DATA MODELS TO SUPPORT METROLOGY

by

Saeed Heysiattalab

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Mechanical Engineering

Charlotte

2017

Approved by:

____
Dr. Edward P. Morse

____
Dr. Robert G. Wilhelm

____
Dr. Jimmie A Miller

____
Dr. M. Taghi Mostafavi

____
Dr. Jing Xiao

ABSTRACT

SAEED HEYSIATTALAB. Data models to support metrology (Under the supervision of Dr. EDWARD P. MORSE)

The Quality Information Framework (QIF) is a project initiated in 2010 by the Dimensional Metrology Standards Consortium (DMSC is an American Standards Developing Organization) to address metrology interoperability. Specifically, the QIF supports the exchange of metrology-relevant information throughout the product lifecycle from the design stage through manufacturing, inspection, maintenance, and recycling / end-of-life processing. The QIF standard is implemented through a set of XML schemas called "Application Schemas" along with common core "XML schema libraries". Of these schemas, QIF Measurement Resources and QIF Rules are the two application schemas on which this research is focused. QIF Measurement Resources is an application schema developed to provide standard representations of physical measuring tools and components, and can be used to support measurement planning, statistical studies, traceability, etc. The Resources schema was supported by the creation of a new hierarchy of metrology resources in support of the product lifecycle. The QIF Rules schema is under development to provide the language with which manufacturers can define how dimensional measurement equipment is selected for various tasks, and how this equipment is used during the measurement task. The work on QIF Resources has been published in the current ANSI QIF standard, and the work on QIF Rules is under consideration for the next revision of this standard. In addition to the research and development of these schemas, work on upstream CAD models has included the development of a translator to and from the QIF modeling schema. Interoperability and potential information loss in this process are discussed.

ACKNOWLEDGEMENTS

I would like to express my deepest acknowledgment toward my advisor Dr. EWARD P. MORSE for his support throughout my PhD. I also would like to appreciate Center for Precision Metrology (CPM) affiliates and members for the funding and guidance conducting my research. I should also extend my gratitude for QIF Measurement Resources and QIF Rules working groups members for their support.

I have an amazing family and the words cannot pay the tribute for the sacrifices they have made for me throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

STEP	standard for the exchange of product data model
DME	Dimensional Metrology Equipment
AIAG	Automotive Industry Association Group
MSA	Measurement System Analysis
QIF	Quality Information Framework
ANSI	American National Standard Institute
LVDT	Linear variable differential transducer
DVRT	Differential variable reluctance transducer
ISO	International organization of standardization
GD&T	Geometric dimensioning and tolerancing
CMM	Coordinate measuring machines
PMI	Product and manufacturing information
ASME	American society of mechanical engineers
CNC	Computer Numerical Control

Chapter 1: Introduction

Preface: Gun manufacturing was a turning point in the history of metrology and manufacturing. The original parts of a gun could not always be replaced with spare ones as every gun part was slightly different (Evans 1989). Due to the ever-increasing number of manufacturers and growth of the industrial world, new challenges were raised. Companies would like to have the choice to source any product component from any vendor, making interchangeability and interoperability issues extremely crucial. Interoperability capabilities along with proprietary technologies can assist in products marketing and the metrology community is not an exception to this matter. Interoperability is defined as "successful performance of required tasks by two or more agents requiring the exchange of information" (IMTI 2006) and interchangeable parts are the parts identical for practical purposes (Wikipedia 2017). Lack of Interoperability and information exchange can cause missing data, impose data translation expenses (as data are probably in different formats), and spending resources on not value-added activities. Data translation expenses just in North America Automobile industry has been estimated to be around $600 million annually, and the problem is not limited to the aforementioned issues (Zhao et al. 2012).

Three solutions have been suggested to resolve the interoperability issues. The first choice would be a single vendor product life-cycle support that does not require any digital data translation if the vendor uses the same format all over product life cycle (or the vendor has internal data translator that handles transferring data throughout product life cycle). Depending on a single vendor to support product life cycle would not be a smart choice as vendors go bankrupt, stop supporting their products, and so on. The second solution would be the data translation approach that translates each digital data format to another digital data format. This approach wastes a lot of time and resources (non- value adding activities), especially with increasing the number of data formats. If there exist N proprietary data formats, the number of data translators would be N× (N-1). The third solution would be to define neutral standard data formats -covering product life cycle- that every vendor can translate its own format to that neutral format. Under these circumstances, the number of data translators would be 2×N translators, which is, much less than N× (N-1) (with increasing the number of data formats, N) (Nasr and Kamrani 2007).

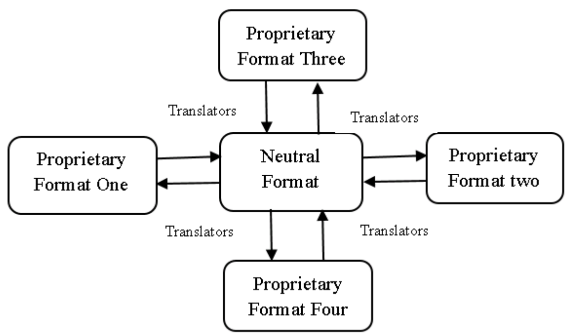


Figure 1 Data Translation System with a Neutral Format

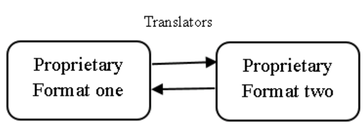


Figure 2 Data Translation System without a Neutral Format

For the sake of discussion, standards will be classified into three types: open standards, industry standards and de facto standards. Open standards are the standards that are used pervasively to solve the interoperability issues. These type of standards (like ISO 10303 or

STEP) are usually developed by international organization like ISO or a group of industrial companies. The industry standards are the standards that are mostly influenced by one or a group of companies and there is not a democratic procedure for the governing committees (like Java technology). The third group of standards (de facto standards) can also be a part of interoperability issues. They are widely used and accepted by the public and any proprietary or non-proprietary format can be a de facto standard (like DMIS or Windows 7) (Peak et al. 2004). This research is mostly about (open) standards as they are developed by joint collaborations between different organizations, companies, universities, etc.

To resolve interoperability issues, we first need to understand product life cycle concepts, activities, and the information required for these activities. For instance in metrology realm, GD&T information may be exchanged between product design and inspection process performed by a CMM. In this case, PMI (GD&T plus other information) is exchanged between two activities (design and inspection). If the GD&T is not defined in each of these activities or the same technology is not used to define the information (different file extensions, etc.), the information is lost or cannot be transferred (both cause problems).

Product life cycle: "Product life cycle is the cycle through which every product goes through from introduction to withdrawal or eventual demise" (economic times). In fact, any activity (including design, manufacturing, and inspections) that product goes through is considered as part of product life cycle. Globalization phenomenon and marketing requirements have made companies to perform product related activities in different parts of the world. A product may be designed in the USA, manufactured in Europe, and assembled in China. The design group needs to send the product information to manufacturing plants in Europe and this information should be read by machine tools (controllers) in Europe and assembly facilities in China. Every product has a different manufacturing method and equipment. For instance, a vehicle part may be manufactured by machining, casting, forming or any other method. This requires enabling different manufacturing devices (machine tools controllers, etc.) to read product data from the design step. In addition, the CNC technology is used in many manufacturing processes (like laser machining, welding, etc.) and the interoperability requires exchanging information between different manufacturing devices (controllers or planners).

Product life cycle activities can be classified as concurrent or sequential engineering methods. In the concurrent engineering methods, an activity can be started before ending the previous activity (they overlap), but in sequential engineering, an activity cannot be initiated before ending the previous activity. For instance in sequential engineering method, measurement planning of a product cannot be started before the product design, but in concurrent engineering, these activities overlap (measurement planning can start before ending the product design). It should be mentioned that sequential and concurrent

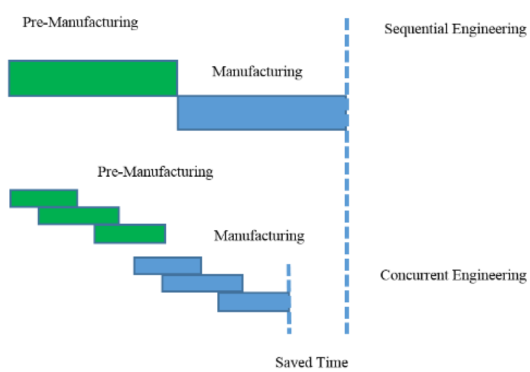


Figure 3 Sequential Engineering (left) vs. Concurrent Engineering (right) (Badin et al. 2012)

engineering product life cycles could have different activities and they do not necessary have to share exactly the same activities. Product life cycle is the most comprehensive cycle that a product can go through and any other group of activities (cycle or framework), like metrology system, is a part of product life cycle (Badin et al. 2012).

Metrology Systems: Metrology systems or quality systems are a part of the product life cycle and it includes all the activities required to ensure that product has the intended quality (all quality related activities). Depending on the type of metrology (surface metrology or dimensional metrology), different activities are included in the metrology system. For dimensional metrology, which is more focused here, a typical metrology system includes product design, measurement planning, measurement execution, results analysis and reporting activities. Results analysis and reports are especially designed for a closed-loop manufacturing enterprise for process improvement. Measurement process planning includes high-level and low level process planning. There is not a distinct separation line between high-level and low-level measurement process planning, but the high-level measurement process planning is a process planning independent of measurement device and the low-level one dependent on the measurement device (Zhao et al. 2011).

There are different data formats that have tried to store product life cycle information or metrology system information (or parts of them). In the next chapter, we will go through these data formats and will explain more about these data formats.

Chapter 2: Digital Data Formats in Dimensional Metrology

Introduction to Digital Data Format in Dimensional Metrology: Computers are one of the greatest inventions of human beings. They have revolutionized the way in which we process and archive data. The product life cycle, including the way we design, manufacture, and inspect the product, have been digitized. Many data formats have been developed for saving product information and in this chapter; we will review some of these data formats. Some data formats are commercial and of course the way the data is stored is proprietary (not open to public), but some are standard data formats developed for data exchange purposes.

There have been numerous interoperability solutions for product life cycle or metrology systems (metrology system is a part of product life cycle). Unfortunately, product life cycle related data formats have not put so much effort in metrology related parts and this requires especial attention. For the sake of discussion, this chapter includes two parts: product life cycle related data formats and metrology systems related data formats.

Product Life Cycle Related Data Formats

This section explains the prevalent data formats used in product life cycle for data exchange. As explained in the previous chapter, product life cycle includes any activity from design to recycling that product goes through it. The interoperability movement started in early 1980 by releasing the first data exchange format, which is IGES.

Initial Graphics Exchange Specification (IGES): IGES is an 80-column ASCII data exchange format used for wireframe, surface and solid models. It was developed by

National Aeronautical and Space Administration (NASA) and national bureau of standards and it was approved as a US national standard by American National Standard Institute (ANSI). IGES (the latest version is 5.3) does not cover any non-geometry information like PMI or measurement resources information. Unlike the modern feature-based CAx systems, IGES is an entity-based system, meaning that IGES can have geometrical entities like circles or lines and annotation entities. An IGES file contains five parts: start section, global section, directory entry section, parameter data section and terminate section (S, G, D, P, and T stands for the aforementioned sections respectively). The start section is an introduction to the IGES file being used and includes basic information. The global section describes the information needed to interpret the IGES file like delimiter characters and the number of significant digits. All geometric entities are described in the directory entry section and the parameter data section. The terminate section is a single line that displays the end of an IGES file with the number of lines for each of the aforementioned sections (US Product Data Association 1996). A sample IGES file is demonstrated as following:

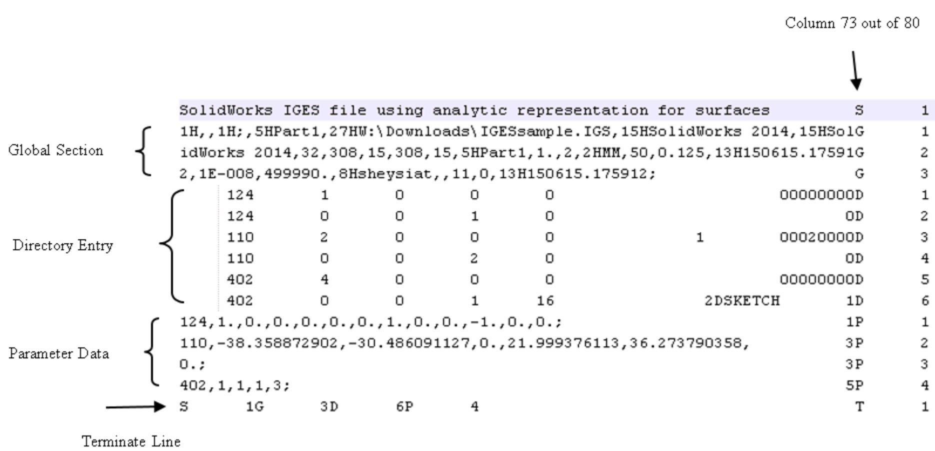


Figure 4 Sample IGES File

IGES had some shortcomings that drove the industrial community to resolve these issues and find another solution. Firstly, IGES did not support the product life cycle and it was mostly product geometry and topology. IGES did not have a schema or any information modeling languages (explained in the next chapter), unlike STEP or QIF (will be explained in the upcoming sections), which lead to ambiguity and having different implementation flavors of IGES. In addition, IGES files are not human readable and thus not easy to revise. Overall, many IGES shortcomings drove international community toward STEP as a successor of IGES (following section).

ISO 10303: STEP or STandard Exchange of Product data model was introduced as an ISO standard in 1993 (development started in 1984 as a successor of IGES). STEP has an information modeling language called EXPRESS, which has object oriented flavor (everything is treated as an object) and it covers all product life cycle, from design to manufacturing, quality control and recycling. STEP integrates computer aided-design (CAD), computer aided-manufacturing (CAM), computer aided-process planning (CAPP), computer aided inspection (CAI), and other processes data format. It allows bilateral seamless (closed loop) exchange of information throughout product life cycle unlike IGES. STEP has also gained significant interest in numerical control (NC) machines used for machine tools and CMMs (ISO 10303-238: 2007).

Because of aforementioned problems with IGES in the last section, software applications had to migrate from IGES to STEP and develop tools to translate data stored in IGES format (with *.igs extension) to STEP format (with *.stp or *.step extension). For this purposes, an EXPRESS schema could be developed to capture the IGES entities and then converting them to STEP format. A visualizer can be used to investigate the proper data translation. An object-oriented environment could be used to read the data from IGES and then store it and write into step format (STEP physical exchange file or Part 21 Format) (Bhandarkar et al. 2000).

STEP is closed-loop data format and a prototype has been developed at Aachen Technical University in Germany for CAPP/CAM/CNC loop. The STEP-NC code is generated by using the WZL-FSP software and then the product is manufactured by two STEP-NC enabled milling machines and after machining, the part is transferred to a ZEISS CMM for inspection. As there were no STEP-NC enabled CMM (controller), the WZL FSP software converts the STEP-NC code (STEP-NC code includes both manufacturing and inspection) into two separate files: STEP AP 203 (which just includes the geometry and topology) and Q-DAS ASCII data transfer format (which contains the characteristics and PMI). The CMM performs the measurement plan and reports the results to WEPROM interface (WEPROM connects the geometry to PMI). After inspection, the measurement results are stored in a WEPROM text file and then integrated with STEP-NC format by WZL SFP STEP-NC software. Integrating CAD, CAM and CAI data format can be a great achievement as CAD technologies use IGES or any proprietary format (just geometry and topology) and CAM technologies use G & M code technology (ISO 6983). In this case, there are two separate files, one for geometry and topology, and the other one for machining (Brecher et al. 2006).

STEP is divided into different engineering domains called Application Protocols (AP) that takes care of that engineering domain. The first released AP was called AP203 or Configuration Controlled 3D Designs of Mechanical Parts and Assemblies. This AP mostly covered product design and geometry and the second version of this application control, AP203, covered product geometry and topology, plus PMI information like GD&T to some extent. It is worth mentioning that AP214 also defines geometry and design similar to AP 203, but it is more automotive industry oriented, and it is called: Core Data for Automotive Mechanical Design Process. STEP includes many other APs and some of them are listed in table 1.

Table 2 Some of STEP APs and their domain

AP224	Mechanical product definition for process plans using machining features
AP219	Dimensional inspection information exchange
AP240	Process plans for machined products
AP223	Exchange of design and manufacturing product information for cast parts
AP238	Application interpreted model for computer numeric controllers
AP210	Electronic assembly, interconnect and packaging design. The most complex and sophisticated STEP AP.
AP215	Ship arrangement
AP233	Systems engineering data representation

A typical STEP AP has three parts: Application Activity Model (AAM), Application Reference Model (ARM), and Application Interpreted Model (AIM). AAM specifies requirements and activities in an AP domain. ARM defines the Application

Objects (AO) necessary for that domain and AIM is the ARM in the form of EXPRESS modeling language (Information modeling languages will be explained in the next chapter).

STEP has saved up to $150M per year for aerospace; shipbuilding and automotive industry although only the geometry part of STEP (AP203) has gained more interest (Zhao et al. 2012). The conventional STEP implementation, Part 21, is pretty much an old technology and ISO is working toward other new technologies such as XML and UML. The part 21 file is complicated, not human readable, and hard to extend (Xu. and Nee, 2009).

STEP has addressed interoperability and seamless exchange of information in different engineering domains, but quality control and inspection has not gained much attention so far. STEP has many shortcomings in the metrology system world. AP219 (Dimensional Inspection Information Exchange) was early developed for dimensional planning, executing, and analyzing dimensional inspection, but many stubs were left empty for future development. ”The latest ISO standard version of AP219 only defines measurement results information” (Zhao et al. 2012).

Among different STEP APs, AP242 requires special attention. STEP AP242 is the first and only STEP AP that has addressed Model-Based Definition (MBD) concepts like 3D annotation and GD&T (still AP242 does not cover measurement planning, execution, results analysis and reports). Following section explains more about STEP AP242.

STEP AP 242: STEP AP 242 (Managed Model Based 3D Engineering) is a relatively new project launched by ISO TC 184/SC 4 in 2009 that merges STEP AP 203 Edition 2 and STEP AP 214. It harmonizes different APs including AP 239 (Product life cycle support), AP209 (multidisciplinary analysis and design), and other APs. AP 242 first version has been published recently (end of 2014), and it has only been implemented by a few vendors (CAPVIDIA products, LKSoftWare, etc.). ASME Y14.5-2009, ISO 5459:2011, and ISO 1101:2011 are all covered by STEP AP 242 (ISO 10303-242: 2014). In addition, STEP AP242 needs to be harmonized with other standards because they may get the product information from STEP AP242 (the standard needs to get more mature). The first version of AP242 does not use machining symbols of PMI and it should be harmonized with STEP-NC to use its machining features for GD&T applications. Chapter 4 has a comprehensive overview of geometry, topology, and PMI modeling in STEP.

STEP AP 242 covers the following engineering domains:

- “Tessellated Geometry and topological information”
- “A product design with a 3D model with all PMI in order to avoid technical drawings”
- “3D composite design”
- “Metrology Information”
- “3D kinematics assembly (based on AP214)”
- “3D shape quality”
- “Providing specifications for exchange and long term archiving”
- “3D assembly”
- “Interoperability with machining features” (www.AP242.org)

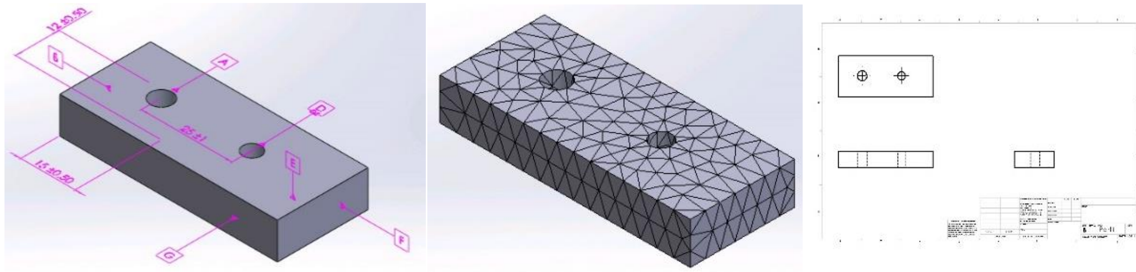


Figure 5 AP 242 Supports 3D PMI (left), Tessellation (Middle), and 2D Draughting (Right)

STEP is a very huge project that industry and governments have spent over $200 million in more than 33 years (PDES Inc. 2017). Besides AP219, many other STEP APs are not yet ready for implementation (many stubs were left empty). Another issue with STEP is the old technology (EXPRESS) used as information modeling language in STEP, even though recently started developing other information modeling languages. One of the most important issues with STEP was lack of proper semantic GD&T support, which is solved recently by AP242 (lack of proper GD&T coverage in STEP from 1993 to 2014). Overall, aforementioned problems drove many users toward other technologies like JT or PLM XML to solve their specific interoperability issues.

Data eXchange Format (DXF): DXF is a format developed by Autodesk Company to facilitate exchanging information (the first version was released in 1982). It does not have any PMI information and it is limited to wireframe, surface and solids. It is based on AutoCAD software library and it became a de facto standard (many companies used DXF as their data exchange format) to exchange wireframe information (AUTODESK 2012). Following is an example of DXF file (not human readable).

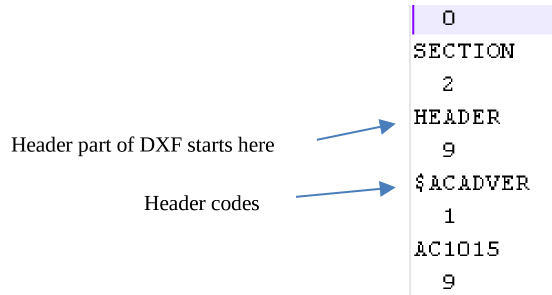


Figure 6 Sample part of a DXF File. Header file full of codes

Jupiter Translation (JT): “The JT format is an industry focused, high-performance, lightweight, flexible file format for capturing and repurposing 3D Product Definition data that enables collaboration, validation and visualization throughout the extended enterprise. JT format is the de-facto standard 3D Visualization format in the automotive industry, and the single most dominant 3D visualization format in Aerospace, Heavy Equipment and other mechanical CAD domains” (Siemens JT Format Reference). JT is mainly used for collaborative product design and development, and it is supported by an industry forum called JT Open. JT is currently an ISO standard (ISO 14306) even though it was created by Engineering Animation Inc. and Hewlett Packard. JT is the standard interoperability format in Siemens PLM software. JT is a complementary technology along with STEP as it can be used for large files (McKenzie-Veal 2012).

PLM XML: PLM XML is a set of XML schemas developed by Siemens PLM to boost interoperability. The idea behind PLM XML is very similar to Quality Information Framework of DMSC as both try to define a collection of schemas to facilitate the seamless data exchange (QIF will be explained in the upcoming sections). PML XML is claimed as open (schemas are published online), lightweight, extensible and it has Software Development Kit (SDK) that allows software developer to adopt PLM XML and read and write data in PLM XML format. It supports product geometric representation (point, curve, and cte.), metadata, data ownership and has many other properties. The major difference between QIF and PML XML, besides the hierarchy of data in schemas and QIF library format, is that QIF just supports metrology systems, but PLM XML supports product life cycle, which is much more comprehensive than QIF (metrology system is a part of product life cycle). PLM XML is being developed and it does not completely support the product life cycle currently (Siemens PLM Software 2011). Another important difference between QIF and PML XML is their developing organizations that can make a big difference in the future for industry to adopt the standard (DMSC is a consortium, but PLM XML is developed by Siemens and open forum).

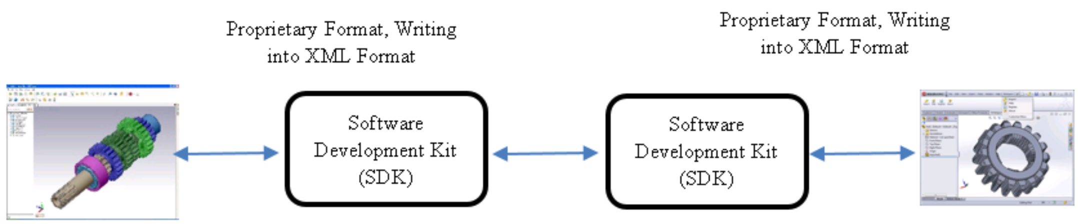


Figure 7 PLM XML and its implementation by Software Development Kit (Siemens PLM Manual)

Metrology Systems Related Data Formats:

As discussed in the previous chapter, metrology systems include four main activity: product design, measurement planning, measurement execution, results analysis and reports. In this section, we will explain the data formats used in metrology systems world to exchange metrology information.

Inspection Plus, Plus Dimensional Measuring Equipment (I++DME): This data format is developed by a group of European automakers (Daimler, Audi, BMW, Opel, Volkswagen, Porsche, and Volvo) -as some of main users of CMMs- to allow them choosing any CMM software for any CMM controller with no concern about interoperability barriers. It is composed of a client and a server. The Client is in fact the metrology software and the server is the measurement plan executer or the CMM controller. It is based on a Transmission Control Protocol/Internet Protocol (TCP/IP) messaging protocol and the measurement results could be saved using Unified Modeling Language (UML), DMIS or any other format (Proctor et al. 2007). A test suite has been developed by the National Institute of Standard and Technology (NIST) to accelerate the adoption process and let the users to see what they should expect in the measurement process. The test suite can read the files in DMIS or I++DME file format and then the client interprets the file and sends the commands to server; the server then executes the commands and sends the results to the client (Proctor at al. 2007).

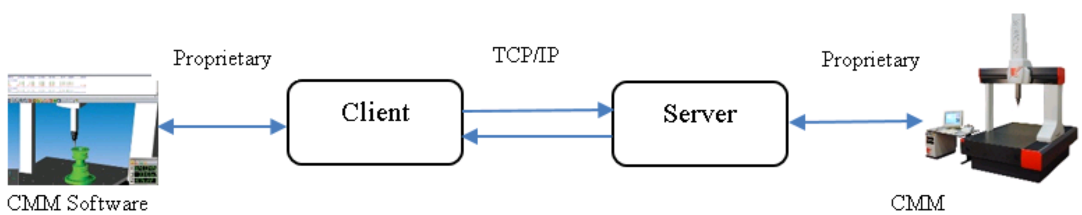


Figure 8 I++DME Working Diagram (Proctor at al. 2007)

Dimensional Metrology Interface Standard (DMIS): DMIS is an ANSI and ISO accredited standard that was funded by Computer Aided Manufacturing International, Inc. (CAM-I) and the first version (DMIS 1.0) was developed by IIT Research Institute under their contract to CAM-I. From the seventh version (DMIS 5.1), DMIS is being developed by the efforts of Dimensional Metrology Standard Consortium (DMSC) and the latest available version is 5.3 (ANSI/ DMIS 5.2, 2009). The main idea behind DMIS was allowing exchanging information within measurement plan executing software and CMM controllers. DMIS has two parts: DMIS part 1(input) and DMIS part 2 (output). DMIS part 1 includes features, tolerances and sensors. DMIS part 2 is more about the execution of commands and the results of measurements. DMIS parts 1 is widely used in industry, but DMIS part 2 which is equivalent to I++DME (mentioned before this) has not been implemented or used widely and its European counterpart is more welcomed (I++DME is not an official standard!). DMIS is currently a part of QIF V2.1. (QIF will be explained later)

Dimensional Markup language (DML): DML is a standard data format to report measurement results. It is based on XML technology and the idea behind it is to gather the results from DME and sending to a statistical process control (SPC) for further data analysis. DML committee has announced their mission as following: “The mission of the DML specification committee is to define and document the data format and content requirements for dimensional inspection results for reporting and database application” (Schafer 2013). DML defines a single eXtensible Mark-up Language (XML) schema to report measurement results and so data exchange will be limited to measurement results and not beyond that. Following picture illustrates a snippet of DML file. DML development was the base for QIF Results (part of QIF), which will be explained more in the upcoming sections.

```
<Feature id="F704" name="HOLE704">
    <AppliedTolerances>
        <Tolerance_id id="T7"/>
        <Tolerance_id id="T8"/>
    </AppliedTolerances>
</Feature>
```

Figure 9 Snippet of a DML File (Schafer 2013)

Q-DAS ASCII Transfer Format: Beside the official standards, there are de facto standards such as Q-DAS ASCII transfer format that are used industry wide. Q-DAS ASCII is utilized just to transfer the product characteristics (especially statistics of them) information and the geometry is usually transferred as a separate STEP file because the formats like IGES and STEP (first version of AP 203) do not provide semantic relationship for the PMI and geometry (Imkamp 2005). The metrology software could read the PMI and associate the characteristics with proper geometry by using the numbers in STEP file and Q-DAS ASCII file. Q-DAS ASCII uses K-fields (K codes) to define the information. For example, K2101, K2112, K2113 are used for nominal geometry, upper tolerance and lower tolerance respectively. For creating a measurement plan by any CMM software, the geometry is imported to software as a STEP file, and then the second file, which is the list of characteristics (K-Fields list), is imported along the STEP file. After choosing the proper sampling strategies and probing information, measuring plan could be simulated for collision test and send to CMM controller for execution. Figure 11 shows how Q-DAS ASCII transfer works (Q-DAS ASCII Transfer Format Version 6, 2015).

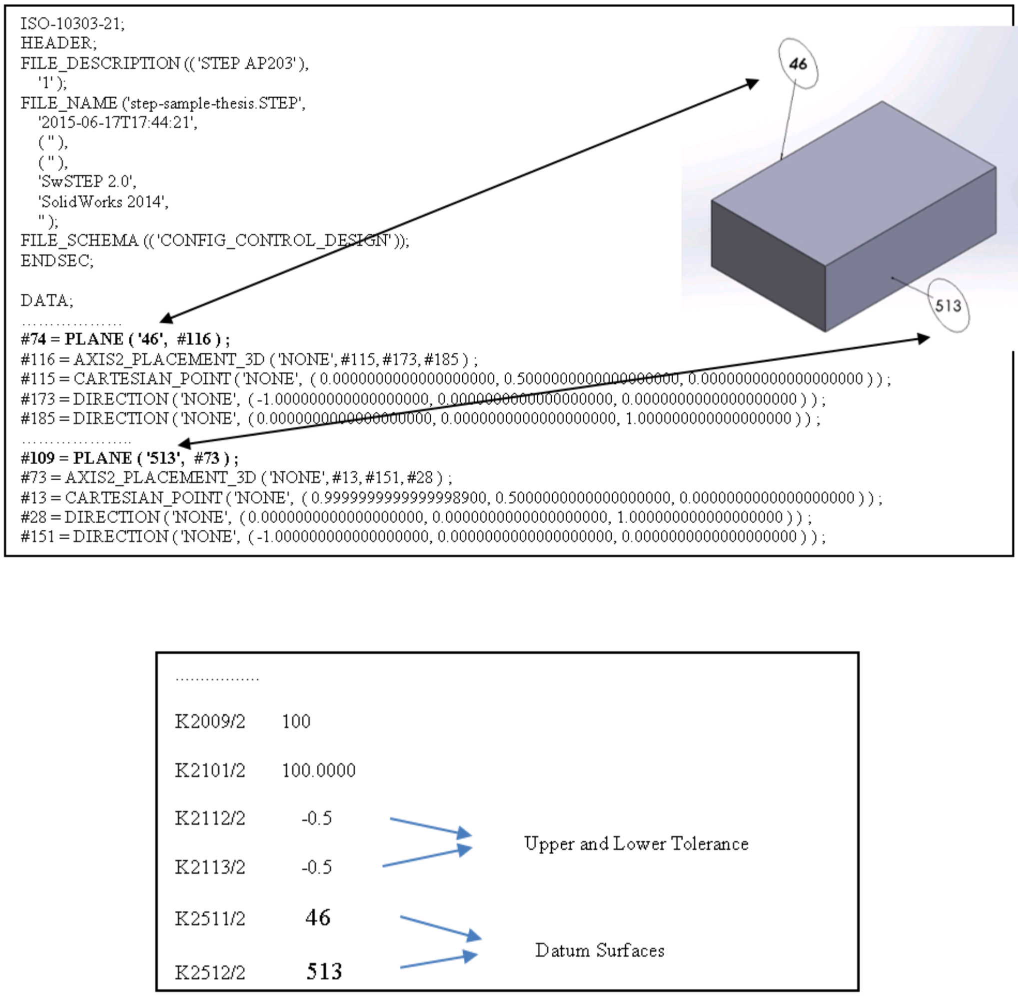


Figure 10 Sample STEP File connected with QDAS ASCII Format to Transfer Characteristics

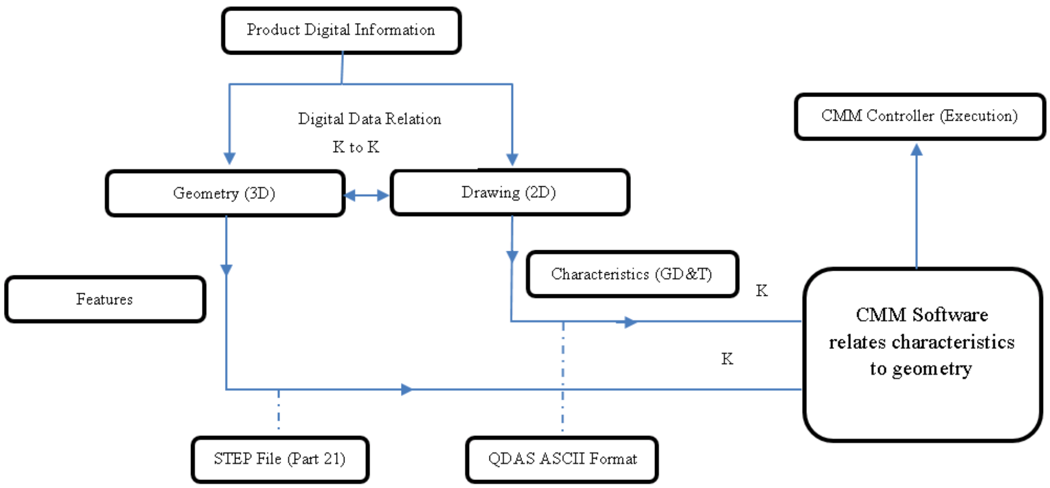


Figure 11 How QDAS ASCII Works (Imkamp 2005)

All standards mentioned so far have been successful to resolve interoperability issues partially; however, none of them solves all of interoperability issues, specifically quality systems interoperability issues. Splitting the metrology process into four steps of product definition, measurement process planning, measurement process execution and analysis and reporting, interoperability barriers are discussed for any of these steps.

For the product definition, PMI is not properly available in the aforementioned data formats (except AP242). PMI (GD&T) needs to have a semantic relationship with the measuring features and not just as annotations. In addition, GD&T standards need to be more harmonized, as there could be different interpretations for GD&T concepts. There is not a non-proprietary data format that covers PMI comprehensively (GD&T, surface finish, annotation, etc.) at the time this research was initiated (AP242 does not cover surface finish).

For the measurement process planning, there is not enough metrology resource information to create measurement plans. Measurement resource information such as accuracy, calibration history, and available sensors are mandatory to create measurement plans. The creation rules and methods of measurement plan are vague or not available. In addition, there is not a standard method to capture the GD&T information.

In the measurement process execution part, two main standards are I++DME and DMIS. DMIS part one supports high-level process plan execution and part two and I++DME supports low-level process plan execution. I++DME is not an official standard and DMIS part two has no industrial implementation. In addition, DMIS part one needs to be expanded to support more equipment.

Finally, for the analysis and reporting of results, the interoperability issues are uniform traceability support, no closed-loop chain with the design intent and no uniform statistical methods for dealing with measurement results (Zhao et al. 2011).

