Quality Information Framework – Integrating Metrology Processes

Yaoyao Fiona Zhao*, John A Horst*, Thomas R. Kramer*, William Rippey*, Robert J. Brown**

**National Institute of Standards and Technology, Gaithersburg, MD 20899*
USA (Tel: 301-975-3417; e-mail:william.rippey@nist.gov).
***Mitutoyo America Corporation, Aurora, IL 60502,*
USA (Tel: 630-723-3581; e-mail: Robert.Brown@mitutoyo.com)

Abstract: As defined by major dimensional metrology system users and suppliers, the Quality Information Framework (QIF) is an integrated and holistic set of information models which, if widely adopted, can enable the effective exchange of metrology data throughout the entire manufacturing quality measurement process – from design to planning to execution to analysis. This paper introduces the philosophy and rationale behind QIF, as well as some of its detailed content. Past standards efforts in manufacturing quality systems have had a variety of shortcomings, which QIF plans to overcome. The QIF data models have been encoded in Extensible Modeling Language (XML) schemas. Schemas developed during the first year of the QIF project include the QIF schema library and the Quality Measurement Results (QMResults) schema. QIF models quality characteristics and measurement features as defined in the ASME Y14.5 specification, and is able to cover use cases including reverse engineering, batch quality measurement, and discrete quality measurement. Correct semantic associations between measurement feature and quality characteristics, and between nominal values and actual values, are guaranteed by implementing strong typing using identifiers. The next step of the QIF project is to conduct a set of pilot tests to validate the information models.

Keywords: Quality Information Framework (QIF), manufacturing quality, interoperability, data model, XML schema, measurement features, quality characteristics, ASME Y14.5

1. INTRODUCTION

This paper describes a new non-industry-specific framework of interoperable standards to facilitate the exchange of manufacturing quality data between components of manufacturing quality systems. It is named the Quality Information Framework (QIF), shown in Fig. 1. QIF will provide metrology information interoperability using industry consensus standards to communicate data between component producers and consumers of manufacturing quality systems. QIF will provide a data model for the streamlined development, integration, and support of manufacturing quality systems and components, while maintaining the scalability necessary to adapt to an ever-changing manufacturing quality landscape.

Past quality standards and specifications have been scoped narrowly, each targeting a single dimension of a quality system. QIF is different, in that it will consist of individual application area standards (black boxes in Fig. 1) supported by common data types and generic structures to promote re-use and inheritance throughout the QIF, thus ensuring compatibility between the standards. QIF appears to be the first standards effort seeking to model the **entire** quality measurement process (planning through analysis) for **all** types of quality measurements (dimensional, non-dimensional, and attribute), and which also plans to be consistent with upstream standards such as ISO STEP design models (ISO 1994) and AS9102 first article inspection (SAE 2004).

One of the unique features of the QIF data model is that the semantic connections between features and characteristics, and also between nominals and actuals are guaranteed by implementing strong type identifiers. The QIF data model also models measurement features and characteristics with four main elements: instance, definition, nominal, and actual. This method ensures that QIF library and Quality Management Results (QMResults) schema can be used for both a reverse engineering process where actual measurement data is stored without the presence of nominal information because it is unknown, and a conventional measurement process where actual measurement data is stored and nominal data is known and present.

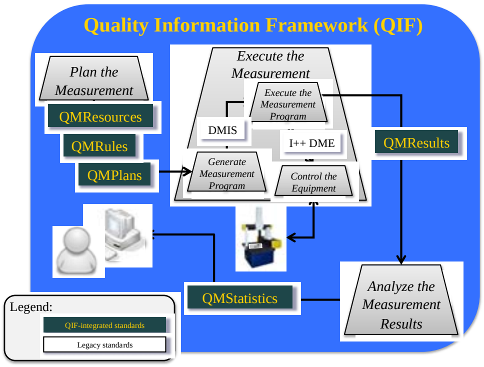


Fig. 1 A quality measurement activity diagram with QIF application area standards (black boxes).

Solving the metrology interoperability problem will benefit manufacturers by avoiding wasted resources spent on non-value-added costs related to the translation of data between the different components of manufacturing quality systems. The automotive industry alone reported that costs due to translation of measurement data between manufacturing quality systems amounted to over $600 million annually (IMTI, 2006). Other cost benefits include a reduction in the price of quality system components through newly introduced competition between vendors of interoperable components.

A primary benefit of adoption of QIF by solutions providers is seen as reduced resources needed for systems integration in all industries that implement dimensional metrology systems. Further, QIF can facilitate commercially available components that are interoperable, allowing users to buy the products that suit their individual business models. Interoperability through standard interfaces may also allow providers of niche quality applications to compete with larger vendors.

