NIST Advanced Manufacturing Series
NIST AMS 500-1

Enabling FAIR Data in Additive Manufacturing to Accelerate Industrialization

Shengyen Li
Kareem S. Aggour
Yan Lu
Peter Coutts
Brennan Harris
Alex Kitt
Afina Lupulescu
Michael McNair
Luke Mohr
Mike Vasquez

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AMS.500-1

NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Advanced Manufacturing Series
NIST AMS 500-1

Enabling FAIR Data in Additive Manufacturing to Accelerate Industrialization

Shengyen Li
Systems Integration Division
Engineering Laboratory

Kareem S. Aggour
GE Research

Yan Lu
Systems Integration Division
Engineering Laboratory

Peter Coutts
The Penn State Applied Research Lab

Brennan Harris
Idaho National Lab

Alex Kitt
EWI

Afina Lupulescu
ASM International

Michael McNair
SAE Industry Technologies Consortia

Luke Mohr
EWI

Mike Vasquez
3Degrees

This publication is available free of charge from:
https://doi.org/10.6028/NIST.AMS.500-1

July 2023

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

NIST AMS 500-1
July 2023

NIST Technical Series Policies

Publication History

Approved by the NIST Editorial Review Board on 2023-05-04

How to Cite this NIST Technical Series Publication

Li S, Aggour KS, Lu Y, Coutts P, Harris B, Kitt A, Lupulescu A, McNair M, Mohr L, Vasquez M (2023) Enabling FAIR Data in Additive Manufacturing to Accelerate Industrialization. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Advanced Manufacturing Series (AMS) NIST AMS 500-1. https://doi.org/10.6028/NIST.AMS.500-1

Contact Information

shengyen.li@nist.gov

Abstract

Additive manufacturing (AM) is an important enabler of Industry 4.0 but there are several hurdles that need to be overcome to realize the full potential of AM. These challenges include the need for a data infrastructure that supports the sharing of data generated by the technologies as they mature, and data interoperability is critical to a sustainable data infrastructure that is widely accessible. This paper outlines the need for building a common data stack for an interoperable AM data infrastructure. At the foundation is a common data dictionary (CDD), which defines a primary technical vocabulary for the AM community in logical buckets. On top of the CDD is a common data model (CDM), which defines the hierarchy and relationships of the terms in the CDD and enables the data to be linkable in a complex data system. The CDM empowers information integration and sharing throughout the lifecycles and value chains for different AM technologies. The comprehensiveness of the properties defined in the CDM makes AM data reusable. To enable the exchange of data between different systems, common data exchange formats (CDEFs) transform the CDM into targeted data packages. The design philosophy of the CDD, CDM, and CDEF are described to accelerate the development and adoption of AM technologies across alliances.

Keywords

Additive Manufacturing; Common Data Dictionary; Common Data Model; Common Data Exchange Format; data management.

Table of Contents

List of Figures

Acknowledgments

The authors appreciate the discussions and feedbacks from Dr. Milos Drobnjakovic, Dr. Joy Gockel, Dr. Jiaze He, Dr. Hyunwoong Ko, Richard Resseguie, Tristan Uribe, Dr. Bob Zollo, and Dr. William "Pilgrim" Frazier.

1. INTRODUCTION

Industry 4.0 is a concept that describes the integration of artificial intelligence, cyber-physical systems, internet of things (IoT), big data and advanced data analytics into manufacturing processes and other industrial sectors to increase efficiency, productivity and innovation. Additive manufacturing (AM) is an important enabler of Industry 4.0 by reducing person-hours, increasing asset efficiency, and enabling high levels of customization, among other benefits [1]. AM can require thousands of parameters to prepare a material and a build may involves hundreds of thousands of control steps, which can result in high variability of the process. Acquiring knowledge from data generated throughout the AM lifecycle is needed to improve the stability and reproducibility of the overall process. A plethora of technologies have been developed to address the myriad challenges of AM, but standards for data management are now needed to enable AM scalability and sustainability as they mature.

Building parts from the scales of micrometer and millisecond, AM uses and generates high-volume and high-dimensional data including the characteristics of the source materials, scan strategies, pedigrees of the feeding or recoating processes, and the observations from in-process monitoring systems, to name a few. To identify the root cause of variations at the present state of AM technology, the merging of disparate databases is required since siloed efforts are still performed to gather information across the AM lifecycle [2–4]. Today, this information is still archived using specific ontologies and data models in different database systems to better present the unique contributions from different AM processes [2,5]. The unique informatics systems create technical barriers to find, integrate and compare data. A common agreement and standards are needed to create a comprehensive data alliance for representing the state-of-the-art data and driving knowledge exploration across the AM lifecycle.