All of the aforementioned problems lead the metrology community to develop a comprehensive standard data format (QIF) that will solve all the interoperability issues for metrology systems and integrate the quality systems information framework.

Quality Information Framework (QIF): QIF is one of the latest efforts to resolve interoperability issues within metrology and quality systems. It was launched by DMSC, which is a consortium formed by members from industry, government, and academia, in late 2010. It plays a roll similar to STEP (integrating product life cycle), but in a limited domain and it is restricted to quality and metrology systems. QIF covers domains like metrology rules (sampling rules) and measurement resources information (like sensors) that STEP AP219 did not cover. STEP AP 219 had limited capabilities and has a lot stubs left empty for future development as mentioned.

QIF is consistent with other metrology standards such as DMIS and in fact it explicitly uses DMIS (part 1) and I++DME, rather than duplicating the standards (DML was also a foundation for QIF Results). QIF splits the metrology process into four steps: Product Design, Planning, Programming, Execution, Results and Analysis Report. For each step, some XML schemas along with a QIF library schemas are defined to allow seamless exchange of information (next chapter will give more information about XML schema definition). The applications schemas are illustrated in the following figure in a data flow diagram (figure 12).

Figure 12 QIF application and library schemas in a data flow diagram (ANSI/QIF V2.1)

The QIF library is composed of fifteen schemas that are accessible throughout metrology systems. The Units.xsd schema defines the units required in the metrology system activities. Primary units include area, angle, force, length, mass, pressure, speed, temperature and time. Beside the primary units (SI units, unit name, and conversion factor for that unit), the end user or the software developers can define their own User Defined Units (UDU).

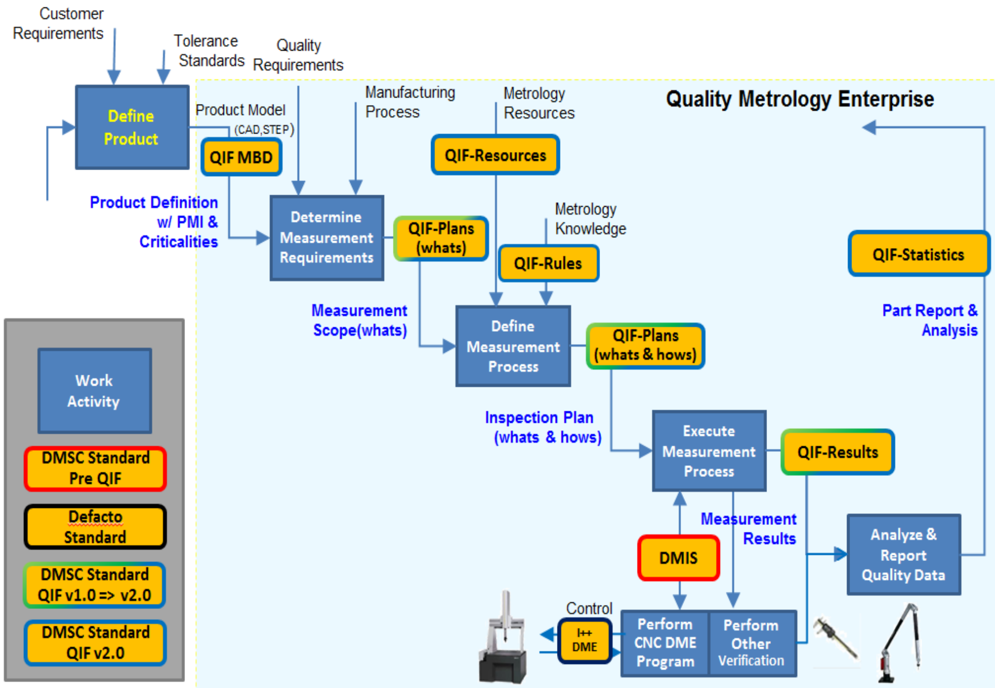


Figure 13 QIF Activity Model (ANSI/QIF V2.1.)

The Characteristics.xsd, IntermediatePMI.xsd, and PrimitivePMI.xsd define different PMI information in different levels. Using these schemas, QIF V2.1 represents ISO 1101 (Version 1983, 2004, 2012), BS_8888_2004, JIS, DIN, ASME Y14.41-2003, ASME Y14.36– 1996, ASME Y14.6- 2001, and ASME Y14.5M (Version 1982, 1994, 2009). The Characteristics.xsd covers high level PMI such as different dimensional tolerances (flatness, straightness, etc.). The IntermediatePMI.xsd models intermediate level PMI like datum translation, different tolerance zones, etc. The PrimitivePMI.xsd schema defines low-level PMI including type of coordinates, manufacturing methods, etc.

The Geometry.xsd and Topology.xsd schemas are used to define product geometry and topology based on Boundary Representation (B-Rep) method. The Traceability.xsd, Features.xsd, Statistics.xsd, and Visualization.xsd schemas model traceability, measurement features (not machining or geometrical features), low-level statistical information, and product view information respectively. These schemas will be explained more lately. For the measurement execution step, QIF takes advantage of I++DME and DMIS (part 1) rather than defining a measurement execution schema.

QIF defines four types of features and characteristics (tolerances): Definition (Feature definition and Characteristics definition), Nominal (Feature Nominal and Characteristics Nominal), Item (Feature Item and Characteristics Item), and Actual (Feature Actual and Characteristics Actual).

Feature definition and Characteristics definition: This aspect refers to a feature or characteristics without defining its position or implying an instance of that feature or characteristics.

Feature Nominal and Characteristics Nominal: This aspect defines a nominal feature or characteristics with its position (defines an instance of a feature or characteristic).

Feature Item and Characteristics Item: This aspect for a feature refers to a feature at any stage of metrology that could be before or after measurements. It can refer to a nominal feature, upstream of CAD data, etc. For a characteristic, item aspect is a method to apply a characteristic to a feature.

Feature actual and characteristics actual: This aspect defines an actual feature or characteristics after measurements (or constructed).

QIF defines 32 features that most of them are consistent with DMIS features and as you see, the measuring features can be different from machining features like pocketing. These features (measurement features) are listed in the following table:

Table 3 DMIS and QIF Features (DMIS 5.2 and QIF)

Feature in QIF	Feature in DMIS	Feature in QIF	Feature in DMIS
Arc	Arc	Generic	Object
Circle	Circle	Opposite Planes	Opposite symmetric planes
Cone	Cone	Opposite Planes	Parallel planes
Conical Segment	Conical Radial Segment	Pattern	Pattern
Opposite Lines	Centered parallel line	Plane	Plane
Cylinder	Cylinder	Point	Point
Cylindrical Segment	Cylindrical Radial Segment	Cuboid	Rectangle
Edge Point	Edge Point	Sphere	Sphere
Ellipse	Ellipse	Spherical segments	Spherical radial
Elongated cylinder	Elongated cylinder	Surface-of-revolution	Surface-of-revolution
Point Defined Curve	Generic curve	Toroidal segments	Toroidal radial segments
Generic	Generic feature	Torus	Torus
Point Defined Surface	Generic surface	Run out Group	No Equivalent
Line	Line	Extruded Cross Section	No Equivalent
Profile Group	No Equivalent	Threaded	No Equivalent
Compound	COMPOUND	EllipticalArc	No Equivalent

QIF and DMIS both represent different ASME Y14.5 symbols and their rules to some extent. Table 3 shows the tolerances covered in DMIS 5.2 and QIF (ANSI/QIF V2.1, 2015).

Table 4 Covered Tolerances (DMIS 5.2 and QIF)

Angle between features	Composite position	Distance between features	Profile of a line	Total run out
Angle	Composite line profile	Distance with respect to another feature	Profile of a point	Width
Angle with respect to another feature	Composite surface profile	Flatness	Profile of a surface	Circularity
Angularity	Concentricity	Parallelism	R Radius	Diameter
Circular run out	Cylindricity	Perpendicularity	Straightness	Position
Symmetry				

Summary: There are a couple of requirements for a data exchange standard to be successful. These criteria are:

- Information modeling languages (EXPRESS and XML Schema)
- Implementation methods (Like STEP Part 21 or Part 28)
- Not limited by intellectual property and being available to everyone
- Democratic process to vote for standards

- Specific consortium or organization responsible for updating standard

Some of the standards discussed so far, like STEP meet these requirements, but some standards like IGES did not meet all of these requirements. QIF, STEP AP242, and PLM XML are being developed and it is a long journey for them to support product life cycle management (PLM) or intended part of product life cycle. QIF is the only comprehensive standard trying to represent quality systems, and all the quality systems are expected to implement QIF, as it gets more mature. It also uses some of other standards (like I++DME, DMIS, and DML) rather duplicating them. Table 4 has summarized advantage and disadvantages of different data exchange formats.

One of the important factors in developing the standards is the information modeling languages that they use for representing product data. For instance, XML schema allows data exchange over the World Wide Web (WWW) and it is a new technology comparing with EXPRESS schema language or UML. Information modeling languages do not allow a standard to have different flavors (or different implementations) and make it easy for companies to implement the standards. Next chapter will be more about how information-modeling languages capture product (life cycle) data.

Table 5 Summary of Discussion

Format Name	Modeling Language	Engineering Domain	Type of Standard
STEP	EXPRESS	PLM	Open (ISO)
QIF	XML Schema	Quality Systems	Open (ANSI)

Table 5, continued

I++DME	No Language (TCP/IP)	CMMs (Execution)	De Facto
DMIS	No Language	CMMs (Execution And Planning)	Open (ANSI and ISO)
JT	No Language	Visualization	Open (ISO)
PLM XML	XML Schema	PLM (Developing)	Industry (Siemens)
IGES	No Language	Geometry	Open (ANSI)
DXF	No Language	Wireframe	Industry (Autodesk)
DML	XML Schema	Measurement Results	Open (DMIS)
STL	No Language	Geometry	Industry (3D Systems)
SAT	No Language (ASCII)	Geometry and PMI	Industry (Spatial Corporation)

Chapter 3: Information Modeling Languages

Preface: This chapter is about the methods and applications used for representing and storing information. Sharing the information in an unambiguous (unique data interpretation), effective, and secure structure is one of important concepts of digital manufacturing enterprise. Digital manufacturing enterprise requires an easy flow of information without any interoperability issues. Information modeling languages are one of the tools to store data in an unambiguous, effective, and secure structure.

What is an 'Information Modeling Language'? "Information modeling language is a technique for specifying the data requirements that are needed within the application domain" (Lee 1999). Information modeling languages are used to describe semantic relationships, restrictions, hierarchy, logics, and rules. These languages are not like programming languages that execute a code or performs a task, but acts as tool for programming languages that may need concepts, rules, or hierarchies. Data and information could be interpreted in different ways that could cause ambiguity and ambiguity could be a barrier to exchange information, and that is why information-modeling languages are an inseparable part of the open standards. Information modeling languages are closely tied to Data Base Management Systems (DBMS). "A database management system (DBMS) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data". Developing information modeling languages dates back to 1970s that the DBMS were increasing and the neutral interpretation of information was an issue as the information was shared between different parties. ANSI conducted a research on database management systems that resulted in developing three schemas. These schemas are external schema (human interpretation of data), internal schema (machine interpretation of data), and conceptual schema (neutral interpretation of data) (Tsichritzis and Klug, 1978).

The first generation of DBMSs were navigational or hierarchical in terms of queries, which is very similar to information modeling languages that develop a hierarchy and structure for instance files. The second generation was called Relational Database Management Systems (RDBMS) that were conducting queries based on the content looking for rather than following the structure to find the information. The third generation of DBMSs are called post-relational DBMS, they use key-value to find the data, and they are document-oriented database. DBMS and information modeling languages are two complimentary technologies. DMBS store the data and whenever needed, information is retrieved through queries and transformed to the structure defined by information modeling language structure.

Information Modeling Methodologies: For the sake of discussion, information modeling methodologies will be classified into three types

- Object-oriented methodology (O-O)
- Entity-Relationship methodology (E-R)
- Functional-modeling methodology (F-M)

Using each of these languages depend on the implementation methods, requirements and priorities. O-O methods are easy to implement if using object oriented programming languages as they treat every piece of information like an object. The E-R method is based on entities and the relationship between entities and it usually uses graphical representation.

It was introduced by Peter Chen in 1976 as a method combining three other methods (network model, relational model, and entity set model) (Chen 1976). The F-M method emphasizes on functional properties of information modeling and it usually illustrates information flow diagram, activity models, and processes.

There are a couple of information modeling languages that have been used in digital enterprise and manufacturing systems, and these are IDEF1X, EXPRESS, XML Schema, and Unified Modeling Language (UML). In the following section, we review these languages, their advantages, disadvantages, and implementation methods.

IDEF1X: IDEF (Integration DEFinition) is a series of methods (IDEF0, IDEF1, and IDEF2) developed by US Air Force (USAF) Integrated Computer Aided Manufacturing (ICAM) program and "was derived from a well-established graphical language, the Structured Analysis and Design Technique (SADT)" (IDEF website). The IDEF0, as a graphical language and functional modeling method, includes a sequence of activity functions (boxes) with arrows showing input, output, control, and mechanisms that display system functionality. IDEF1X is the developed version of IDEF1 that is used to manage information requirements in an enterprise. IDEF1X is an entity- relationship (ER) information modeling language that could be used for data base systems and if the target system is object- oriented or functional, the IDEF1X is not a wise choice. The syntax and semantics of IDEF1X includes entities, relationships, attributes/keys, and notes. Entities are real world objects or anything that we are interested to model like employees, employers, places, etc. The relationships allow illustrating the relationship between entities (connection relationships) and categorizes them (categorizing relationships). Attributes are unique and describe entities throughout the model. A group of attributes is called keys.

The other members of IDEF group are IDEF3, IDEF4, and IDEF5. IDEF3 is a method used for process description capturing. IDEF4 is the object- oriented version of IDEF and IDEF5 is an ontology description capturing method.

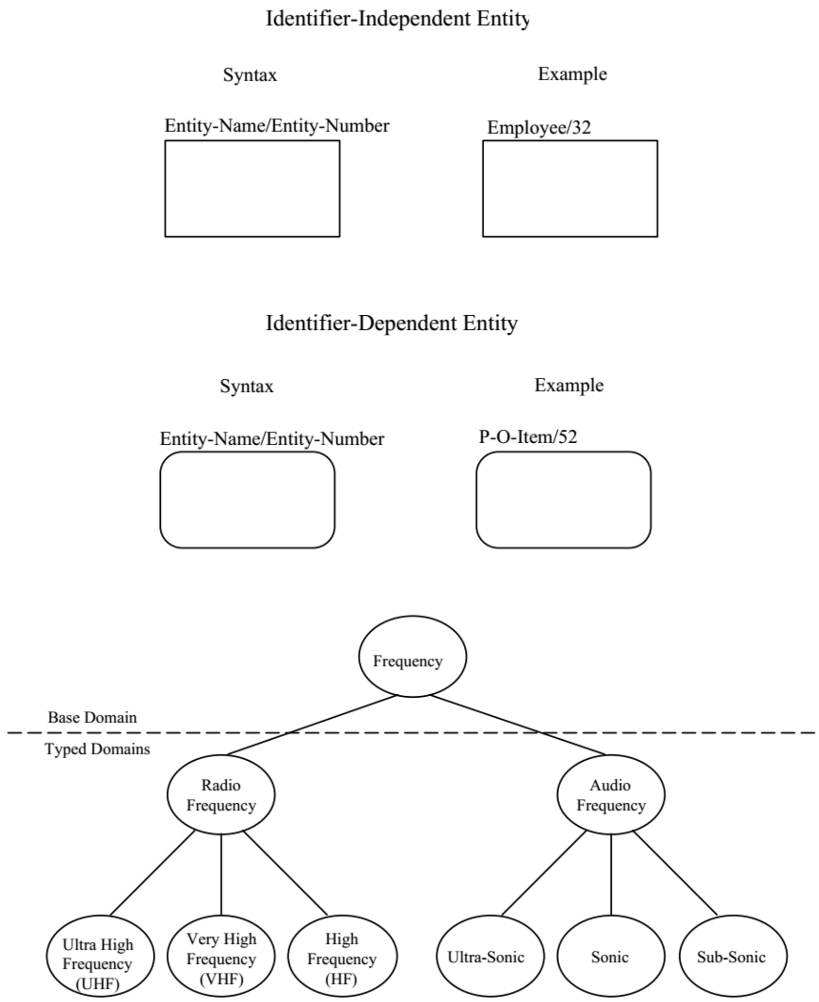


Figure 14 Entity Syntax (left) and domain hierarchy (right) in IDEF1X (IDEF1X manual)

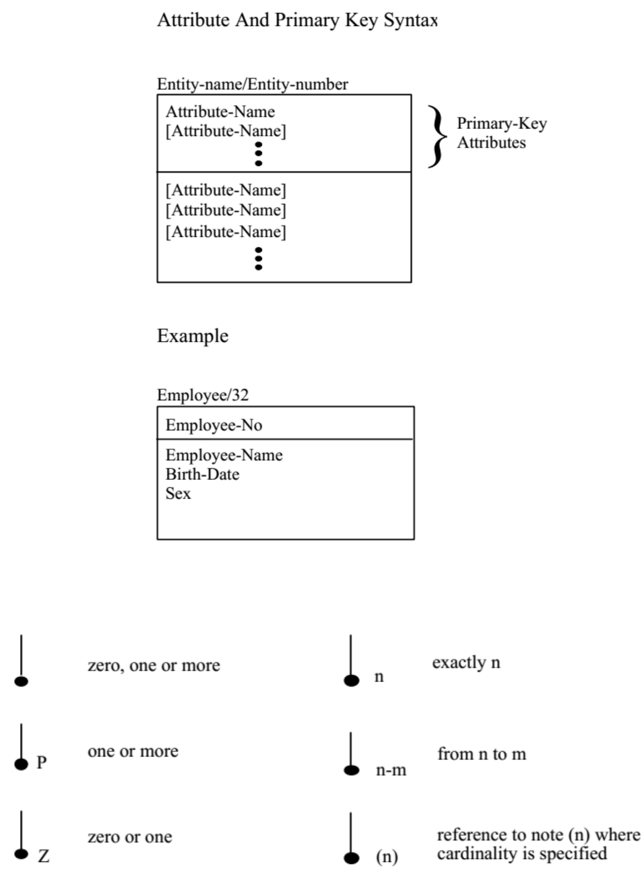


Figure 15 Attribute and primary key syntax (left), Relationship Cardinality Syntax (right) in IDEF1X (IDEF1X manual)

EXPRESS: EXPRESS is the official part 11 or description methods for ISO 10303 or STEP. It is a textual and object-oriented flavor language that allows implementing complex modeling rules and concepts. A couple of programming languages have contributed to EXPRESS development such as C, C++, SQL, PASCAL, etc. Modeling in EXPRESS includes four phases that are basic objects, relationships and attributes, completion of constrains, and model integration (Schenck and Wilson, 1994). First, the modeling objects should be identified. These objects can be a vehicle tag number, color, or any other entities. Next, the relationship between these entities and attributes should be considered. This relationship could be any rule or constraint. Third, the global constrains are addressed, and finally for the large models like STEP the model is integrated as the development process was split into pieces to make it easy. EXPRESS supports SUBTYPE, SUPERTYPE, arithmetic operations, enumerations, logical operations, mathematical function (e.g. tan(x)), etc.

Figure 16 is an example of EXPRESS schema. As you see, schema starts with the schema name “Example”. The entities are defined with the key word “ENTITY” and the definition ends with the key word “END_ENTITY”. The key word “ABSTRACT” is used to define the entities that cannot be instantiated (abstracts). The type of each attribute is defined after the entity attribute. For example, the “Name” and “Familyname” attributes of entity “person” are defined after attributes (“STRING”).

```
SCHEMA Example
ENTITY university
    ABSTRACT SUPERTYPE OF (UNCC) ;
    NumberofStudents  :  INTEGER ;
    AREA   :  REAL ;
    StartDate  :  date ;
    Semester  :  semesterType ;
    Student    :  person
END_ ENTITY
Type  semesterType=ENUMERATION OF (FALL, SPRING, SUMMER) ;
END_ Type
ENTITY  person
     Name : STRING ;
    Familyname : STRING ;
END_ ENTITY
ENTITY UNCC SUBTYPE OF (university)
  motto : STRING
```

Figure 16 Sample EXPRESS Schema

For EXPRESS schema implementation purposes, NIST launched a project called: STEP Class Library (SCL). SCL inputs the EXPRESS schemas and translates them into standard C++ libraries that could be used in different applications. The SCL facilitates and accelerates the adoption process by third parties and CAD vendors (by transforming them to standard C++ classes). In fact, there are two implementation methods called: Late-Bound (compile time binding) and Early- Bound (run time binding). Late- binding uses a data-dictionary that makes it hard to program, and pushes the load toward post- compiling process, but in early binding the load is more for compile- time process. Early- binding is easy to program and fast versus late- binding that is hard to program, easy to prototype and it is slower. SCL uses both of these methods (early and late binding). It implements the early- binding for the EXPRESS schema (compile time), and late binding and early-binding for some of the entities and attributes instances (Loffredo 1999).

EXPRESS also has a graphical form that allows navigating through schemas and develop them easily. This graphical form is called EXPRESS-G. The rules of EXPRESS- G is represented in the following figure:

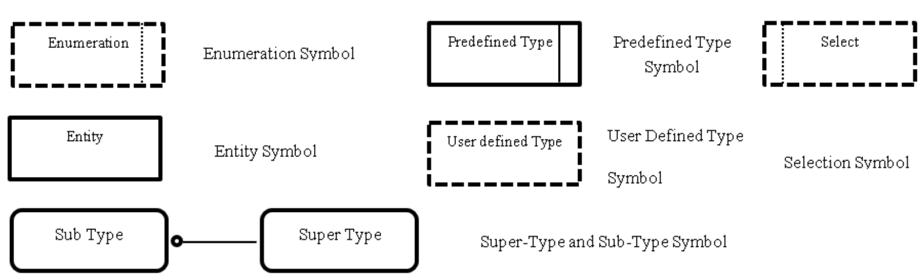


Figure 17 Guide to EXPRESS-G Symbols (NIST Magazine)

The high-level structure of STEP is made of five areas: Description Methods, Implementation Methods, Conformance Testing, Application Protocols (APs), and Application Resource Model. The description methods are the information modeling languages that are used to describe methods. For example part 11 and part 12 are EXPRESS and EXPRESS-I. Implementation methods describe how the schemas are implemented. For example, part 21 is the STEP physical file (figure 8) and part 28 is implementation by using XML schema. Conformance testing is the methods used to examine if the schemas and instances of schemas comply with STEP. The application protocols or APs, which are the most important part of STEP, are assigned to describe each part of product life cycle. Each AP is made of three segments: Application Activity Model (AAM), Application Reference Model (ARM), and Application Interpreted Model (AIM), which in fact is the encoded ARM in EXPRESS schema language. AAM is the graphical representation of the process (EXPRESS-G) and the flow of information. Every box in AAM defines an activity in product life cycle, and ARM is just AAM in the form of application objects (Xu and Nee, 2009).

Integrated environments have been developed for creating EXPRESS schemas. These tools can model (EXPRESS-G), visualize (EXPRESS-Ed), edit (EXPRESS-Ed), and translate data into object oriented database management (OODB) environment and this OODB could read and write STEP (physical) files. The other applications (software) could have access to OODB through STEP data access interface (SDAI) or STEP physical file (Sauder and Morris, 1995).

eXtensible Markup Language Schema (XML Schema): XML schema is an information modeling language that allows defining the structures, hierarchy, rules, and restraints for a

XML document. Markup languages like Hyper Text Markup Language (HTML) and especially XML have been the standard tool for representing information throughout World Wide Web (WWW) and this gives great opportunity to XML as being a standard tool to exchange information [Brás et al. 2008]. XML is also used for data storage, extraction, representation of entities on the internet, and data acquisition applications. A XML document is a text document that has many elements and every element has two tags (start tag and end tag) and defines information between these tags. For example, <name> John <name\> is a XML element that <name> is the start tag, <name\> end tag, and “John” is the tag content (information) populated for this tag. This element (a start tag plus end tag) can also have an attribute like an identification code inside the start tag. For example “Student-ID=”125”” is an attribute for <name Student-ID=”125”> start tag. It is obvious that this textual, human, and machine-readable language could be very flexible that cause ambiguity and syntax error. To resolve these issues, a couple of institutions and organizations have published their rules and methods (sometimes standards) for XML schema. Practices published by World Wide Web Consortium (W3C) and Document Type Definition Language (DTDs) are the two main and important standards used frequently. Because of different XML schema languages and naming confusion, the syntax and recommendation of W3C is referred to as XSD and the schema file extension is “*.xsd”. XML schema can also be implemented like EXPRESS of STEP. There are (commercial) tools that create C++ or .NET libraries from XML schemas, and these libraries could be used to create different applications (Loffredo 1996).

A sample XML schema (figure 20) is a collection of “SimpleType” and “ComplexType” (especially in industrial applications) that these types have sub- elements like “<xs:

Sequence> </xs: Sequence>" and "<xs: element> </xs: element>. The "name" attribute of element provides the name of element and the "type" attribute provides the type of it (string, integer, complex, etc.). The "base" attribute in complex types is used for inheritance purposes and is incorporated with <xs: restriction> </xs: restriction> and <xs: extension> </xs: extension> to restrict and extend the base respectively. The xmlns: xs= "http://www.w3.org/2001/XMLSchema" declares that elements and data types come from "http://www.w3.org/2001/XMLSchema" namespace and every element declaration should start with "xs" (like <xs: element name="Name" type="xs: string"/>). The "maxOccurs" attribute could be added to an element to define the number of time that an element could be repeated ("unbounded" for this example).

An example of XML document (XML instance) and XML Schema are presented in the following figures (Figures 19 and 20). As you see, the example has a root element (UNCC) and other elements are sub- element of root element. Any element containing sub- element is a complex element (like <student> </student>) otherwise, a simple element (like <semester> </semester>). XML documents should be validated against XML schemas. Validation here means that hierarchy, syntax, and restraints of the XML document (instance) should be according to its corresponding schema. As example, "Semester" element should be chosen from an enumeration ("fall", "spring", and "summer"), sequence should be enforced inside a complex element (university element), and "motto" element should be string type. The XML schema, that the XML document should be validated against it, can be referenced by xsi: schemalocation= "" from the XML document.

The name space (xmlns) and target name space are used to distinguish between two different XML schemas and XML documents. For example, if the first XML document has an element named <address> with “ns1” as the name space and second XML schema has an element named <address> with “ns2” as name space, if you combine these two schemas it would not cause any identification problem because the <address> element of the first and second XML documents have different name spaces.

An XML schema and the corresponding XML document should have the same hierarchy but this hierarchy is defined by XML schema and followed by the XML document. The analogy used here is illustrated in the following figure (the XML schema and the XML document are similar to a mold and part respectively).

Figure 18 the analogy used to compare the XML document with an XML Schema (www.concretenetwork.com)

```xml
<?xml version="1.0" encoding="utf-8"?>
<UNCC xmlns="university">
  <NumberofStudents>12265</NumberofStudents>
  <AREA>1636546</AREA>
  <StartDate>1959</StartDate>
  <Semester>fall</Semester>
  <motto>49ers</motto>
  <Student>
    <Name>Saeed</Name>
    <FamilyName>HEYSIATTALAB</FamilyName>
  </Student>
  <Student>
    <Name>STUDENTNAME</Name>
    <FamilyName>STUDENTFAMILYNAME</FamilyName>
  </Student>
</UNCC>
```