QIF is championed by the Dimensional Metrology Standards Consortium (DMSC) (DMSC 2011), a consortium of businesses and experts representing a wide range of manufacturing industries. The DMSC is an ANSI accredited standards developing organization and has successfully maintained, enhanced, and progressed standards such as the Dimensional Measuring Interface Standard (DMIS) (ANSI 2004) as ANSI 105.2-2009 and ISO 22093.

2. BACKGROUND

Several past standards efforts addressing manufacturing quality data interoperability have focused only on pieces of the total manufacturing quality system (Zhao et al., 2011). Manufacturing quality systems can be generally categorized into four sub-systems, namely product definition, measurement process planning, measurement process execution, and measurement results reporting as shown in Fig. 2. The past standards efforts primarily focused on modeling information passing across one of the interfaces among the four manufacturing quality sub-systems. This section will give a brief overview of these standards efforts.

In the product definition process, the most accepted standards are: the Initial Graphics Exchange Specification (IGES) (IGES, 1980) and the Standard for the Exchange of Product model data (STEP) (ISO 1994) standards. Within STEP, various Application Protocols (APs) were developed to describe product data for different sections of manufacturing processes. STEP AP 203 (ISO 2007a) models 3D product design information. The first edition of STEP AP 203 does not have sufficient geometric dimensioning and tolerancing (GD&T) information to support automated processing of information by downstream quality processes. A newer version of STEP AP 203 – AP 203 edition 2 (ISO 2009a), which models both annotated and semantic GD&T information in 3D product design, is close to publication to address these deficiencies.

However, the data model still needs to be tested and validated before it is released to the public. The most recent validation test of STEP AP 203 edition 2 was carried out by some major CAD vendors. The GD&T definition from AP 214 (ISO, 2001a) (Core data for automotive mechanical design process) was harmonized with AP 203 edition 2. These GD&T definitions are mainly for annotation purposes; therefore they are not sufficient for automatic generation of dimensional measurement process plans. These definitions should be harmonized and eventually adopted into AP 203 in a form usable by computer-aided or automated process planning systems. Only in this way will AP 203 be able to provide adequate information for generating measurement process plans.

DMIS is the only standard that combines measurement features and operation instruction information within the same measurement process definition. It is a language for controlling dimensional measuring equipment and includes an input and an output language. Part of the DMIS input language defines features, tolerances, sensors, etc. The output language serves both as a log of action commands and settings and a report of results, with actual and nominal point data, features, and tolerances. However, it does not define complete measuring equipment resources. Measuring equipment resource data is necessary to complete the effectiveness of DMIS.

STEP AP 219 (ISO 2007b) specifies an application protocol for the exchange of information resulting from the dimensional inspection of solid parts, which includes administering, planning, and executing dimensional inspection as well as analyzing and archiving the results. AP 219 is inadequate in providing complete definitions of dimensional measurement features, dimensional measurement results collections, and analysis methods. There are many entities in AP 219 that were left empty for further development. There is no known industry implementation of AP 219.

As for the interface between measurement process execution and equipment control, there are two publicly available specifications/standards, one of which is formalized as an official ANSI and ISO standard – the equipment module of DMIS Part 2 (ANSI 2003). The other is the I++DME Interface Specification (I++DME, 2005) which is a specification for dimensional measuring equipment information exchange developed by several European automakers and measuring equipment vendors. There are no known product implementations of DMIS Part 2. There are many software implementations of I++DME worldwide, but it is not yet ubiquitous for either coordinate measuring machine (CMM) software or CMM systems to offer I++DME in their published product offerings. Several vendors have I++DME simulators available to enable quick and accurate development of I++DME implementations within measurement process execution software.

