to large number of features that each component has, it will be time consuming to inspect each and every feature. Also, many of the features may not be helpful from designer's perspective while redesigning the engine. The criteria considered while selecting the features of the components of the single cylinder engine are explained as follows:

a) Boundary definition: certain components like piston and slip rings can be combined together as they undergo wear and tear at the same rate. In case of flywheel, as it does not affect the working of the engine under normal circumstances, inspection data of its features has been excluded.

b) Form consideration: As piston, connecting rod and crankshaft are critical to the smooth working of the engine it is essential to inspect their features that interact with each other. The external features of fin can also be excluded from the inspection, as under form consideration, changes in its dimensions won't affect the working of other components of the engine.

c)Functional analysis: As the function of a single cylinder engine is to generate kinetic energy, it is essential to inspect the features of the components that are subjected to constant wear as a result of the motion performed by the engine to generate the energy. The list of components and their functions include piston (reciprocating motion), connecting rod (rotational motion), crankshaft (rotational motion), pin (oscillating motion), fin's inner bore (reciprocating motion).

Based on these considerations, feature selection is carried out and tabulated below.

Table 13. Feature selection to narrow down the number of features

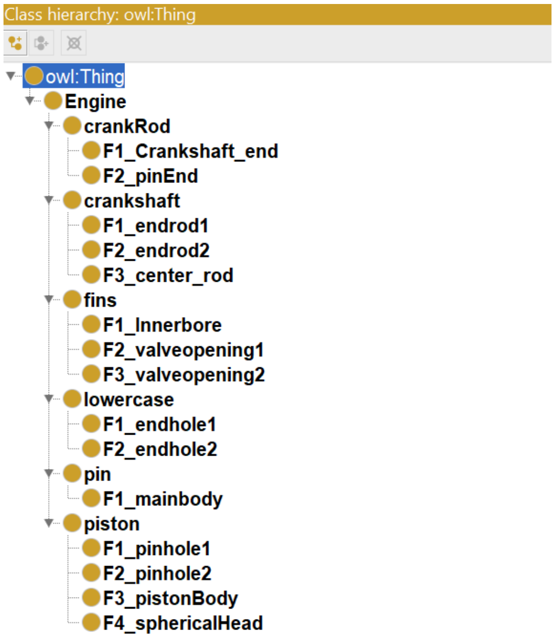
Part	Total number of surface features	Surface features selected	Rationale
Sliprings	4	0	Excluded from the boundary condition of piston

Uppercase	20	3	Excluded as the functional analysis determined not much significant change in form.
Pin	3	1	The flat ends on either side are not detrimental to functional analysis, thus excluded.
Crank rod	12	2	Excluded as the functional analysis determined not much significant change in form.
Crankshaft	17	3	Excluded as the functional analysis determined not much significant change in form.
Lowercase	11	2	Excluded as the functional analysis determined not much significant change in form.
Flywheel	5	0	Excluded from the boundary condition of crankshaft
Piston	4	4	The features satisfy boundary condition, and functional analysis criteria for feature selection.

There are 76 (excluding components such as fasteners) tolerance features in the single cylinder assembly. The number of tolerance features to be inspected were brought down to 15, through feature selection.

5. Defining classes and their hierarchy

As in the case previous metrontology, the OntoSTEP plugin facilitates import of STEP file data of the CAD model into Protégé. Along with OntoSTEP plugin, Cellfie plugin is used for importing keywords for creating the classes and subclasses of the metrontology. This ensures the classes are defined according to the name of components and surface features. In case of metrontology, as the information regarding component features is obtained after the information of component is known, the top-down approach of designing ontology is followed. The Figure 31 depicts the image of class hierarchy for single cylinder engine in Protégé.



Class hierarchy: owl:Thing

- owl:Thing
  - Engine
    - crankRod
      - F1_Crankshaft_end
      - F2_pinEnd
    - crankshaft
      - F1_endrod1
      - F2_endrod2
      - F3_center_rod
    - fins
      - F1_Innerbore
      - F2_valveopening1
      - F3_valveopening2
    - lowercase
      - F1_endhole1
      - F2_endhole2
    - pin
      - F1_mainbody
    - piston
      - F1_pinhole1
      - F2_pinhole2
      - F3_pistonBody
      - F4_sphericalHead

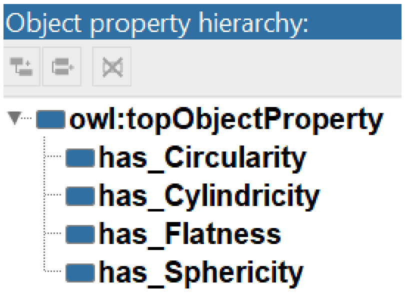
Figure 31. Class hierarchy for Single Cylinder Engine

6. Defining individuals & properties

Individuals and properties are selected from the keywords that are defined in the first step for creation of metrontology.