Figure 19 Sample XML Document Created Based on Schema

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="university"
  xmlns:t="university"
  targetNamespace="university"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:complexType name="persontype">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="FamilyName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="university" abstract ="true">
    <xs:sequence>
      <xs:element name="NumberofStudents" type="xs:int"/>
      <xs:element name="AREA" type="xs:decimal"/>
      <xs:element name="StartDate" type="xs:integer"/>
      <xs:element name="Semester" type="SemesterEnumType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="SemesterEnumType">
    <xs:restriction base="xs:token">
      <xs:enumeration value ="fall"/>
      <xs:enumeration value ="spring"/>
      <xs:enumeration value ="summer"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="UNCC">
    <xs:complexContent>
      <xs:extension base="university">
        <xs:sequence>
          <xs:element name="motto" type="xs:string"/>
          <xs:element name="Student" type="persontype" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="UNCC" type="UNCC"/>
</xs:schema>
```

Figure 20 Sample XML Schema

XML is closely tied with other technologies such as XSLT, XPath, XQuery, HTML, etc. These technologies allow an efficient application of XML to exchange data. For instance, XPath allows to search in XML documents or XSLT technology translates XML documents to other formats such as HTML. XML documents are used in WWW applications and XPath, XSLT, and HTML are essential techniques combined with XML for boosting XML capabilities.

Unified Modeling Language (UML): UML, as an information modeling language, is a tool created during 1990s to replace object- oriented analysis and design methods. Object management group (technology standards consortium) then adopted this tool to "provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software based systems as well as for modeling business and similar processes" [quoted from standard]. UML, or modified version of UML like SysML, has been used frequently in systems engineering for system modeling [Fowler 2004]. UML is made of two parts: Notations and Meta- Models. The notation describes the graphical syntax of the model and Meta- Model is made of three parts: Classifier (the hierarchy of objects), Events (what happens in the process), and Behaviors (a set executed algorithms). UML has two types of diagrams (structural diagram and behavior diagram) each having seven sub categories. The structural diagrams are like the subjects in English grammar that fulfils an action and behavior diagrams are the measures that the subjects performs. UML is a very complex object oriented modeling language comparing with other languages like XML schema and could be a choice for object oriented programming environments like C#, Java, C++, etc.

Summary: Information is asset for an enterprise and manufacturing companies require data exchange between different departments of a company and between different companies. Data could be analyzed and interpreted in different ways and this could be a barrier for interoperability. Information modeling languages are created for neutral and unbiased data interpretation. There are many different information-modeling languages, but the most frequent ones (especially in industrial applications) are EXPRESS, IDEF1X, XML Schema, and UML. Each of these methods have an advantage and disadvantage that are classified in the following table.

Table 6 Comparing Different Modeling Languages

Language	Methodology	Developer	Graphical/Text	Application Area
IDEFX	E-R	Air Force	Graphical	ICAM
EXPRESS	O-O	ISO	Text	ISO 10303
XML Schema	O-O	W3C	Text	QIF, DML, etc.
UML	O-O	OMG	Graphical	Systems and Software Engineering

Chapter 4: QIF Measurement Resources V2.1.

Introduction: This chapter is focused on developing QIFMeasurementResources from V2.0 into V2.1. The version 2.0 was very limited in terms of supporting metrology devices and sensors to create a measurement plan, traceability, and statistical studies. Version 2.0 was limited to devices such as CMMs (Cartesian CMMs), caliper, gage, and comparator. Among, these devices, the hierarchy for gage and comparator were just empty stubs left for future developments. For CMM performance tests, ISO 10360 was developed for acceptance and verification of only Cartesian CMMs. It did not include imaging systems for CMMs, and it covered ISO 10360 parts two (accuracy of CMM structure), three (rotary axis), four (scanning probes), and five (tactile probe in discrete mode). The highest level of hierarchy in schema V2.0 had six sub-elements that include Version, Carriages, Fixtures, MeasurementDevices, Sensors, and Tools (Carriage is removed in V2.1.).

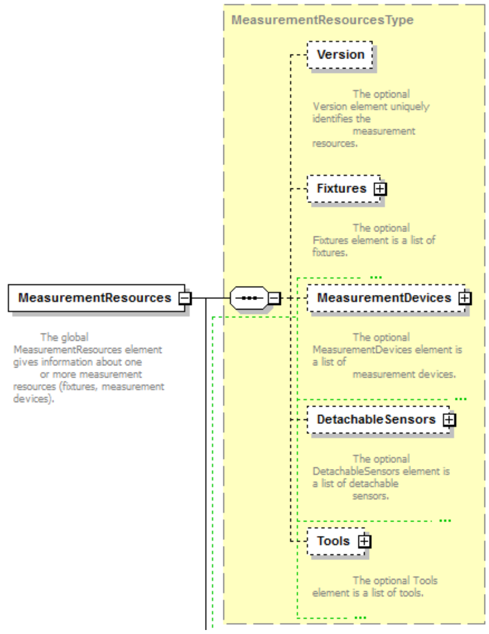


Figure 21 Highest level of hierarchy in QIF Measurement Resources schema V2.1

The “Version” element is type of “VersionType” complex type and it covers information like the time that version was created and the name of employee that signs off the version element. All the types throughout the QIF inherit from their base types if they exist. In the “QIFMeasurementResouces” schema, most of the types and elements inherit from “QIFMeasurementResourcesBaseType” type, which has elements such as “Name”, “Description”, “Manufacturer”, “ModelNumber, “SerialNumber”, and etc. (so almost every type has aforementioned elements). The “Fixtures” element allows defining infinite number of fixtures that all inherit from “MeasurementResourceBaseType”. The optional “PersistentId” element stores identification number of measurement resources anywhere outside QIF (a fixture element, defining a fixture, includes “Description”, “Manufacturer”, “ModelNumber”, “SerialNumber”, etc.).

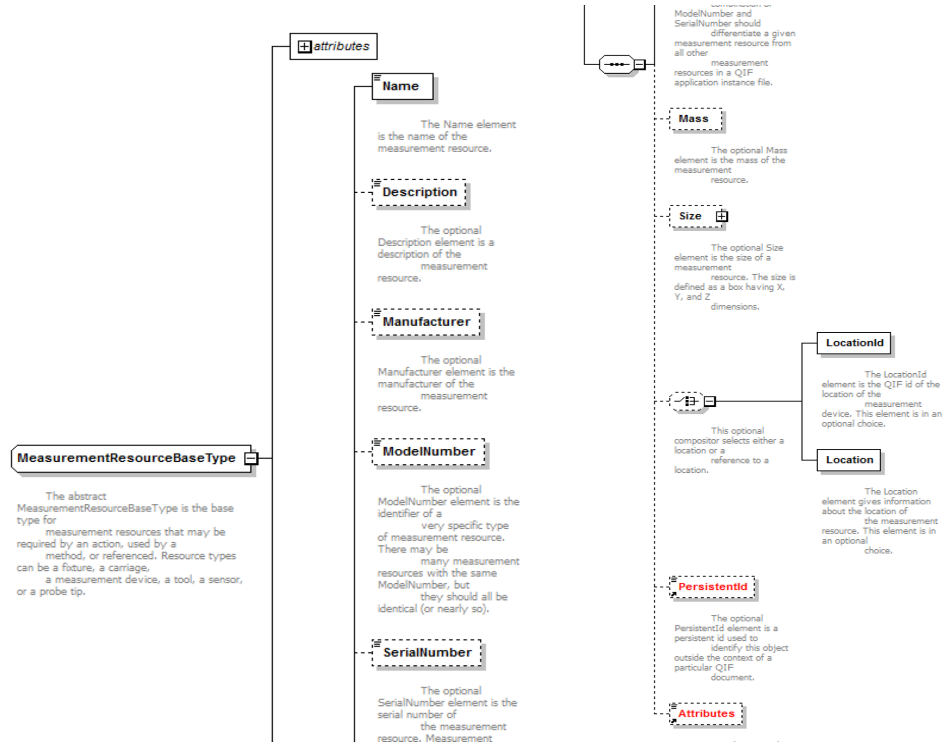


Figure 22 MeasurementResourceBaseType as the base for any measurement resource

The “MeasurementDevices” element is type of “MeasurementDevicesType” and can be used to define infinite number of measurement devices. The devices supported in version 2.1 (added to version 2.0) are autocollimator, different type of CMMs (light pen CMM, Articulated Arm CMM (AACMM), Parallel Link CMM, equator, computed tomography, laser tracker, micrometer, sinebar, microscope, profile projector, theodolite, and universal length measuring machine. Each of these devices has a complex type and includes many sub-elements for each device (will be explained in the upcoming sections).

Any measurement device can have different types of sensors mounted on a tool or use a separate sensor. Version 2.0 did not cover any type of sensors therefore so much effort had been put into defining different types of sensors. These sensors include Charged Coupled Device (CCD) Camera, capacitive sensor, confocal chromatic sensor, DVRT sensor, draw wire sensor, Eddy current sensor, LVDT Sensor, laser triangulation sensor, magneto inductive Sensor, structured light sensor, tactile probe sensor, and ultrasonic sensor. Each of these devices has a complex type (for instance "CCDCameraSensorType") that defines the type of corresponding sensor element. Tools with detachable sensor ("ToolWithDetachableSensorsType") refer to the sensor's IDs mounted on the tool and extra elements that they inherit from "ToolType".

In terms of performance standards, B89.4 test for acceptance and verification of articulated arm CMMs ("AACMMTestType") is added to the Cartesian version of B89.4 test (AACMM). This test includes three sub-tests for AACMMs that are effective diameter performance test, single-point articulation performance test, and volumetric performance test (Hocken and Pereira 2004).

Besides hierarchical development of QIFMeasurementResources, many global complex types are defined that are used in different measurement device types (Appendix A). These complex types include working volumes, effective working volumes, resolution, laser type, etc. Working volume is the volume range that the axes of the measuring devices can travel and the effective working volume is the working volume (volume without any obstacle) that can be used for measurements. This volume is less than the working volume and considers any tool turret, which could be an obstacle for the moving sensor. Four types of working volume and effective working volume are defined that includes Cartesian Working Volume, spherical working volume, closed shell set working volume, and user defined working volume. Cartesian working volume usage is for devices with three perpendicular linear axes, and spherical working volume for devices like laser tracker that uses spherical coordinate system for measurements. Closed shell set working volume and user defined working volume are customized working volumes and the closed shell set one uses boundary representation model defined in Geometry.xsd schema to define working volume.

For defining resolution in measurement devices, a substitution group containing four types of resolution is developed. This substitution group includes Cartesian resolution (for three perpendicular linear axes of measurement devices), linear resolution (for manual devices with one axes), spherical resolution (devices having spherical coordinate system like laser tracker), and user defined resolution (for user defined measuring coordinate system.). Any working volume system or resolution system that uses an axis can be a rotary axis or linear axis that is defined by “Axis” global element substitution group.

Measurement Devices Complex Types: All the measurement devices complex types (at the highest level) inherit from ManualMeasurementDeviceType (for manual devices including caliper, micrometer, sine bar, and gage) or UniversalDeviceType complex types (for non-manual measurement devices). In addition, CMMs (four types of CMMs are defined) inherit from CMMType complex type and CMMType complex type inherits from UniversalDeviceType complex type (more information in Appendix A).

Autocollimator: Autocollimator is an optical measurement device used for angle measurements especially for polygons. The important optical properties of this device is defined in elements like “LightSource”, “FieldOfView”, “ObjectiveFocalLength”, and “ApertureSize”. Figure 23 illustrates how this device is defined in QIF V2.1.

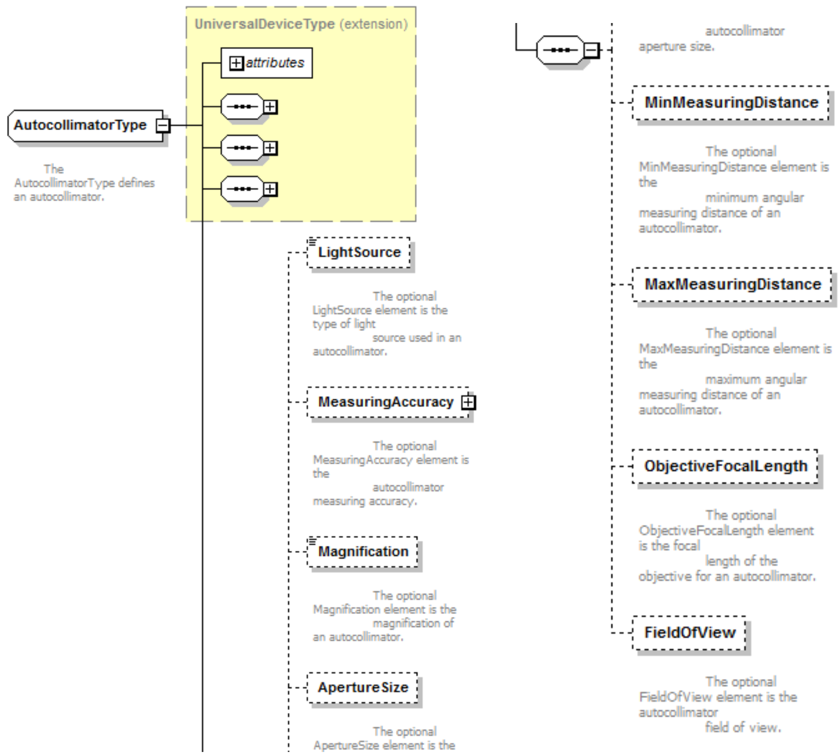


Figure 23 AutocollimatorType complex type definition in V2.1

Coordinate Measuring Machines (CMMs): Four types of CMM are defined and the "CMM" element is the head of this substitution group. All these devices inherit from "CMMType" and "CMMType" inherit from "UniversalDeviceType" which is one of global complex types defined for measurement devices. "UniversalDeviceType" shares the common "Resolution", "WorkingVolume", "EffectiveWorkingVolume", and "TemperatureCompensation" elements that all types of CMMs share these elements. Each of these CMM types have proprietary elements to define their measuring capability that may be used in support of a measurement plan.

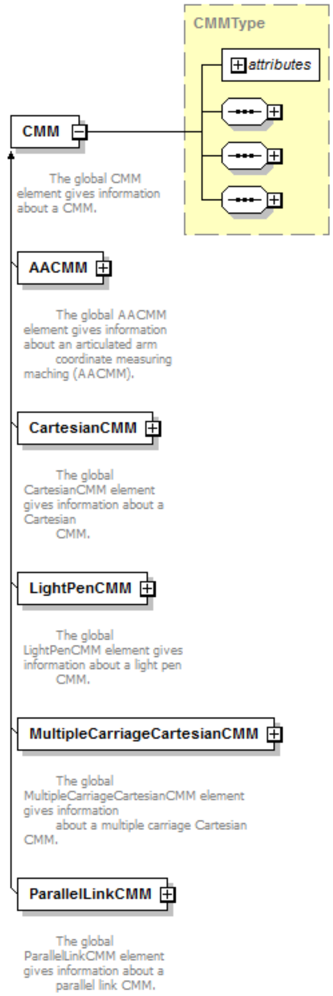


Figure 24 CMM global element substitution group in QIF Measurement Resources

Comparator: Comparator (equator in this case) is an inspection device that compares the part to a master part. It uses a parallel link mechanism to run the measurements and it is modeled according to Figure 25.

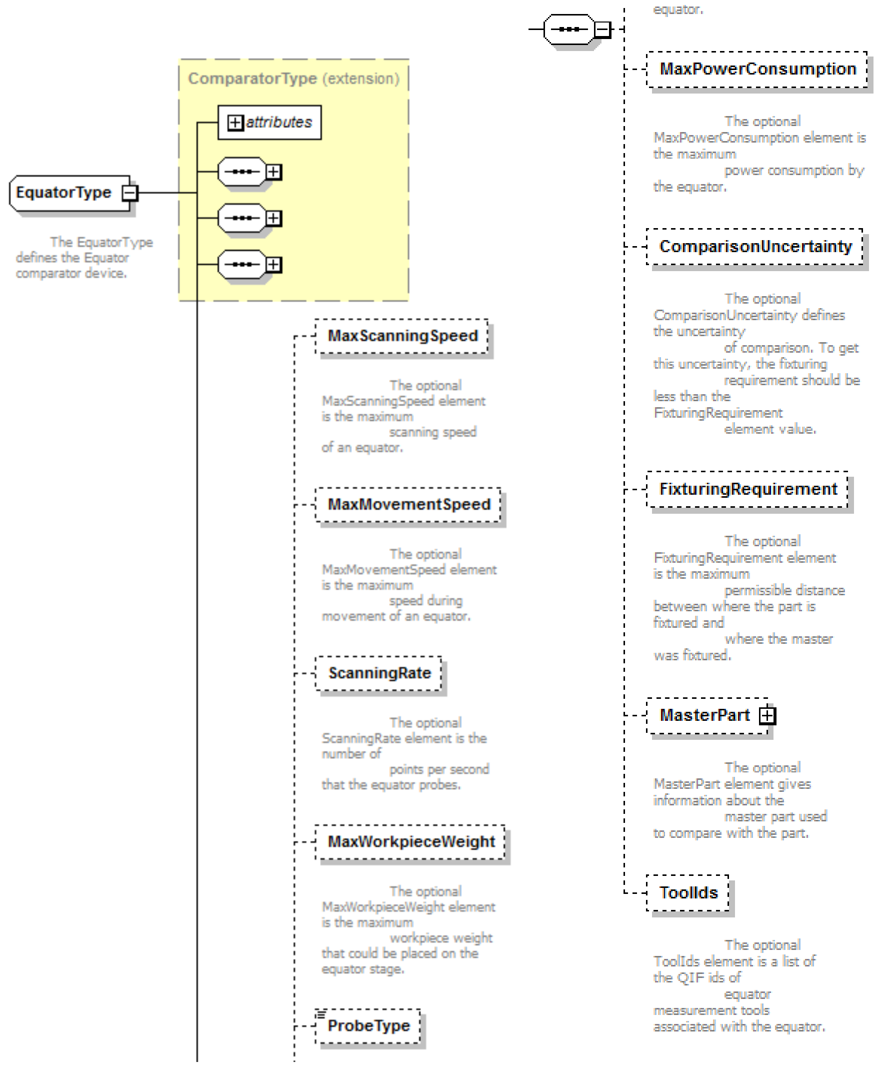


Figure 25 EquatorType complex type definition in V2.1

Computed Tomography: Computed tomography complex type defines a computed tomography DME that inherits from “UniversalDeviceType” complex type. The computed tomography machine works based on X-Ray penetration principle, which is dependent on the properties of the part. The part is mounted on a stage and rotated after each time shooting X-Ray to the part and recording the penetration results by the detector. “ComputedTomographyType” complex type is illustrated in figure 26.

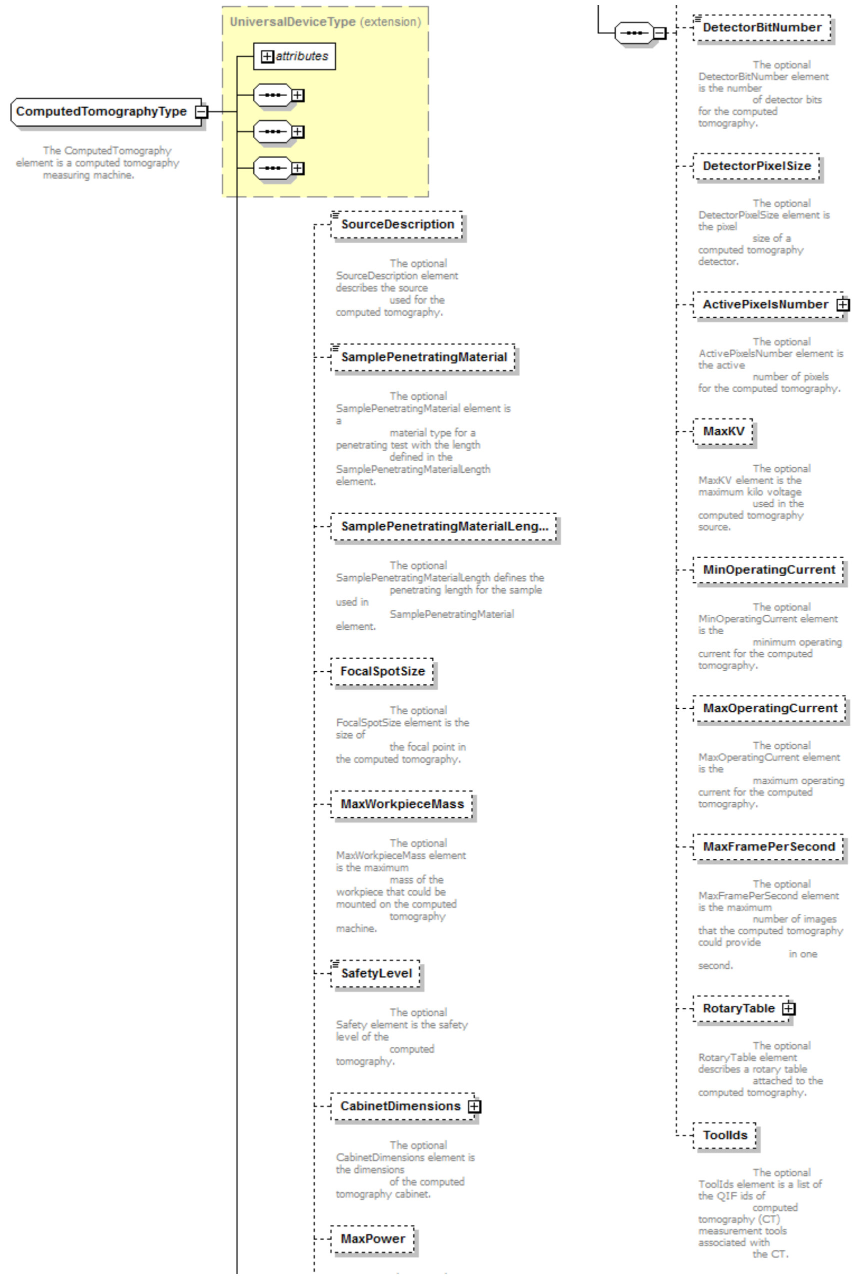


Figure 26 ComputedTomographyType complex type defined in V2.1

Laser Tracker: "LaserTrackerType" complex type defines a laser tracker measurement device and it inherits from "UniversalDeviceType". Laser tracker uses a retroreflector and tracking system to calculate the distance. Laser working amplitude, laser type, linear and angular accuracy are some of important specifications of a laser tracker.

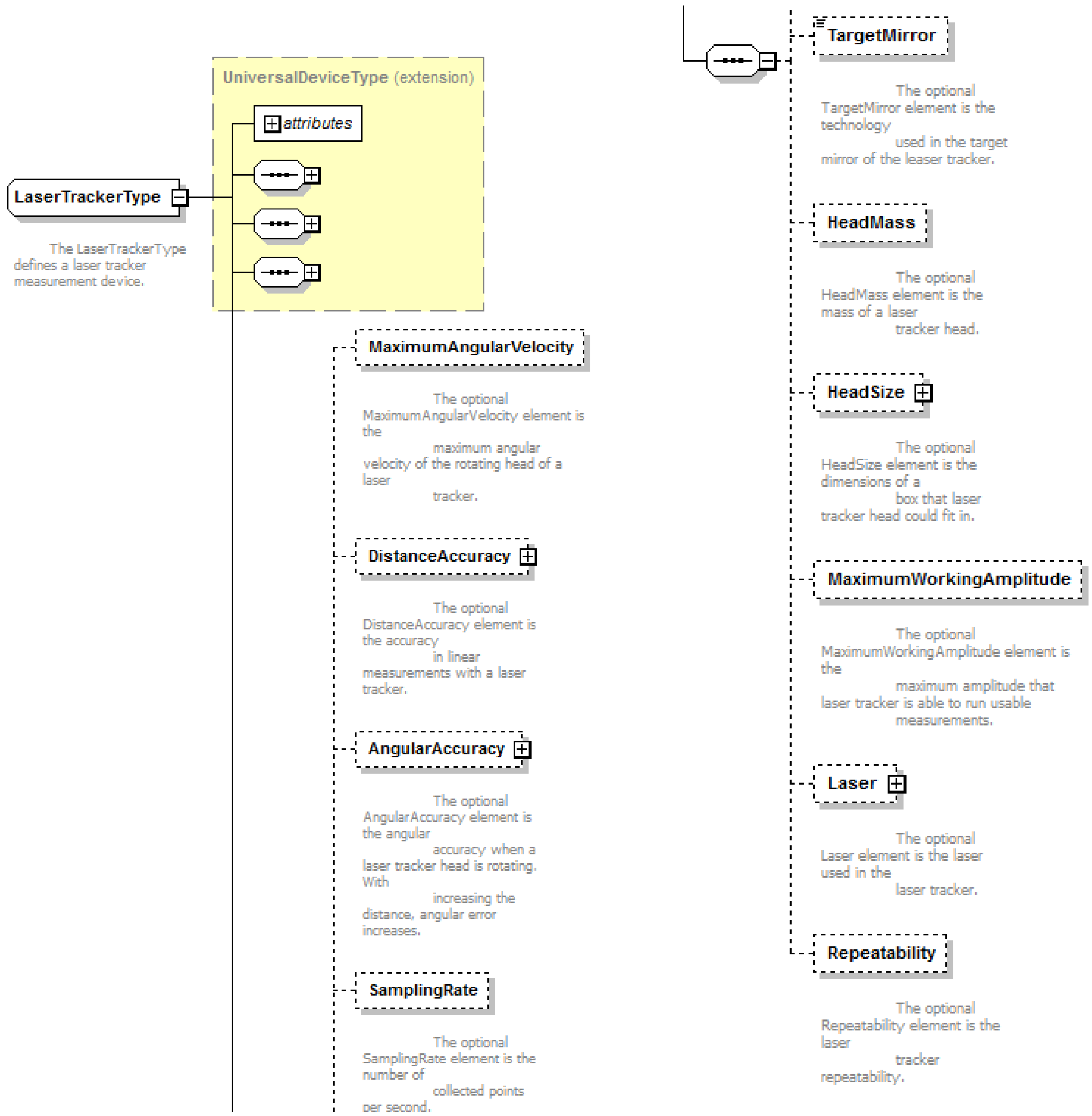


Figure 27 LaserTrackerType complex type defined in V2.1

Manual Measurement Devices: There are four types of manual measurements devices, but Sine bar is the only device added in V2.1. Sine bar is a device made of two cylinders connected to each other and it is used for angular measurement such as measuring a cone angle. “SineBarType” complex type is illustrated in figure 28.

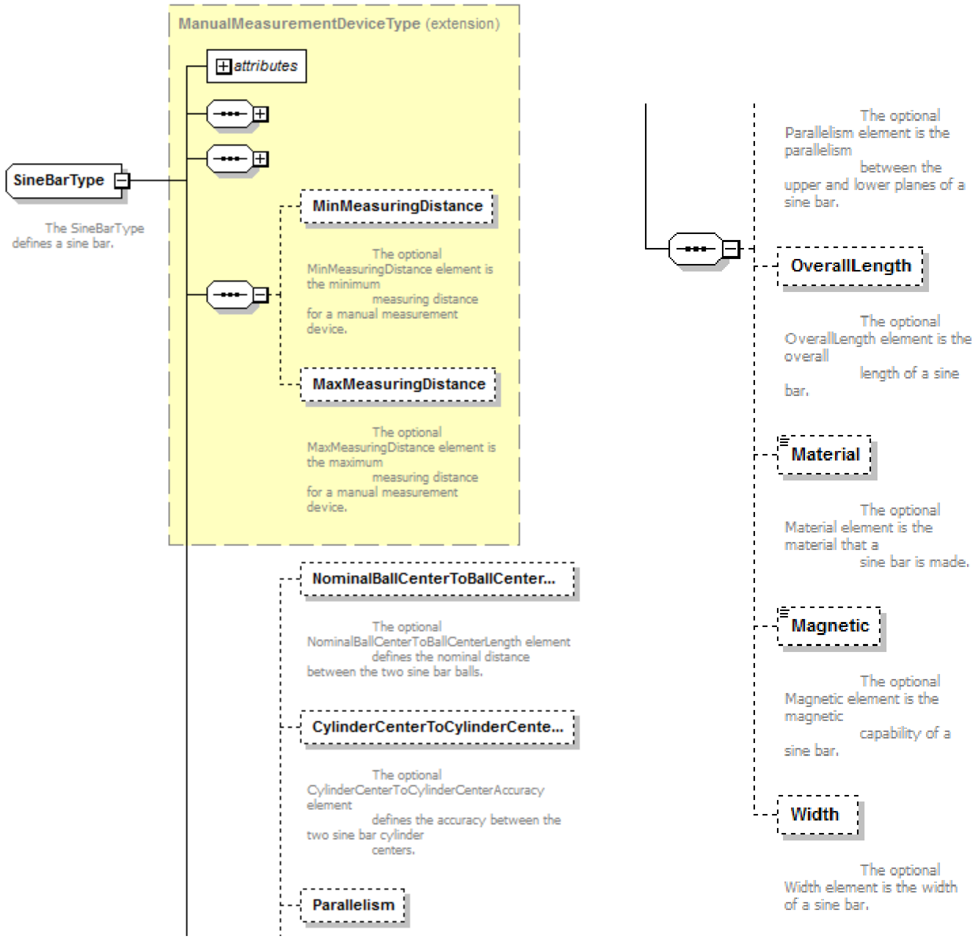


Figure 28 SineBarType complex type definition in V2.1

Microscope: There are many different types of microscopes available in the market used for different applications. The one defined in this schema is a dimensional metrology microscope with a two-dimensional stage (X and Y), an eyepiece and objective lenses. The "MicroscopeType" base type is "UniversalDeviceType" and is illustrated in the following figure.

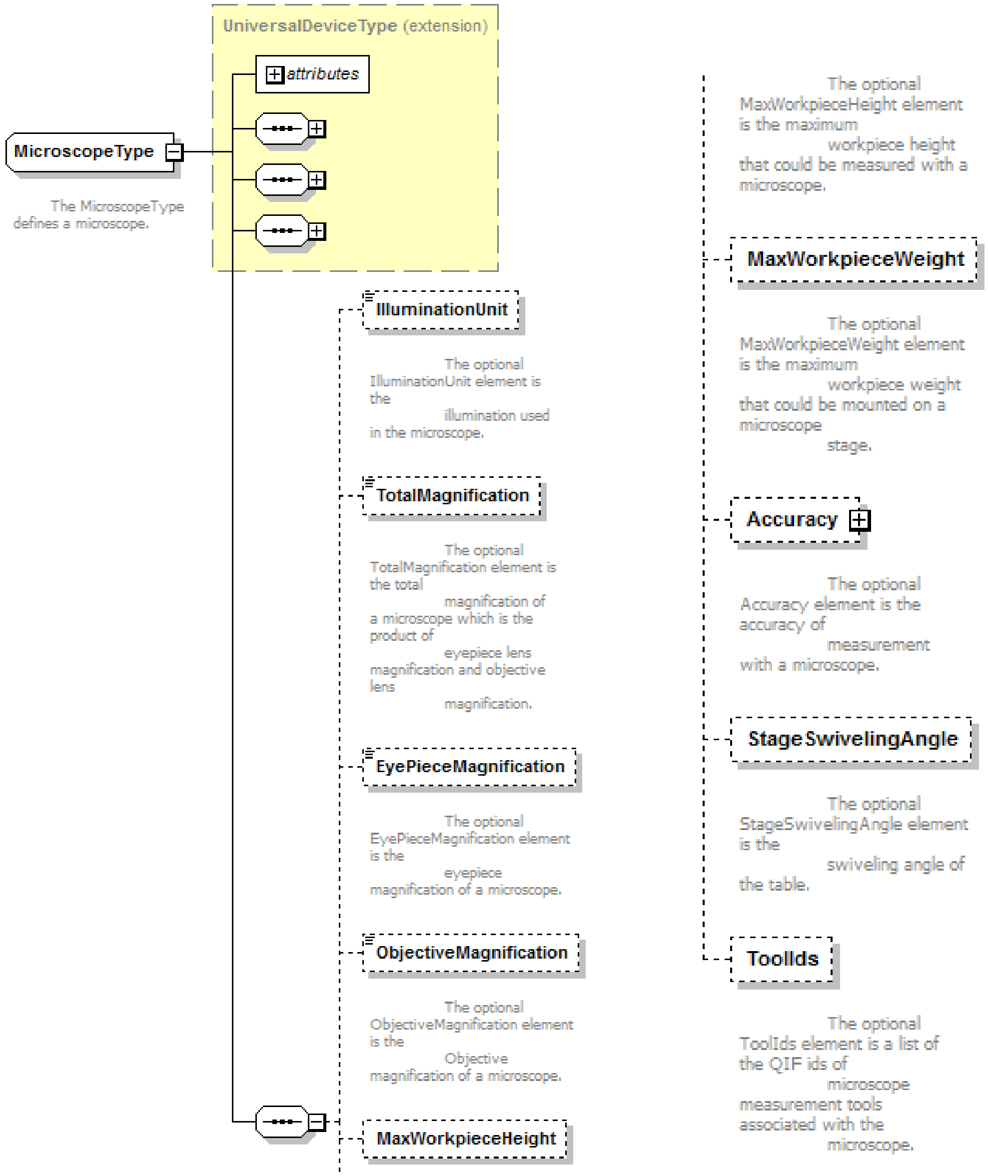


Figure 29 MicroscopeType complex type definition in V2.1