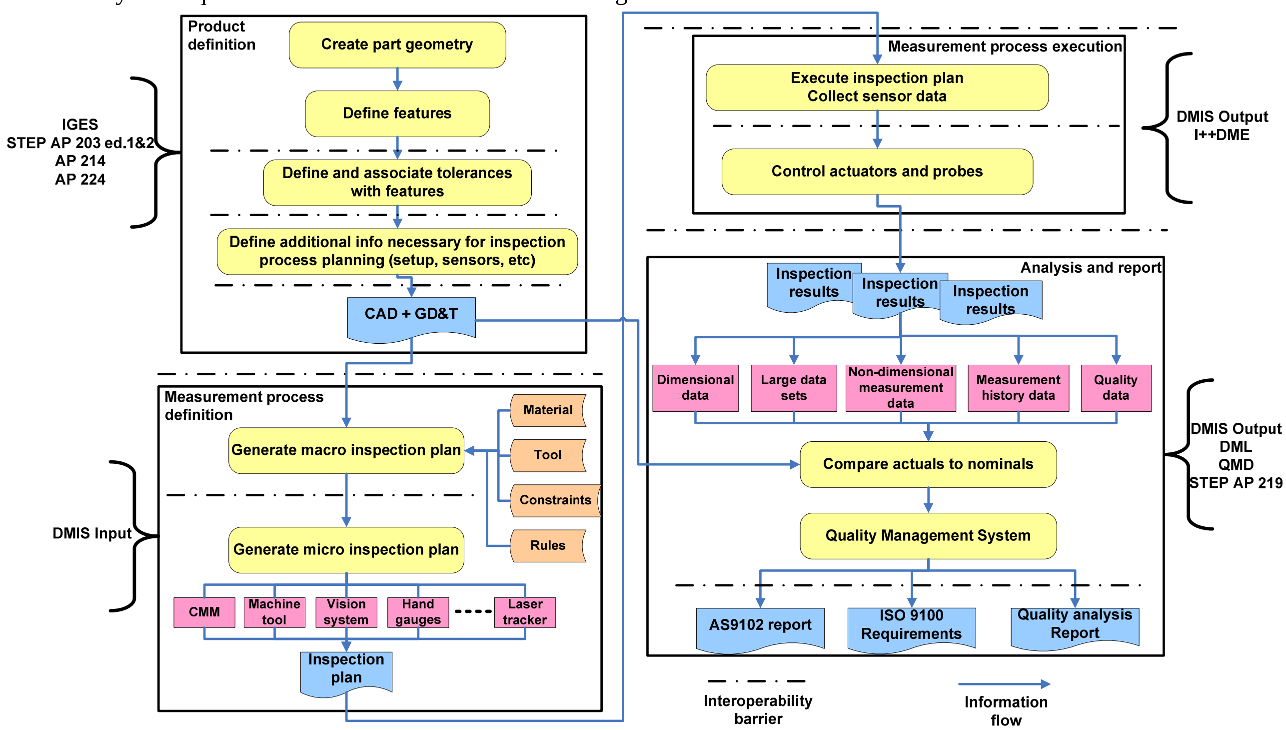


Fig. 2 Existing standards efforts in manufacturing quality systems

The ASME B5.59 (ANSI 2009a, ANSI 2009b) series should be assessed to explore the applicability to defining CMM configurations. CMM machine types and configurations are defined in the ISO 10360 series (ISO 2000a, ISO 2009b, ISO 2000b, ISO 2000c, ISO 2010, ISO 2001b). However, these definitions are in human readable format. A standard digital model in compliance with these standards needs to be developed and validated, enabling industry to develop implementations in software modules.

Dimensional Markup Language (DML) (DML 2009), DMIS Output, AP 219, and Quality Measurement Data (QMD) (QMD 2009) are specifications/standards for measurement results reporting. DML is having moderate usage largely in North America. A format for CMM measurement results is defined within DMIS, and has enjoyed some usage, wherever DMIS is used. Within the STEP effort, AP 219 was defined to cover all important metrology information, including, but not limited to, measurement results. As mentioned earlier, the latest ISO standard version of AP 219 only defines measurement results information. The QMD Data Model describes a non-proprietary and open standard XML schema for variable, attribute, and binary quality measurements, including non-dimensional measurements and gage measurements. QMD targets quality measurements from measurement devices other than CMMs. The standard is unidirectional – it defines the measurement export only. There are multiple standards and/or specifications that define traceability data such as DMIS, DML, and ISO 10303 AP 238 (ISO 2004). However, the link between traceability and measurement data is insufficient.

3. USE CASES FOR QIF

As mentioned earlier, the metrology process can be modeled as a set of distinct components each with specific behavior and specific information in/out. The upstream information coming into a metrology planning process is from the product definition processes (e.g., STEP) and quality management activities.

To help determine the precise nature of the information landscape for quality measurement, use cases have been created for several activities. Use cases, as defined by knowledgeable metrologists and quality engineers, explicate the various ways quality measurement is actually done in real and varied manufacturing operations. With this knowledge, an information modeller can define a set of schema definitions that will be useful to quality measurement systems users. Use case development is a proposed method to ensure full information coverage and that information boundaries for different conformance classes are clearly defined and validated against real-world use cases.