Figure 1 shows the broad landscape for AM data generated from the AM lifecycle and value chain activities, both of which cover four various domains—material, machine, design, and process. These data, appropriately integrated, play a critical role in streamlining the AM development process, from design, fabrication to part certification. AM data, especially those generated from in-situ monitoring and ex-situ inspections, embody all the four V characteristics of Big Data—volume, velocity, variety, and veracity. For example, the amount of data produced is estimated at 3000 TB for the qualification of an additively built aircraft component [6]. In-situ process monitoring during a powder bed fusion process can produce data at GB/sec.

These Big Data are usually collected, archived, and analyzed for build monitoring and control, part qualification and iterative design, as well as for AM process understanding, process, machine, and material qualification and improvement. The latter can rely on a broad spectrum of data beyond an individual stakeholder's data management capability. For example, in-process monitoring data generated for part quality control can be used for material and machine property understanding and reduce the need for experiments by the material and machine vendors for material and machine development.

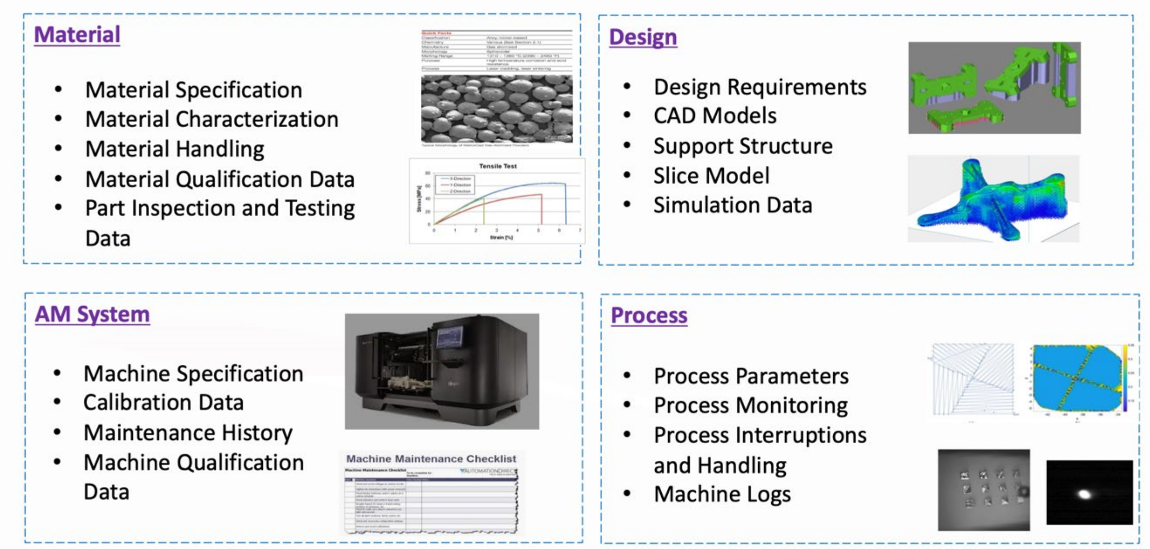


Figure 1 Representative data generated from Additive Manufacturing lifecycle and value chain activities.