In case of metrontology, the keywords used for defining object properties as well as data properties remain the same. Certain properties will be excluded from the metrontology of the product, based on the components and features of each component. The single cylinder assembly has six components which together have about 76 surface features subjected to four different types of tolerances. Therefore, the object properties that will be used for developing the metrontology are *has_Flatness*, *has_Cylindricity*, *has_Circularity*, and *has_Sphericity*. In case of data property, we use all the data properties defined previously.

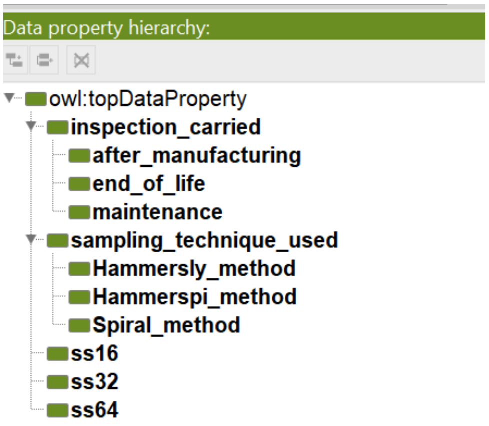
Figure 32 and Figure 33 show the object properties and data properties implemented in Protégé.



Object property hierarchy:

- owl:topObjectProperty
  - has_Circularity
  - has_Cylindricity
  - has_Flatness
  - has_Sphericity

Figure 32. Object properties of Single Cylinder Engine



Data property hierarchy:

- owl:topDataProperty
  - inspection_carried
    - after_manufacturing
    - end_of_life
    - maintenance
  - sampling_technique_used
    - Hammersly_method
    - Hammerspi_method
    - Spiral_method
  - ss16
  - ss32
  - ss64

Figure 33. Data properties of Single Cylinder Engine

7. Populating and Maintenance of EO

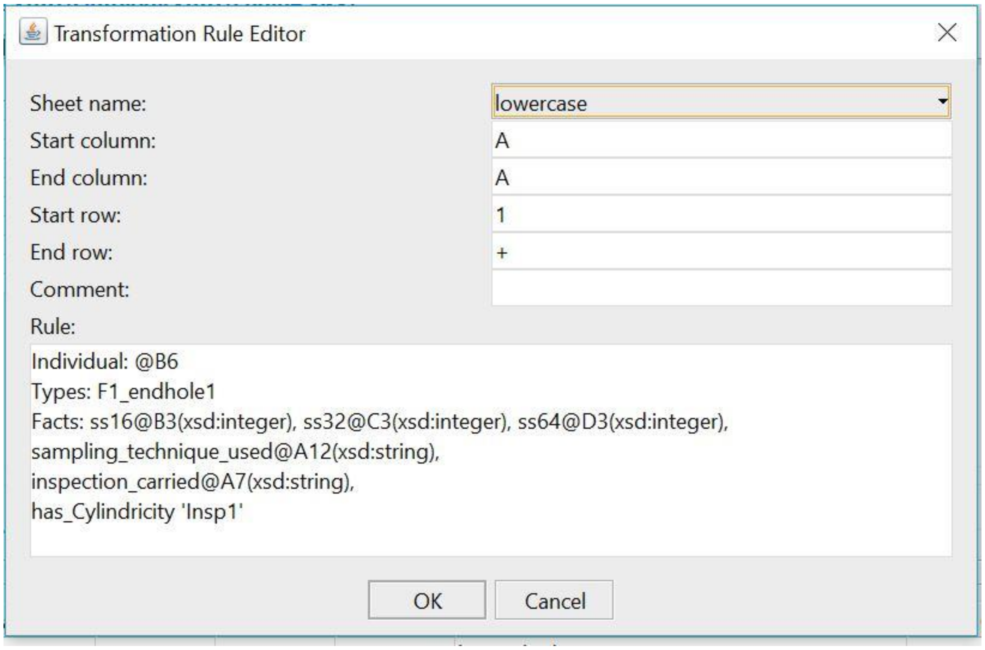
For populating the instances in the metrontology, the tolerance data for the features of various components was generated using Monte Carlo simulation. According to (Yan, Wu, and Yang 2015) simulated values for part tolerances can be derived used Monte Carlo simulation. Various criteria like manufacturing process used, sample size for inspection, wear and tear for the product, etc., are considered while determining tolerance zones for the simulation.

The knowledge base, stored in MS Excel, also has information regarding inspection techniques used. For this study, Aligned Systematic, Hammersley, Spiral and Hamspi methods were considered (Collins et al. 2007). It is to be noted that the tolerance values generated using Monte Carlo simulation are assumed to be within the least count of the Coordinate Measuring Machine. The tolerance values in this table are based on the research work of (Recker 2015), and (Lathan 1998). Also the tolerance chart used for reference was adopted from (DeGarmo et al. 1997).