4. STRUCTURE OF QIF

The contents of the QIF schemas were determined by building use case scenarios coupled with full coverage of ASME Y14.5 2009 (ANSI 2009c) and DMIS 5.2. The high level architecture of the schemas includes (see Fig. 1):

- QMPlans.xsd - inspection planning information
- QMResults.xsd - single part measurement results, typically from coordinate measuring machines, manual instruments, or on-machine inspection
- QMStatistics.xsd - processed measurement data from more than one part inspection
- QMRules.xsd and QMResources.xsd, which define external metrology resource and method information,
- and a set of nine library schemas (the QIF library).

QIF library schemas are referenced by the application area schemas using the XML schema's "include" statement. The first release of QIF specifications, V0.9, included QMResults.xsd and the QIF library, as shown in Figure 3. The other application area standards are under development.

XML technology was chosen for QIF encoding because the basic XML specifications are supported as open, public domain, royalty-free standards, and because XML is very widely used: educated experts and users are easy to find. Furthermore, tools for incorporating XML into software projects are widely available.

QMResults.xsd was released in 2011, and is believed to be technically complete. It is expected that testing and debugging will add only minor items. Testing the QMResults schema began in October 2011, when sample test files were first built. A second top-level schema, QMPlans.xsd, has been drafted but, as of February 2012, is not technically complete..

A large information model is analogous to a large computer program in that the first complete version is rarely bug-free. Processing the initial test files for QMResults revealed a number of bugs that were subsequently fixed. Proving that QMResults works as intended and making it robust requires that quality components be tested. This will undoubtedly reveal additional bugs and needs for tweaks in functionality. XML schema processing tools were used to automatically validate QIF schema files. Automatic validity checking is extremely valuable as a strong first line of defense against bugs, but the XML schema language, by virtue of providing a great deal of flexibility, also allows semantic errors that do not show up in automatic validations. Those errors need to be found by examination and testing.

Since schemas may be combined using XML schema's "include", schemas may be modularized (split up) in almost any way. The modularization of the QIF schemas was done based on two main principles (1) keep groups of types together that serve the same function, and (2) try to anticipate what the top level schemas will need to include. A minor principle has also been applied: "includes" should not form loops (where a chain of includes leads back to the beginning). After the initial design of modules, a units schema was added and some remodularization was done. The current module structure for QMResults.xsd is shown in Figure 3. In the figure, a schema at one level of indentation includes all the schemas below it at the next level of indentation. There are no loops, but some schemas are included multiple times; this is allowed by XML schema rules and does not confuse XML tools.

QMResults.xsd

- **CharacteristicTypes.xsd**
  - ***PrimitiveTypes.xsd***
  - ***QIFTypes.xsd***
- **ConstructedFeatureTypes.xsd**
  - ***FeatureTypes.xsd***
  - ***PrimitiveTypes.xsd***
- **FeatureTypes.xsd**
  - ***PrimitiveTypes.xsd***
  - ***QIFTypes.xsd***
- **MeasurementDevices.xsd**
  - ***PrimitiveTypes.xsd***
- **PrimitiveTypes.xsd**
  - ***Units.xsd***
- **QIFTypes.xsd**
  - ***PrimitiveTypes.xsd***
- **Traceability.xsd**
  - ***MeasurementDevices.xsd***
  - ***QIFTypes.xsd***
- **Transforms.xsd**
  - ***PrimitiveTypes.xsd***

Fig. 1 Structure of QMResults.xsd

4.1 *QIF Schema Conventions and Documentation*

Certain design conventions have been used in the QIF schemas:

- All type definitions are declared globally, i.e., as direct children of a root schema element. In other words, no type definition is embedded inside another type definition. This convention is commonly called using venetian blind format.
- Names are descriptive and formed by concatenation without abbreviation. All concatenated words except possibly the first start with an upper case letter. Example name: ArcFeatureNominalType
- XML Type and Element names start with an upper case letter. XML Attribute names start with a lower case letter.
- Most type names end in "Type".
- No namespaces are used.

Explanations of the semantics of the QIF schema is provided using the XML schema "documentation" node. More detailed documentation is planned for future revisions.

4.2 QIF Units Schema

All quantities except tolerances in the QIF schemas have a specified type of unit. The Units.xsd schema defines the following value types that have units:

- AngleValueType
- AreaValueType
- ForceValueType

- LengthValueType
- MassValueType
- PressureValueType
- SignedLengthValueType
- SpeedValueType
- TemperatureValueType
- TimeValueType

The Units.xsd schema is implemented such that quantities using any of the unit types can be represented efficiently in data files. This is enabled in a top level schema such as QMResults.xsd or QMPlans.xsd by using the FileUnits element defined in Units.xsd. The FileUnits element, whose type is FileUnitsType, specifies a primary unit for each type. Quantities expressed in the primary unit can be written in a data file without any mention of a unit. Units in QIF are based on SI units. The default unit of any type is the SI unit. If it is desired to have a primary unit type not be a SI unit, the declaration of the primary unit must (1) specify the name of the primary unit and (2) give numerical factors that may be used in converting values of that unit type into values of the SI type.