Optical Comparator: This is an optical device with a two-dimensional stage that uses lenses to magnify the profile of objects and projects the image on the screen for measurements. Important parameters in creating a measurement plan are collected in “OpticalComparatorType” complex type and illustrated in figure 30.

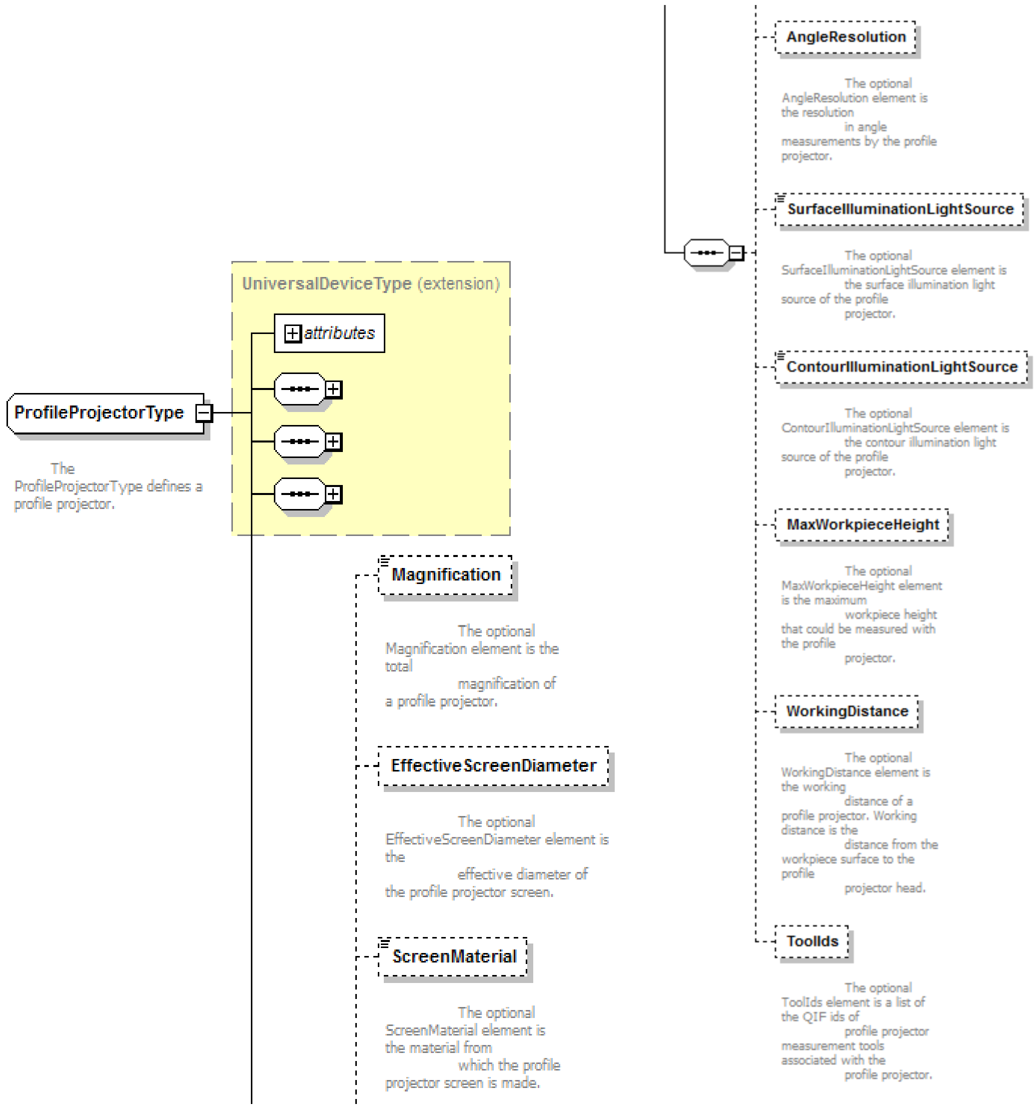


Figure 30 ProfileProjectorType complex type definition in V2.1

Theodolite: “TheodoliteType” complex type defines a theodolite measurements device. Theodolite is an optical measuring device that uses laser as a light source and its optical properties are addressed in its complex type. The measurement system uses two or more theodolites and calculates the distance using triangulation methods. The modern theodolites may have a CCD camera which is an optional element in the schema (most elements are optional in the schema). “TheodoliteType” is illustrated in figure 31.

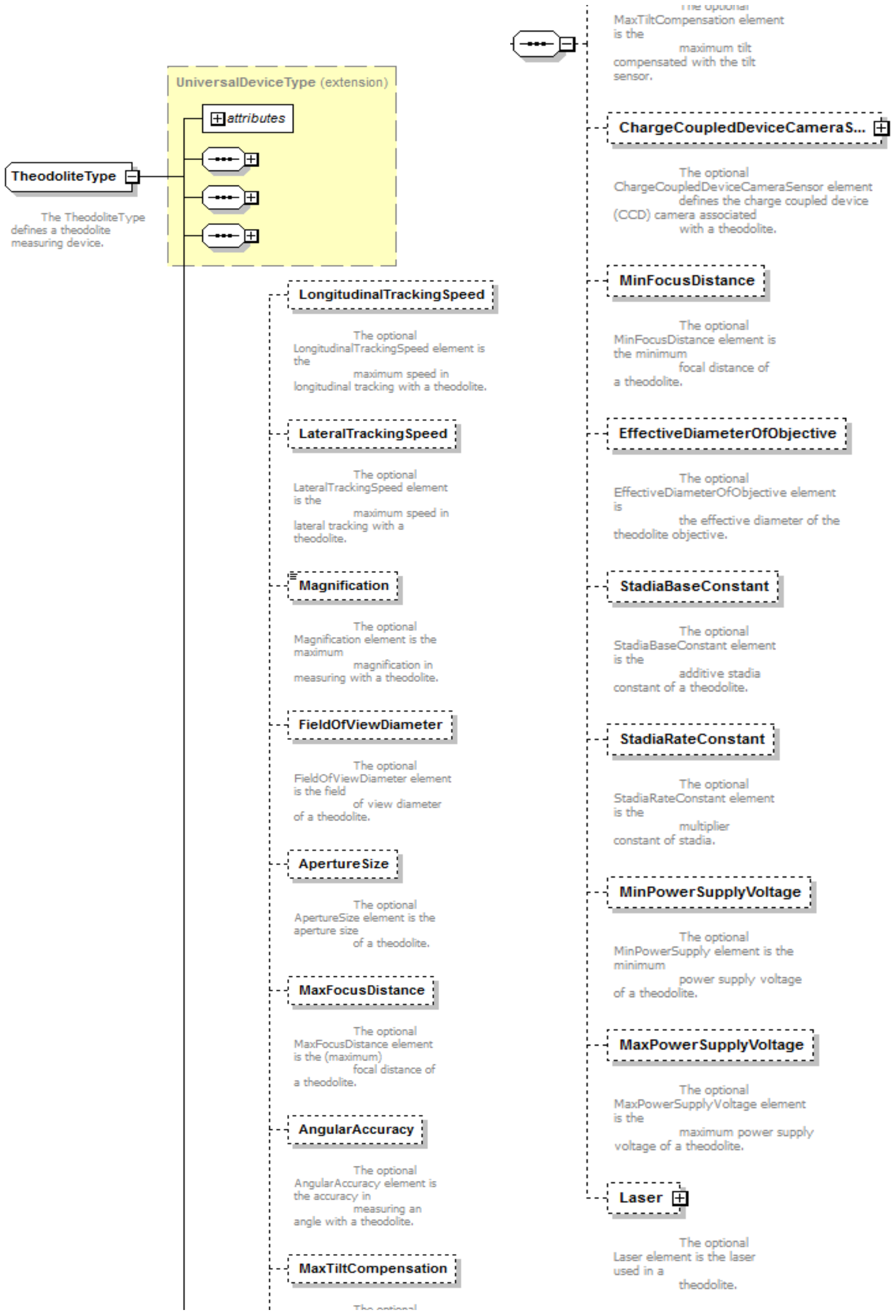


Figure 31 TheodoliteType complex type definition in V2.1

Universal Length Measuring Machine (ULM): ULM is a machine for measuring internal and external lengths, especially for different gauge calibrations. The device typically has a stage for mounting the parts and two knobs to move the part as wished.

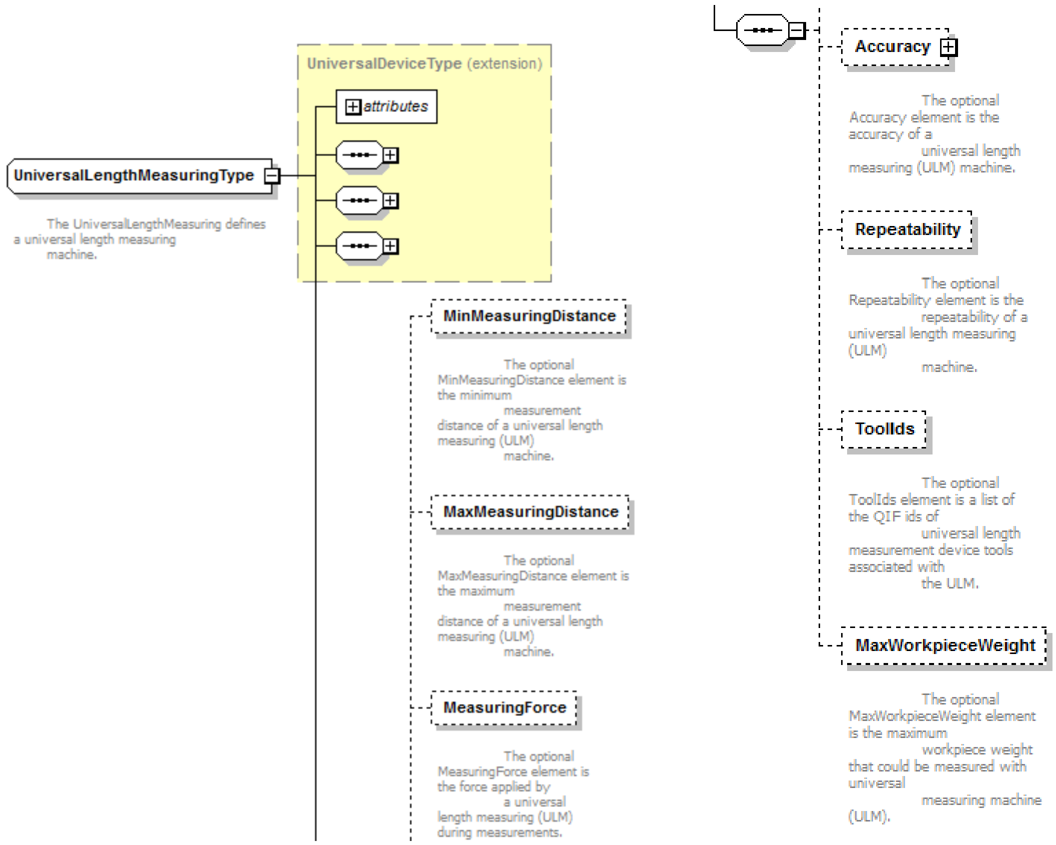


Figure 32 UniversalLengthMeasuringType complex type definition in V2.1

Sensors and Tools Complex Types: All the sensors are derived from "SensorType" which is a base type for all sensors. It is expected that any sensor in the QIF Measurement Resources schema inherit the characteristics of SensorType complex type (such as repeatability, sensitivity, etc.). For instance, confocal chromatic sensor or draw wire sensors all have the aforementioned elements of the SensorType and SensorType inherits (has) all the elements of MeasurementResourceBaseType complex type.

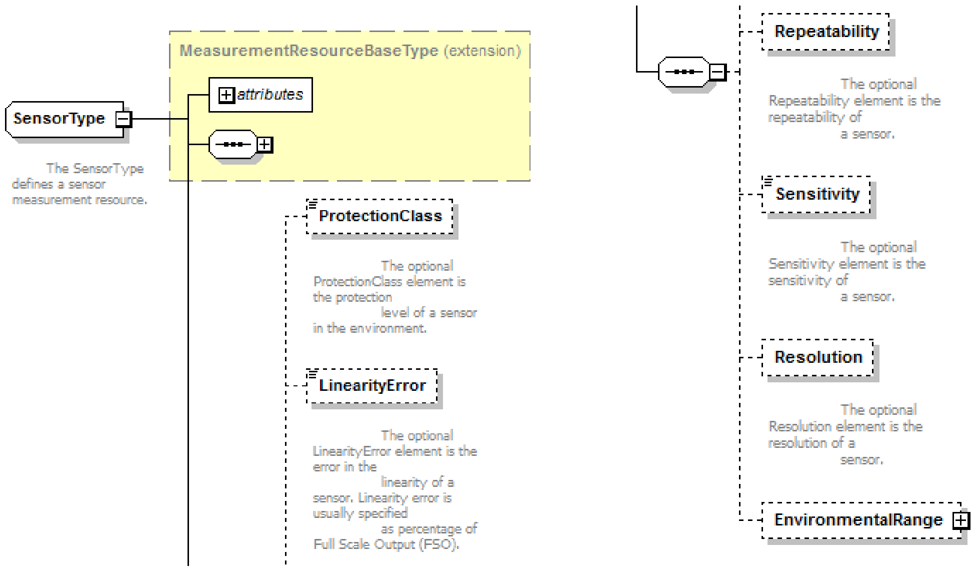


Figure 33 SensorType definition in QIF Measurement Resources

Charged Coupled Device (CCD) Camera: CCD cameras are one of most common detectors used in modern measurement devices. This sensor is made of three conductive strips, which are called gates, a silicon dioxide layer, n-channel layer, and p-type silicon layer. Any positive voltage applied to gates creates potential wells beneath the gates and the incident light is absorbed through accumulating electrons in the potential well. The amount of accumulated electrons is proportional to the intensity of incident light.

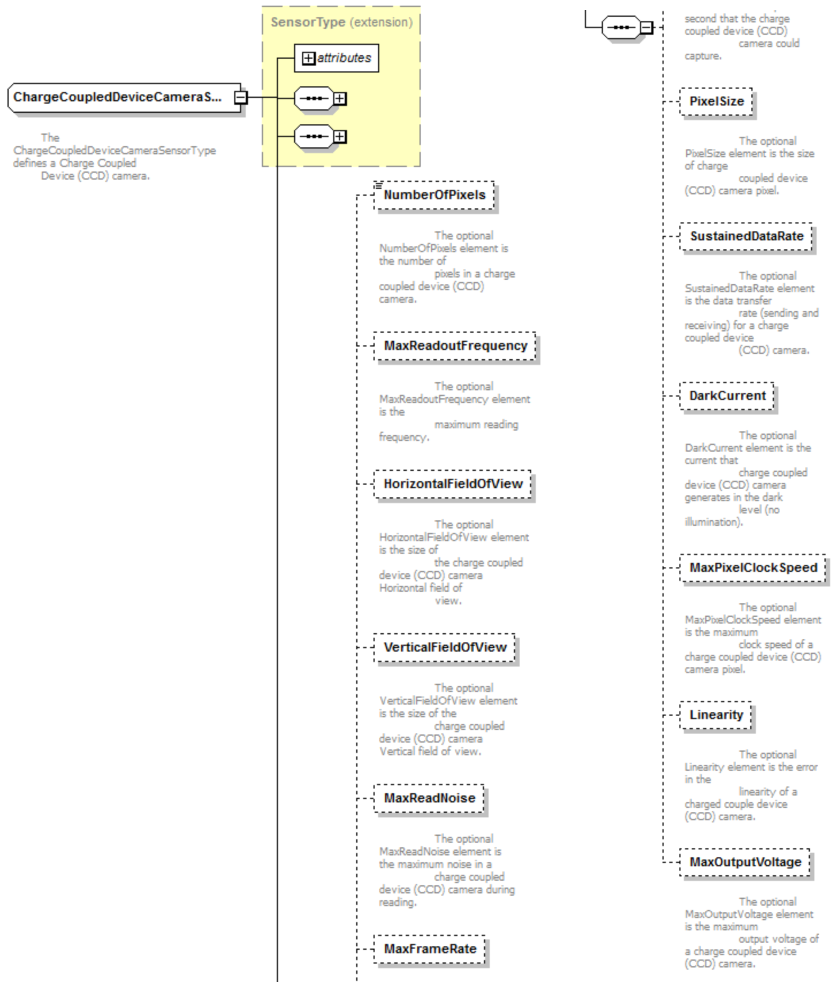


Figure 34 ChargeCoupledDevice Camera in QIF Measurement Resources V2.1

Capacitive Sensor: Capacitive sensors are very common for non-contacts displacement measurements especially for CMM and machine tool metrology applications. The principle is based on the measurements of capacity of a capacitor that changes according to the displacement. Important parameters to create a measurement plan using a capacitive sensor are illustrated in the following figure.

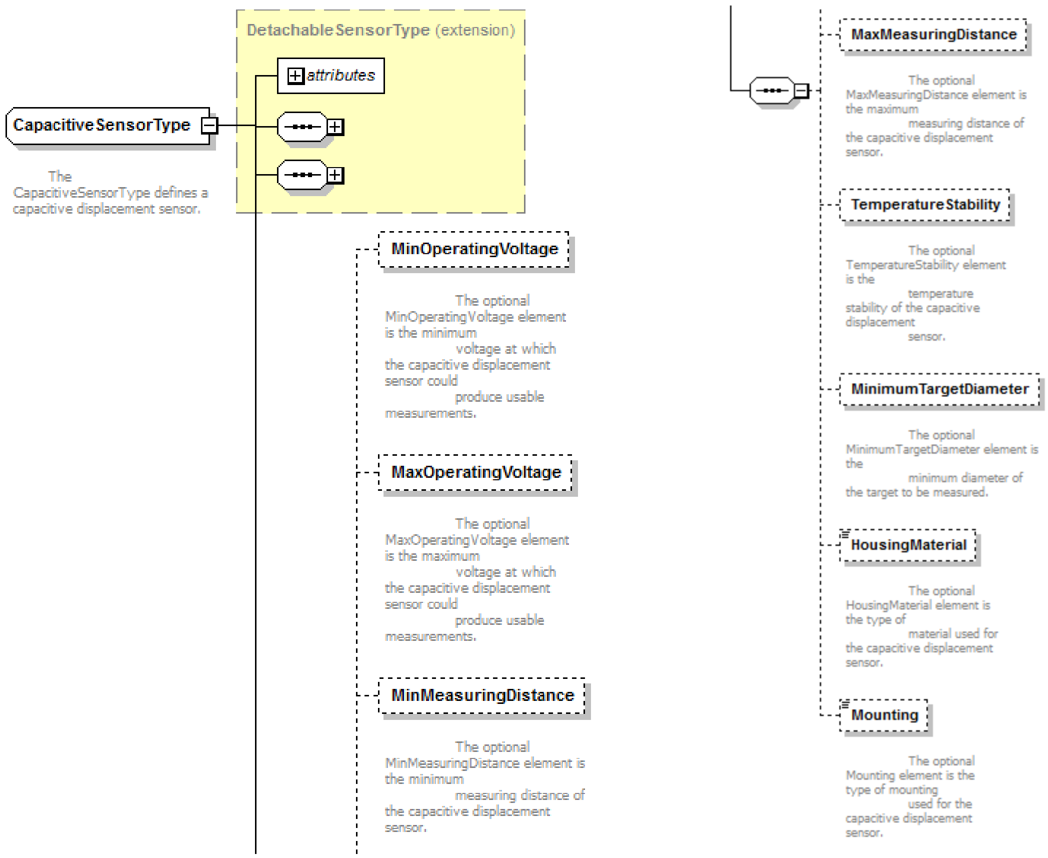


Figure 35 CapacitiveSensorType complex type defined in V2.1

Confocal Chromatic Sensor: This sensor calculates the distance based on optical chromatic aberration of different wavelengths and it is mounted on many modern CMM turrets. This sensor can even be used to read transparent coating (“MultiPeakMeasurement” element) and optical properties of the surface plays an important role in the measurement process. There are some limitations for this sensor like maximum slope of the surface (“MaxTilt” element) and all the important parameters about this sensor are illustrated in the following figure.

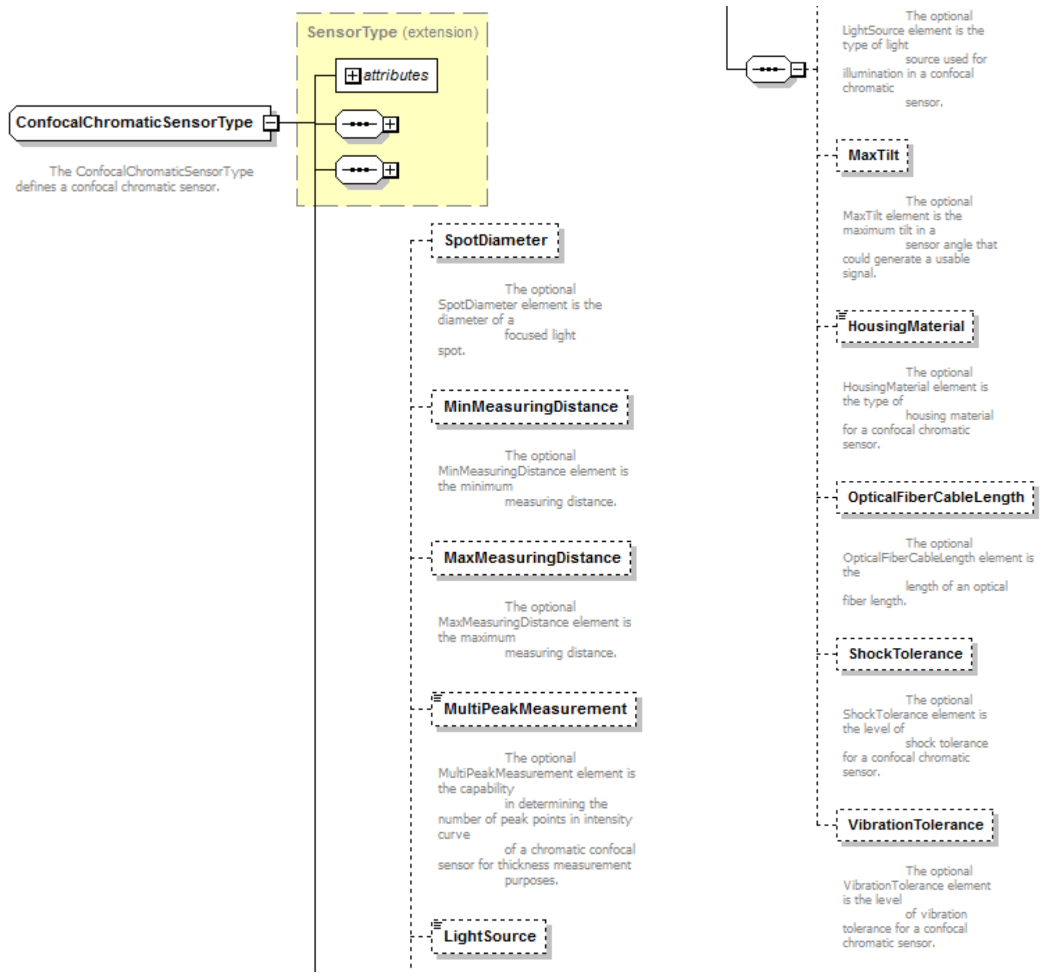


Figure 36 ConfocalChromaticSensorType complex type definition in V2.1

Linear Variable Displacement Transducer (LVDT) Sensor: This is a very common sensor in many high precision applications. This sensor has three solenoidal coils around a core that generates a linear signal for displacement measurements. By moving the core and driving alternating current (AC) through primary coil, induction creates voltage difference between secondary and tertiary coils, which can be used for displacement measurements.

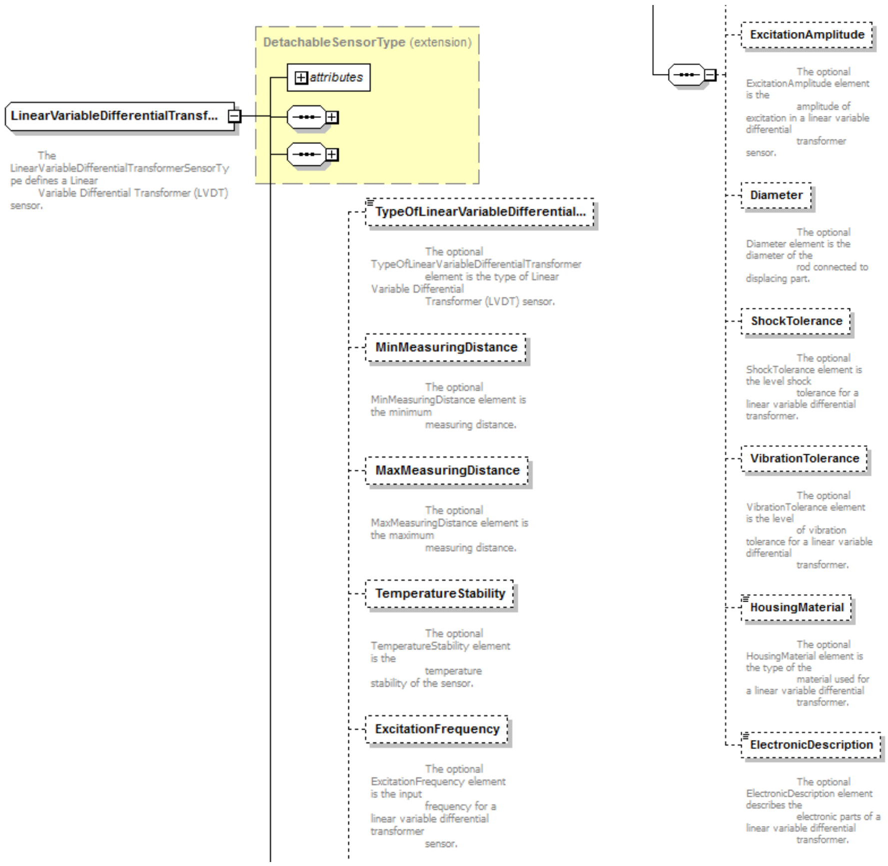


Figure 37 LinearVariableTransducerSensorType complex type defined in V2.1

Differential Variable Reluctance Transducer (DVRT) Sensor: DVRT sensor has a very similar configuration to LVDT except that it does not have the primary coil and it is a half bridge.

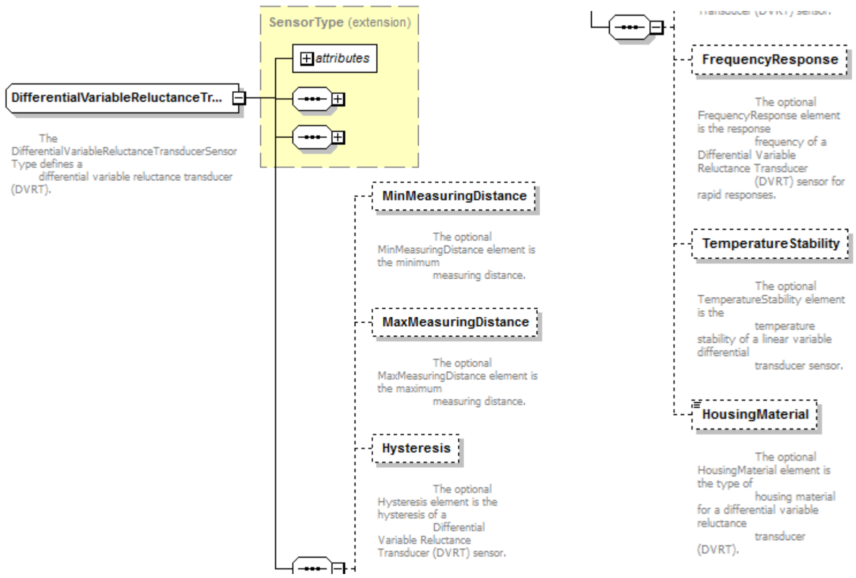


Figure 38 DifferentialVariableReluctanceTransducer complex type definition in V2.1

Draw Wire Sensor: Draw wire sensor is based on a wire displacement that its one head is connected to the moving device and the other head winded on a capstan and connected to a potentiometer or encoder. By displacing the object connected to a wire, the potentiometer or encoder rotates and generates an electric signal that can be used to calculate the displacement and dimensions.

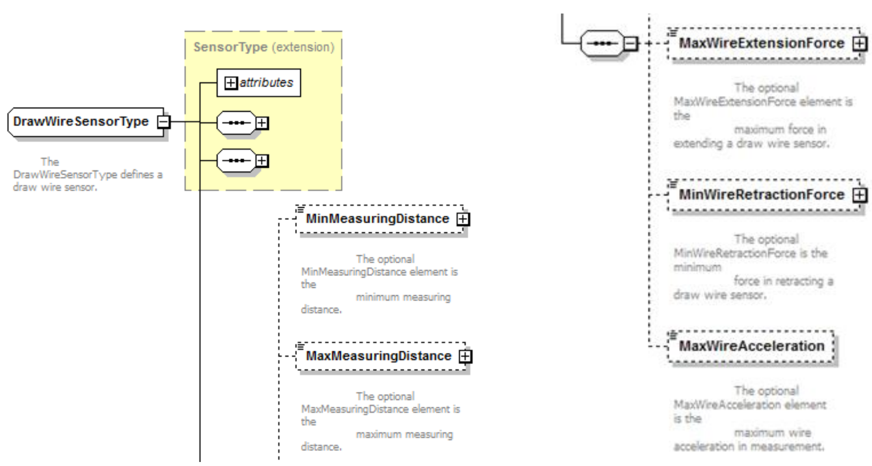


Figure 39 DrawWireSensorType complex type defined in v2.1

Eddy Current Sensor: Eddy current sensors measures the displacement or distance based on Eddy current generation. Eddy current is created when a coil that has AC current approaches a conductive surface. This AC current and alternating magnetic field creates AC currents on the part surface. These AC currents generate opposite magnetic fields that is proportional to the distance between the probe and part.

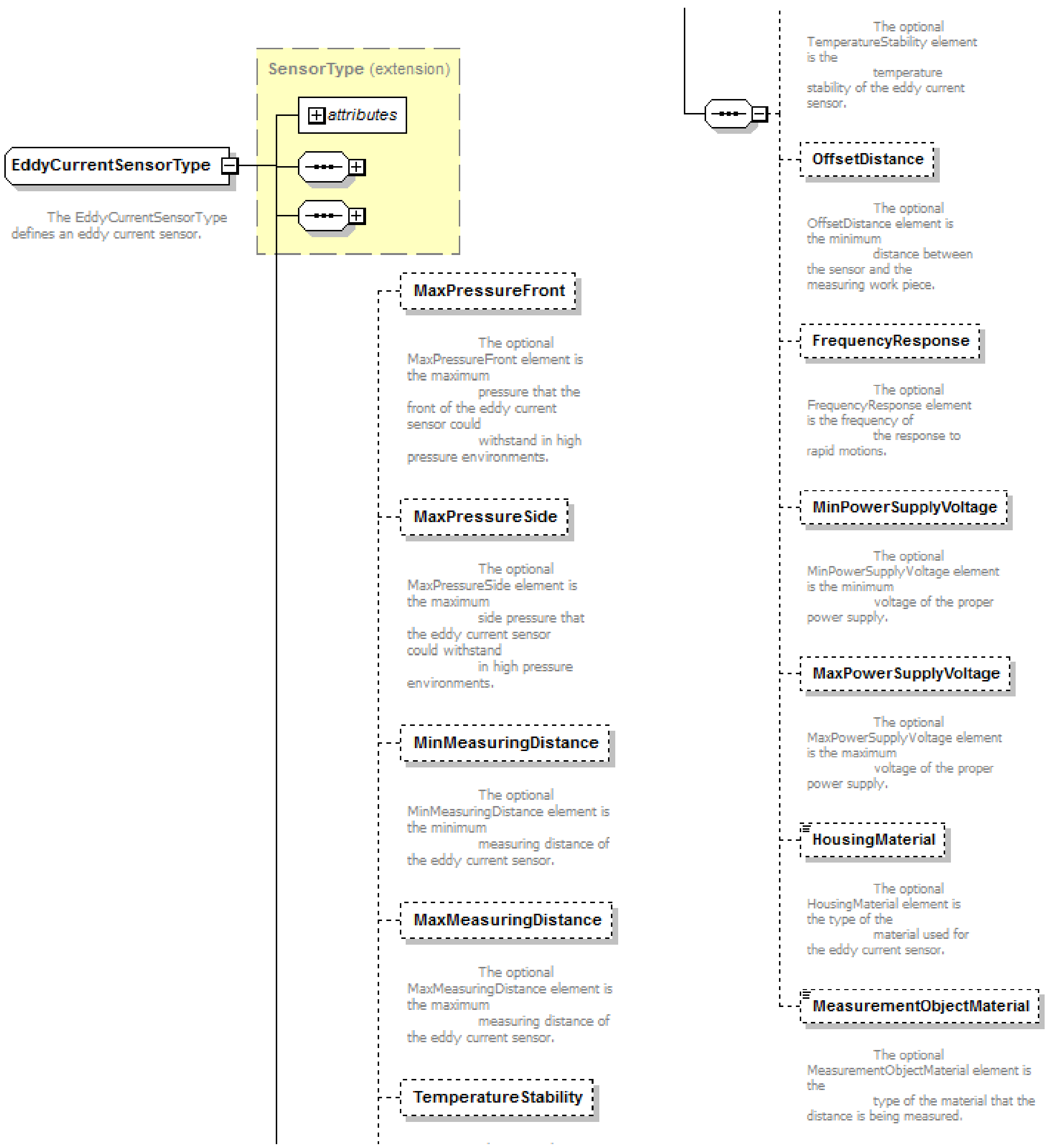


Figure 40 DrawWireSensorType complex type defined in V2.1

Laser Triangulation Sensor: This sensor includes two main parts, laser as source, and a detector like CCD camera. The laser projects the beam to the surface of the parts and the detector measures the maximum intensity of the reflected light in two different positions and then calculates the displacement. Optical properties of the part surface plays an important role in sensor performance and specific designs need to be considered for the specular surface applications.

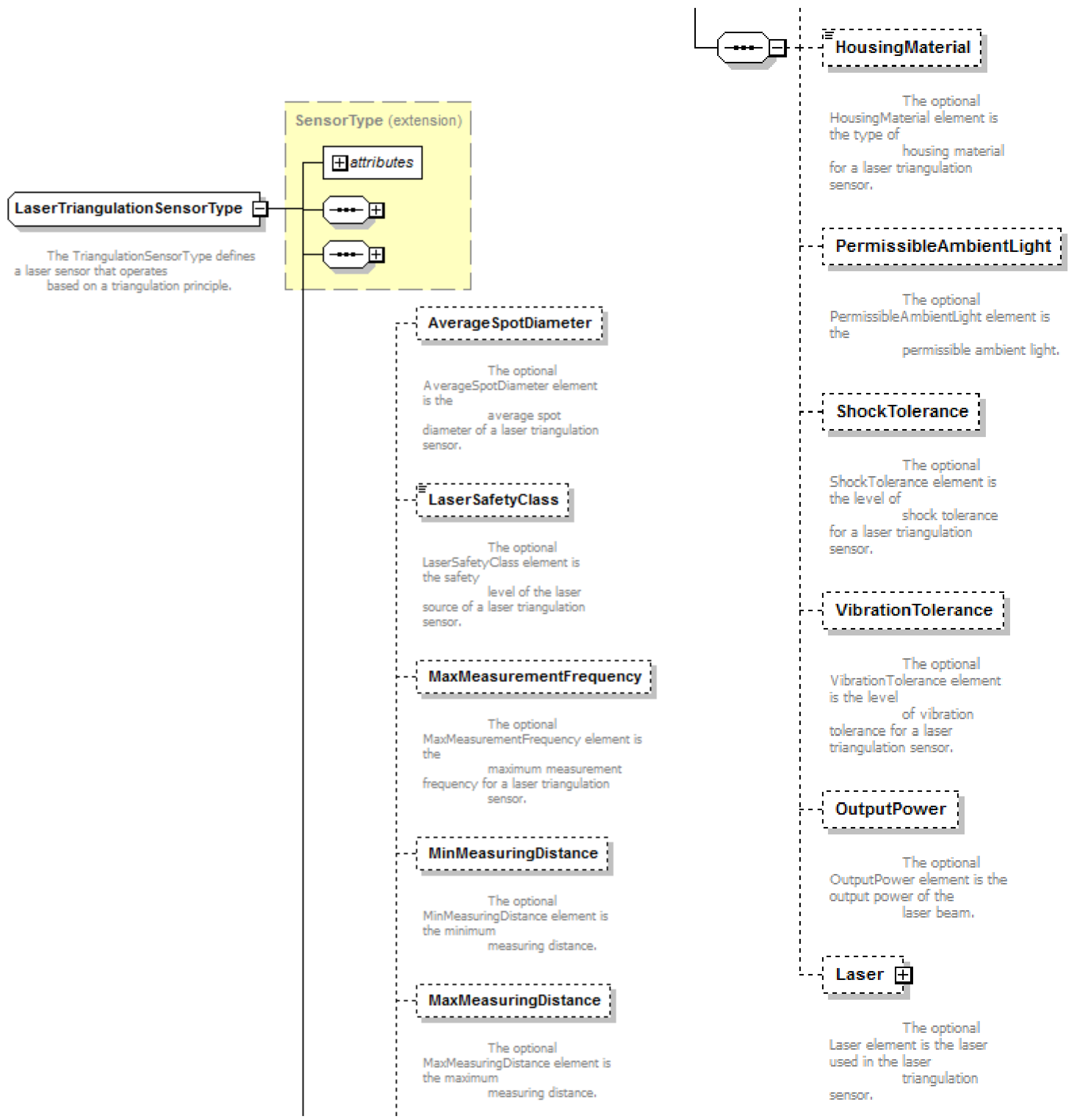


Figure 41 LaserTriangulationSensorType complex type defined in V2.1

Magneto-Inductive Sensor: Magneto-Inductive sensors use the concept of magnetic and inductive sensors. The magnet is attached to the moving object and the primary coil is supplied with the AC current that creates Eddy currents. The generated electric signal is linearly proportional to the distance between the magnet and the sensor.

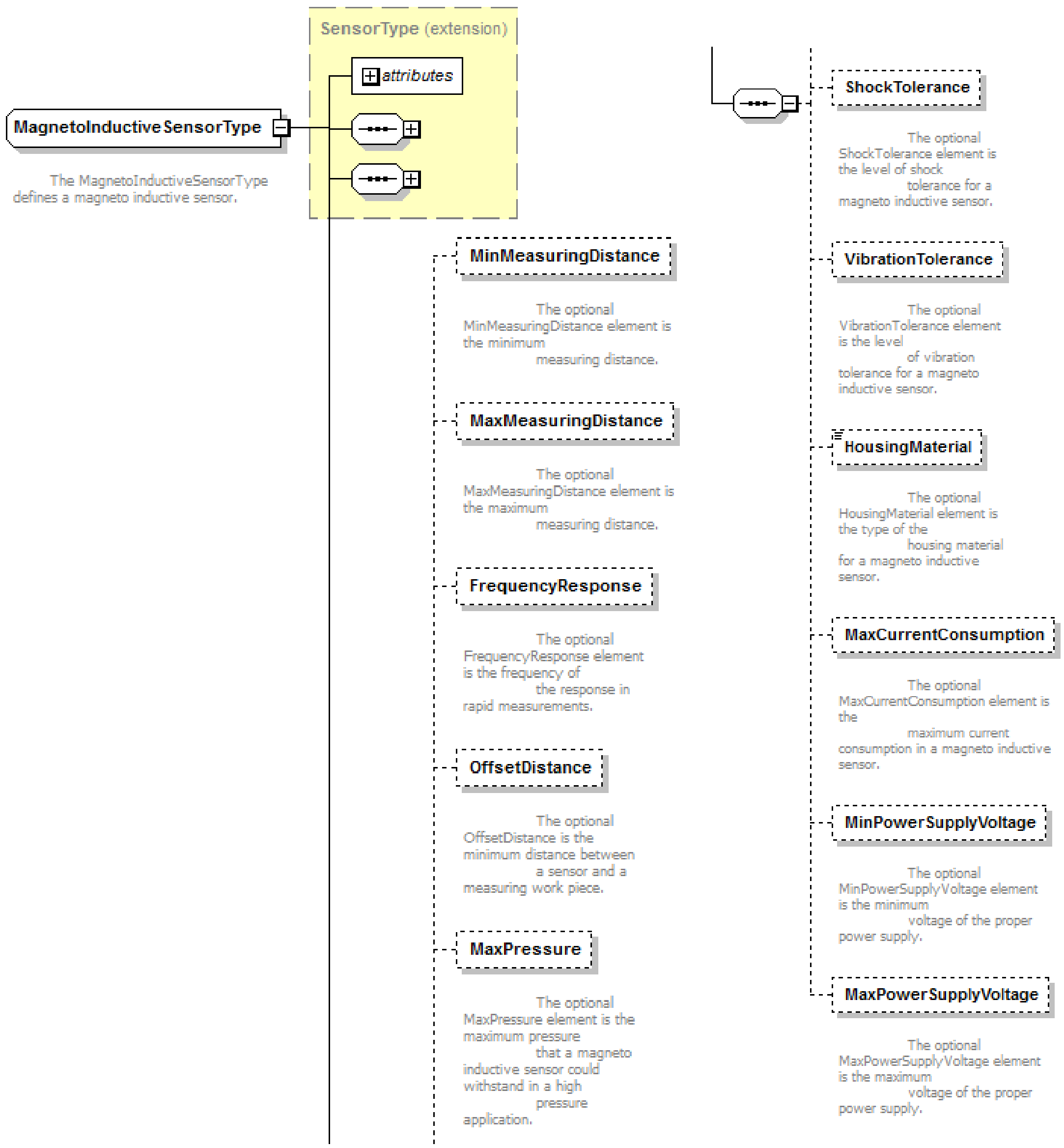


Figure 42 MagnetoInductiveSensorType complex type defined in V2.1

Structured Light Sensor: The Measurement principle in this sensor is based on projecting a pattern of light on an object (measurands) and measuring the deformation of that pattern. The light pattern is created by laser interference or a projector and the camera(s) measures the bended light pattern and calculates the coordinates of the points on the object by calculating the deformation of the pattern. The structured light sensor is one of the common accessories mounted on many modern CMM tool racks (Hocken and Pereira 2011).

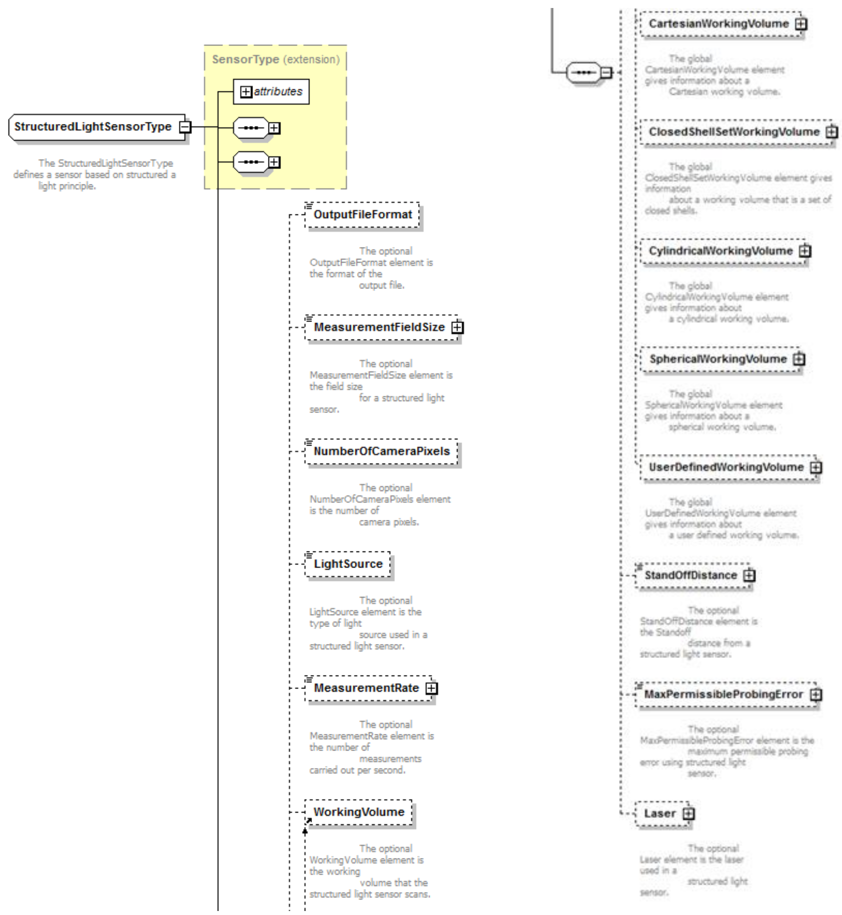


Figure 43 StructuredLightSensorType hierarchy in QIF Measurement Resources

Tactile Probe Sensor: The tactile probe is the most common probe mounted on most of the CMM tools and it is typically a tip (sphere shape or other kinds of shape) attached to a stem. Tactile probe can have one tip ("SimpleTactileProbeSensorType") or a combination of tips ("ComplexTactileProbeSensorType") and these are defined as two separate individual complex types. The only major difference between these two complex types is the "TipEndLocation" element that allows definition of the coordinates of multiple tip locations for "ComplexTactileProbeSensorType".

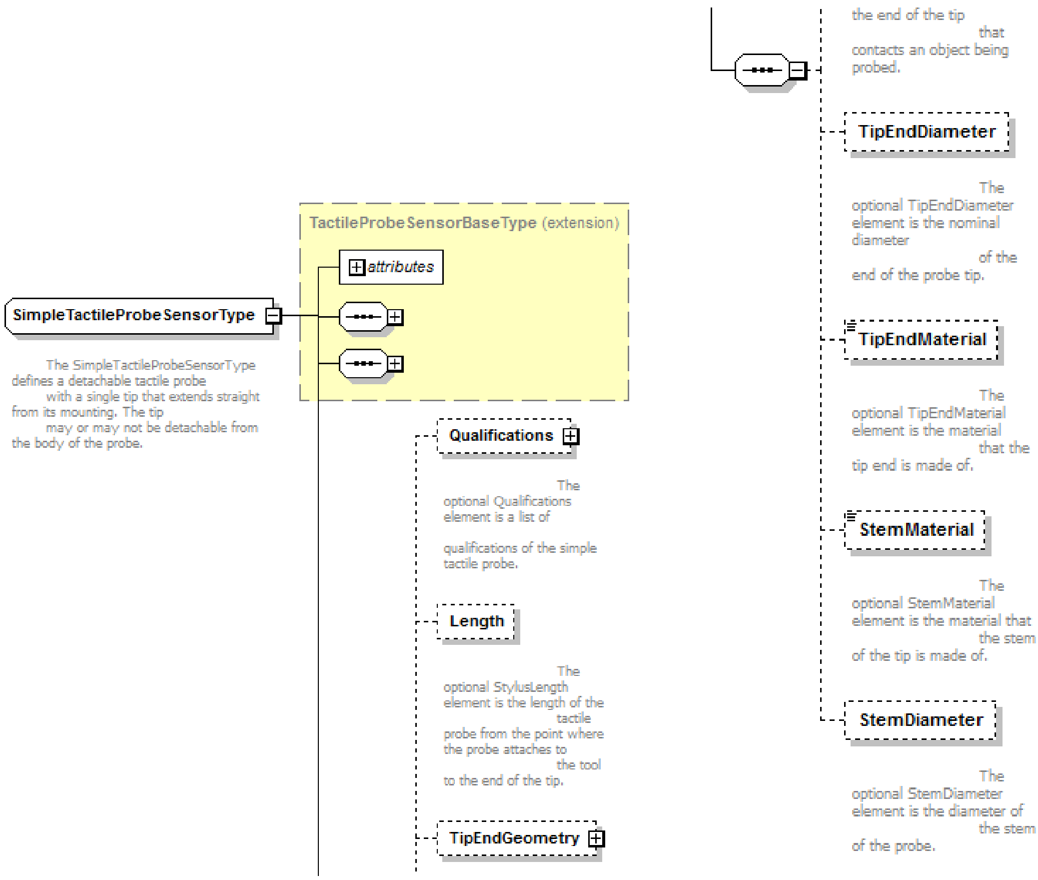


Figure 44 SimpleTactileProbeSensorType complex type defined in V2.1

Ultrasonic Sensor: The ultrasonic sensor uses ultrasonic waves to measure the displacement based on the time of flight method. The sensor sends an ultrasonic wave and detects the reflected wave and then by reporting the time of flight and multiplying it by the wave speed, the displacement is calculated. Ultrasonic sensors are so sensitive to environmental changes.

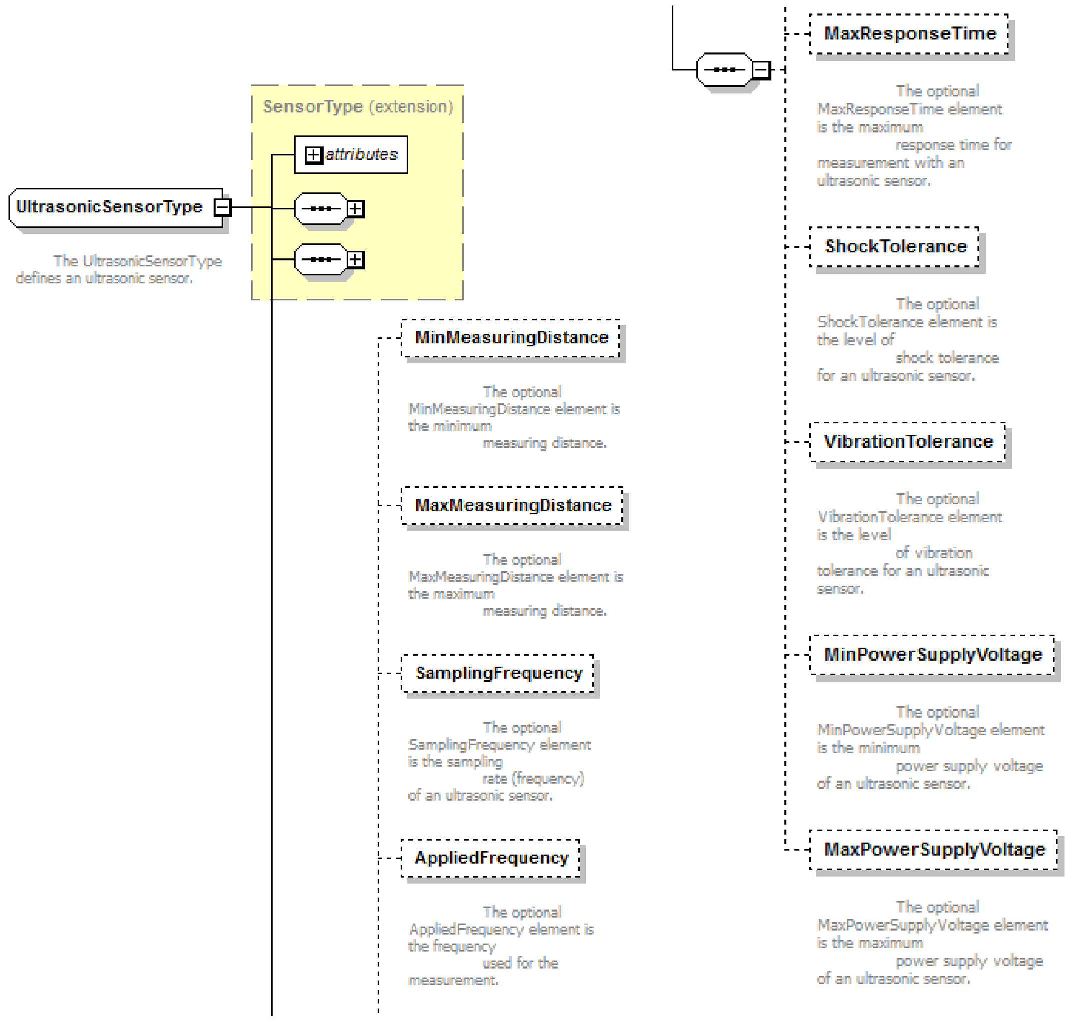


Figure 45 UltrasonicSensorType complex type defined in V2.1

Auxiliary Complex types: Many complex types are defined as auxiliary types that are used in definition of measurement devices or sensors. Some of these complex types are "LaserType", "StiffnessType", "MeasurementDeviceScaleType", etc. Full list of developed auxiliary types are illustrated in the appendix A.

Chapter 5: Developing a translator for STEP to QIF

Preface: This chapter focuses on explaining the architecture of a translator that converts QIF (*.xml extension) files to STEP (*.stp extension) files and vice versa. STEP and QIF both are neutral data formats. One (STEP) is ISO accredited standard and the other one ANSI (QIF). STEP is the most comprehensive standard supporting product lifecycle and the main data format competing with QIF that especially covers metrology framework. A software that does not read *.qif extension, but reads STEP files can benefit from this translator. For the case studies, Calypso from Carl Zeiss and Creo parametric student version from PTC have been chosen for representing translated files (for *.qif to *.stp or *.step).

Any (proprietary) software translates data format to its own proprietary format for representation or any other purposes. In this case, Calypso or Creo can translate the *.qif extension to their own proprietary format but (Calypso does not read *.qif extension files) this requires having access to the copy righted software (Calypso or Creo). Finding a solution for this problem would be a translator that reads and writes *.qif and *.stp (or *.step) extensions, and translates one format to the other one. A typical STEP and QIF file have three types of information: topology, geometry, and PMI. Topology is more about defining how the vertices of a part are connected to each other or part edges, but geometry is more about defining a geometry of a part without talking about edges or vertices of the part (more mathematics). PMI includes GD&T, welding signs, 3D PMI annotation, material properties, and surface finish. GD&T is the main part of PMI and is more emphasized in this translator. Geometry and Topology are modeled using boundary representation (B-Rep) techniques. PMI (or GD&T) in QIF and STEP is mostly ASME Y14.5 and ISO GD&T standards (ISO 1101, etc.). Figure 46 illustrates different ways to read and write neutral data formats (like *.qif and *.stp) or different proprietary formats (like *.sldprt or *.prt).

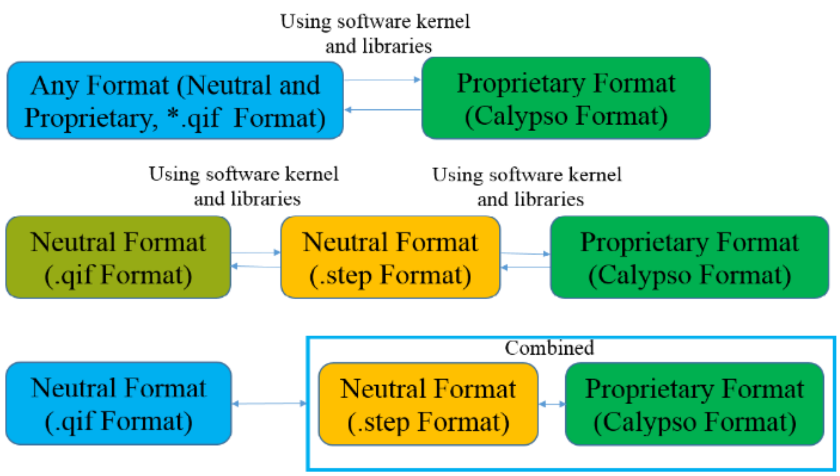


Figure 46 Different ways to read a *.qif extension file

Data Format in STEP

Hierarchy, structure, and relationships can be defined in EXPRESS schema and these concepts are enforced to the corresponding part 21 file. Entities in EXPRESS schema are defined by key word ENTITY and the elements of that entity are defined between ENTITY and END_ENTITY key words. Looking at Table 1, a cylindrical feature is defined as entity #1 with 10.5 as radius and entity #2 as the center of cylindrical feature (center is defined as the coordinate of X, Y, and Z).

Table 7. EXPRESS Schema and Part 21 File Comparison

SCHEMA Contact_schema	#1= CylindricalFeature(10.5, #2)

Table 7, continued

ENTITY CylindricalFeature Radius : Real; Center : Center; END_ENTITY ENTITY Center coordinates : LIST [1:3] END_ENTITY END_SCHEMA	#2= Center(1.5, 6.3, 5.1)
Fragment of an EXPRESS schema (a)	Fragment of a part 21 file (b)

Data Format in QIF

Data in QIF are defined in XML format, a hierarchy based format with a start and end tag that the information is stored between start and end tag. For instance, for a cylindrical feature, there is start tag (<CylindricalFeature>) and end tag (</CylindricalFeature>) and other elements (other start and end tags) are defined between these two tags.

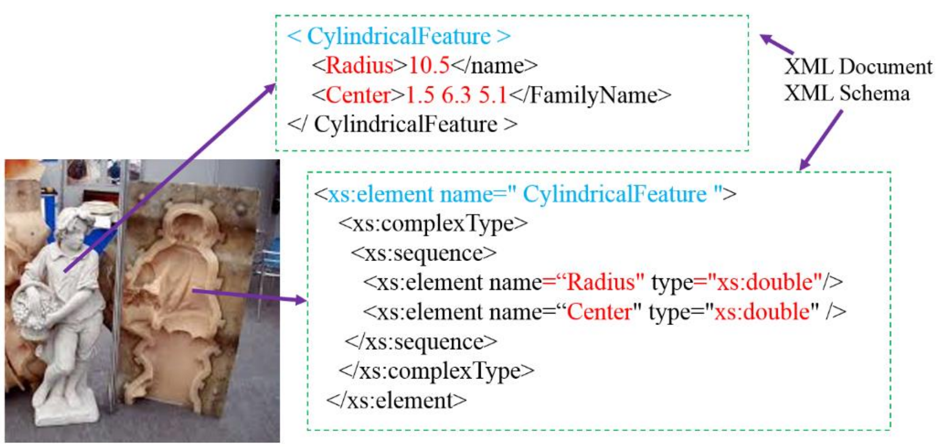


Figure 47 XML and XSD Comparison

Methodology Used in the Translator

This translator is developed on Java ECLIPSE platform (ECLIPSE Kepler) in Java environment by adding java binding of XML schemas developed by QIF project and java libraries generated by EXPRESS java compiler. EXPRESS java compiler is a patch developed by LKSoftWare GmbH that can be integrated with ECLIPSE to read and write STEP files or any other EXPRESS schema file. The java binding of XML schemas (QIF) are generated by Java Architecture XML Binding (JAXB) and then was modified manually (figure 48). JAXB and EXPRESS compiler generate a class for each complex type or entity of QIF and STEP respectively.

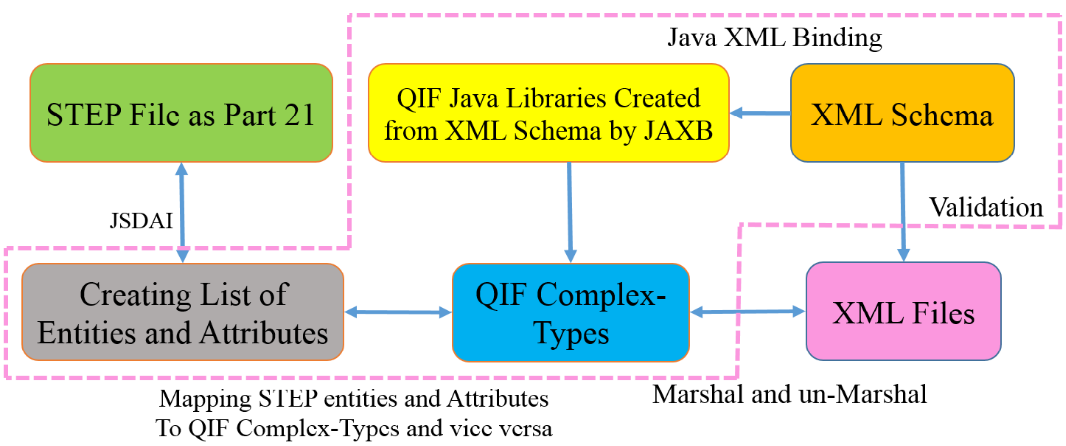


Figure 48 Methodology Used for QIF to STEP and Vice Versa Translation

QIF classifies geometry and topology into six types including point, 2D curves, 3D curves, surfaces, mesh curve, and mesh surface. There is no equivalent entity for mesh curve and mesh surface as meshing is not supported in STEP AP203 Ed2. For instance, figure 52 illustrates how a circle can be defined in STEP and QIF (The ArcCircular13 complex type is compared to “circle” entity). In both formats, the same ID, radius, center, reference direction and normal direction are defined in different ways. The other geometry and topology QIF complex types are correlated with STEP entities according to Tables 8 and 9. In addition to topology and geometry, STEP and QIF can be compared in terms of PMI (table 9).

QIF