As the ontology is being developed to capture the entire lifecycle of the piston-cylinder engine, Monte Carlo simulation is also used for generating tolerance values for two more phases of products lifecycle; the maintenance stage (mid-life phase) and end of life phase.

To take these different time-series instances into consideration, additional random numbers were added to previously calculated simulations with an assumption that as the parts of a product age, it will cause wear and tear of them and therefore loosen up their tolerances. Randomness ensures that it’s not true in case of all tolerances. In order to import these data as instances, Cellfie plugin is used. The advantage of Cellifie plugin lies in the fact that several instances can be created at one time in the ontology.

Maintenance of this ontology will include updating classes and sub-classes as the designer changes the product definition of the CAD model of single cylinder engine. Also, properties would need to be updated as and when metrology knowledge like tolerance types, and sampling techniques used would change



Sheet name: lowercase
Start column: A
End column: A
Start row: 1
End row: +
Comment:
Rule:

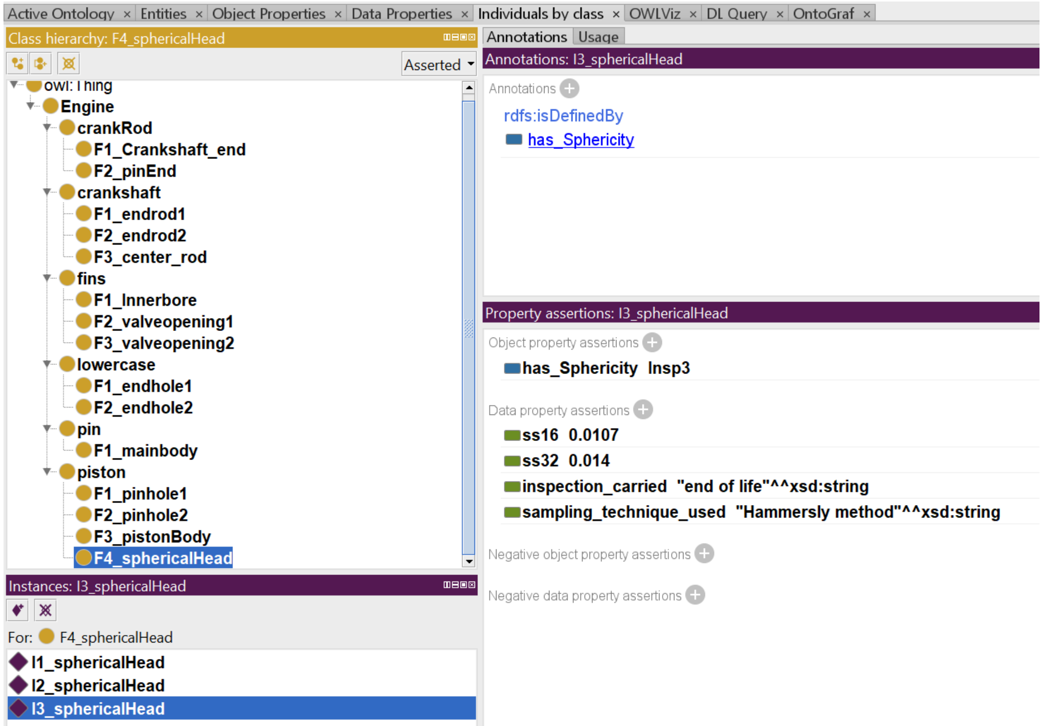
```
Individual: @B6
Types: F1_endhole1
Facts: ss16@B3(xsd:integer), ss32@C3(xsd:integer), ss64@D3(xsd:integer),
sampling_technique_used@A12(xsd:string),
inspection_carried@A7(xsd:string),
has_Cylindricity 'Insp1'
```

Figure 34. Cellfie plugin denoting the creation of instances for 'Lower case' feature named '*F1_endhole1*'

8. Interpreting the developed metrontology

In order to make sure that the developed EO is semantically correct, we check this aspect of the ontology using reasoners. A reasoner is a JAVA based plugin, which when implemented, scans through the axioms of the ontology and provides inference of a subsumption hierarchy for the classes described in the ontology. These reasoners use descriptive logic that help them in validating the semantic correctness of the classes and their associated instances. In this case FaCT++ reasoner was used for checking the logical and hierarchical correctness of the developed metrontology.

The Figure 35 shows the metrontology for Single Cylinder Engine in Protégé. The object property assertions indicate that the feature 'F4_shpericalHead' has sphericity at the inspection carried out during end of life instance of the time-series.