4.3 *Features*

The FeatureTypes.xsd schema defines 28 feature types. Almost all of these have equivalents in DMIS 5.2. Comparison of feature types in QIF and DMIS 5.2 is shown in Fig. 4. QIF features are described using Cartesian coordinates.

QIF Feature	Equivalent DMIS Feature
Arc	ARC (format 1)
Attribute	GEOM, OBJECT (possibly)
Circle	CIRCLE
Composite	
Compound	COMPOUND
Cone	CONE
ConicalSegment	CONRADSEGMNT
ConstantCrossSection	*no equivalent*
Cuboid	RCTNGL
Cylinder	CYLNDR
CylindricalSegment	CYLRADSEGMNT
EdgePoint	EDGEPT
Ellipse	ELLIPS
ElongatedCylinder	ELONGCYL
Line	LINE
Pattern	PATERN
Plane	PLANE
PointCurve	GCURVE
Point	POINT
PointSurface	GSURF
Slot2D	CPARLN
Slot3D	PARPLN
Slot3DWithDraft	SYMPLN
Sphere	SPHERE
SphericalSegment	SPHRADSEGMNT
SurfaceOfRevolution	REVSURF
ToroidalSegment	TORRADSEGMNT
Torus	TORUS

Fig. 4 Comparison of feature definitions in QIF and DMIS

Features are defined using four ***aspects***: definition, nominal, actual, and instance. These have identifiers and are connected by references to the identifiers. The linking references among the feature aspects are shown in Fig. 5. Notes may be attached to any of the four aspects.

- A **feature definition** includes information that is independent of the position of the feature - the diameter of a circle, for example. A single definition can be referenced by many nominal features. Only nominal features reference feature definitions.
- A **feature nominal** defines a nominal feature by referencing a feature definition and providing position information - the center of a circle and the normal to the plane of the circle, for example.
- A **feature actual** defines an actual feature that has been measured or constructed. Data files typically also include a related nominal feature definition, with a reference linking the two. If feature measurements were made during a reverse engineering process, the data file may not contain a nominal feature related to the actual feature.
- A **feature instance** represents an instance of a feature at any stage of the metrology process - before or after a feature has been measured. The feature instance must reference a nominal feature or an actual feature. If an actual feature is referenced, the corresponding nominal feature (if there is one) may be found through the actual feature. If a feature is measured several times, it is expected that a feature instance will be defined for each measurement and will have a different actual feature for each measurement.

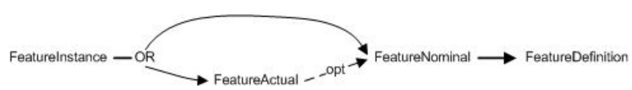


Fig. 5 The reference connections among the four aspects of a feature. Solid lines show required references, dashed lines show optional references.

4.4 Semantic Connecting Types Using Identifiers

Many connections between types in the QIF schemas are made using XML elements that are identifiers rather than types. For example, an instance of a FeatureActualBaseType is connected to an instance of a FeatureNominalBaseType by using an element named FeatureNominalId whose type is FeatureNominalReferenceType and whose value is the identifier for an instance of a FeatureNominalBaseType. Identifiers of the FeatureNominalIdType are exclusive to and are required to be attached to instances of the FeatureNominalBaseType and its derived types. Fig. 6 shows a typical set of connections found in FeatureTypes.xsd. The arrow on the right side of the FeatureActualBaseType box indicates that in an instance of FeatureActualBaseType, the value of the FeatureNominalId (which is of FeatureNominalReferenceType) must be identical to the (FeatureNominalIdType) value of the id of an instance of FeatureNominalBaseType.

ArcFeatureActualType is derived from FeatureActualBaseType. In the derivation, the first four items (elements and attributes) are inherited from FeatureActualBaseType, but the FeatureNominalId element is restricted to being of the ArcFeatureNominalReferenceType (a subtype of

FeatureNominalReferenceType). In addition, the type of id is restricted and some other elements, such as CenterPoint and Normal are added.

Similarly, ArcFeatureNominalType is derived from FeatureNominalBaseType. In the derivation, the first four items are inherited from the FeatureNominalBaseType, but the id attribute is restricted to being of the ArcFeatureNominalIdType (a subtype of FeatureNominalIdType). In addition, the type of FeatureDefinitionId is restricted and some other elements, such as CenterPoint and Normal are added.