```
<ArcCircular13 id="220">
 <ArcCircular13Core domain="0 3.14">
  <Radius>12.5</Radius>
  <Center>1.3 5.8 9.5</Center>
  <DirBeg>1.0 0.0 0.0</DirBeg>
  <Normal>0.0 0.0 1.0</Normal>
 </ArcCircular13Core>
</ArcCircular13>
```

STEP

```
#2232=CIRCLE('220',#2233,12.5);
#2233=AXIS2_PLACEMENT_3D($,#2234,#2236,#2235);
#2234=CARTESIAN_POINT($,(1.3, 5.8, 9.5));
#2235=DIRECTION($,(0.0, 0.0, 1.0));
#2236=DIRECTION($,(1.0, 0.0, 0.0));
```

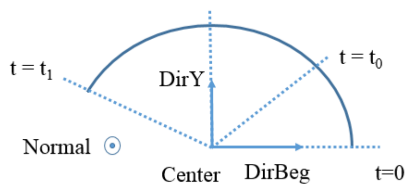


Figure 49 QIF and STEP Comparison for a 2D Circle

Figures 51 and 52 illustrate two case studies with QIF formats that have been translated to STEP format along with their Q-DAS ASCII file and opened in Calypso or Creo Parametric. There are a few challenges in this translation as following:

Splines Translation Techniques

One of challenging parts in this translation is Spline13 complex type defined by QIF that needs to be translated to b_spline_curve_with_knots entity of STEP. Spline13 is actually piecewise polynomials that can be defined by equation 1. These piecewise polynomials are connected in the knots and have first order continuity (first order derivatives are equal). To do this translation, two points in the middle of each polynomial are found (four points for each polynomial) and then a b-spline curve is fitted through the points from polynomials.

$$Spline13 = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{Degree,\ p} \left(C_i^p\right)_x \left(t_p\right)^i \\ \sum_{i=0}^{Degree,\ p} \left(C_i^p\right)_y \left(t_p\right)^i \\ \sum_{i=0}^{Degree,\ p} \left(C_i^p\right)_z \left(t_p\right)^i \end{bmatrix} \quad (1)$$

$$t_p \in \begin{Bmatrix} \left[Knot_p\,, Knot_{p+1}\right] & Normalized = False \\ [0-1] & Normallized = True \end{Bmatrix} \quad (2)$$

$$t_p = \begin{Bmatrix} \left[t - Knot_p\right] & Normalized = False \\ \left[t - Knot_p\,/Knot_{p+1} - Knot_p\right] & Normallized = True \end{Bmatrix} \quad (3)$$

The $\left(C_i^p\right) and\ Knot_p$ are the coefficients and the knots of splines respectively. After plugging different values of knots and coefficients into spline, equations and coordinates of the points are retrieved.

$C(t) = \sum_{i=0} N_{i,3}\,(t)\,P_i$ (Cubic b-spline equation) (4)

To interpolate n points, n+6 knots and n+2, control points are required. In order to pass the first and last data points, the first and last knots are repeated four times and then uniformly distributed from zero to n.

$N_{i,3}\,(t)\,P_i + N_{i+1,3}\,(t)\,P_{i+1} + N_{i+2,3}\,(t)\,P_{i+2} = D_i$ (5)

By using the above equation for different "i" iterations:

$P_0 = D_0$ $\quad \frac{1}{4}P_1 + \frac{7}{12}P_2 + \frac{1}{6}P_3 = \mathrm{D}_1$

$\frac{1}{6}\mathrm{P}_2 + \frac{2}{3}\mathrm{P}_3 + \frac{1}{6}\mathrm{P}_4 = \mathrm{D}_2$ $\quad \frac{1}{6}\mathrm{P}_{n-2} + \frac{2}{3}\mathrm{P}_{n-1} + \frac{1}{6}\mathrm{P}_n = \mathrm{D}_{n-2}$……..

$\ldots\frac{1}{6}\mathrm{P}_{n-1} + \frac{7}{12}\mathrm{P}_n + \frac{1}{4}\mathrm{P}_{n+1} = \mathrm{D}_{n-1}$ $\quad \mathrm{P}_{n+2} = \mathrm{D}_n$ (6)

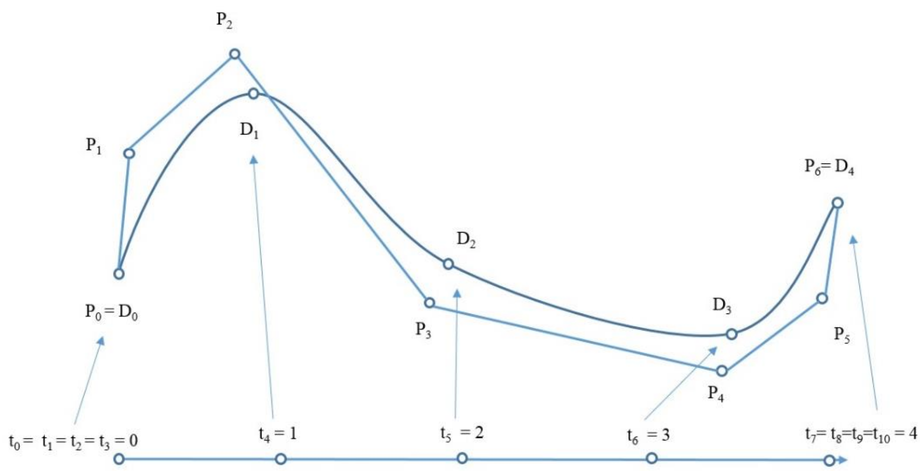


Figure 50 Illustration of knots and their value along with their multiplicity

While having (n-1) set of equations (ignoring the first and last equations) for (n+1) points, two more equations are required to solve this set of equations. There could be different type of extra conditions such as setting the second derivative of the curve at start and end point to be zero or setting the first two points ($P_0=P_1$) and the last two points ($P_n=P_{n-1}$) to be equal. For the case studies, the latter extra conditions are applied (figure 50). By forming the following form of matrix system, the coordinates of control points can be acquired.

$$\begin{bmatrix} 1 & \ldots.. & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ \ldots. \\ \ldots. \\ P_{n-2} \\ P_{n-1} \\ n \end{bmatrix} = \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ \ldots. \\ \ldots.. \\ D_{n-2} \\ D_{n-1} \\ D_n \end{bmatrix} \qquad (7)$$

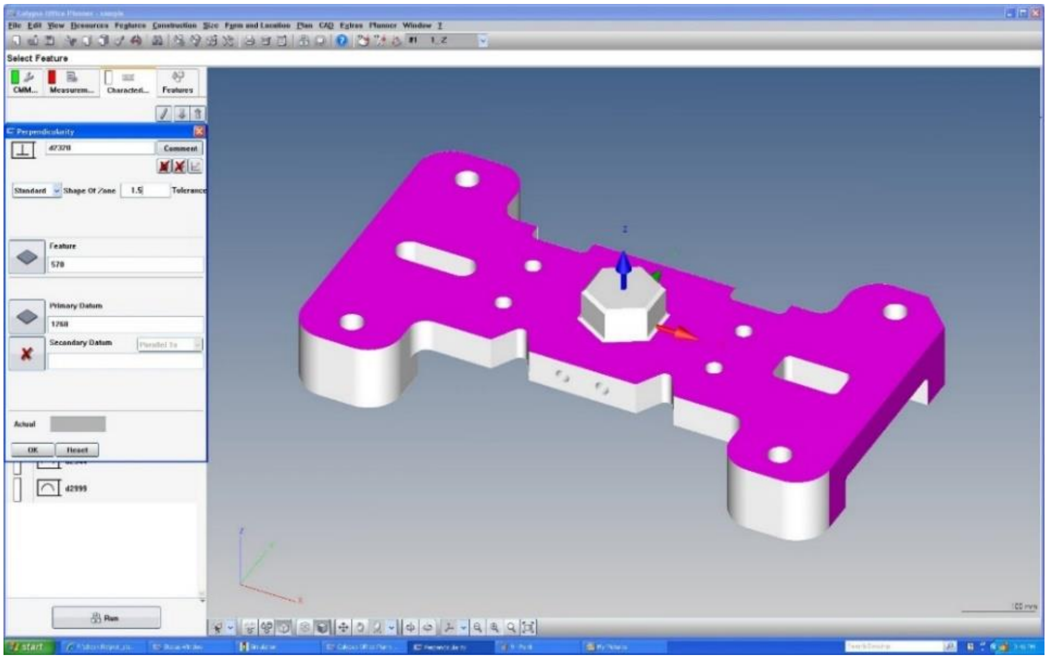


Figure 49 First Case Study of QIF to STEP

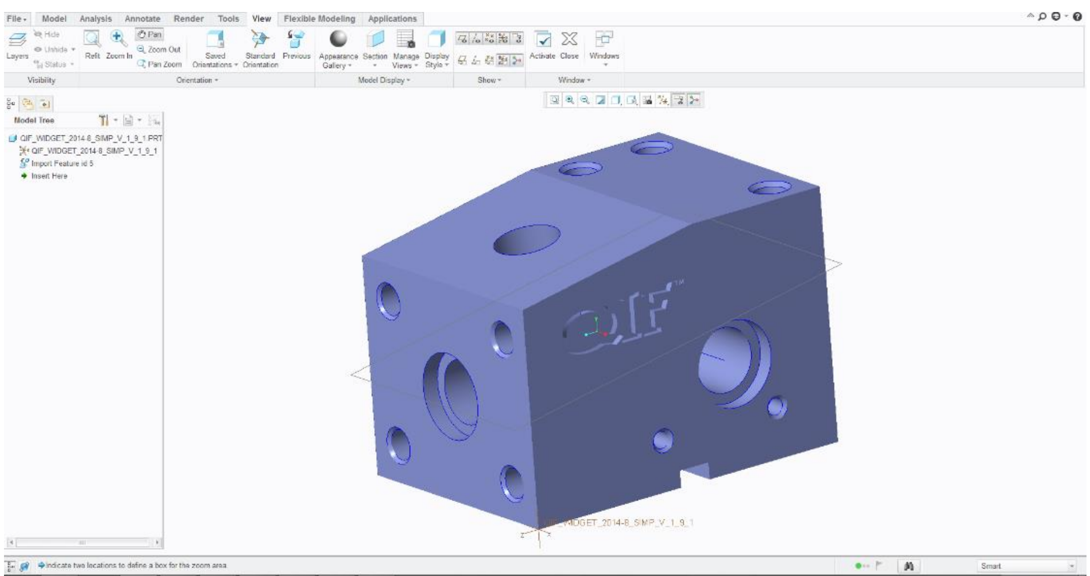


Figure 50 Second Case Study of QIF to STEP

Encrypted Control Points of QIF Files

QIF allows encrypting data (for the purposing of contracting file size) including coordinates of different points, control points, etc., into binary format using RFC 2045 alphabet. For this purpose, the control points are first encoded into IEEE 754 binary 64 double precision floating-point format (little –endian that reads bits from right to left) and then these bits are encoded by using base 64 alphabet in RFC 2045. RFC 2045 alphabet is illustrated in the following table.

Table 8 RFC 2045 Alphabet for Base 64

Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.	Value	Char.
0	A	8	I	16	Q	24	Y	32	G	40	O	48	W	56	4
1	B	9	J	17	R	25	Z	33	H	41	P	49	X	57	5
2	C	10	K	18	S	26	A	34	I	42	Q	50	Y	58	6
3	D	11	L	19	T	27	B	35	J	43	R	51	Z	59	7
4	E	12	M	20	U	28	C	36	K	44	S	52	0	60	8
5	F	13	N	21	V	29	D	37	L	45	T	53	1	61	9
6	G	14	O	22	W	30	E	38	M	46	U	54	2	62	+
7	H	15	P	23	X	31	F	39	n	47	v	55	3	63	/

Complex Entities in STEP

STEP allows combining different entities and creating complex entities (figure 59). This can create challenges in translaing as they may be different combiniation of entities for the same concept. To avoid this problem, it is important to follow the guidelines and recommended practices publishd by the standard committee (AP203 recommended practices).

PMI in QIF and STEP

PMI in QIF and STEP mostly includes GD&T and 3D annotations. Among many different GD&T standards, ASME Y14.5 and ISO GD&T standards are the main ones frequently used in industry. ASME Y14.5 has three major revisions, ASME Y14.5-1982, ASME Y14.5-1994, and ASME Y14.5-2009. Unlike ASME, ISO GD&T standards are not consolidated and different GD&T standards are published that form ISO GD&T standards (Table 8).

Table 8 Some of ISO GD&T Standards

Description	ISO Standard Number
Tolerances of Form, Orientation, Location, and Runout,	ISO 1101
Fundamental Tolerancing Principle	ISO 8015
Datums and Datum System for Geometrical Tolerances	ISO 5459
Position Tolerancing	ISO 5458
Least Material Requirement	ISO 2692

ISO and ASME tolerances are different and similar in many aspects and the differences and similarities should be discussed (and considered in translation) in four aspect of principles, interpretation, symbols, and terminology.

In terms of similarities, both standards support size, form, orientation, location, and runout tolerances. They both support material modifiers (maximum and minimum material condition and regardless of feature size), datum definition, datum target symbols and definition, free state, basic dimension, projected tolerance zone, and many other common symbols or concepts. In addition, ISO and ASME both define the same tolerance zone in most cases. For instance, the defined tolerance zone for a flatness or straightness is the same tolerance zone in ISO and ASME.

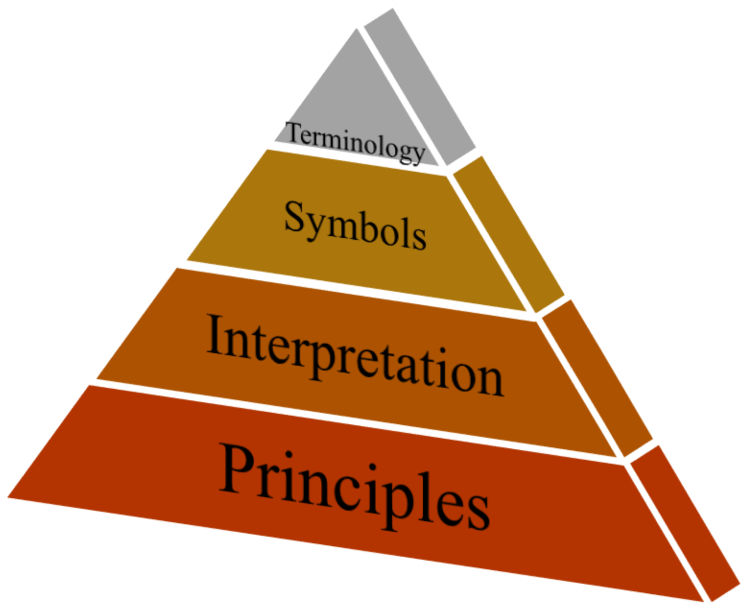


Figure 51 ISO and ASME GD&T Similarities and Differences Aspect

In terms of principles, ASME defines rule #1, which size controls form according to that. ISO does not recognize rule #1 as a principle and uses Ⓔ symbol whenever the size should control form. To violate rule #1 (size and form being independent), ASME uses Ⓘ symbol, but ISO does not need any symbol and size and form are considered independent automatically. Form tolerances in ISO and ASME are a little different. Using flatness tolerance, in ISO tolerance zones are two separate tolerance zones each having the same specified width but in ASME, there is only one tolerance zone with the specified width (Figure 54).

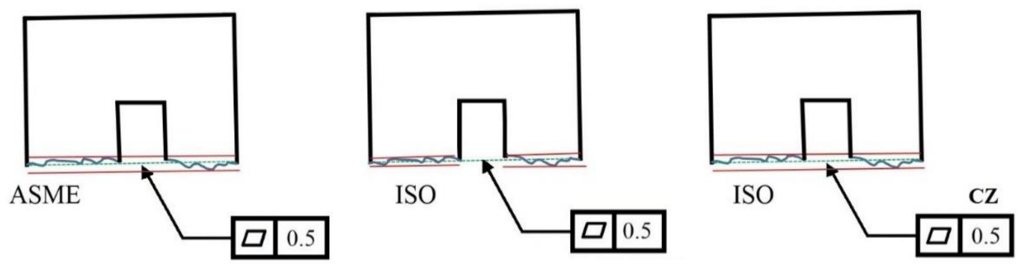


Figure 52 ISO and ASME Flatness Tolerance Comparison

ASME and ISO have different interpretation for orientation tolerances (Figure 55). ASME uses the axis (line, or median plan) of related actual mating envelope but ISO uses the actual or extracted axis (line, or median plane) to inspect the feature being in or out of specifications.

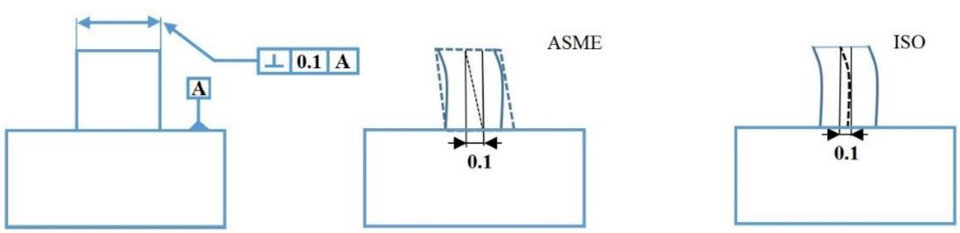


Figure 53 Different ISO and ASME Orientation Tolerance Interpretations

ISO and ASME are also different in terms of position tolerancing. ASME does not allow to define a position tolerance for a non- feature of size (Figure 56).

ASME and ISO define a different tolerance zones for a profile tolerance (Figure 57). Profile tolerance zone in ISO are defined by two equally or unequally disposed surface or lines enveloping a circle or sphere, but ASME defines two equally or unequally disposed profiles around the true profile created by extension at the corners.

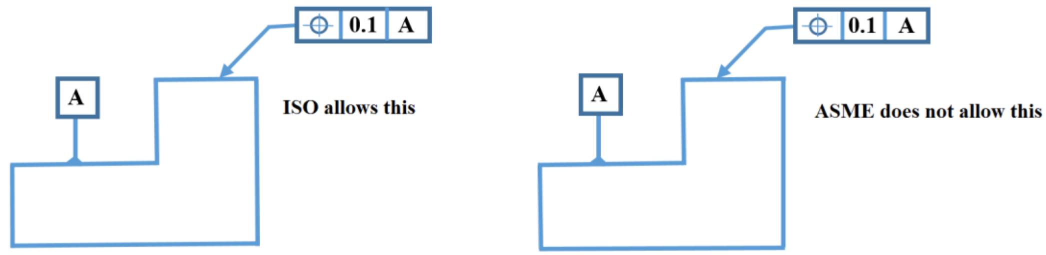


Figure 54 Different ISO and ASME Position Tolerance Interpretations

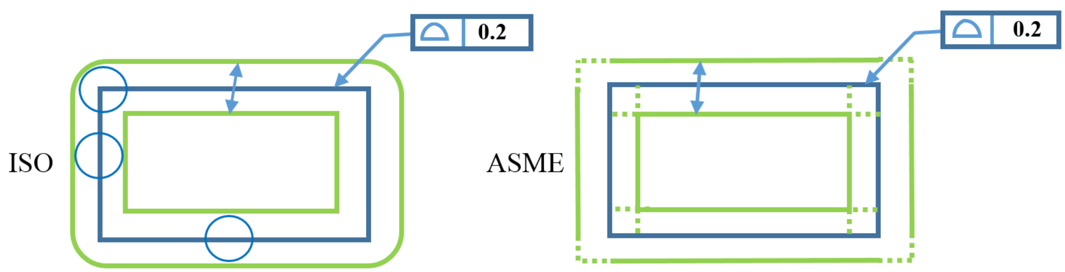


Figure 55 Different ASME and ISO Profile Tolerance Interpretations

Symbols

ASME and ISO have many common and some uncommon symbols. Table 9, 10, and 11 show the common ASME and ISO symbols, ASME proprietary symbols, and ISO proprietary symbols respectively.

Table 9 ISO and ASME common symbols and their definition in STEP and QIF

Symbol For	Symbols	QIF	STEP
Form Tolerances	— ⏥ ○ ⌭	☑	☑

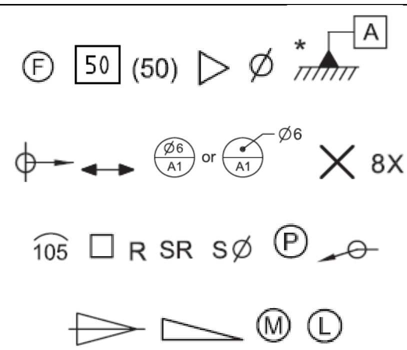
Orientation Tolerances	∠ ⊥ //	☑	☑
Location Tolerances	⌖ ◎ ⌯	☑	☑
Profile Tolerances	⌒ ⌓	☑	☑
Run-out Tolerances	↗ ⌰	☑	☑
Modifying Symbols		☑	☑

Table 10 ASME proprietary symbols and their definition in QIF and STEP

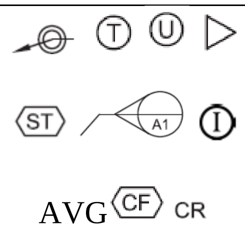
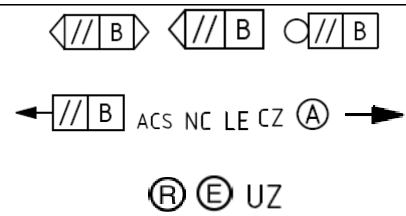
Description	Symbols	QIF	STEP
Modifying Symbols		No AVG?	☑
Dimensioning Symbols		☑	☑

Table 11 ISO additional symbols in QIF and STEP

Description	Symbols	QIF	STEP
ISO Additional Symbols		☑	No → Ⓐ

Terminology

ASME and ISO sometimes use different terminologies that a few of them are collected in table 14.

Table 12 Different Terminology Used in ISO and ASME

ASME	**ISO**
Feature Control Frame	Tolerance Frame
Basic Dimension	Theoretically Exact Dimension (TED)
Reference Dimension	Auxiliary Dimension
True Position	Theoretically Exact Position (TEP)
Inner Boundary	-
Outer Boundary	-
Circularity	Roundness

PMI Definition in QIF and STEP

PMI definition method in STEP and QIF are quite different due to their different information modeling languages. For instance, a cylindrical feature with a specified size and position tolerance (Figure 58) can be defined according to figures 57 and 61.

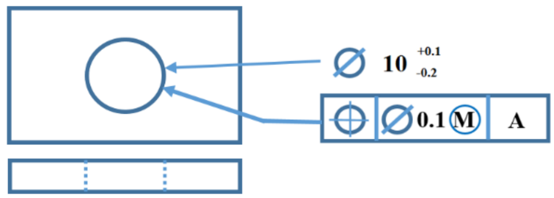


Figure 56 defining a size and position tolerance for a cylindrical Feature

To define a position tolerance with material modifier and datum definition, STEP uses complex entities by combining different simple entities. In this case, (Figure 59), "LENGTH_MEASURE_WITH_UNIT" entity defines the tolerance value, "TOLERANCE_ZONE_FORM" defines tolerance zone form (cylindrical), "MODIFIED_GEOMETRIC_TOLERANCE" entity defines material modifier and "GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE" defines datum.