- owl:Thing
  - Engine
    - crankRod
      - F1_Crankshaft_end
      - F2_pinEnd
    - crankshaft
      - F1_endrod1
      - F2_endrod2
      - F3_center_rod
    - fins
      - F1_Innerbore
      - F2_valveopening1
      - F3_valveopening2
    - lowercase
      - F1_endhole1
      - F2_endhole2
    - pin
      - F1_mainbody
    - piston
      - F1_pinhole1
      - F2_pinhole2
      - F3_pistonBody
      - F4_sphericalHead

rdfs:isDefinedBy
has_Sphericity

has_Sphericity Insp3

ss16 0.0107
ss32 0.014
inspection_carried "end of life"^^xsd:string
sampling_technique_used "Hammersly method"^^xsd:string

I1_sphericalHead
I2_sphericalHead
I3_sphericalHead

Figure 35. Metrontology template view for instance *I3_sphericalHead*

From Figure 36 for the Ontograf image of the metrontology, we can get an idea of its structure and how the classes and individuals are related to each other. The essence of the proposed EO is to aid the designers with metrology information to perceive the expected tolerances of various parts in the post design phase. When it comes to designing the next version of the engine, the knowledge captured by the ontology can give the designers an idea of how the part dimensions have changed over the product’s lifecycle, the sampling size and techniques used, and whether the part tolerances conform to the tolerance type (i.e. object property) associated with it. This will help the designers in taking decisions related to selection of materials, tolerance allocation, and manufacturing process to be used for next product being designed.