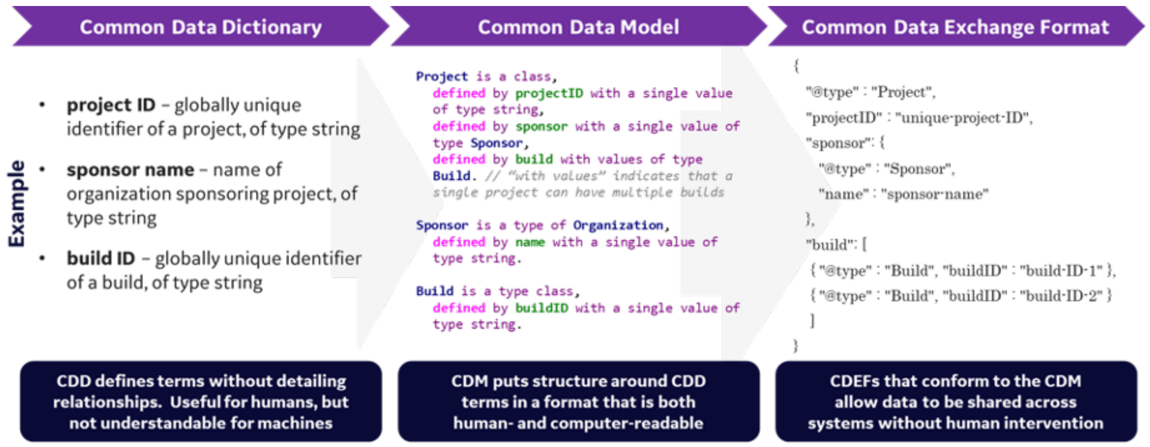
2. Prior Art

A variety of computer tools have been developed to query, visualize, and analyze high dimensional AM datasets [2,7,8]. Some platforms are designed for sharing scientific datasets, which wrap up data with a generic set of metadata [9,10]. Because the parameter space from an AM project is enormous, significant effort is required to explore the material, processing, structure, and properties relationships at different phases of technology development. Data fusion and format translation can effectively use the present knowledge to support the analytics to identify technical gaps. For example, ISO 10303 has been applied to exchange the 3D geometry data for different machines [3,11]. Standards for data management are critical to mature AM technology.

Figure 2 shows the relationships between the CDD, CDM, and CDEF, in which a Common Data Dictionary defines the vocabulary that will be used in the development of a Common Data Model. The Common Data Model puts a structure around the terms, enabling the searchability of data stores without human intervention. Common Data Exchange Formats that conform to the Common Data Model allow for reading data from or loading data into a variety of databases that support the CDM. This paper clarifies the definitions of the major components of FAIR principles (Findable, Accessible, Interoperable, and Reusable) [12], as highlighted in Figure 3, for AM and outlines the need for such standards.

The remainder of this paper is organized as follows. The next section summarizes the prior art in data management and integration frameworks, in particular for additive manufacturing applications. The following section then describes AM common data dictionary ASTM standard F3490-21 about the vocabularies for the descriptions of the AM data. The following section then describes a common data model that puts a structure around this common data dictionary. This working group consensus enables the accessibility and reusability of the data with the detailed provenance from metadata. The next section then describes common data exchange formats that can be used to enable the interoperability of data in a variety of formats, such as JSON and/or XML, for different data systems, followed by a Conclusions and Future Work section summarizing the vision and progress towards the development of these standards, and outlining next steps to make these standards widely adopted.

We hope it is clear to the reader that throughout this document when we refer to the CDD, CDM, and CDEF, we are referring specifically to the common data dictionary, common data model, and common data exchange formats being developed for additive manufacturing.



- **project ID** – globally unique identifier of a project, of type string
- **sponsor name** – name of organization sponsoring project, of type string
- **build ID** – globally unique identifier of a build, of type string

```
Project is a class,
  defined by projectID with a single value
  of type string,
  defined by sponsor with a single value of
  type Sponsor,
  defined by build with values of type
  Build. // "with values" indicates that a
  single project can have multiple builds

Sponsor is a type of Organization,
  defined by name with a single value of
  type string.

Build is a type class,
  defined by buildID with a single value of
  type string.
```

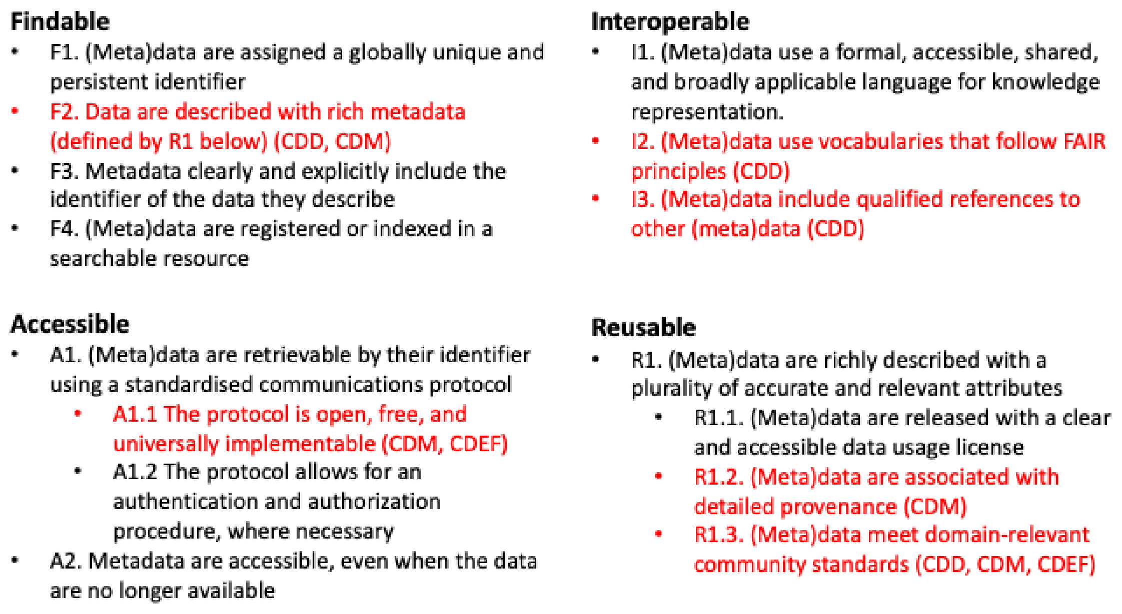
```
{
  "@type" : "Project",
  "projectID" : "unique-project-ID",
  "sponsor": {
    "@type" : "Sponsor",
    "name" : "sponsor-name"
  },
  "build": [
   { "@type" : "Build", "buildID" : "build-ID-1" },
   { "@type" : "Build", "buildID" : "build-ID-2" }
   ]
}
```