```
#1000= (GEOMETRIC_TOLERANCE ('position', 'Positional tolerance for hole 1', #1045, #1008)
        GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE ((#1002, #1003, #1004))
        MODIFIED_GEOMETRIC_TOLERANCE (.MAXIMUM_MATERIAL_CONDITION.)
        POSITIONAL_TOLERANCE ());
#1045=LENGTH_MEASURE_WITH_UNIT (LENGTH_MEASURE (0.1), #41);
#1055=TOLERANCE_ZONE_FORM ('cylindrical or circular');
```

Figure 57 Feature Control Frame Definition in STEP

PMI definition in QIF is very different because of additional QIF capabilities in reporting measurement results. Features (defined as a part of geometry) and characteristics can have four aspects including "Definition", "Nominal", "Actual", and "Item" in QIF. The feature definition can be stored in "Definition" and "Nominal" sector and measurement results can be stored in "Item" and "Actual" sections. "FeatureNominal" defines a nominal feature without exactly specifying the feature location and allows "FeatureDefinition" to reference any "FeatureNominal" without redefining the feature. Characteristics and features correlations and referrals are illustrated in figure 60.

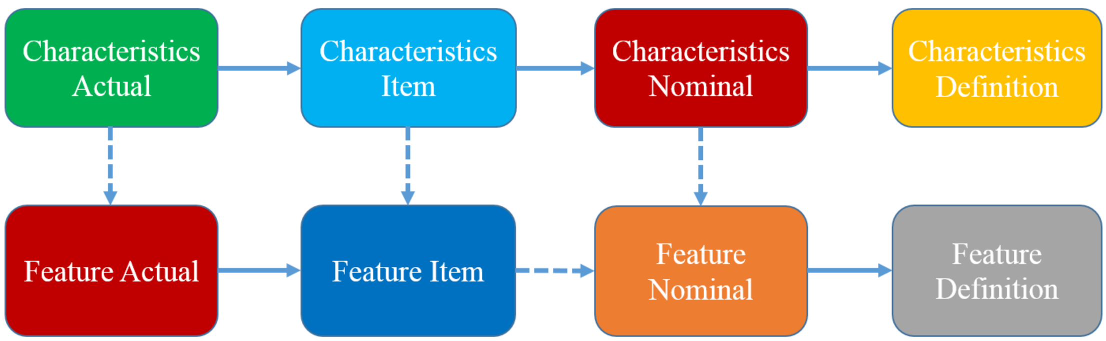


Figure 58 Characteristics Referral in QIF

The "Actual" and "Item" aspects of the characteristics and features are used whenever a characteristic or feature is measured more than once, the measurement results are stored in feature or characteristic "Item" and the result including pass or fail is measured in the characteristic "Actual" section. The case studies just include the nominal and definition aspect of a feature and characteristic as STEP AP203 is not used to report measurement results. In case studies, the software first loads the complex types or entities to the repository and creates an array of them. Then, a definition and nominal aspect for each entity of STEP is created. For instance in a circle feature case, a STEP "circle" entity is equal to a "CircularArcFeatureDefinitionType" feature definition aspect plus "CircularArcFeatureNominalType" feature nominal aspect. The circle feature diameter and length are stored in definition aspect and the location in nominal aspect (this also works for PMI). In PMI cases for instance, the software reads all STEP Perpendicularity tolerances (PERPENDICULARITY_TOLERANCE) and creates an array of that entity. According to this tolerance definition in STEP, the first element is name (or ID) and second element is description. The third, fourth, and fifth elements refer to tolerance value, part tolerance being applied and datums reference frame. For any PMI entity, correspondent definition and nominal aspect of that entity is created in QIF XML format. The tolerance value is always stored in definition aspect.

Table 13 PMI Comparison in QIF and STEP

QIF	STEP
……………….. <DatumDefinition id="1435"> <DatumLabel>**A**</DatumLabel> <FeatureNominalIds N="1"> <Id>2134</Id> </FeatureNominalIds> </DatumDefinition> ………………….. <DatumReferenceFrame id="**1437**"> <Datum> <SimpleDatum> <DatumDefinitionId>1435</DatumDefinitionId> <MaterialModifier>NONE</MaterialModifier> <ReferencedComponent>NOMINAL</ReferencedComponent> </SimpleDatum> </Datum> </DatumReferenceFrame> …………………………………… <PerpendicularityCharacteristicDefinition id="**2319**"> <ToleranceValue>1.5</ToleranceValue> <DatumReferenceFrameId>**1437**</DatumReferenceFrameId>…	#2000=PERPENDICULARITY_TOLERANCE (‘**2319’**, 'Perpendicularity tolerance for top face', #2045, #2008,(#2003)) ; #2045=LENGTH_MEASURE_WITH_UNIT (LENGTH_MEASURE (**1.5**),#41) ; #2008=SHAPE_ASPECT ('Top face', 'Top face of part', #40, .T.); #2018=PROPERTY_DEFINITION ('Shape','Shape of Feature',#2008); #2009=SHAPE_DEFINITION_REPRESENTATION (#2018,#2001) ; #2001=SHAPE_REPRESENTATION ('Representation of face', (#142),#49) ; #142=ADVANCED_FACE ('', (#132,#141),#98,.T.) ; #98=PLANE (‘**2134’**,#97) ; #2003=DATUM_REFERENCE (1,#1006); #1006=DATUM (**'1435'**,'Datum A',#40,.F.,**'A'**); #40=PRODUCT_DEFINITION_SHAPE ($,'Shape of part ',#14) ;

Figure 59 illustrates how datums are defined in STEP. Geometric_tolerance and Geometric_tolerance_with_datum_reference entities reference datum_reference entity and datum_reference entity refers datum A. In addition, Datum A is connected to a geometrical entity like “plane” through a chain of entities. This diagram also shows how the software writes different entities into STEP file. Geometric_tolerance can be any type of geometric tolerances like perpendicularity, flatness, etc.

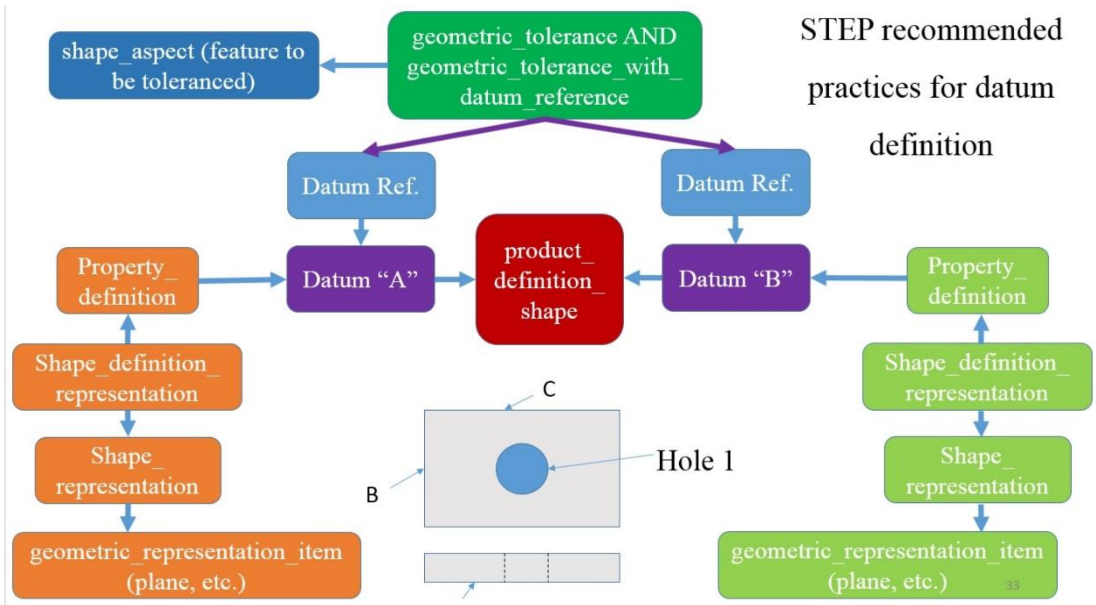


Figure 59 STEP Recommended practices for Datum definition (AP203 Practices)

STEP has a guideline for different GD&T concepts (STEP AP203 recommended practices). These concepts include geometric tolerances with datums, geometric tolerances with modifiers, tolerance zones, movable datum targets, datum targets, and many other concepts that can found in the reference STEP AP203 PMI recommended practices (STEP AP242 recommended practices is also published recently). For instance, you can find STEP guidelines in associating a tolerance to a feature, which is equal to figure 58 in QIF.

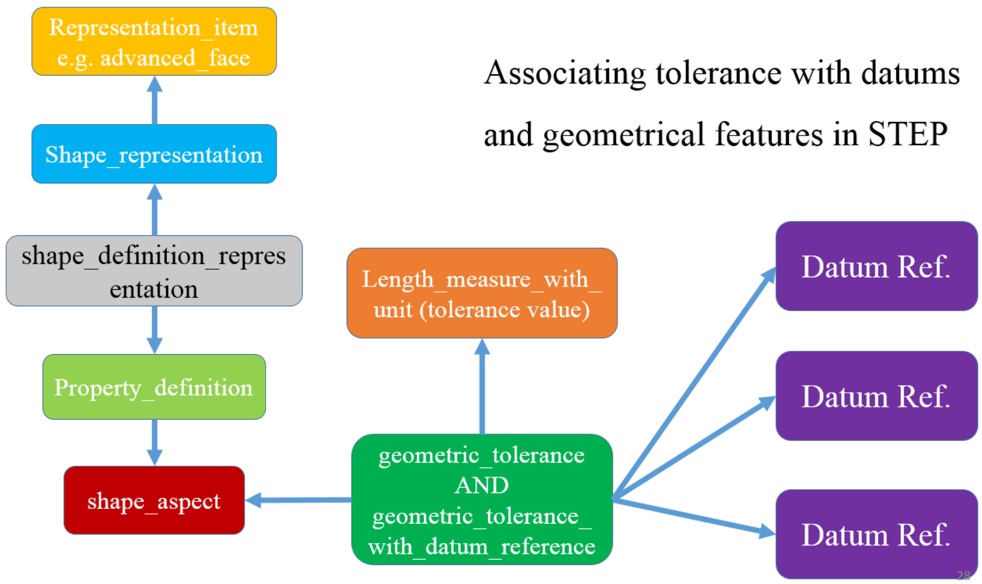


Figure 60 Guidelines associating tolerances with features (AP203 practices)

In the case of material modifier, two separate classes should be generated for geometric tolerances that can have material modifiers, one java class with material modifier and another java class without material modifier. Besides material modifiers, many other modifiers (Common Zone, Free State, Statistical Tolerance, Tangent Plane, etc.) can be modeled in STEP using a complex entity combining Modified_geometric_tolerance entity with the desired tolerance entity (like position_tolerance entity). Many of these modifiers (common zone, Free State, and statistical tolerance) are defined as elements of "CharacteristicDefinitionBaseType" that all characteristics inherit from that.

For composite tolerancing, STEP simply creates two separate geometric tolerances and connecting them with geometric_tolerance_relationship entity. QIF has a different approach and it has multiple complex types for composite tolerance definition.

In case of applying a characteristic to multiple features, QIF just references multiple features in characteristic nominal section, but STEP does this by combining different shape_aspect and referencing them through composite_shape_aspect entity.

A datum can have different modifiers (Basic, Translation, Maximum or Least Material Requirement, etc.). STEP uses referenced_modified_datum entity description (instead of datum_reference entity) and QIF defines all of these modifiers as elements of “DatumType” complex type.

In another example, a position tolerance can be modelled in QIF according to Figure 61. Comparing figure 59 with figure 61, PositionCharacteristicNominal references the feature in “FeatureNominalIds” tag with the id of “2177” (CylinderFeatureNominal tag) and PositionCharacteristicDefinition with the id of “2349”. Looking at cylinder feature nominal and definition aspects, “EntityInternalIds” tag references the entities defining the cylindrical feature in the boundary representation model of the part. Information such as diameter and length are defined in the feature definition tag. Material condition, tolerance value, datum reference(s), and zone shape are defined in the characteristic definition part

(PositionCharacteristicDefinition).

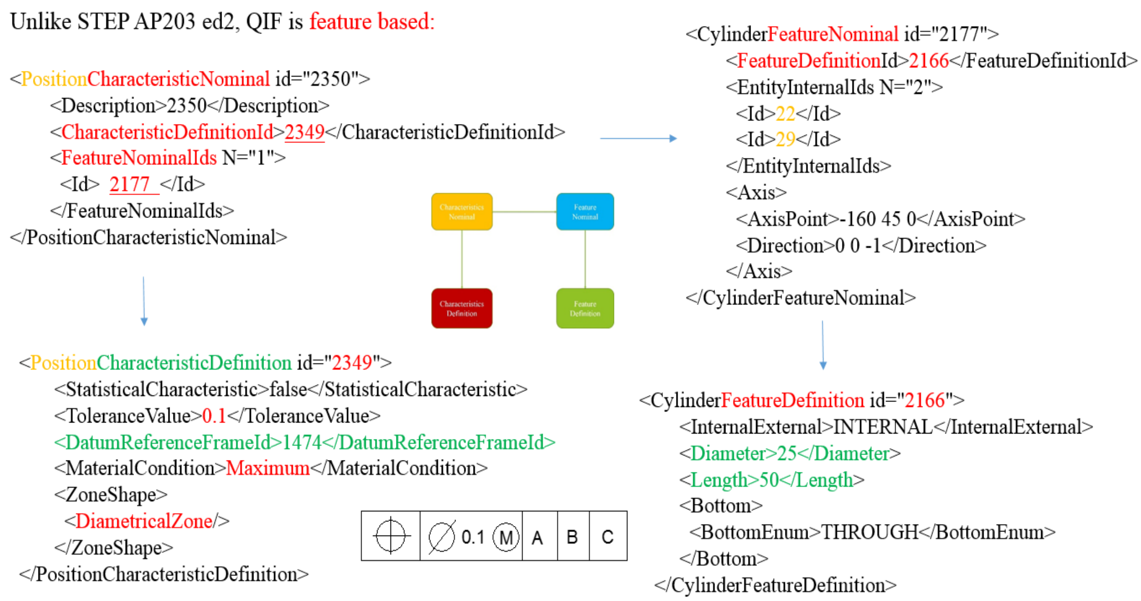


Figure 61 Different Feature Aspects in QIF

Q-DAS ASCII Transfer Format

Q-DAS ASCII transfer format has been used to transfer the PMI from QIF format to STEP format for the sole purpose of PMI illustration in the CMM software. Q-DAS is a statistically based format consisting many codes (k-fields) to store PMI and report measurement results.

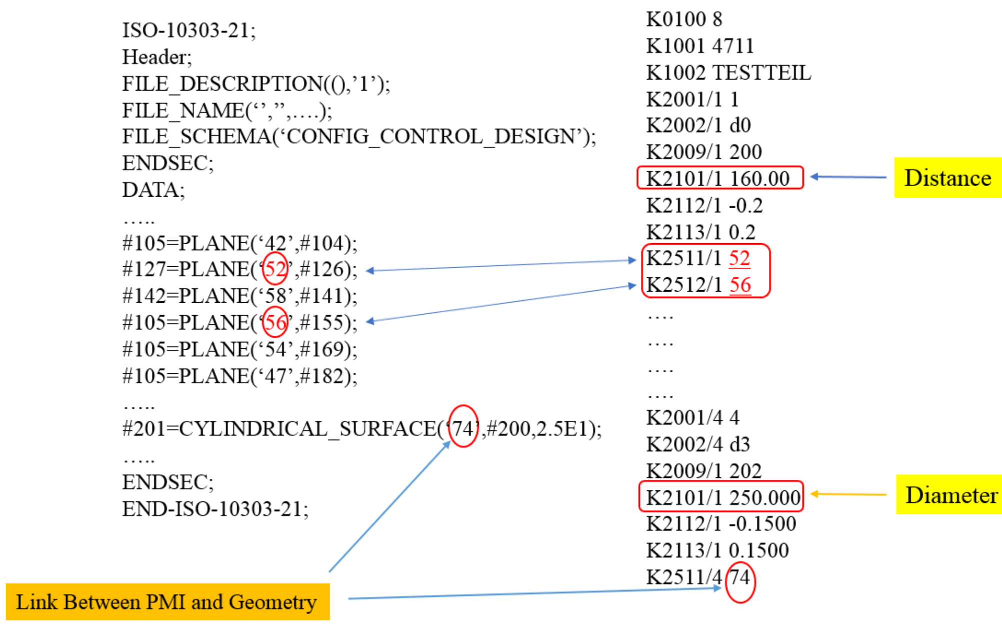


Figure 62a QDAS ASCII Transfer Format Used for PMI

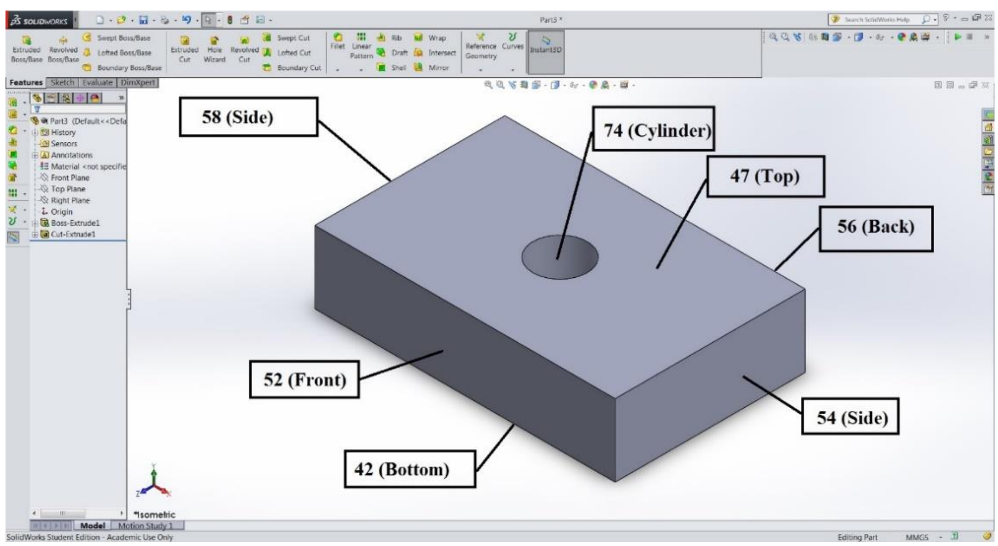
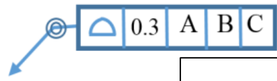


Figure 62b Solid Model of Part Used as Q-DAS ASCII Example

Q-DAS is used along a STEP file to import PMI and correlate PMI (including GD&T) to part geometry. For instance, if the cylindrical surface ID is assigned to be "74" in the STEP file the same ID is used in the k-filed to reference the geometry. In this case, k2511, k2101, k2112 (with k2113), and k2009 (with the value of 202) are used for referencing the geometry, defining the nominal value of the geometry (diameter), upper and lower tolerance, and diameter (characteristic) respectively.

Challenges and Interoperability Problems

STEP and QIF have different intrinsic properties that many of them originate from different information modeling languages that they use. STEP is using EXPRESS information modeling language and this allows creating complex entities (creating a java class for each complex entity). In some cases, this allows great flexibility to STEP that may result in creating different complex entities with the same concept (adding different flavors to STEP) or creating entities that does not exist in the real world. In addition, STEP uses a pool of entities to connect to other pool of entities while QIF uses a hierarchy to save the information and it does not seem to have this problem. Recommended practices for STEP is a key to resolve having different flavors for STEP. For instance to define the all-over symbol for a profile tolerance, QIF uses the hierarchy illustrated in figure 56 to define the extent that profile tolerance is applied and the extent is chosen from an enumeration, but STEP references “product_definition_shape” entity to define all-over for a profile tolerance. All-over symbol and its definition in STEP and QIF are illustrated in figure 63. Table 15 summarizes some of the issues that may be faced during this data translation.



STEP

#214=geometric_tolerance ($, $, #143, **#175**);

#175=product_definition_shape (‘’, $, #125);

QIF

```
<SurfaceProfileCharacteristicDefinition id="39">
    <ToleranceValue>0.3</ToleranceValue>
    <Extent>

<ExtentEnum>ALLOVER</ExtentEnum>
    </Extent>
………….
</SurfaceProfileCharacteristicDefinition>
```

Figure 63 Comparing all-over symbol (ASME Y14.5- 2009, section 8.3.1.6) definition in QIF and STEP

STEP

#110=position_tolerance (PT, $, #115, **#111**);

#111=dimensional_size (#112, ‘#111’);

#112=shape_aspect ($, $, #120, #121);

#112 used to define cylindrical feature in this STEP file snippet (not completely represented)

QIF

```
<PositionCharacteristicNominal id="323">
<FeatureNominalIds n="1">
    <Id>5012</Id>
</FeatureNominalIds>
  <Name>PT</Name>
</PositionCharacteristicNominal>
```

FeatureNominal (along with FeatureDefinition) is used to define a feature (cylindrical feature in this case).

Figure 64 Comparing STEP and ISO methods in associating FOS with a (position) tolerance

Table 14 Comparing PMI in AP203 with QIF

	STEP AP203	QIF			STEP AP203	QIF
Datums	✓	✓		Different geometric tolerance modifiers (free state, not convex)	✗	✓
Location	✓	✓		Multiple datum features	✗	✓
Orientation	✓	✓		All-over / All-around modifier	✗	✓
Size	✓	✓		Tolerance per unit area or length	✗	✓
Profile	✓	✓		Different modifiers of Datums (basic, translation, etc.)	✗	✓
Unequal or non-uniform tolerance zones	✗	✓		Composite Tolerancing	✓	✓

Chapter 6: Dimensional Measuring Equipment Selection Rules in QIF Rules

QIF Rules is an application schema developed for uniformity in measurement strategies across a company by proving the recommended or best practices for a DME programmer. The current version of the QIF Standard (version 2.1) has a basic schema that includes rules for the measurement of features and characteristics.

The structure of a rule is the composition of a Boolean expression and an action that is based on the evaluation result (i.e. true or false) of that expression. This relationship is shown in Figure 65.

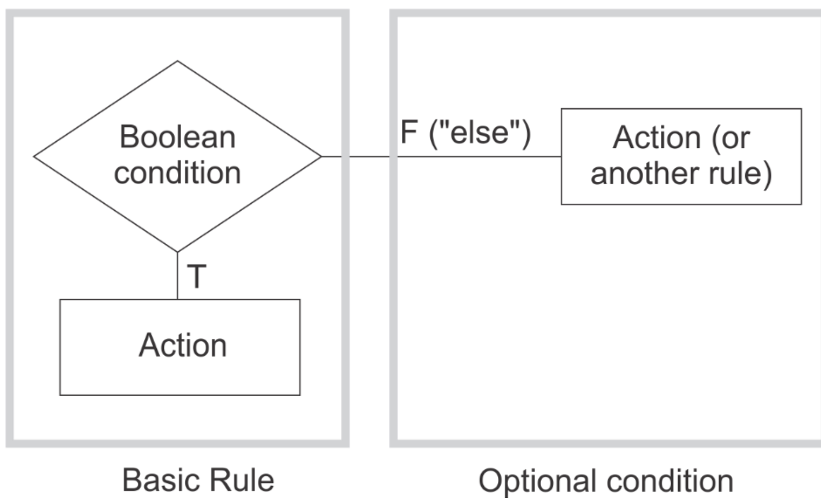


Figure 65 Structure of the rules

The Boolean condition portion of the rule can be one of only eight types, listed below.

- Characteristics type (if the characteristics is equal to a certain type, for instance length)
- Tolerance value (comparing tolerance value with a given value)
- Feature type (if feature type is equal to a certain feature type, for instance circle)

- Shape class (prismatic, thin walled, etc.)
- Is feature a datum?
- Feature size (comparison with a given value)
- Feature surface area or length (comparison with a given value)
- And user defined sampling rigor

The conditions listed above are from three distinct flavors: **isequal**, as in the characteristic type condition; **numerical comparison**, as in the tolerance value condition; and **iselement**, as in the datum condition. These conditions allow the user to define the following actions based on the Boolean condition evaluation. Note that for the current version 2.1 of the standard, only simple actions related to measurements with a CMM are considered.

- Number of measurement points
- Measurement point density
- Point sampling strategy (according to ISO–14406 : 2010)
- Feature fitting algorithm to use

The actions described above fall into two classes: **assignment** of a numerical value (either the number of measuring points or the density of points), and **selection** from an enumerated list of sampling strategies or fitting algorithms.

QIF Rules is dependent on two other schemas that are Expressions.xsd and GenericExpressions.xsd. These two schemas define the QIF-specific Boolean and Arithmetic operations and expressions. In addition to aforementioned Boolean expressions, “Not”, “And”, “Or”, “Equal” (both arithmetic and Boolean), “Less Than”, “Less than or equal”, “Greater”, “Greater than or equal”, and “Constant is” Boolean expressions are defined as general Boolean operations. The figure below shows an example implementation of a measuring rule, defined according to QIF Rules V2.1.