The connection between the ArcFeatureActualType and the ArcFeatureNominalType is still made by the FeatureNominalId of former pointing to the id of the latter, but now the types of both are restricted so that the connection is sure to be between entities of the correct type.

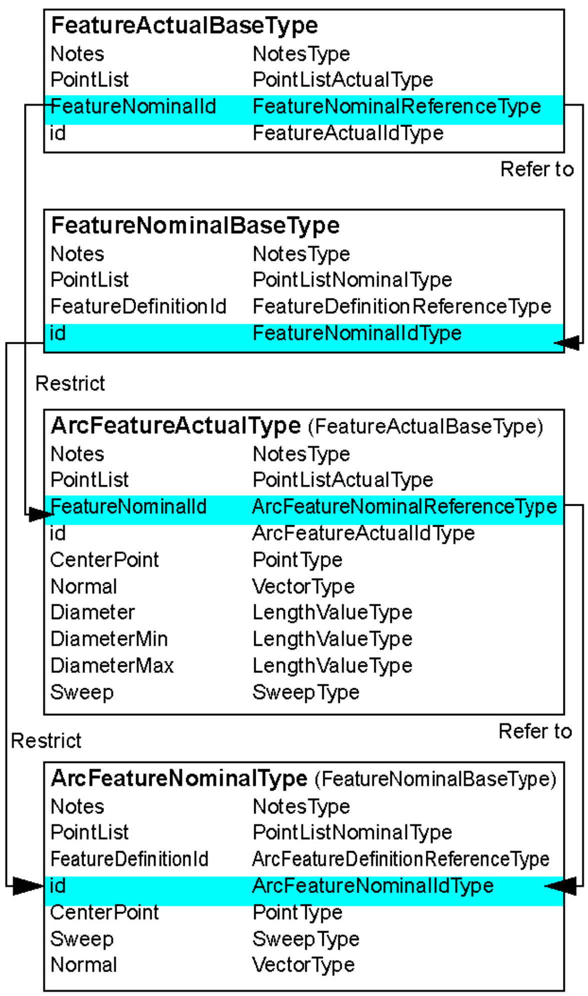


Fig. 6 Semantic connection between feature actual and feature nominal, specialized to the connection between arc feature actual and arc feature nominal

Restricting the type of an element must be done inside an XML *restriction*, while adding new elements to a complex type must be done inside an XML *extension*. In the QIF schemas, this is actually done in two stages as shown in Fig 7. For the ArcFeatureActualType, for example, first the ArcFeatureActualSemiType is derived via restriction from the FeatureActualBaseType and then the ArcFeatureActualType is derived via extension from the ArcFeatureActualSemiType. It would be possible to do the restriction and extension simultaneously by nesting one inside the other in a complexType definition, but that would be much more complex and only slightly less verbose. Deriving the ArcFeatureNominalType from the FeatureNominalBaseType is done similarly in two stages. The patterns exemplified by Fig. 6 and Fig. 7 are repeated throughout FeatureTypes.xsd and CharacteristicTypes.xsd.

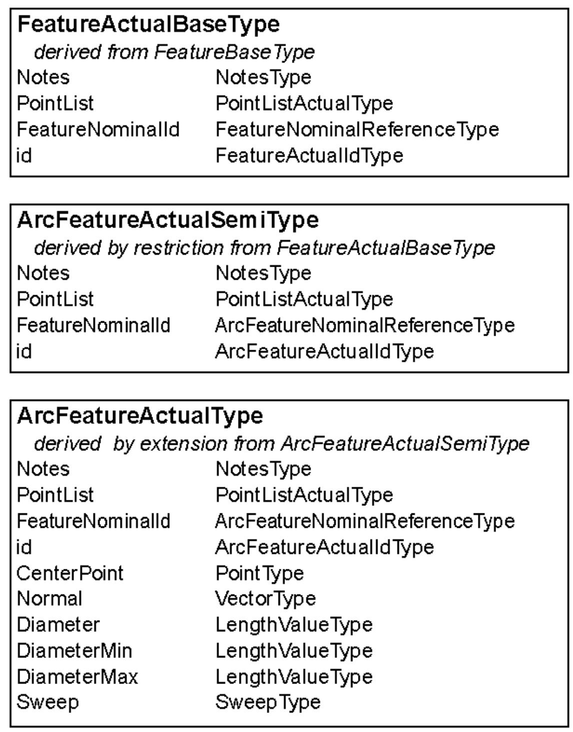


Fig. 7 Derivation of Types in Two Stages

The XML Schema language definitions of features and aspects form a type hierarchy. FeatureBaseType and FeatureInstanceBaseType are at the top of the hierarchy. At the second level, FeatureDefinitionBaseType, FeatureNominalBaseType, and FeatureActualBaseType are derived from FeatureBaseType.