CDD defines terms without detailing relationships. Useful for humans, but not understandable for machines

CDM puts structure around CDD terms in a format that is both human- and computer-readable

CDEFs that conform to the CDM allow data to be shared across systems without human intervention

Figure 2 Descriptions and relationships between the Common Data Dictionary, Common Data Model, and Common Data Exchange Formats.



Findable

- F1. (Meta)data are assigned a globally unique and persistent identifier
- F2. Data are described with rich metadata (defined by R1 below) (CDD, CDM)
- F3. Metadata clearly and explicitly include the identifier of the data they describe
- F4. (Meta)data are registered or indexed in a searchable resource

Accessible

- A1. (Meta)data are retrievable by their identifier using a standardised communications protocol
  - A1.1 The protocol is open, free, and universally implementable (CDM, CDEF)
  - A1.2 The protocol allows for an authentication and authorization procedure, where necessary
- A2. Metadata are accessible, even when the data are no longer available

Interoperable

- I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2. (Meta)data use vocabularies that follow FAIR principles (CDD)
- I3. (Meta)data include qualified references to other (meta)data (CDD)

Reusable

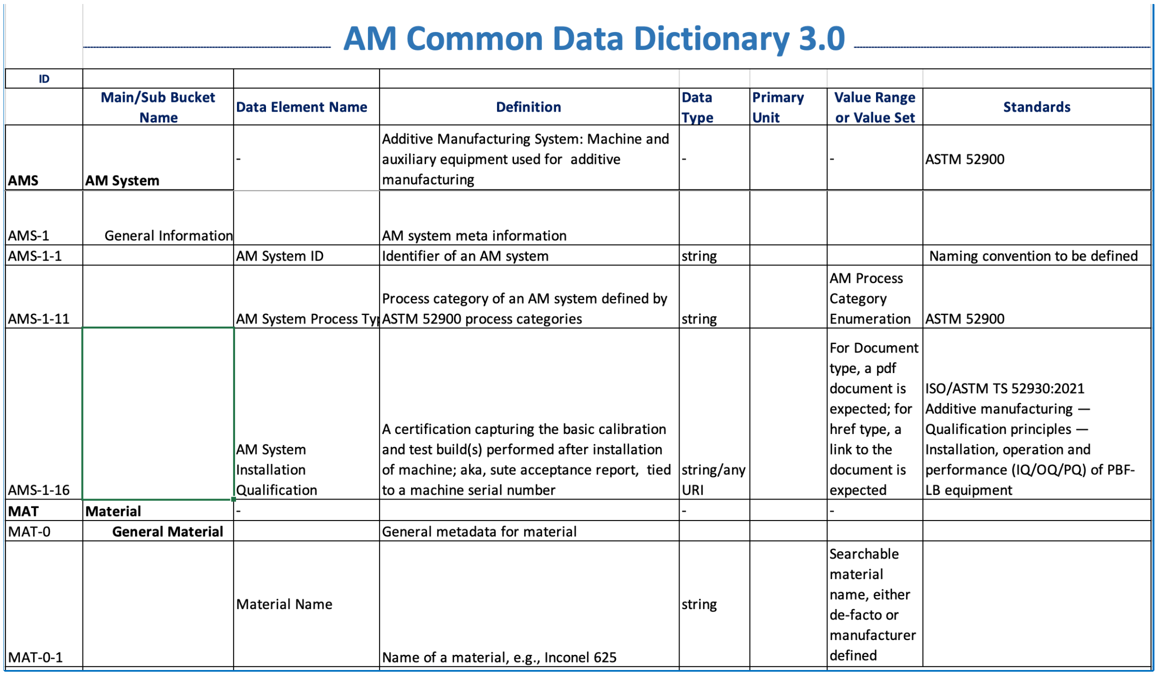
- R1. (Meta)data are richly described with a plurality of accurate and relevant attributes
  - R1.1. (Meta)data are released with a clear and accessible data usage license
  - R1.2. (Meta)data are associated with detailed provenance (CDM)
  - R1.3. (Meta)data meet domain-relevant community standards (CDD, CDM, CDEF)

Figure 3 The highlighted items of the FAIR (Findable, Interoperable, Accessible, and Reusable) Principles [12] that are addressed in this work.