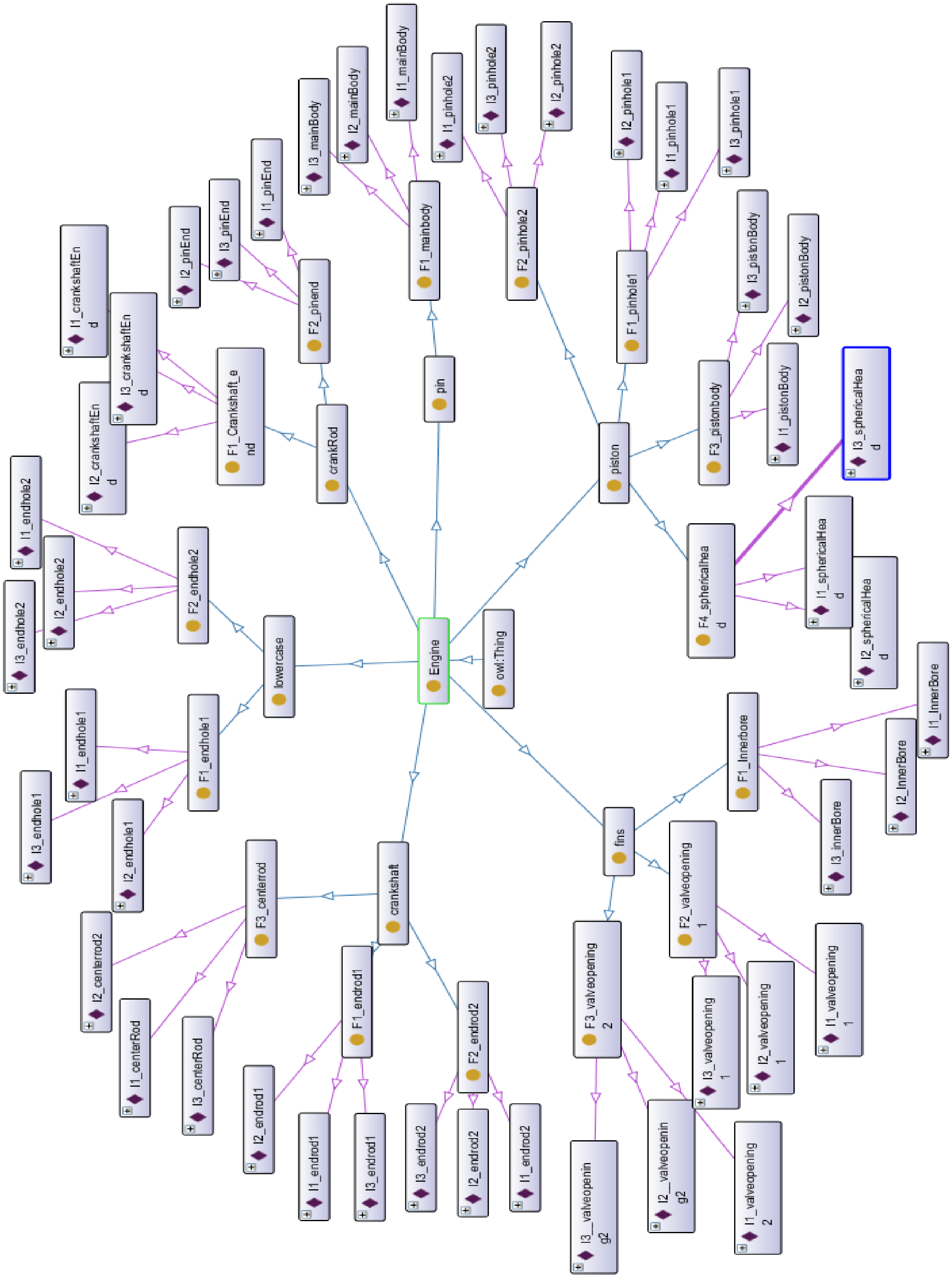


Figure 36. Visualization of developed Metrontology in Ontograf

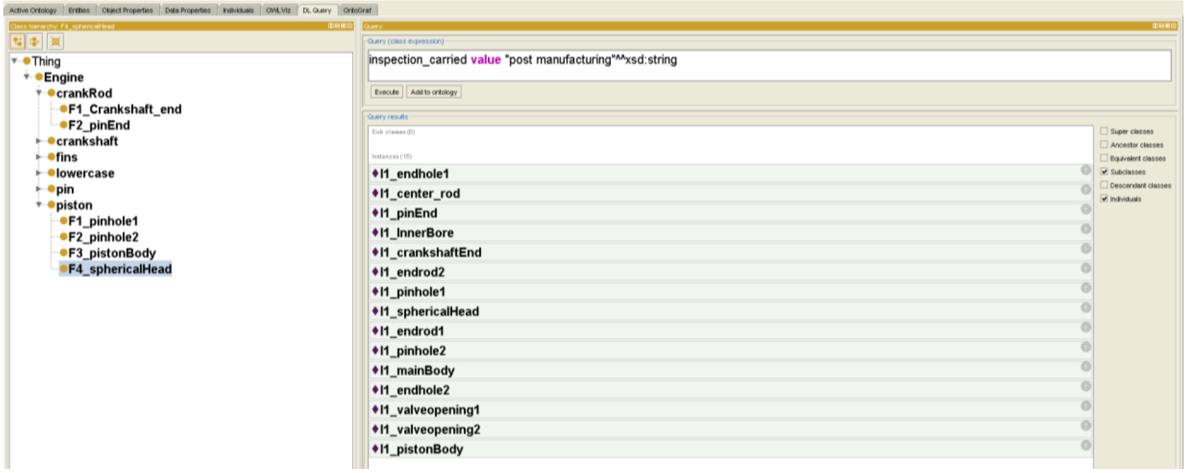
Chapter 6: Conclusions and Recommendations for future study

6.1 Conclusions and Observations

As a first step for development of metrology markup language a metrontology was developed. This metrontology can now be either maintained as an independent ontology that will serve as a knowledge base for tracking metrology features of an assembly (example used is an engine) over its entire lifecycle, or it can also be integrated with another ontology or PLM software enclave. This is possible as the ontology has been developed in extensible markup language (XML) format. XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

The approach suggested in this research work is unique as it also includes capturing time series data pertaining to the product metrology at three instances over the entire life of the product. This could aid in making design modifications and suggestions, even for products designed for the very first time based purely on past experiences with similar features, materials, and geometries. Also, feature selection methodology incorporated in this work can ensure that the designers nor metrologists would have to spend excess time on analyzing and recording large amount of metrology data which may not be relevant to product development.

One of the ways in which the metrology information can be searched by the product designers within the EO is via queries. Queries can only be performed on the metrontology if the reasoner does not return any error for logical and hierarchical correctness of the metrontology. FaCT++ reasoner used for verifying the logical and hierarchical correctness of the Single Cylinder Engine metrontology does not return any logical or hierarchical error. Ontology queries can be initiated in the DL query tab in the software Protégé. Queries can be performed on finding metrology information regarding which surfaces features went under common sampling method, and the tolerances of post-manufactured parts. Queries can be performed at different levels of hierarchy within the metrontology. Figure 37 shows a query example for find the instances that were created during the post manufacturing inspection of the single cylinder engine components.



inspection_carried value "post manufacturing"^^xsd:string

I1_endhole1
I1_center_rod
I1_pinEnd
I1_InnerBore
I1_crankshaftEnd
I1_endrod2
I1_pinhole1
I1_sphericalHead
I1_endrod1
I1_pinhole2
I1_mainBody
I1_endhole2
I1_valveopening1
I1_valveopening2
I1_pistonBody