At the bottom level a type is defined for each of the 28 feature types and each of the four aspects. The types for the definition aspect (CircleFeatureDefinitionType, for example) are derived from FeatureDefinitionBaseType. The types for the nominal aspect are derived from FeatureNominalBaseType, and the types for the actual aspect are derived from FeatureActualBaseType. The types for the instance aspect are derived from FeatureInstanceBaseType.

4.5 Characteristics

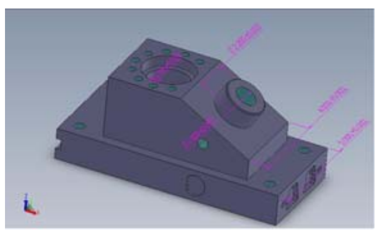
Characteristics (primarily tolerances) are defined in CharacteristicTypes.xsd, the largest of the QIF library schemas. As with features, characteristics have four aspects: definition, nominal, actual, and instance. Also as with features, the XML Schema language definitions for characteristics form a type hierarchy, but the characteristics hierarchy is longer and deeper. There are four hierarchies, one for each aspect. Each of the four is headed by CharacteristicType. The CharacteristicType and all of the types whose name includes "Base" are abstract types and cannot be instantiated. The hierarchy has 39 leaf nodes that are not abstract and can be instantiated.

5. CASE STUDIES

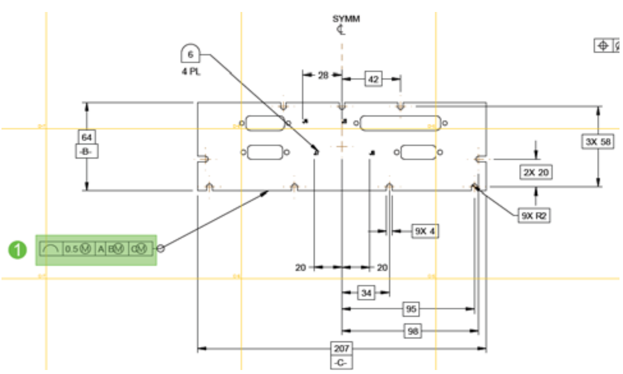
In September 2011, the QMResults.xsd Version 0.9 was published by the QIF QMResults working group, and a series of validation tests were carried out to evaluate correctness and completeness of the specification. This validation analyzed QMResults data files created for sample parts. Currently, the QMResults.xsd schema has been validated to the following three case studies shown in Figure 8:

1) a CMM calibration master ball (Fig. 8 (a)),
2) ANC101 example part (Fig. 8 (b)),
3) Sheet metal scanning measurement example part (Fig. 8 (c)).

(a) Case study 1



(b) Case study 2



(c) Case study 3

Fig. 8 Case studies to validate QMResults.xsd

These validation case studies showed that the QMResults.xsd schema is able to represent different types of measurement features and characteristics. Since the format of results data is independent of how the data was collected, the QMResults.xsd schema is able to convey measurement results from common CMMs and optical or vision scanning devices. This is one of the key capabilities that the QMResults schema must have in order to support interoperability between different types of measurement equipment. For major end users, such as airplane manufacturers and automobile manufacturers, more than one type of measurement device is used on any shop floor. Therefore, QMResults.xsd provides a solution to exchange and collect measurement results seamlessly without costly data translation.

6. CONCLUSIONS AND NEXT STEPS

Based on the development of QIF to date, the DMSC believes that a complete set of specifications will facilitate effortless integration of commercial software solutions for manufacturing quality systems. Exchange of quality data in standard formats is judged to be a good solution to achieving interoperability of multivendor components. Benefits accrued to manufacturers should include flexibility in configuring quality systems and in choosing commercial components, and effortless and accurate flow of data within factory walls as well as with contractors and suppliers. Solution providers should be able to eliminate their efforts previously spent in data translations, and there should be increased opportunities to sell their products and expand features.

In order to facilitate and encourage software applications that manipulate QIF formated data, DMSC proposes to create software libraries (with C++, Java, or .NET) to write data into QMResults XML files (serialization), and/or read data from QMResults XML files (deserialization). This software development will be an open source, public domain project so members of the manufacturing quality community can suggest improvements or improve the code for the benefit of all users. Once the open source libraries are developed, it will be easy for existing software developers to access the data, allowing them to link in the libraries, and avoid the effort of developing code that does import and export to the QMResults schema and QIF structures. The libraries would be a toolkit for extended functionality to better use the base libraries, with continuous improvement through the open source model.