3. COMMON DATA DICTIONARY

A Common Data Dictionary sits at the foundation of a semantic data stack and is developed to provide a consistent technical vocabulary of AM concepts and attributes for the community to communicate and collaborate. A common data dictionary not only allows AM system developers to design or update a data store that meets business and process requirements using standard definitions of data elements, but also enables AM data sharing among organizations and personnel with legacy proprietary data systems using neutral definitions for essential AM data terms that can be mapped to proprietary data.

Figure 4 shows an extract of the AM common data dictionary developed by a joint industry-government-academic working group.



AM Common Data Dictionary 3.0

ID							
	Main/Sub Bucket Name	**Data Element Name**	**Definition**	**Data Type**	**Primary Unit**	**Value Range or Value Set**	**Standards**
AMS	**AM System**	-	Additive Manufacturing System: Machine and auxiliary equipment used for additive manufacturing	-		-	ASTM 52900
AMS-1	General Information		AM system meta information				
AMS-1-1		AM System ID	Identifier of an AM system	string			Naming convention to be defined
AMS-1-11		AM System Process Ty	Process category of an AM system defined by ASTM 52900 process categories	string		AM Process Category Enumeration	ASTM 52900
AMS-1-16		AM System Installation Qualification	A certification capturing the basic calibration and test build(s) performed after installation of machine; aka, sute acceptance report, tied to a machine serial number	string/any URI		For Document type, a pdf document is expected; for href type, a link to the document is expected	ISO/ASTM TS 52930:2021 Additive manufacturing — Qualification principles — Installation, operation and performance (IQ/OQ/PQ) of PBF-LB equipment
MAT	**Material**	-		-		-	
MAT-0	**General Material**		General metadata for material				
MAT-0-1		Material Name	Name of a material, e.g., Inconel 625	string		Searchable material name, either de-facto or manufacturer defined	

Figure 4 AM Common Data Dictionary extract.

The working version of the CDD is defined in an Excel spreadsheet that consists of several tabs to ease its development by a large, distributed team. The first tab provides an overview of the top fifteen AM concepts (also known as "buckets") and their relationships. These concepts are used to group AM data items into information modules, for example, AMS (AM System), BLD (Build), Mat (Material), PRC (Process Control), PRD (Process Data) and PTD (Part Design), etc. The second tab captures the definitions of about 830 AM data items that were considered essential for the community to manage and exchange information. The column titled "ID" represents the unique identifier for the data element corresponding to that row. The second column, contains the names of the main or sub-buckets. The "Data Element Name" column contains the given name of a particular data element, followed by a column containing the definition of the data element. This is then followed by the "Data Type" column, which contains the preferred kind of data that a particular data element should contain. The values in this column reference a list of data types in the "Data Type" tab of the spreadsheet. Next is the "Primary

Unit", which represents the preferred unit that the data type should be reported as listed in the "Unit" tab. The "Value Range or Value Set" column refers to either the values contained within the "Value Set" tab or to a single foreign key or multiple foreign keys. The final main column of this tab is the "Standards" column, where any known applicable standard related to this data element is listed.

When the CDD reached relative maturity, it was transferred to the ASTM F42.08 Data subcommittee for standardization. The subcommittee decided that the process-agnostic elements of the CDD, representing a core of common attributes across all AM processes, should be included in the first AM data dictionary standard. ASTM F3490-21 "Standard Practice for Additive Manufacturing — General Principles — Overview of Data Pedigree" was released, which includes 395 general AM data items, was defined as the first AM common data dictionary standard.

4. COMMON DATA MODEL

The CDD described previously is incredibly important in that it standardizes the vocabulary of the additive manufacturing community. However, the CDD by itself only allows members of the community to communicate more effectively. To achieve truly FAIR AM data, a Common Data Model and Common Data Exchange Formats are required to be established and adopted by the AM community, as well. Common Data Models are critically important in making the CDD practical because they put a structure around the CDD so that computing systems can begin to use the same vocabulary to make data Findable, Accessible, Interoperable, and Reusable.