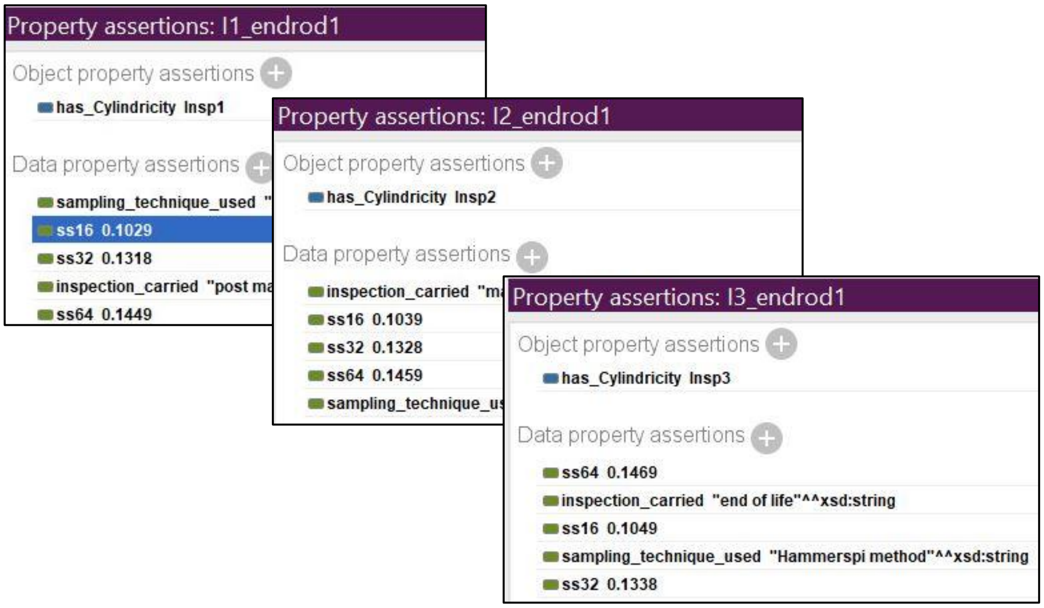
Figure 37. Example of Query for Single Cylinder Engine metrontology

From the ontology, some common observations about part tolerances can be made:

a) As the sample size of the inspection increases, the tolerance value increases. This indicated that large number of sample points are able to define the tolerance more robustly.

b) Also, with every inspection, the tolerance value increases. This trend reflects common wear and tear that occur as the engine is used over a long period of time.

c) If the tolerance value indicated by object properties ss16, ss32, or ss64 of the feature instance no longer complies with tolerance value, its object property is dropped for the feature for that respective tolerance type. Also, the ss16, ss32 and/or ss64, are categorized as negative object properties to indicate their noncompliance with the tolerance values determined by the inspector during the time of inspection.


Property assertions: I1_endrod1
Object property assertions
has_Cylindricity Insp1
Data property assertions
sampling_technique_used "
ss16 0.1029
ss32 0.1318
inspection_carried "post ma
ss64 0.1449

Property assertions: I2_endrod1
Object property assertions
has_Cylindricity Insp2
Data property assertions
inspection_carried "ma
ss16 0.1039
ss32 0.1328
ss64 0.1459
sampling_technique_us

Property assertions: I3_endrod1
Object property assertions
has_Cylindricity Insp3
Data property assertions
ss64 0.1469
inspection_carried "end of life"^^xsd:string
ss16 0.1049
sampling_technique_used "Hammerspi method"^^xsd:string
ss32 0.1338

Figure 38. Depiction of gradual increase in tolerance value of surface feature over its life cycle

6.2 Recommendations for future study

This research work focused on capturing metrology information of the entire product over its lifecycle. From perspective of future study, development of an ontology that can capture both detailed inspection planning as well as capturing metrology information over its entire life-cycle is recommended. Also, of interest for future study can be the actual demonstration of integrating this metrontology with other ontologies that have non-geometric information such as material, manufacturing costs, etc. regarding the product.

Currently, the feature selection process developed for narrowing down the number of features to be inspected is a manual process. It is possible to explore ways in which this step can be automated, using concepts of artificial intelligence and machine learning (ML), so that it will further save decision time for selecting surface features. Also, the three criteria discussed can be translated into categorical variables for better simplification and automation of the feature selection process. Employing machine learning would also help in grouping together products with similar set of metrology concepts. This will help in systematic organizing the metrology knowledge of several products into the proposed Metrology Information Enclave.

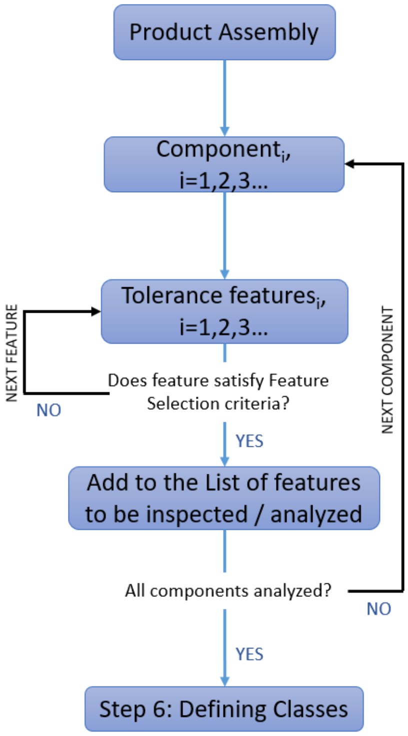


Figure 39. Logical steps of ML algorithm for Feature Selection

Further research is also needed in use of these metrontology for development of metrology markup language (MML) that will encapsulate extensive information regarding product dimension and tolerancing as well as product inspection. This would result in further enrichment of standard Model based Definition and PLM knowledge of the product.