```
<MaxFeatureRules n="2">
  <IfThenSurfaceRule>
    <GreaterThan>
      <FeatureArea/>
      <ArithmeticConstant val="10"/>
    </GreaterThan>
    <ThenPoints>
      <MinPoints>25</MinPoints>
    </ThenPoints>
  </IfThenSurfaceRule>
  <Else>
    <ThenPoints>
      <MinPoints>10</MinPoints>
    </ThenPoints>
  </Else>
</MaxFeatureRules>
```

Figure 66 QIF Rules sample for feature rules (QIF/ANSI V2.1)

According to this rule, if the feature area are more than 10 units, then the number of measurement points will be no less than 25, otherwise (“Else” section) the minimum number of points is 10.

In addition to feature rules, the schema has an empty stub for DME Selection Rules that will be discussed. This chapter includes proposed developments of DME Selection Rules for the next version of QIF. The proposed version will be submitted for approval to QIF annual summit and then ANSI approval as US national standard. In order to maintain backwards- compatibility with earlier versions of QIF Rules, the new development described in this chapter are tied closely to the syntax and structure of version 2.1.

The introduction of DME (Dimensional Measuring Equipment) Selection Rules is a new part of QIF Rules; the research supporting the development of these selection rules will be the focus of this chapter. The feature rules discussed above are very specific to a particular type of measuring equipment, and the development of the DME selection rules is a much broader area. The overall intent of these rules is to capture the decision process where a particular piece of measuring equipment is chosen to validate workpiece characteristics based on the precision required (tolerances), the available equipment, and other information that will be described below. In the upcoming sections, the selection criteria will be reviewed and QIF Rules schema modeling will be explained.

DME selection rules and factors: There are different methods and criteria for DME selection and these methods depend on instrument type. The Measuring Systems Analysis (MSA) reference manual is one of documents published by the Automotive Industry Action Group (AIAG) describing the evaluation of measuring equipment performance. In addition to industrial manuals, different journal and conference papers have been published describing DME selection methods (Martínez-Pellitero et al. 2012). The DME selection criteria and methods proposed for the next QIF version are described next.

Uncertainty of measurements and tolerance: Uncertainty of measurement is an absolute parameter associated with the result of measurement that characterizes the dispersion of true value [VIM 2004]. True value resides somewhere between the measured value plus and minus the uncertainty and the true value must be inside the tolerance range to accept the part or characteristic. With regard to maximum and minimum value of tolerance, uncertainty (if it is large enough) may result in failing to accept a part manufactured in tolerance (Hocken and Pereira 2011) or conversely, accepting a part that is out of tolerance. As you see in figure 65, arbitrary true value (selected in the uncertainty band) and measured value are in the tolerance range and the uncertainty is small enough that the uncertainty band remains inside the tolerance range.

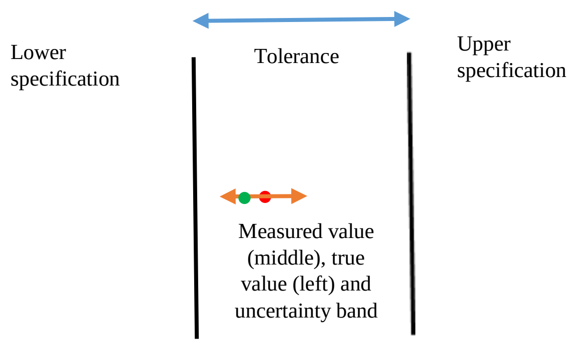


Figure 67 Tolerance and uncertainty role in DME selection

Uncertainty is task specific and depends on many parameters including maximum permissible error of the DME, room temperature, resolution, etc. The aforementioned parameters contribute to the uncertainty budget and can influence DME selection rules. It is worth mentioning that there may be different uncertainties for the same characteristic on the same part measuring with the same device (which may be because of vibration, room temperature fluctuation, etc.). There is an uncertainty associated with a characteristic (result of measurement) every time being measured. For CMMs selection rules, task-specific uncertainty is very complex and it depends on non-physical parameters like sampling strategy and mathematical algorithm selected for substitute geometry computations. In addition to software, measurement device, and environmental conditions, uncertainty analysis also depends on part errors (including form errors, surface finish errors, etc.) which makes uncertainty analysis very complicated. In general five factors contribute to uncertainty analysis for CMMs and these factors are coordinate measuring system hardware (temperature, vibration, probing, etc.), workpiece, sampling strategy, algorithms, and extrinsic factors (operator, fixturing errors, etc.).

Statistical studies of DME performance: Many measurement process indices have been defined for measurement process assessment (like C_p as tolerance range over six sigma of the measuring process). In addition to capability indices, gage repeatability and reproducibility (gage r&r) is a tool to study measurement system performance. The most common gage r&r studies (or methods) are Analysis of Variance (ANOVA) and average and range method. The proposed version of QIF Rules does not include gage r&r and capability indices role in DME selection.

Cost, skill, labor and location: Cost, skill, labor, and location of DME are some of parameters in selecting DMEs, but they are considered out of scope in this research.

Feature orientation: Feature orientation is another criterion that should be considered in creating measurement rules. For instance, an Indexable probe is a choice if the tactile probe axis is normal to the feature axis (for instance a cylindrical feature). Feature orientation should also be considered in measurement planning for potential probe or sensor collision with the part or fixture (Vafaeesefat and Elmaraghy, 2000).

Characteristic and feature type: QIF defines twenty-nine types of features and supports different GD&T (ISO, ASME, etc.) and PMI standards. The characteristics and feature type can play an important role in DME selection requirement. Inspecting a cylinder diameter with a caliper should be more convenient than a CMM, unless the caliper is not accurate enough. Tight tolerances can affect the DME selection process because tighter tolerance requires higher resolution and lower uncertainty (better resolution does not necessarily imply lower uncertainty). Moreover, some characteristics cannot measured with every DME. For instance, straightness cannot be measured with a caliper!

Dimensions of the parts: Any DME has a limited capability in terms of maximum and minimum size measurement. Feature maximum and minimum sizes should be in range of effective working volume of DME and range of measurement device. Feature minimum and maximum size are part of "if" sections of next QIF Rules version. The maximum size of the part that should be measured must be less than maximum range of DME. Depending on the shape of the part (cylindrical, prismatic, etc.), this dimensions can be compared with working volume and effective working to make sure that the part can be inspected.

Measuring time: Measuring time, including setup time, is one of major rules for DME selection. If the measurement is performed for a batch production or a massive number of parts are being measured, measuring time can be an important parameter. The measuring time can also depend on the number of required measurement points for a specific characteristic and the number of points can be measured simultaneously.

Environmental Conditions: DMEs work in a certain environmental range of temperature, humidity, pressure, and carbon dioxide (creating useful measurement results in operational limits of environmental condition). Environmental conditions are important factors in DME selection process as some DMEs may be more sensitive to environmental changes and require a more stable environment. In addition to creating useful measurement results, effects of environmental condition should be studied on DME performance and measurement results, as they are huge contributing factors to uncertainty budget.

QIF Rules DME selection hierarchy: In the highest level of QIF Rules hierarchy, four elements are defined "Version", "RulesUnits", "FeatureRules" and "DMESelectionRules".

The first three elements are already developed in previous versions and they are not the topic of this research (DMESelectionRules is developed in this version).

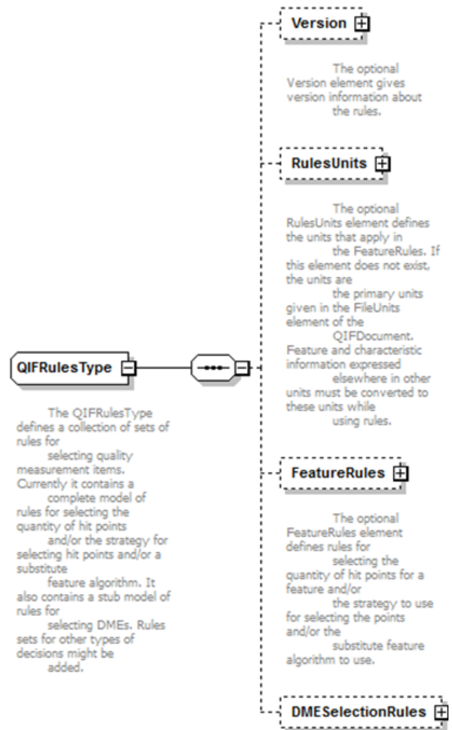


Figure 68 Highest level of hierarchy in QIF Rules

In the DME Selection Rules section, one sub category is defined "PhysicalMeasurementSystemRules" (element or complex types) (figure 67). Physical measurement system includes any DME that requires selection rules.

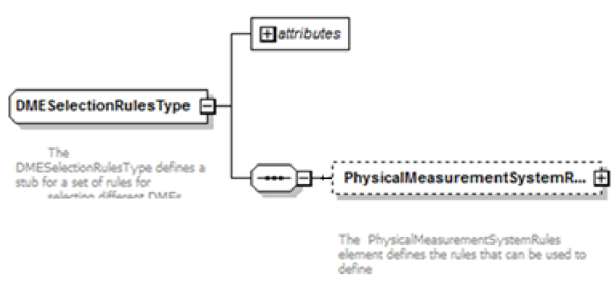


Figure 69 DME Selection Rules element hierarchy

The proposed version of QIF Rules supports four types of measuring equipment (DMEs) that are subcategory of “PhysicalMeasurementSystemRules”. These are coordinate measuring systems, single- characteristic devices, attribute gages and non-destructive test equipment.

•**IfThenCoordinateMeasurementEquipment**: This equipment verify the characteristics by measuring the coordinates of multiple points and using them to construct a substitute geometry. Because almost all attributes of a workpiece can be reconstructed from coordinate data, these instruments are very flexible. However, they may be expensive and slower to operate than dedicated systems, and are limited by the points that can be collected from the workpiece surface. This equipment type includes CMMs, computed tomography machines, laser scanners, and other devices.

•**IfThenDirectCharacteristicMeasurementEquipment**: These devices directly measure the characteristics without any coordinate measurement. Three types of displacement, linear and angular categories are defined for direct characteristics measurement device. While roundness-measuring equipment is dedicated to measuring the form of circles and cylinders, this equipment is not currently supported in this class.

•**IfThenDirectToleranceVerificationDevice:** This element models the devices that do not measure the characteristics and just verifies if the part is in tolerance or out of tolerance. The attribute gages, or hard gages, include snap gages, gage pins, and fixture gages.

•**IfThenDefectDetectionInstrument**: This element models the devices used to detect surface or beneath surface defects. This class of (typically) non-destructive testing (NDT) equipment often evaluates criteria not specified in the geometric PMI discussed in the earlier chapters of this document.

Figure 70 Physical measurement system hierarchy in proposed QIF Rules

IfThenCoordinateMeasurementEquipment: This element or complex type defines five sub categories as following figure. In addition, IfThenCMM includes four other types of CMMs as its subcategory (according to QIF Measurement Resources). IfThenCMMTool is a type defined to model the CMM tool and sensor requirement.

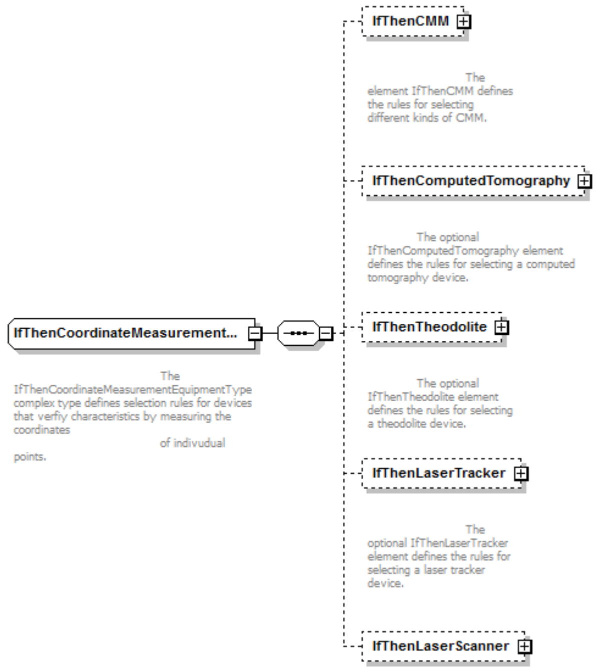


Figure 71 Coordinate measurement equipment hierarchy in QIF Rules

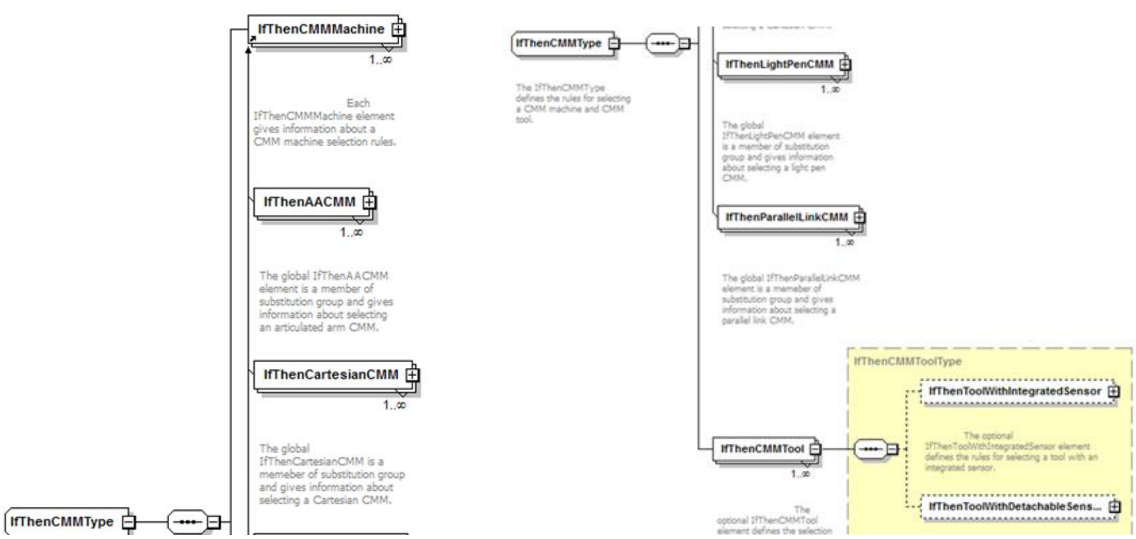


Figure 72 IfThenCMMType hierarchy in QIF Rules

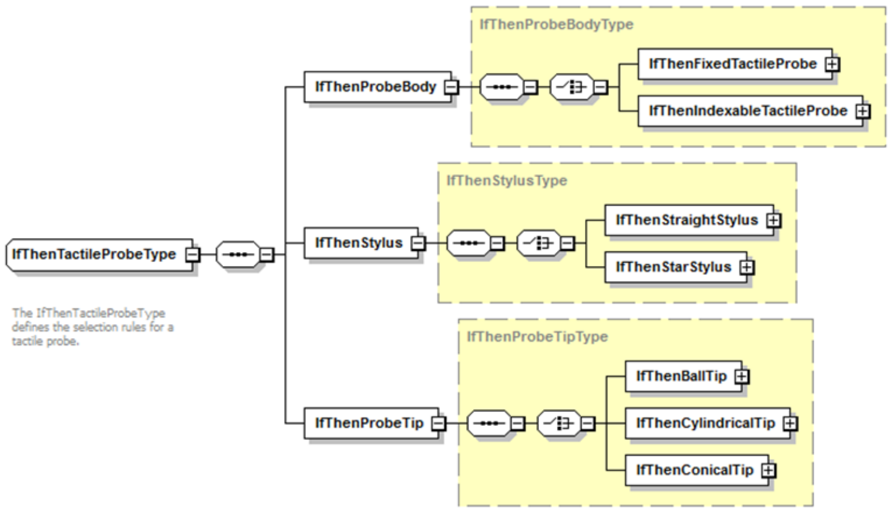


Figure 73 IfThenTactileProbeType hierarchy in QIF Rules

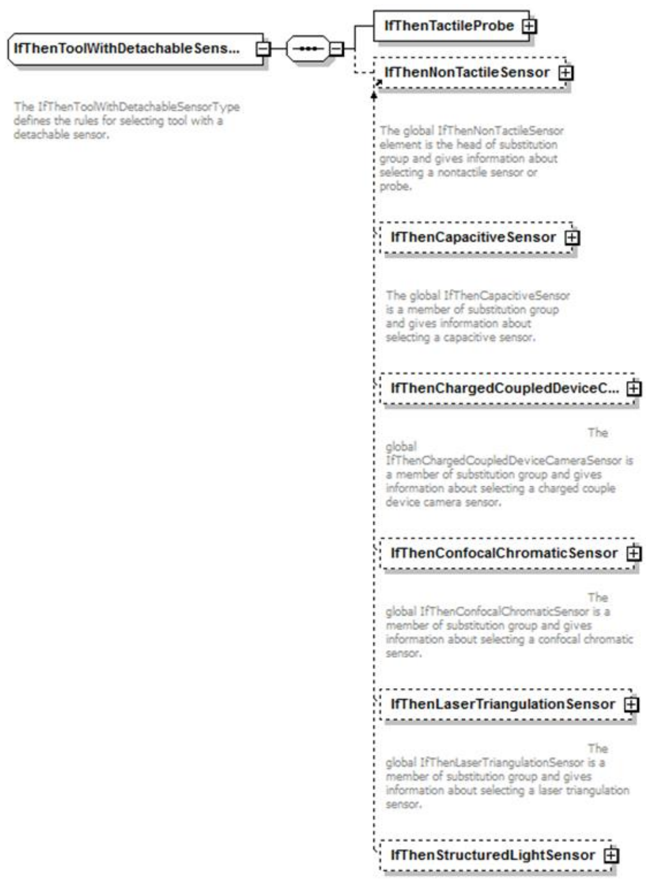


Figure 74 IfThenToolWithDetachableSensor hierarchy in QIF Rules

IfThenDirectCharacteristicMeasurement: This category of devices is split into displacement, linear, and angular measurements. For the displacement measurement four LVDT, DVRT, magneto inductive, and draw wire sensors are considered as subtype of IfThenDisplacementMeasurement. The IfThenLinearDimensionMeasurementType includes five devices, ULM, micrometer, caliper, bore gage, and optical comparator. For angular measurement devices, optical comparator, autocollimator and sine bar are defined (optical comparator is a common device in a few different categories).

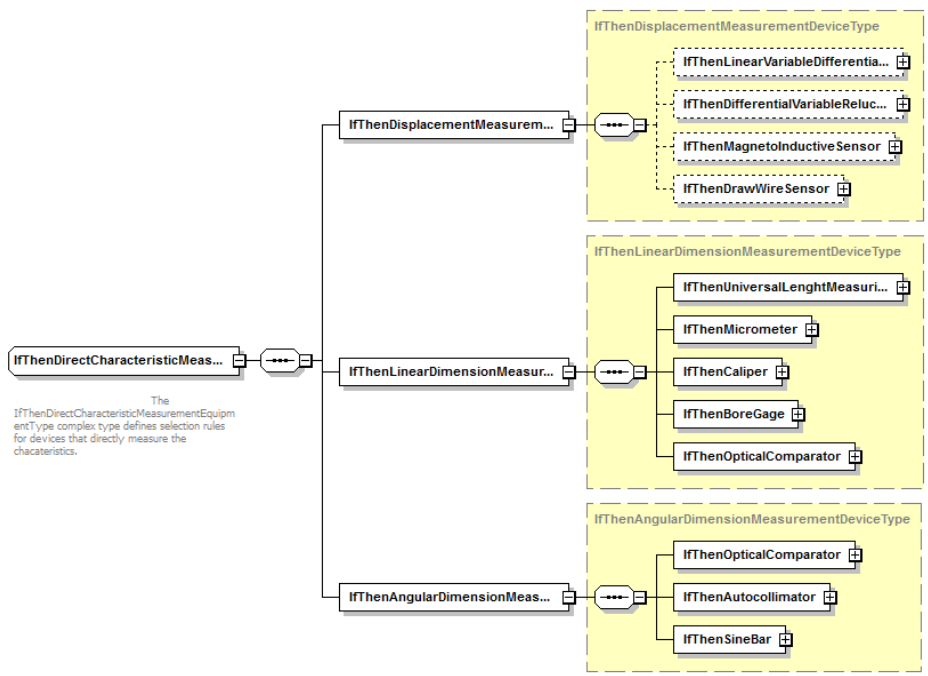


Figure 75 IfThenDirectCharacteristicsMeasurement Equipment hierarchy in proposed QIF Rules

IfThenDefectDetectionInstrument: This complex type defines the sensors and devices used for surface and beneath surface defects. Two sensors are defined for each of each sub-categories, eddy current and microscope for surface defect detections, and ultrasound and computed tomography for beneath surface defects. This category is based on available DMEs in QIF Measurement Resources catalogue.

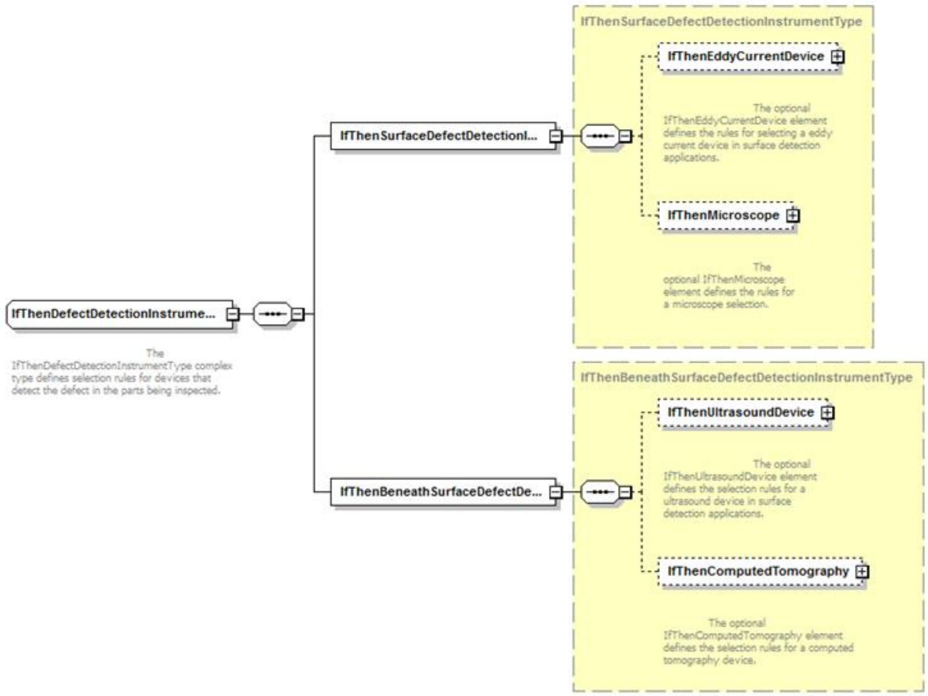


Figure 76 IfThenDefectDetectionInstrument hierarchy in QIF Rules

IfThenDirectToleranceVerificationDevice: This complex type defines the devices used for tolerance verification without any measurement. They just claim if the part is in tolerance or out of tolerance. Two measurement devices are defined in this section go/no-go gage and optical comparator.

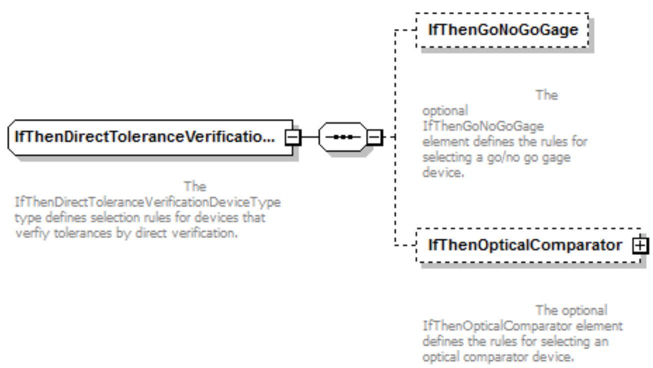


Figure 77 IfThenDirectToleranceVerificationDevice hierarchy in QIF Rules

DME Selection Rules Boolean Expressions: Beside general Boolean expressions ("And", "Or", etc.), other Boolean expressions are defined for DME selection rules

- Characteristic is of a certain type
- Feature is of a certain type
- Tolerance value compared to certain value
- Feature orientation compared to certain value
- Measuring time compared to a certain value
- Work space size compared to a certain value
- Feature size compared to a certain value

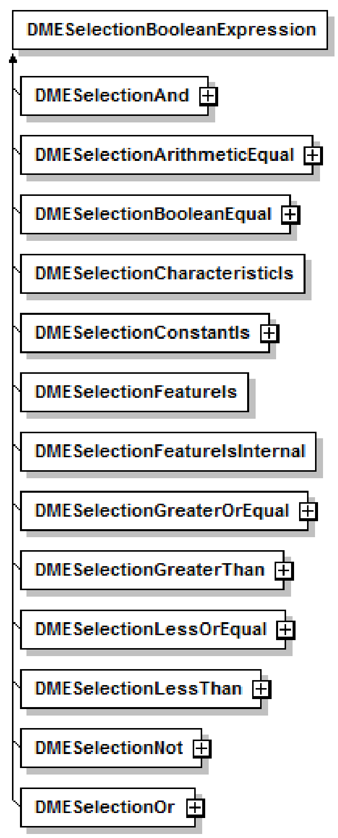


Figure 78 DME Selection Rules Boolean Expressions

All Boolean conditions are sub-elements of any DME in the QIF Rules and can be used to define any rules for any DME device. DMEs are selected and modeled based on QIF Measurement Resources V2.1.

DME Selection Rules Arithmetic Expression: Besides general Arithmetic expressions ("Plus", "Minus", etc.), different DME selection arithmetic parameters are defined

- DME Selection Arithmetic Feature Parameter (could be feature maximum and minimum size)
- DME Selection Arithmetic Characteristic Parameter (characteristic value, etc.)
- DME Selection Feature Orientation
- DME Selection Arithmetic Parameter (measuring time and work space)

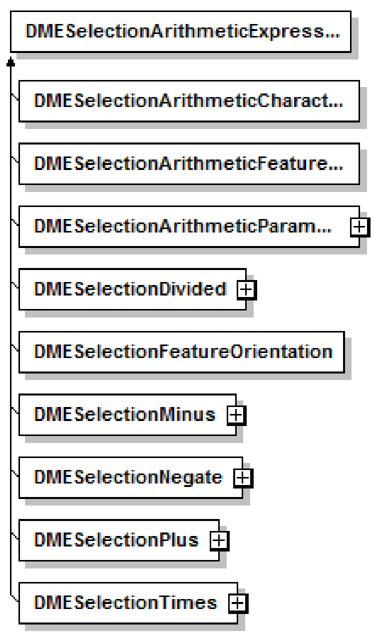


Figure 79 DME Selection Rules Arithmetic Expressions

Arithmetic and Boolean Expressions Connection: Arithmetic comparison expression can be Boolean true or false. Two base types are defined for connecting Arithmetic and Boolean expressions. Arithmetic comparison (greater than, etc.) and arithmetic equal (arithmetic equal is a member of Boolean expression substitute group) complex types inherit from them.

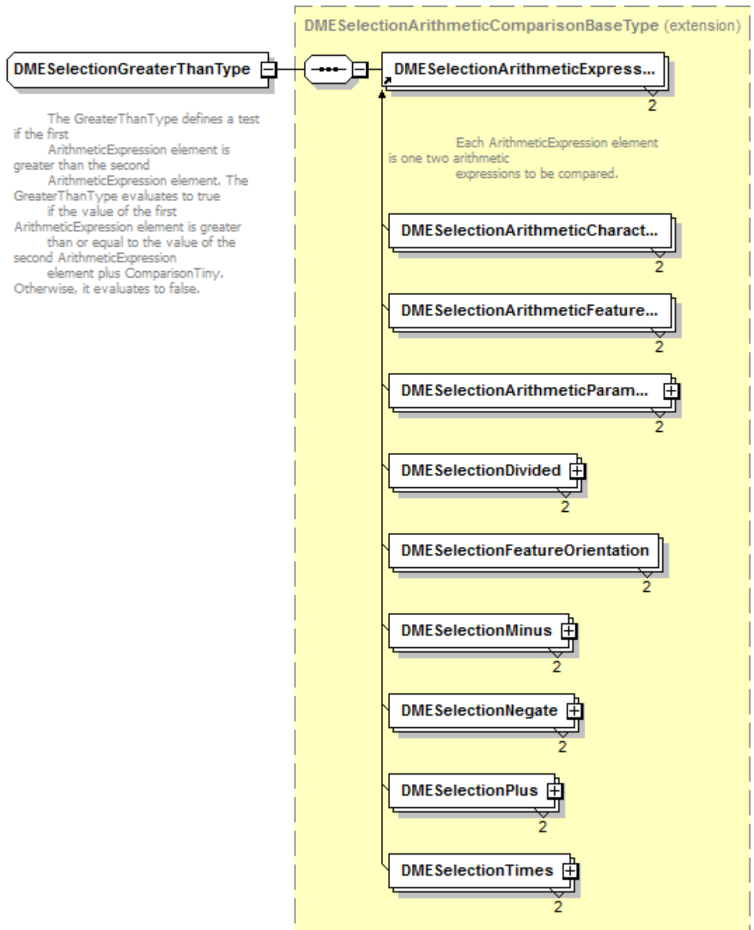


Figure 80 DMESelectionGreaterThan inherits from DME SelectionArithmeticComparisonBaseType

DME Selection Rules Examples

In this section, two QIF Rules examples are presented. The first example declares that a caliper should be selected that its resolution is less than or equal to 0.1 of tolerance value.

<Rules >

<DMESelectionRules n="1">

<PhysicalMeasurementSystemRules>

```
                <IfThenDirectCharacteristicMeasurementEquipment>
                    <IfThenLinearDimensionMeasurementDevice>
                        <IfThenCaliper>
                            <DMESelectionLessOrEqual>
                                <DMESelectionArithmeticDMEParameter>
                                    <Parameter>Resolution</Parameter>
                                </DMESelectionArithmeticDMEParameter>
                                <DMESelectionTimes>
                                    <DMESelectionArithmeticCharacteristicParameter>
                                        <Parameter>ToleranceValue</Parameter>
                                    </DMESelectionArithmeticCharacteristicParameter>
                                    <DMESelectionArithmeticConstant val="0.1"/>
                                </DMESelectionTimes>
                            </DMESelectionLessOrEqual>
                        </IfThenCaliper>
                    </IfThenLinearDimensionMeasurementDevice>
                </IfThenDirectCharacteristicMeasurementEquipment>
            </PhysicalMeasurementSystemRules>
        </DMESelectionRules>
</Rules>
```

The second example as following declares that a caliper with Id=3166513 should be selected if the characteristic being measured is a thickness.