By defining a formal class structure and defining the relationships between the terms within the CDD, the precise linking between the attributes becomes apparent in a computable model. This computable model can be used by software systems to map entries in the CDM to specific fields within a data store. This allows those data stores to become searchable using the vocabulary defined in the CDD even if the underlying data stores do not store their data using the vocabulary specified in the CDD.

This concept is shown in Figure 5, in which a user can explore and select a set of attributes of interest in terms defined within the CDM. This query can then be pushed to one or more underlying data storage systems across many different potential AM collaborators. Each of these collaborators may have their own unique underlying storage technologies with their own data schemas and structures, they do not have to adopt the CDM as their internal data representation to make their data FAIR. Collaborators must, however, map their data to the CDM so that when a query is sent to their data infrastructure, their system can automatically parse the CDM-based query, translate it into a query executable against their internal data structure, retrieve whatever data is shareable that matches the search criteria, and then return the data in the structure of the CDM.

Figure 5 mentions four specific systems (from GE, NIST, AFRL, and Hexagon), and a generic fifth 'System X' to highlight the fact that this approach is flexible and extensible to any participant that would like to make their additive manufacturing data FAIR. These four specific systems are highlighted as examples because members of these organizations have been partnering on the development of the CDD, CDM, and CDEF and have (e.g., through AFRL's CAMDEN effort) and/or are actively developing proofs-of-concept to demonstrate that these kinds of data exchanges can work.

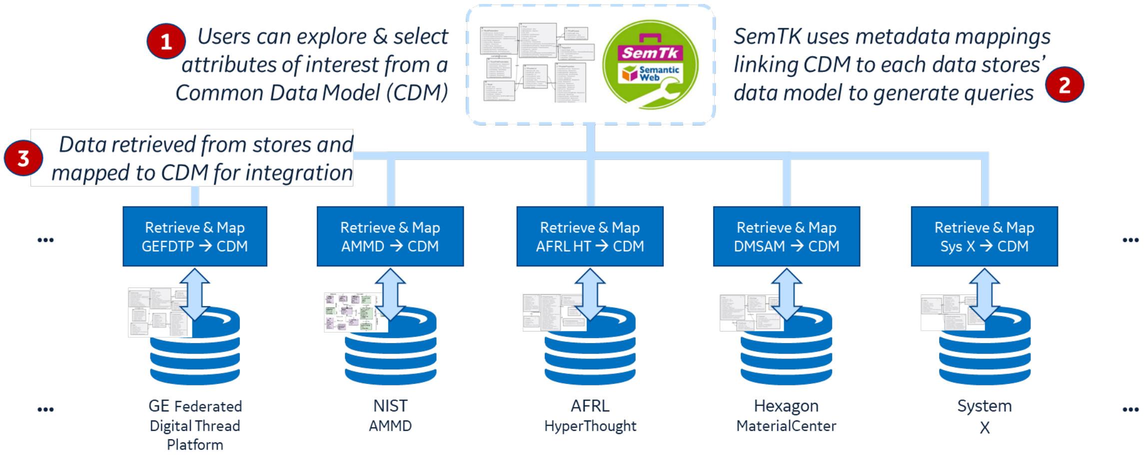


Figure 5 Vision of a Common Data Model being used to enable access to multiple source systems, each with their own internal data structures. A user can specify attributes of interest within some query system (as represented within the dotted box at the top), potentially using a graphical user interface to visually explore classes within the CDM to select specific attributes. An open-source tool such as the Semantics Toolkit [13] could transform the CDM-based request into one or more queries against the underlying source systems, which would in turn run the queries, retrieve their local data, and map it back into a format that conforms to the CDM so that the data from the many systems could be seamlessly merged.

5. COMMON DATA EXCHANGE FORMATS

Local data models are generally designed based on different design philosophies for unique purposes. The integration of data across different infrastructures requires the identification of desired attributes and then the retrieval and physical merging of data between multiple distinct local data models. The design and development of common data exchange formats is an effort to eliminate the technical barriers for the transmission and integration of data across infrastructures.

An intelligent data exchanger should be capable of fusing datasets from different sources, like SemTK [13] shown in Figure 5. Integration using APIs and open-source software libraries can satisfy the needs at the executive level of data federation. The major challenge is that the local data models may be implemented using different computer languages and different file formats for data curation, which requires additional steps to compare and align the information. A common data exchange format is needed to translate the ontologies and taxonomy into a unified format and align relevant information from multiple sources.