References

ASME.Y14. 2009. *Dimensioning and Tolerancing* (American Society of Mechanical Engineers: U.S.A.).

Barbau, Raphael, Sylvere Krima, Sudarsan Rachuri, Anantha Narayanan, Xenia Fiorentini, Sebti Foufou, and Ram D Sriram. 2012. 'OntoSTEP: Enriching product model data using ontologies', *Computer-Aided Design*, 44: 575-90.

Brunnermeier, Smita B, and Sheila A Martin. 1999. *Interoperability cost analysis of the US automotive supply chain* (DIANE Publishing).

Collins, CE, EB Fay, JA Aguirre-Cruz, and S Raman. 2007. 'Alternate methods for sampling in coordinate metrology', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 221: 1041-52.

DeGarmo, Ernest Paul, J Temple Black, Ronald A Kohser, and Barney E Klamecki. 1997. *Materials and process in manufacturing* (Prentice Hall).

Drouillard, Gerald. 2001. 'Bill of Materials', Accessed Feburary 7th. http://www.inventoryinterface.com/bill_of_materials.html.

Engineers, The American Society of Mechanical. 2009. "Dimensioning and Tolerancing." In. New York, USA.

Fenves, Steven J, Sebti Foufou, Conrad E Bock, N Bouillon, and Ram D Sriram. 2005. "Cpm 2: A revised core product model for representing design information." In.

Fenves, Steven J, Sebti Foufou, Conrad Bock, and Ram D Sriram. 2008. 'CPM2: a core model for product data', *Journal of computing and information science in engineering*, 8: 014501.

Goldsmith, Mike. 2010. *A Beginner's Guide to Measurement* (National Physical Laboratory).

Gruber, Thomas R. 1995. 'Toward principles for the design of ontologies used for knowledge sharing?', *International journal of human-computer studies*, 43: 907-28.

Grüninger, Michael, and Mark S Fox. 1995. 'Methodology for the design and evaluation of ontologies'.

Horridge, Matthew, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, and Chris Wroe. 2009. 'A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 2', *The University of Manchester*, 107.

Huang, Zhicheng, Lihong Qiao, Nabil Anwer, and Yihua Mo. 2015. "Ontology model for assembly process planning knowledge." In *Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014*, 419-23. Springer.

Institute of Systems Engineering, HES-SO VALAIS-WALLIS. 2014. 'S-TEN', Accessed 3/26/2018. https://www.hevs.ch/en/rad-instituts/institut-systemes-industriels/projects/s-ten-3448.

Johardi. 2018. 'Cellfie', GitHub, Accessed 04/17. https://github.com/protegeproject/cellfie-plugin.

Kissick, Christopher. 2014. 'Feature selection process for First Article Inspection', The University of Oklahoma, Norman.

Krima, Sylvere, Raphael Barbau, Xenia Fiorentini, Rachuri Sudarsan, and Ram D Sriram. 2009. 'Ontostep: OWL-DL ontology for step', *National Institute of Standards and Technology*, *NISTIR*, 7561.

Kurkin, O, and M Januška. 2010. 'Product life cycle in digital factory', *Knowledge management and innovation: a business competitive edge perspective. Cairo: International Business Information Management Association (IBIMA)*: 1881-86.

Lad, Subham. 2016. 'Single Cylinder Engine Model', GrabCAD.com, Accessed 01/27/2018. https://grabcad.com/library/single-cylinder-engine-model-2.

Lathan, Charles. 1998. 'Construction of an Internal Combustion Engine.' in, *Construction Mechanic Basic* (Naval Education and Training Professional Development and Technology Center).

Li, Zhanjun, Maria C Yang, and Karthik Ramani. 2009. 'A methodology for engineering ontology acquisition and validation', *AI EDAM*, 23: 37-51.

Lu, Wenlong, Yuchu Qin, Xiaojun Liu, Meifa Huang, Liping Zhou, and Xiangqian Jiang. 2015. 'Enriching the semantics of variational geometric constraint data with ontology', *Computer-Aided Design*, 63: 72-85.

Narayan, K. Lalit 2008. *Computer Aided Design and Manufacturing* (Prentice Hall of India: New Delhi).

Patil, Lalit, Debasish Dutta, and Ram Sriram. 2005. 'Ontology-based exchange of product data semantics', *IEEE Transactions on automation science and engineering*, 2: 213-25.

ProtegeWiki. 2018. 'Protege_Ontology_Library', MediaWiki, Accessed 04/17. https://protegewiki.stanford.edu/wiki/Protege_Ontology_Library#OWL_ontologies.