```
<Rules>
        <DMESelectionRules n="1">
                <PhysicalMeasurementSystemRules>
                        <IfThenDirectCharacteristicMeasurementEquipment>
                                <ThenDeviceID>3166513</ThenDeviceID>
                                <IfThenLinearDimensionMeasurementDevice>
                                        <IfThenCaliper>
                                                <DMESelectionCharacteristicIs val="THICKNESS"/>
```

</IfThenCaliper>

</IfThenLinearDimensionMeasurementDevice>

</IfThenDirectCharacteristicMeasurementEquipment>

</PhysicalMeasurementSystemRules>

</DMESelectionRules>

</Rules>

Chapter 7: Conclusion

Quality control is an important concept to increase customer satisfaction with a product and information technology is a tool that has revolutionized everything including the way we inspect such products and ensure their quality. In a modern manufacturing enterprise, manufacturing and inspection information are captured and saved in different proprietary digitized formats. The differences in these formats impose many expenses for the companies to translate the data from one proprietary format to another proprietary format. An integrated data framework across the product lifecycle, but focused on metrology (quality) data, can be a solution to this problem. There have been many different data formats proposed, trying to resolve interoperability problems and integrate part of product lifecycle. Some of these data formats include IGES, DMIS, DML, SAT, I++DME, VDAFS (German Acronym), STEP, etc. Many of these data exchange formats (Except STEP) cover only part of product life cycle and QIF is a unique data format trying to integrate metrology framework by developing a suite of information models. Using an XML structure and XSD as information modeling language in QIF also gives it advantages over other data formats as the XML structure is a standard format for data exchange over the World Wide Web. In addition, XML is a human- and machine-readable data format.

QIF splits quality data framework into a series of steps (or activities) and defines XML Schema(s) for each of these activities– in addition to the common core library schemas – to enhance data exchange throughout quality systems. The activities of the quality framework include product definition, measurement requirements determination, measurement process definition, measurement process execution, and quality data analysis and reporting. Each of these activities have at least one application schema and they include QIF Plans, QIF Resources, QIF Rules, QIF Execution , QIF Results, QIF Statistics, and QIF MBD.

The QIF Measurement Resources schema has been developed such that it models the hierarchy of information necessary to support creating a measurement plan in terms of measurement resources information (e.g. calibration history, available sensors, measurement device accuracy, resolution, etc.). In QIF Measurement Resources, measurement resources are classified into four types: fixtures, measurement devices, detachable sensors, and tools. Sensors and tools are classified as integrated or detachable. Detachable sensors are listed in the sensors section and the tools can be defined either with integrated sensors or with detachable sensors (from the sensors section). Different measurement devices are modeled in the measurement devices section that include autocollimator, Cartesian CMM, Articulated Arm CMM (AACMM), light pen CMM, parallel link CMM, multiple carriage CMM, equator, computed tomography, laser trackers, caliper, micrometer, sine bar, microscope, optical comparator, theodolite, and ULM machines. Many measurement devices, such as CMM or other devices can have a tool rack that can be loaded with different sensors and tools. QIF Measurement Resources V2.1 also supports different optical and non-optical sensors that include CCD camera, capacitive sensor, structured light sensor, tactile probe, draw wire sensor, confocal chromatic sensor, LVDT sensor, laser triangulation sensor, eddy current sensor, DVRT sensor, and ultrasonic sensor. In addition to the specific hardware metrology resources (devices, sensors, fixture, etc.), performance testing using the major CMM standards are modeled in QIF V2.1. These standards (ISO 10360 and B89.4) are necessary to evaluate the performance of the different types of CMM, including Cartesian and articulated arm CMMs. This is an important upgrade, as the performance of a CMM will influence its selection for a given measuring task.

In addition to QIF measurement Resources, metrology rules are developed to create a uniform measurement strategy throughout a company (DME selection strategies). A rule in QIF Rules is a combination of conditions and actions. The conditions (if section) include parameters like tolerance value, characteristics type, part feature type and sizes, feature orientation, work space, measuring time, and the action is selecting the measurement device according to the developed hierarchy. The developed DME selection hierarchy in the highest level classifies selection items to the selection of physical measurement systems. A physical measurement system has four sub categories including coordinate measurement equipment, direct-characteristics measurement equipment, direct tolerance verification device, and defect detect instrument. The coordinate measurement equipment category includes devices that measure coordinates of point being measured for evaluation purposes. These devices include CMM, computed tomography, theodolite, laser tracker and laser scanner devices. The direct- dimensional characteristics measurement devices measure the characteristics directly and they include linear- dimension measurement devices (ULM, micrometer, caliper, bore gage, and optical comparator) and angular-dimension measurement devices (Sine bar, autocollimator, and optical comparator). The direct-tolerance verification devices includes go/no go gages and optical comparator devices. The defect detection instrument category consists surface defect detection instrument (eddy current and microscope) and beneath surface defect detection instrument (ultrasonic device).

QIF is not the only format used in the industry and there should be ways to get the information from other data formats as well. The best-known data exchange format (in both industry and academia) is STEP. For implementation purposes, and to show how QIF can be integrated with STEP (STEP AP 203 Ed2), a translator has been developed to read and write data (including topology, geometry and PMI information) from QIF to STEP, and vice versa. Eclipse is used as the platform to encode the algorithms. For the topology and the geometry, the QIF complex types are correlated with the STEP entities to find the equivalent ones. For PMI, QDAS ASCII transfer format is used for QIF to STEP translation for transferring the PMI to the CMM software that is capable to read PMI through Q-DAS ASCII format (for the sole purpose of CMM software reading Q-DAS). PMI in STEP to QIF is transferred through correlating different PMI entities and complex types including different geometric tolerances (size, form, orientation, location, and runout) and other GD&T symbols (QDAS ASCII not used). Interoperability problems exist between QIF and STEP AP203 and some information may be lost. In terms of PMI, it is been investigated how effectively STEP AP203 Ed2 and QIF are able to model both ASME Y14.5 and ISO GD&T information, as these are the two main GD&T standards. STEP is an ISO standard and QIF is an ANSI standard; differences in the default interpretation of various tolerances may cause the loss of information during data translation.

For the future development, QIF should transform DMIS and I++DME to XSD data formats (DMIS and I++DME are a part of QIF suite). This will allow a uniform data format throughout metrology framework and would remove the only data format bottleneck which is DMIS and I++DME. In addition, the metrology resources information should be stored in a database and this requires developing databases and queries to store and retrieve the measurement resources information for XML files (using languages like SQL and MySQL). Measurement process planning requires information from manufacturing side about how a part is manufactured (like if the part is a casting part, machining part, etc.). Currently, there is not a manufacturing XML schema to convey manufacturing related information to support QIF Plans. QIF also needs to be harmonized with other major data exchange standards including STEP AP242, which has been published recently.

References

Peak, R.S., Lubell, J., Srinivasan, V. and Waterbury, S.C., 2004. STEP, XML, and UML: complementary technologies. Journal of Computing and Information Science in Engineering, 4(4), pp.379-390.

Tsichritzis, D. and Klug, A., 1978. The ANSI/X3/SPARC DBMS framework report of the study group on database management systems. *Information systems*, *3*(3), pp.173-191.

Zhao, Y., Xu, X., Kramer, T., Proctor, F. and Horst, J., 2011. Dimensional metrology interoperability and standardization in manufacturing systems. Computer Standards & Interfaces, 33(6), pp.541-555.

Brecher, C., Vitr, M. and Wolf, J., 2006. Closed-loop CAPP/CAM/CNC process chain based on STEP and STEP-NC inspection tasks. International Journal of Computer Integrated Manufacturing, 19(6), pp.570-580.

Srinivasan, V., 2008. Standardizing the specification, verification, and exchange of product geometry: Research, status and trends. Computer-Aided Design, 40(7), pp.738-749.

Zhao, Y.F., Kramer, T., Rippey, W., Horst, J. and Proctor, F., 2012, July. An integrated data model for quality information exchange in manufacturing systems. In 37th MATADOR Conference.

Zhao, Y.F., Horst, J.A., Kramer, T.R., Rippey, W. and Brown, R.J., 2012. Quality Information Framework–Integrating Metrology Processes. IFAC Proceedings Volumes, 45(6), pp.1301-1308.

Schafer J.; Proper Use of DML to Haul Dimension Data and Results, PowerPoint presentation by Joe Schafer chair of DML committee.

Brás, N.B., Ramos, P.M. and Serra, A.C., 2008. Extensible mark-up language file definition for structured acquisition data storage and transfer. Measurement, 41(3), pp.320-326.

Goh, A., Hui, S.C. and Song, B., 1996. An integrated environment for product development using STEP/EXPRESS. Computers in Industry, 31(3), pp.305-313.

Loffredo, D., 1999. Fundamentals of STEP implementation. *STEP Tools, Inc*.

Toll, D.G. and Cubitt, A.C., 2003. Representing geotechnical entities on the World Wide Web. Advances in Engineering Software, 34(11), pp.729-736.

Williams, M.E., Consolazio, G.R. and Hoit, M.I., 2005. Data storage and extraction in engineering software using XML. Advances in Engineering Software, 36(11), pp.709-719.

Zhao, Y.F., Habeeb, S. and Xu, X., 2009. Research into integrated design and manufacturing based on STEP. The International Journal of Advanced Manufacturing Technology, 44(5), pp.606-624.

Sauder, D. and Morris, K.C., 1995, October. Design of a C++ software library for implementing EXPRESS: the NIST STEP Class Library. In Proc. of EUG '95—the Fifth EXPRESS Users Group Conference (pp. 21-22).

Lee YT., An overview of Information Modeling for Manufacturing Systems Inspection. NISTIR. National Institute of Standards and Technology.

Chen, P.P.S., 1976. The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), pp.9-36.

Zhao, X., Pasupathy, T.K. and Wilhelm, R.G., 2006. Modeling and representation of geometric tolerances information in integrated measurement processes. Computers in Industry, 57(4), pp.319-330.

Siemens PLM Software (2011); Opening Product life-Cycle Data Sharing Using XML, Germany, Siemens.

Imkamp, D., 2005. POTENTIALS OF STANDARDIZED INSPECTION DATA TRANSFER FROM CAD TO METROLOGY. In Joint International IMEKO TC1+ TC7 Symposium.

Proctor, F., Rippey, B., Horst, J., Falco, J. and Kramer, T., 2007, August. Interoperability testing for shop floor measurement. In Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems (pp. 275-279).

Zhao, Y., Kramer, T., Brown, R. and Xu, X., 2011. Information modeling for interoperable dimensional metrology. Springer Science & Business Media.

IDEF Family of Methods. 2017. IDEF. [ONLINE] Available at: http://www.idef.com/. [Accessed 14 May 2017].

AP242. 2017. STEP AP242 Project. [ONLINE] Available at: http://www.ap242.org/. [Accessed 14 May 2017].

Fowler, M., 2004. UML distilled a brief guide to the standard object modeling language. Addison-Wesley Professional.

Booch, G., 2005. The Unified Modeling Language User Guide. Pearson Education India.

Xu, X. and Nee, A.Y.C. eds., 2009. Advanced design and manufacturing based on STEP. Springer Science & Business Media.

McKenzie-Veal, D., 2012. An Analysis of Step, Jt, and Pdf Format Translation Between Constraint-based Cad Systems with a Benchmark Model.

Nasr, E.A. and Kamrani, A.K., 2007. Computer Based Design and Manufacturing. Springer Science & Business Media.

US Product Data Association, 1996. Initial Graphics Exchange Specification. IGES 5.3. *ANSI. Retrieved July*, *12*, p.2008.

AUTODESK, I., 2012. Autodesk autocad dxf reference 2012.

ANSI (2009) Dimensional measuring interface standard, DMIS 5.2 Standard, Part 1, ANSI/CAM-I 105.0-2009, Part 1.

Quality Information Framework part1-7 (2016), QIF V2.0., Quality Information Framework- An Integrated Model for Manufacturing Quality Information, Burleson, Texas, USA. Dimensional Metrology Standard Consortium.

Willard, B., 2007. UML for systems engineering. Computer Standards & Interfaces, 29(1), pp.69-81.

QDAS ASCII Transfer Format Version 6 (2015), QDAS Gmbh, Weinheim, Germany, QDAS Gmbh.

Schenck, D.A. and Wilson, P.R., 1994. Information modeling the EXPRESSway. Oxford University Press.

W3Schools. 2017. W3School Online Web Tutorials. [ONLINE] Available at: https://www.w3schools.com/. [Accessed 14 May 2017].

Thompson, H.S., Beech, D., Maloney, M. and Mendelsohn, N., 2004. XML schema part 1: structures second edition. W3C Recommendation.

IMTI (2006) A roadmap for metrology interoperability. Integrated Manufacturing Technology Initiative (IMTI Inc.), Oak Ridge

Biron, P., Malhotra, A. and World Wide Web Consortium, 2004. XML schema part 2: Datatypes. World Wide Web Consortium Recommendation REC-xmlschema-2-20041028.

JT File Format Reference Version 9.5 Rev-A, 2010

Carl ZEISS, Calypso CMM Programming Manual User Guide http://www.dmsc-inc.org/

Curtis W. Brown, Karen Kontos (2014); Manufacturing Quality Information Framework, an Interoperability Standard for Digital Product Verification, Burleson, Texas, USA, Dimensional Metrology Standard Consortium.

Bhandarkar, M.P., Downie, B., Hardwick, M. and Nagi, R., 2000. Migrating from IGES to STEP: one to one translation of IGES drawing to STEP drafting data. Computers in industry, 41(3), pp.261-277.

Walmsley, P., 2001. Definitive XML schema. Pearson Education.

Morse, E., Heysiattalab, S., Barnard-Feeney, A. and Hedberg, T., 2016. Interoperability: linking design and tolerancing with metrology. Procedia CIRP, 43, pp.13-16.

Morse, E., Heysiattalab, S., 2016. From STEP to QIF: Product and Manufacturing Information. ASPE 31st Annual Meeting, pp. 312-317.

Evans, C., 1989. Precision engineering: an evolutionary view. Bedford: Cranfield Press, 1989.

Hocken, R.J. and Pereira, P.H. eds., 2011. Coordinate measuring machines and systems. CRC Press.

Java STEP Data Access Interface. 2017. JSDAI. [ONLINE] Available at: http://www.jsdai.net/. [Accessed 14 May 2017].

Quality Information Framework part1-7 (2016), QIF V2.1., Quality Information Framework- An Integrated Model for Manufacturing Quality Information, Burleson, Texas, USA. Dimensional Metrology Standard Consortium.

Standard, I.S.O., 2012. 1101: 2012 Geometrical Product Specifications (GPS)–geometrical tolerancing–tolerances of form, orientation, location and run-out. International Organization for Standardization, Geneva.

Standard, I.S.O., 2004. 1101: 2004 Geometrical Product Specifications (GPS)–geometrical tolerancing–tolerances of form, orientation, location and run-out. International Organization for Standardization, Geneva.

Standard, I.S.O., 1992. 1101: 1992 Geometrical Product Specifications (GPS)–geometrical tolerancing–tolerances of form, orientation, location and run-out. International Organization for Standardization, Geneva.

American Society of Mechanical Engineers (2009), ANSI Y14.5. Dimensioning and tolerancing ANSI Y14.5, New York, NY; USA, ASME.

American Society of Mechanical Engineers (1994), ANSI Y14.5.M Dimensioning and tolerancing ANSI Y14.5.M, New York, NY; USA, ASME.

Henzold, G., 2006. Geometrical dimensioning and tolerancing for design, manufacturing and inspection: a handbook for geometrical product specification using ISO and ASME standards. Butterworth-Heinemann.

Toteva, P., Vasileva, D. and Mihaylova, N., 2014. Methods for Selection of Measuring Instruments. In-Applied Mechanics and Materials (Vol. 657, pp. 1006-1010). Trans Tech Publications.

Martínez-Pellitero, S., Barreiro, J., Cuesta, E. and Raya, F., 2015. Categorization of Inspection Elements in Coordinates Measurement for a KBE Implementation. Procedia Engineering, 132, pp.1037-1044.

Muelaner, J.E., Cai, B. and Maropoulos, P.G., 2010. Large-volume metrology instrument selection and measurability analysis. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 224(6), pp.853-868.

Vafaeesefat, A. and Elmaraghy, H.A., 2000. Automated accessibility analysis and measurement clustering for CMMs. International Journal of Production Research, 38(10), pp.2215-2231.

Vim, I.S.O., 2004. International vocabulary of basic and general terms in metrology (VIM). International Organization, 2004, pp.09-14.

AIAG, A., 2010. Measurement systems analysis (MSA). Reference Manual, Fourth Edition, The Automotive Industries Action Group, Troy.

APPENDIX A: QIF MEASUREMENT RESOURCES AUXILIARY TYPES

QIF Measurement Resources define many auxiliary complex types that are demonstrated in this appendix.

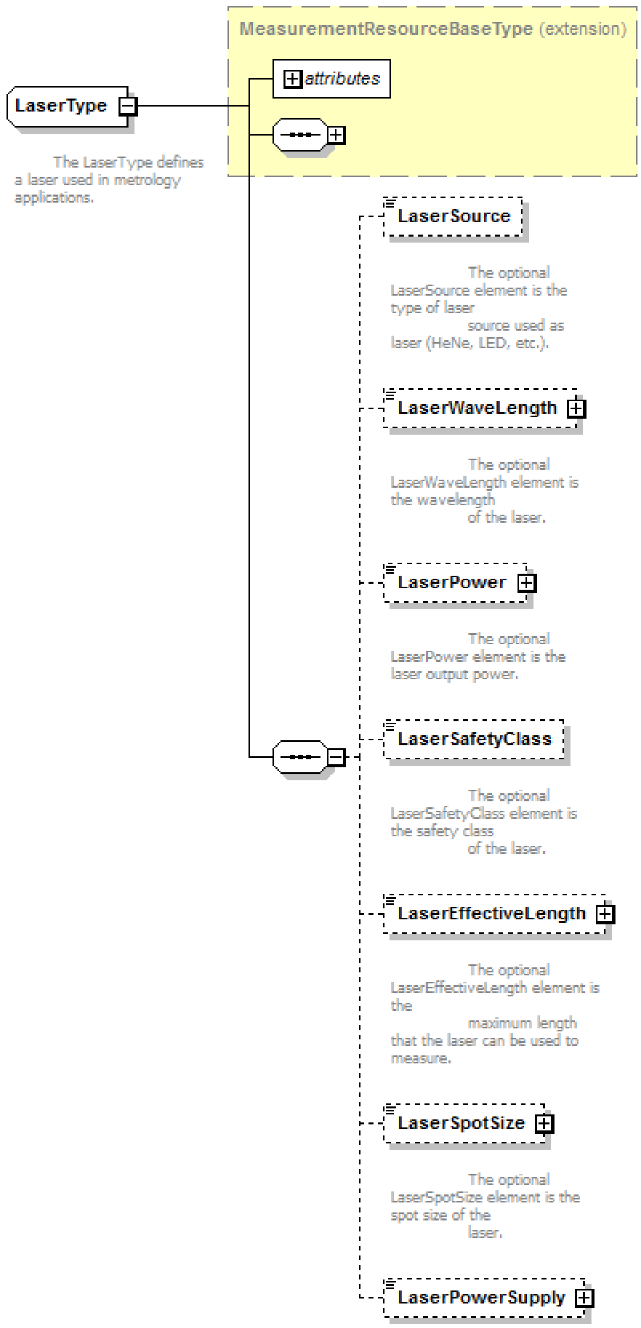


Figure 81 LaserType auxiliary type

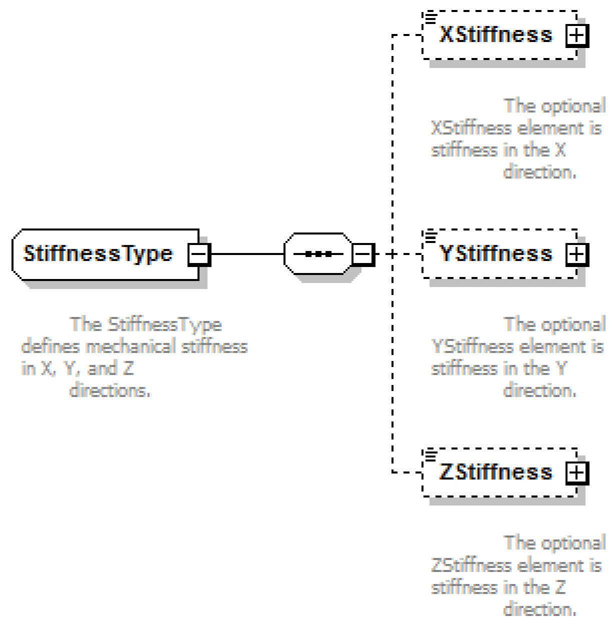


Figure 82 StiffnessType axillary type


MeasurementDeviceType (extension)
attributes
UniversalDeviceType
The UniversalDeviceType defines a universal measurement device. Any nonmanual device that has a working volume, an effective working volume, and a resolution is considered to be a universal device.
Calibrations
The optional Calibrations element is a list of calibrations for a measurement device.
EnvironmentalRange
The optional EnvironmentalRange element describes the range of environments in which the device may be used for producing usable measurements.
Resolution
The optional Resolution element is the resolution of a universal measurement device.
CartesianResolution
The global CartesianResolution element gives information about a Cartesian coordinate system resolution.
LinearResolution
The global LinearResolution element gives information about a linear device resolution.
SphericalResolution
The global SphericalResolution element gives information about a spherical coordinate system resolution.
UserDefinedResolution
The global UserDefinedResolution element gives information about a user defined coordinate system resolution.
WorkingVolume
The optional WorkingVolume element is the working volume of a universal measurement device.
CartesianWorkingVolume
The global CartesianWorkingVolume element gives information about a Cartesian working volume.
ClosedShellSetWorkingVolume
The global ClosedShellSetWorkingVolume element gives information about a working volume that is a set of closed shells.

Figure 83 UniversalDeviceType auxiliary type

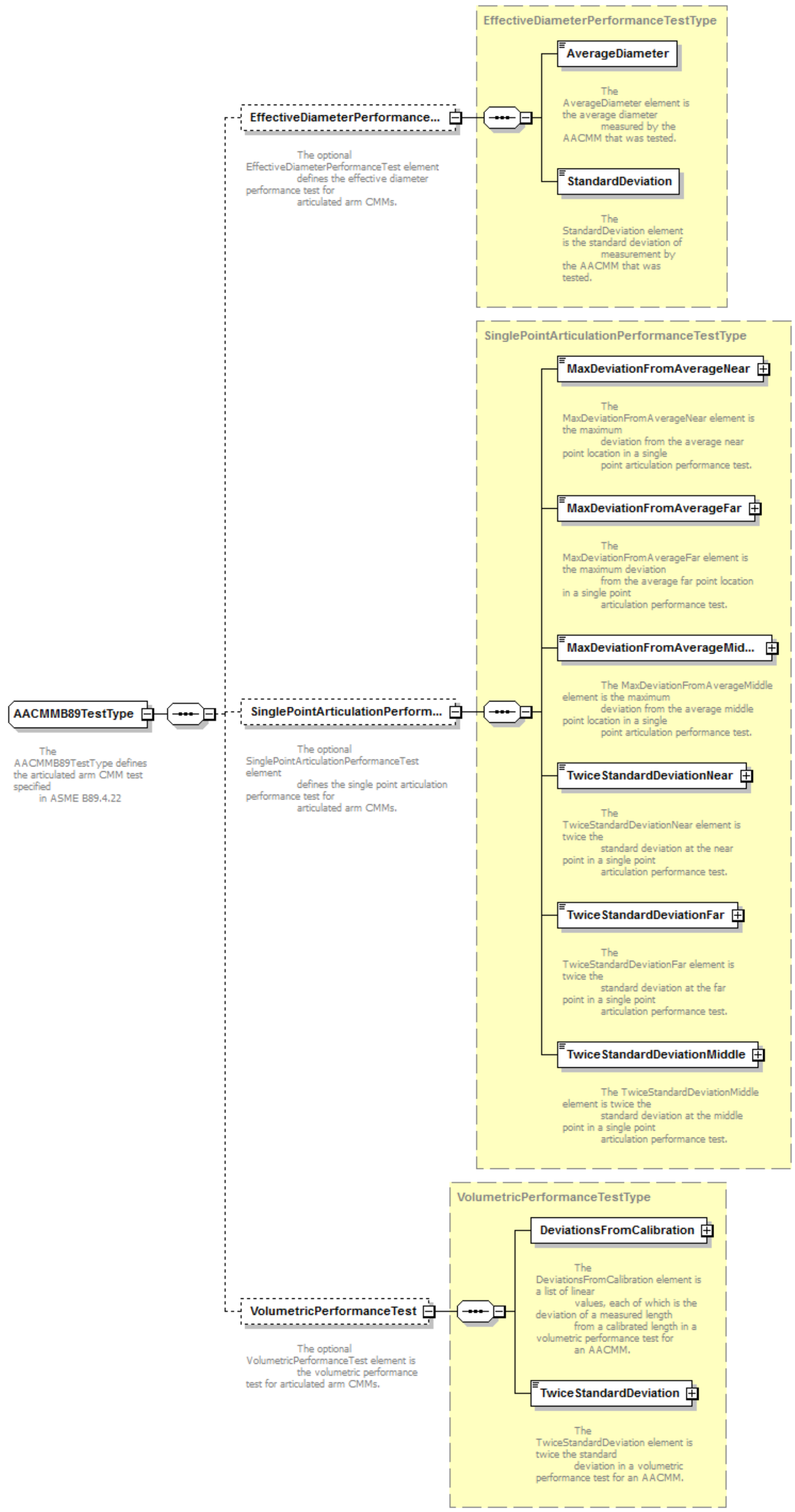


Figure 84 AACMMB89TestType auxiliary type

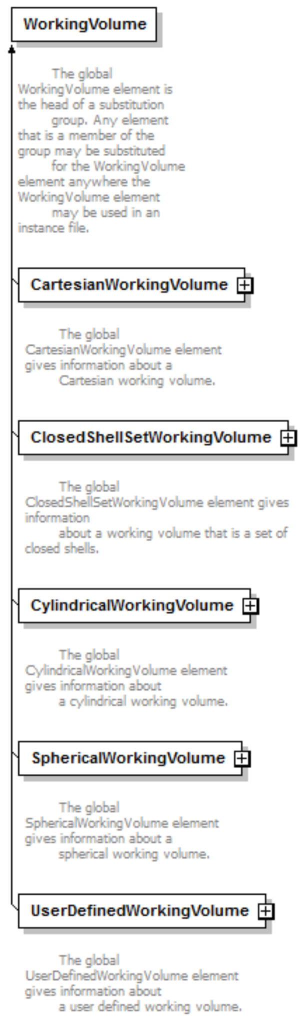


Figure 85 WorkingVolume element substitution group

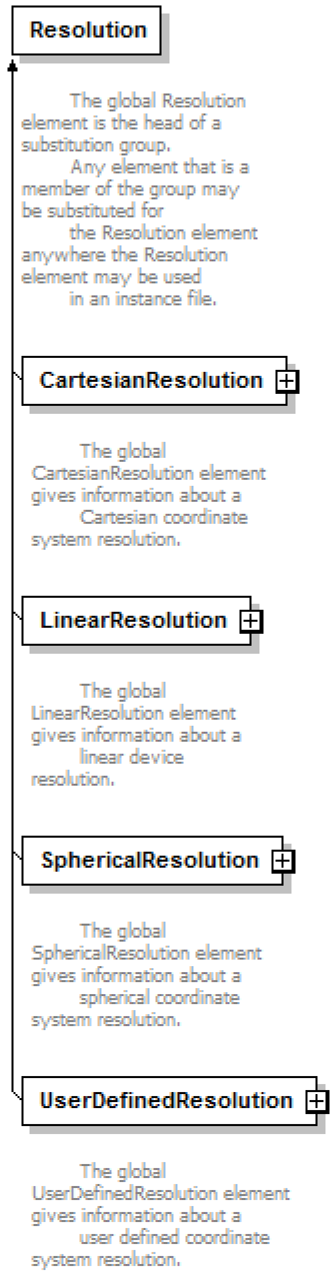


Figure 86 Resolution element substitution group

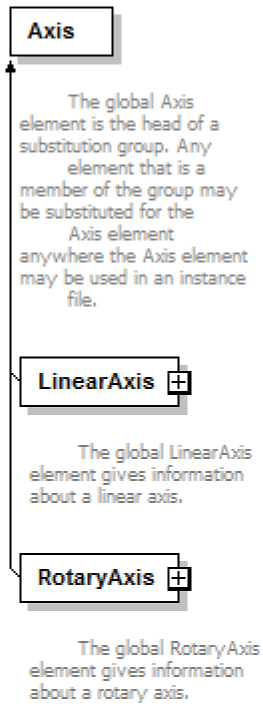


Figure 87 Axis element substitution group

APPENDIX B: QIF MEASUREMENT RESOURCES EXAMPLES

This appendix includes a few examples of dimensional metrology devices defined according to QIF Measurement Resources.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2015 rel. 4 sp1 (x64) (http://www.altova.com)-->
<AACMM xmlns="http://qifstandards.org/xsd/qif2" xmlns:t="http://qifstandards.org/xsd/qif2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="" xsi:schemaLocation="http://qifstandards.org/xsd/qif2
QIFMeasurementResources.xsd">
	<Name>Romer_Arm</Name>
	<Description>AACMM_Located_at_UNCCharlotte</Description>
	<Manufacturer>Hexagon_Metrology</Manufacturer>
	<ModelNumber>RA_7525SEI</ModelNumber>
	<SerialNumber>String</SerialNumber>
	<Mass>8</Mass>
	<Size>
		<XAxisLength>5</XAxisLength>
		<YAxisLength>5</YAxisLength>
		<ZAxisLength>2.5</ZAxisLength>
	</Size>
	<LocationId>1</LocationId>
	<EnvironmentalRange>
		<MaxAmbientTemperature>50</MaxAmbientTemperature>
		<MinAmbientTemperature>0</MinAmbientTemperature>
		<MaxAmbientRelativeHumidity>90</MaxAmbientRelativeHumidity>
		<MinAmbientRelativeHumidity>10</MinAmbientRelativeHumidity>
	</EnvironmentalRange>
	<WorkingVolume xsi:type="t:SphericalWorkingVolume">
		<SphericalWorkingVolume>
			<RadialAxisLength>2.5</RadialAxisLength>
			<MinAzimuthalAxisAngle>0</MinAzimuthalAxisAngle>
			<MaxAzimuthalAxisAngle>360</MaxAzimuthalAxisAngle>
```

```xml
<MinPolarAxisAngle>0</MinPolarAxisAngle>
<MaxPolarAxisAngle>180</MaxPolarAxisAngle>
</SphericalWorkingVolume>
</WorkingVolume>
<TemperatureCompensation>
<TemperatureCompensationEnum>NONE</TemperatureCompensationEnum>
</TemperatureCompensation>
<HomeLocation>0 0 0</HomeLocation>
<NumberOfJoints>3</NumberOfJoints>
<MinMeasuringDistance>0</MinMeasuringDistance>
<MaxMeasuringDistance>2.5</MaxMeasuringDistance>
<Accuracies n="1">
<AACMMAccuracy>
<AACMMB89Test>
<VolumetricPerformanceTest>
<TwiceStandardDeviation>0.058</TwiceStandardDeviation>
</VolumetricPerformanceTest>
</AACMMB89Test>
<AccuracySource>

<AccuracySourceEnum>MANUFACTURER_SPECIFICATION</AccuracySourceEnum>
</AccuracySource>
</AACMMAccuracy>
</Accuracies>
</AACMM>
```


Hexagon metrology articulated arm CMM model RA 7525 SEI


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2015 rel. 4 sp1 (x64) (http://www.altova.com)-->
<LaserTracker xmlns="http://qifstandards.org/xsd/qif2" xmlns:t="http://qifstandards.org/xsd/qif2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="" xsi:schemaLocation="http://qifstandards.org/xsd/qif2
QIFMeasurementResources.xsd">
<Name>Leica_Interpherometer</Name>
<Description>Interpherometer_Located_at_UNCCharlotte</Description>
<Manufacturer>Leica</Manufacturer>
```

```xml
<ModelNumber>AT901-B</ModelNumber>
<SerialNumber>String</SerialNumber>
<Mass>39</Mass>
<Size>
	<XAxisLength>620</XAxisLength>
	<YAxisLength>290</YAxisLength>
	<ZAxisLength>240</ZAxisLength>
</Size>
<LocationId>1</LocationId>
<EnvironmentalRange>
	<Name>NMTOKEN</Name>
	<Description>String</Description>
	<MaxAmbientTemperature>0</MaxAmbientTemperature>
	<MinAmbientTemperature>40</MinAmbientTemperature>
	<MaxAmbientRelativeHumidity>90</MaxAmbientRelativeHumidity>
	<MinAmbientRelativeHumidity>10</MinAmbientRelativeHumidity>
</EnvironmentalRange>
<Resolution xsi:type="t:SphericalResolutionType">
	<RAPZResolution>
		<RadialResolution>0.00032</RadialResolution>
		<AzimuthalAngleResolution>0.14</AzimuthalAngleResolution>
		<PolarAngleResolution>0.14</PolarAngleResolution>
	</RAPZResolution>
</Resolution>
<WorkingVolume xsi:type="SphericalWorkingVolumeType">
	<SphericalWorkingVolume>
		<RadialAxisLength>160</RadialAxisLength>
		<MinAzimuthalAxisAngle>0</MinAzimuthalAxisAngle>
		<MaxAzimuthalAxisAngle>360</MaxAzimuthalAxisAngle>
		<MinPolarAxisAngle>-45</MinPolarAxisAngle>
		<MaxPolarAxisAngle>45</MaxPolarAxisAngle>
	</SphericalWorkingVolume>
</WorkingVolume>
<EffectiveWorkingVolume xsi:type="EffectiveSphericalWorkingVolumeType">
	<MaxRadialAxis>160</MaxRadialAxis>
```

<MinAzimuthalAxisAngle>0</MinAzimuthalAxisAngle>
<MaxAzimuthalAxisAngle>360</MaxAzimuthalAxisAngle>
<MinPolarAxis>-45</MinPolarAxis>
<MaxPolarAxis>45</MaxPolarAxis>
</EffectiveWorkingVolume>
<TemperatureCompensation>
<TemperatureCompensationEnum>NONE</TemperatureCompensationEnum>
</TemperatureCompensation>
<MaximumAngularVelocity unitName="meter per second">6</MaximumAngularVelocity>
<DistanceAccuracy>
<BaseError>0</BaseError>
<ErrorRate>0</ErrorRate>
</DistanceAccuracy>
<AngularAccuracy>
<BaseError>0</BaseError>
<ErrorRate>0</ErrorRate>
</AngularAccuracy>
<SamplingRate unitName="PointsPerSecond">3000</SamplingRate>
<TargetMirror>CUBE_CORNER</TargetMirror>
<Laser id="Laser">
<Name>LASER</Name>
<Description>String</Description>
<Manufacturer>String</Manufacturer>
<ModelNumber>String</ModelNumber>
<SerialNumber>String</SerialNumber>
<Mass>22</Mass>
<LocationId>1</LocationId>
<LaserWaveLength>0.633</LaserWaveLength>
<LaserSafetyClass>IEC 60825-1-Second Edition (2007-03)</LaserSafetyClass>
</Laser>
</LaserTracker>

Leica Laser Tracker model AT 901-B

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by SAEED HEYSIATTALAB for Cartesian CMMs-->
<CartesianCMM xmlns="http://qifstandards.org/xsd/qif2" xmlns:t="http://qifstandards.org/xsd/qif2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="" xsi:schemaLocation="http://qifstandards.org/xsd/qif2
QIFMeasurementResources.xsd">
	<Name>ZEISS_PRISMO_NAVIGATOR_ULTRA</Name>
	<Description>Located_in_Center_for_Precision_Metrology_at_UNCC</Description>
	<Manufacturer>CARL_ZEISS</Manufacturer>
	<ModelNumber>7/9/5</ModelNumber>
	<Mass>1700</Mass>
	<Size>
		<XAxisLength>1568</XAxisLength>
		<YAxisLength>1750</YAxisLength>
		<ZAxisLength>2940</ZAxisLength>
	</Size>
	<LocationId>1</LocationId>
	<EnvironmentalRange>
		<Description>String</Description>
		<MaxAmbientTemperature>22</MaxAmbientTemperature>
		<MinAmbientTemperature>18</MinAmbientTemperature>
		<MaxAmbientRelativeHumidity>70</MaxAmbientRelativeHumidity>
		<MinAmbientRelativeHumidity>40</MinAmbientRelativeHumidity>
	</EnvironmentalRange>
	<Resolution xsi:type="CartesianResolutionType">
		<XYZResolution>
			<XResolution>0.02</XResolution>
			<YResolution>0.02</YResolution>
			<ZResolution>0.02</ZResolution>
		</XYZResolution>
	</Resolution>
	<WorkingVolume xsi:type="CartesianWorkingVolumeType">
		<CartesianWorkingVolume>
			<XAxisLength>885</XAxisLength>
			<YAxisLength>1220</YAxisLength>
			<ZAxisLength>585</ZAxisLength>
```

</CartesianWorkingVolume>
</WorkingVolume>
<EffectiveWorkingVolume xsi:type="EffectiveCartesianWorkingVolumeType">
<EffectiveCartesianWorkingVolume>
<MinPoint>10 15 10</MinPoint>
<MaxPoint>875 1200 570</MaxPoint>
</EffectiveCartesianWorkingVolume>
</EffectiveWorkingVolume>
<TemperatureCompensation>
<TemperatureCompensationEnum>NONE</TemperatureCompensationEnum>
</TemperatureCompensation>
<HomeLocation>0 0 0</HomeLocation>
<CMMGeometry>
<CartesianCMMGeometryEnum>MOVING_BRIDGE</CartesianCMMGeometryEnum>
</CMMGeometry>
<CMMAxisDirections/>
<Scales/>
<MaxWorkpieceMass>1200</MaxWorkpieceMass>
<JoystickSpeeds>
<MaxXTraverseSpeed>70</MaxXTraverseSpeed>
<MaxYTraverseSpeed>70</MaxYTraverseSpeed>
<MaxZTraverseSpeed>70</MaxZTraverseSpeed>
<MaxXProbingSpeed>350</MaxXProbingSpeed>
<MaxYProbingSpeed>350</MaxYProbingSpeed>
<MaxZProbingSpeed>350</MaxZProbingSpeed>
</JoystickSpeeds>
<RotaryTable>
<LocationOnCMM>3.14159265358979 3.14159265358979 3.14159265358979</LocationOnCMM>
<AxisDirection>3.14159265358979 3.14159265358979 3.14159265358979</AxisDirection>
<ZeroIndexDirection>3.14159265358979 3.14159265358979
3.14159265358979</ZeroIndexDirection>
<TableRadius>0</TableRadius>
<TableErrors>
<AxialError>0</AxialError>
<RadialError>0</RadialError>

```
                <TangentialError>0</TangentialError>
            </TableErrors>
        </RotaryTable>
        <Accuracies n="">
            <CartesianCMMAccuracy>
                <EnvironmentalRange>
                    <Name>NMTOKEN</Name>
                    <Description>AccordingtoDMECatalogue</Description>
                    <MaxAmbientTemperature>18</MaxAmbientTemperature>
                    <MinAmbientTemperature>22</MinAmbientTemperature>
                </EnvironmentalRange>
                <CartesianCMMAccuracyTest xsi:type="t:ISO10360TestType">
                    <LinearError>
                        <BaseError>0.9</BaseError>
                        <ErrorType>0.00285</ErrorType>
                    </LinearError>
                </CartesianCMMAccuracyTest>
                <AccuracySource>

        <AccuracySourceEnum>MANUFACTURER_SPECIFICATION</AccuracySourceEnum>
                </AccuracySource>
            </CartesianCMMAccuracy>
        </Accuracies>
        <ToolIds n="">
            <Id>1</Id>
        </ToolIds>
        <SensorIds n="">
            <Id>1</Id>
        </SensorIds>
    </CartesianCMM>
```


ZEISS PRISMO Navigator

APPENDIX C: STEP AND QIF CORRELATIONS

This appendix includes two tables about different STEP entities and equivalent QIF complex types used in chapter 5 for the translator.

Table 15 Comparing STEP entities with QIF Complex Types

QIF Complex Types	STEP Entities
PointEntityType	cartesian_point
Segment12Type	line
ArcConic12Type	ellipse, parabola, hyperbola
ArcCircular12Type	circle
Nurbs12Type	b_spline_curve_with_knots + rational_b_spline_curve
Spline12Type	b_spline_curve_with_knots
Aggregate12Type	combination of 2D Curves
Polyline12Type	line (Combination)
Segment13Type	line
ArcConic13Type	ellipse, parabola, hyperbola
ArcCircular13Type	circle
Nurbs13Type	b_spline_curve_with_knots + rational_b_spline_curve
Spline13Type	b_spline_curve_with_knots
Aggregate13Type	combination of 2D Curves
Polyline13Type	line (Combination)

Table 15, continued

Nurbs13Type	b_spline_curve_with_knots + rational_b_spline_curve
Spline13Type	b_spline_curve_with_knots
Revolution23Type	surface_of_revolution
Extruded23Type	surface_of_linear_extrusion
Ruled23Type	curve_bounded_surface
Offset23Type	offset_surface
Plane23Type	plane
Cone23Type	conical_surface
Cylinder23Type	cylindrical_surface
Sphere23Type	spherical_surface
Torus23Type	toroidal_surface
PathTriangulationType	---
MeshTriangleType	---
VertexType	vertex_point
EdgeType	edge_curve
LoopType	edge_loop
FaceType	advanced_face
ShellType	closed_shell
BodyType	manifold_solid_brep
LoopMeshType	---
FaceMeshType	---
EdgeOrientedType	oriented_edge

Table 15, continued

PointCloudType	---

Table 16 Comparing PMI in STEP and QIF

QIF Complex Types	STEP Entities
StraightnessCharacteristicAspectType	straightness_tolerance
FlatnessCharacteristicAspectType	flatness_tolerance
CircularityCharacteristicAspectType	circularity_tolerance
CylindricityCharacteristicAspectType	cylindricity_tolerance
AngularityCharacteristicAspectType	angularity_tolerance
ParallelismCharacteristicAspectType	parallelism_tolerance
PerpendicularityCharacteristicAspectType	perpendicularity_tolerance
PositionCharacteristicAspectType	position_tolerance
ConcentricityCharacteristicAspectType	concentricity_tolerance
SymmetryCharacteristicAspectType	symmetry_tolerance
LineProfileCharacteristicAspectType	line_profile_tolerance
PointProfileCharacteristicAspectType	-----
SurfaceProfileCharacteristicAspectType	surface_profile_tolerance
SurfaceProfileNonUniformCharacteristicAspectType	
CircularRunoutCharacteristicAspectType	circular_runout_tolerance
TotalRunoutCharacteristicAspectType	total_runout_tolerance
ThreadCharacteristicAspectType	-----
SurfaceTextureCharacteristicAspectType	-----

AngleCharacteristicsAspectType	Angular_size/ Dimensional_size_with_path
AngleBetweenCharacteristicsAspectType	Angular_location/ Dimensional_location_with_path
AngleFromCharacteristicsAspectType	Angular_location/ Dimensional_location_with_path
WidthCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
ThicknessCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
SquareCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
RadiusCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
LengthCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
HeightCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
DistanceFromCharacteristicAspectType	Dimensional_location/ Dimensional_location_with_path

Table 16, continued

DistanceBetweenCharacteristicAspectType	Dimensional_location/ Dimensional_location_with_path
DiameterCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
DepthCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
CurveLengthCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
ChordCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path
LinearCoordinateCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path / Dimensional_location/ Dimensional_location_with_path
AngularCoordinateCharacteristicAspectType	Dimensional_size/ Dimensional_size_with_path / Dimensional_location/ Dimensional_location_with_path
DatumDefinitionType	datum
LinearToleranceType	Plus_minus_tolerance

Table 16, continued

DatumTargetDefinitionType	Placed_datum_target_feature
ProjectedToleranceZoneValue (as element, not complex type)	Projected_zone_definition
SurfaceProfileNonUniformCharacteristicDefinitionType	Non_uniform_zone_definition

Table 17 QIF and STEP Features correlation

QIF features	STEP entities
Circle	circle
CircularArc	circle
Composite	composite_curve
Cone	conical_surface
ConicalSegment	conical_surface
Cuboid	surface_of_linear_extrusion
Cylinder	cylindrical_surface
CylindricalSegment	cylindrical_surface
EdgePoint	Edge
Ellipse	ellipse
EllipticalArc	ellipse
ElongatedCylinder	surface_of_linear_extrusion
ExtrudedCrossSection	surface_of_linear_extrusion
Generic	-----
Line	line
OppositeAngledLines	item_identified_representation_usage combined with line entity
OppositeParallelLines	item_identified_representation_usage combined with line entity
OppositeAngledPlanes	item_identified_representation_usage combined with plane entity
OppositeParallelPlanes	item_identified_representation_usage combined with plane entity
Plane	plane
Point	cartesian_point
PointDefinedCurve	point_on_curve
PointDefinedSurface	point_on_surface
Sphere	spherical_surface
SphericalSegment	spherical_surface
SurfaceOfRevolution	surface_of_revolution

Table 17, continued

Threaded	------
ToroidalSegment	toroidal_surface
Torus	toroidal_surface

APPENDIX D: QIF MEASUREMENT RESOURCES INHERITANCE DIAGRAM

The following pictures shows the overall inheritance diagram of QIF Measurement Resources.

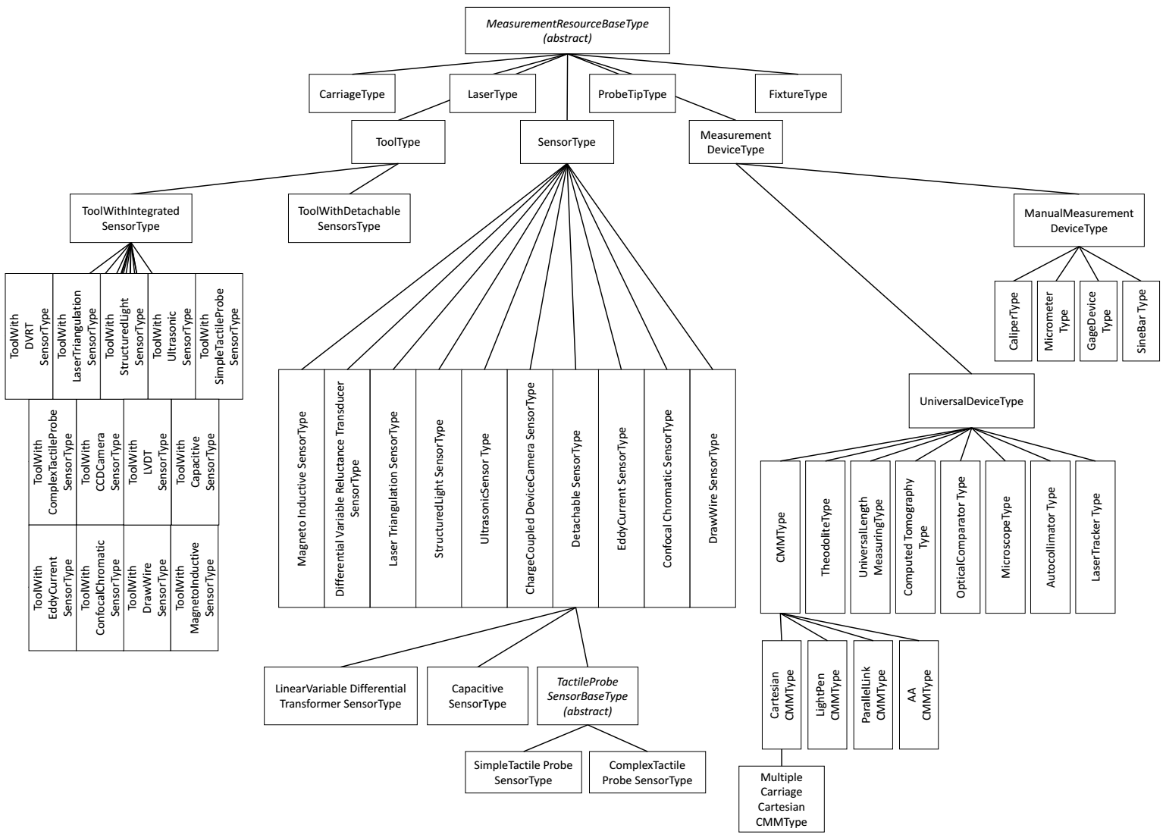


Figure 88 QIF Measurement Resources Inheritance Diagram (From QIF/ ANSI V2.1)