6. CDM MODULES

The CDM has been divided into six logical modules, primarily to facilitate its parallel development. These six modules are: Base, Material, System, Process, Build, and TIC (Test-Inspection-Characterization). Some of the core classes and their connections from these modules are shown in Figure 6.

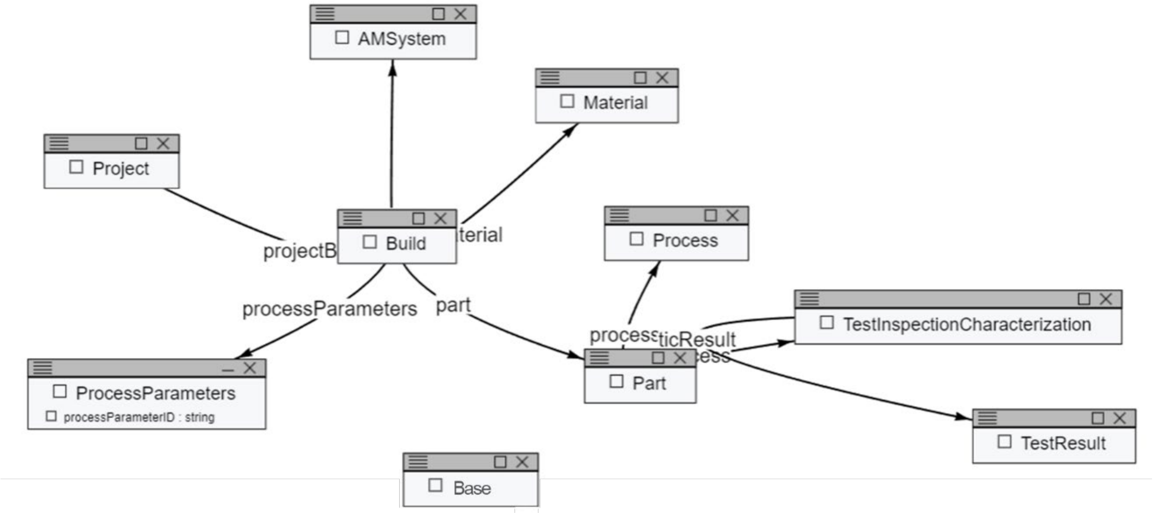


Figure 6 Examples of core classes from CDM and their relationships

The AM common data model is available to the public in a GitHub repository located at: https://github.com/kaggour/AM-CDM. It has been developed in the Semantic Application Design Language (SADL), a formal, English-like language and Eclipse plugin developed to simplify the authoring of semantic models by non-semantic modeling experts [14]. SADL allows experts in additive manufacturing, materials science, and other scientific domains to read and write SADL without having to become experts in semantic modeling. One powerful feature of SADL is it can auto-generate OWL (the Web Ontology Language) and so both SADL and OWL versions of the CDM are available in the Git repository. While we have used a semantic language editor to develop the CDM, the CDM is not an ontology, it is a data model meant to define the structure and relationships of a set of terms that have been defined elsewhere (the CDD).

Each of the six modules are described below.

6.1. Base Module

The Base module consists of over a dozen foundational classes such as Person, Organization, Qualification, Measurement (the primary class used to store data values, including the numerical value and unit, among other meta information), and more. Each of these foundational classes contain a variety of attributes specific to each class. For example, Qualification includes qualificationType, qualificationLevel, and qualifyingOrganization attributes, and Person includes personID, personFirstName, and personQualification, to name a few. Included with the class attribute definitions are their data type and cardinality. In the case of the Person class, each person has a single first name of primitive data type string but can have multiple qualifications of type Qualification. An example of the Person and Qualification Base classes are shown in Figure 7 below. Note that some attributes are defined as single-valued ("with a single value of type…"), and other attributes are defined as potentially multi-valued ("with values of type…").

```
Person (note "Individuals and their primary credentials and contact information.") is a class,
    described by personID (note "Unique identifier of the person.") with a single value of type string,
    described by personFirstName (note "First name of individual performing task.") with a single value of type string,
    described by personLastName (note "Last name of individual performing task.") with a single value of type string,
    //described by personRole (note "Role of individual performing task.") with a single value of type string,
    described by personEmail (note "The email for the individual.") with a single value of type string,
    described by personPhoneNumber (note "The phone number for the individual.") with values of type PhoneNumber,
    described by personQualification (note "Qualifications of person.") with values of type PersonQualification,
    described by personOrganization (note "Links person to an organization") with a single value of type Organization.

Qualification (note "Details of qualification/certification (machine specific operator certificate, UL, ASTM).") is a
    class,
    described by qualificationType (note "The type of qualification/certification.") with a single value of type
    string,
    described by qualificationLevel (note "The level of qualification/certification.") with a single value of type
    string,
         described by qualifyingOrganization (note "The organization from which the qualification or certification
originates.") with a single value of type Organization.
```

Figure 7 Person and Qualification Base classes written in the Semantic Application Design Language, in which each class has multiple attributes of both primitive (e.g., string, double, float) and complex data types associated, which link different classes together.