Qin, Yuchu, Wenlong Lu, Qunfen Qi, Tukun Li, Meifa Huang, Paul J Scott, and Xiangqian Jiang. 2017. 'Explicitly representing the semantics of composite positional tolerance for patterns of holes', *The International Journal of Advanced Manufacturing Technology*, 90: 2121-37.

Qin, Yuchu, Yanru Zhong, Meifa Huang, and Fuyun Liu. 2014. 'An assembly tolerance representation model based on spatial relations for generating assembly tolerance types', *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 228: 1005-20.

Recker, Derek. 2015. 'Design, Development, and Analysis of a Single-Cylinder, Four-Stroke Propane Engine in an Educational Environment', The Ohio State University.

Ryan Mayes, Mr Henry Cuschieri, Mr Brian Stanton. 1994. 'Industrial automation systems and integration -- Product data representation and exchange -- Part 1: Overview and fundamental principles', *Standards Catlogue*, ISO 10303-1: 11s.

Sarigecili, Mehmet I, Utpal Roy, and Sudarsan Rachuri. 2014. 'Interpreting the semantics of GD&T specifications of a product for tolerance analysis', *Computer-Aided Design*, 47: 72-84.

Standardization, International Organization for. 1998. "Industrial automation systems and integration -- Product data representation and exchange -- Part 22: Implementation methods: Standard data access interface." In *ISO TC184 / SC4 resolution*. Bad Aibling, Germany: ISO.

———. 2005. "Industrial automation systems and integration -- Product data representation and exchange." In *Part 239: Application protocol: Product life cycle support.*

International Organization for Standardization. 1994. "Industrial automation systems and integration -- Product data representation and exchange." In *Part 203: Application Protocol: Configuration controlled 3D design of mechanical parts and assemblies.*

———. 2003. "Industrial automation systems and integration -- Product data representation and exchange." In *Part 214: Application protocol: Core data for automotive mechanical design processes*.

———. 2004-11. 'Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual', Accessed 03/25. https://www.iso.org/standard/38047.html.

Stojadinović, Slavenko M, and Vidosav D Majstorović. 2014. 'Developing engineering ontology for domain coordinate metrology', *FME Transactions*, 42: 249-55.

Stojadinovic, SM, and VD Majstorovic. 2012. 'Towards the development of feature-based ontology for Inspection Planing System on CMM', *Journal of Machine Engineering*, 12.

Subrahmanian, Eswaran, Sudarsan Rachuri, Steven J Fenves, Sebti Foufou, and Ram D Sriram. 2005. 'Product lifecycle management support: a challenge in supporting product design and manufacturing in a networked economy', *International Journal of Product Lifecycle Management*, 1: 4-25.

Thilmany, Jean. 2015. 'Digital Tolerance'. https://www.asme.org/engineering-topics/articles/design/digital-tolerance.

Uschold, Mike, and Michael Gruninger. 1996. 'Ontologies: Principles, methods and applications', *The knowledge engineering review*, 11: 93-136.

Wikipedia. 2018. 'Model Based Definition(MBD)', Accessed 3/25/2018. https://en.wikipedia.org/wiki/Model-based_definition.

Wong, Kenneth. 2009. "What PLM Can Learn from Social Media." In. http://www.digitaleng.news/virtual_desktop/2009/07/what-plm-can-learn-from-social-media/: Digitl Engineering.

Yan, Huiwen, Xiangji Wu, and Jiangxin Yang. 2015. 'Application of Monte Carlo method in tolerance analysis', *Procedia CIRP*, 27: 281-85.

Zhong, Yanru, Yuchu Qin, Meifa Huang, Wenlong Lu, Wenxiang Gao, and Yulu Du. 2013. 'Automatically generating assembly tolerance types with an ontology-based approach', *Computer-Aided Design*, 45: 1253-75.

Glossary

Ontology- it is a highly structured system of concepts covering the processes, objects, and attributes of a domain as well as all their pertinent complex relations.

Engineering ontology(EO)- An ontology that has been designed for representing engineering artifacts, processes, and assemblies. In the context of this thesis, the terms EO, ontology, and metrontology have been interchangeably used to describe ontology as well as metrontology.

Dimensional metrology- It is the science of calibrating and using physical measurement equipment to quantify the physical size of the object. Referred to as metrology in this presentation.

Metrology Information Enclave- Proposed smart repository for metrology data that will aid designers, manufacturers and inspectors and maintain/communicate neutral exchange of data.

STEP- Standard for the Exchange of Product model data.

Product tolerance: It is the limit of random (unintentional) deviation of a product dimension from its nominal value.

OntoSTEP plugin- Developed by S. Rachuri et al, this plugin aides in importing information from CAD files in STEP Part 21 format into Ontology Web Language (OWL) file.

Cellfie plugin- A Protégé Desktop plugin for mapping spreadsheets to OWL ontologies. For this thesis, it used for creating subclasses of the EO as well as populating the instances of the EO.