The QIF library and the QMResults schema are at version V0.91. QMResults.xsd has been tested by generating sample part files, to assess correctness and completeness of the specification. DMSC has begun development and testing of the QMPlans schema. The next application areas to be developed are in priority order, QMStatistics, QMRules, and QMResources. The current DMSC working groups are experienced and dedicated, but the pace and quality of QIF development will increase with additional participation by quality system users and solutions providers.

REFERENCES

ANSI 2003. Dimensional Measuring Interface Standard Part 2 : Object Interface Specification, Consortium for Advanced Manufacturing - International.

ANSI 2009d. Dimensional Measuring Interface Standard, DMIS 5.2 Standard, Part 1, ANSI/DMIS 105.2, Part 1-2009.

ANSI 2009a. ANSI/ASME B5.59-1: Data specification for machine tool performance tests (draft).

ANSI 2009b. ANSI/ASME B5.59-2: Data specification for properties of machining and turning centers (draft).

ANSI 2009c. ASME Y14.5-2009: Dimensioning and Tolerancing.

DML 2009. *Dimensional Markup Language (DML)* [Online]. Available: http://www.aiag.org/ [Accessed].

DMSC 2011. *Dimensional Metrology Standards Consortium* [Online]. Available: http://www.dmis.org/ [Accessed].

I++DME 2005. Dimensional Measurement Equipment Interface, the International Association of Coordinate Measuring Machine Manufacturers.

IGES 1980. *Initial Graphics Exchange Specification (IGES)* [Online]. Available: http://ts.nist.gov/standards/iges/ [Accessed].

IMTI 2006. A Roadmap for Metrology Interoperability Integrated Manufacturing Technology Initiative (IMTI Inc.).

ISO 1994. ISO 10303-1: Industrial automation systems and integration - Product data representation and exchange - Part 1: Industrial Automation System and Integration - Product Data Representation and Exchange Part 1: Overview and Fundamental Principles.

ISO 2000a. ISO 10360-1:2000: Geometrical Product Specifications (GPS) -- Acceptance and reverification tests for coordinate measuring machines (CMM) -- Part 1: Vocabulary.

ISO 2000b. ISO 10360-3:2000: Geometrical Product Specifications (GPS) -- Acceptance and reverification tests for coordinate measuring machines (CMM) -- Part 3: CMMs with the axis of a rotary table as the fourth axis.

ISO 2000c. ISO 10360-4:2000: Geometrical Product Specifications (GPS) -- Acceptance and reverification tests for coordinate measuring machines (CMM) -- Part 4: CMMs used in scanning measuring mode.

ISO 2001a. ISO 10303-214: Industrial automation systems and integration - Product data representation and exchange - Part 214: Application Protocol: Core data for automotive mechanical design processes.

ISO 2001b. ISO 10360-6:2001: Geometrical Product Specifications (GPS) -- Acceptance and reverification tests for coordinate measuring machines (CMM) -- Part 6: Estimation of errors in computing Gaussian associated features.

ISO 2004. ISO 10303-238: Industrial automation systems and integration - Product data representation and exchange - Part 238: Application Protocols: Application interpreted model for computerized numerical controllers.

ISO 2007a. ISO 10303-203: Industrial automation systems and integration - Product data representation and exchange - Part 203: Application Protocols: Configuration controlled 3D design.

ISO 2007b. ISO 10303-219, Industrial automation systems and integration - Product data representation and exchange - Part 219: Application protocol: Dimensional inspection information exchange.

ISO 2009a. ISO 10303-203:2009: Industrial automation systems and integration - Product data representation and exchange - Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies.

ISO 2009b. ISO 10360-2:2009: Geometrical product specifications (GPS) -- Acceptance and reverification tests for coordinate measuring machines (CMM) -- Part 2: CMMs used for measuring linear dimensions.

ISO 2010. ISO 10360-5:2010: Geometrical product specifications (GPS) -- Acceptance and reverification tests for coordinate measuring machines (CMM) -- Part 5: CMMs using single and multiple stylus contacting probing systems.

JACOBSON, I. 1992. *Object-oriented software engineering: a use case driven approach*, ACM Press.

QMD 2009. *Quality Measurement Data (QMD)* [Online]. Available: http://www.aiag.org [Accessed].

ZHAO, Y. F., BROWN, R. J., KRAMER, T. R. & XU, X. W. 2011. *Information Modeling for Interoperable Dimensional Metrology*, London, Springer.