6.2. Material Module

Materials are prepared in different shapes, dimensions, phases in thermodynamics, and chemical properties for different AM technologies. It should be addressed that the material mentioned here is different from the material as a part for the assembly of a system, which requires different prospections for the location-specific properties, which is beyond the scope of this work. Being more specific to metal AM, we include a variety of attributes in the Material module such as (1) the alloy composition and (2) many intrinsic properties, such as the specific heat and thermal conductivity, as well as (3) extrinsic properties with some associated variability of materials, such as size distribution of the powder particles, as these are critical to the design of the AM build strategy and to assess the quality of the products.

Also, important to processing history is the initial status of the material, which is vital to all kinds of engineering projects and quality assurance. The information about the manufacturer, production batch, and storage environment help to clarify the root cause of potential sources of variations in an enormous space. The number of reuses/recycles is also a part of the material history affecting the variability of the material status and is captured as attributes of the Material class.

6.3. System Module

The System module of the AM CDM describes the physical equipment and the software associated with additive manufacturing processes, including the 3D printers themselves as well as non-AM auxiliary equipment. All physical, equipment-based classes extend the root ‘System’ class, which includes several generic attributes such as systemID, systemName, systemManufacturer, and systemModel.

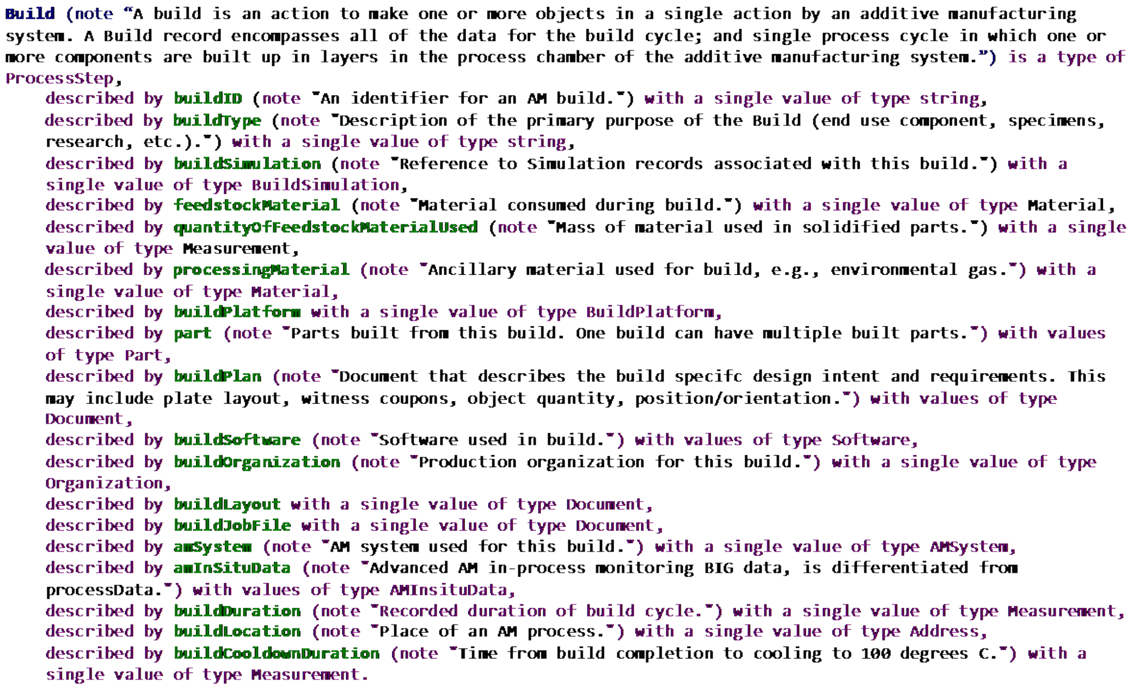
Additionally, the System module includes both AMSystem classes, and NonAMSystem classes. Many modality-specific classes further extend AMSystem, including LaserPowderBedFusionSystem, DirectedEnergyDepositionSystem, and BinderJetSystem, to name a few. The System model also includes several classes for a variety of subsystems found in typical additive manufacturing systems, including LaserSystem, RecoatingSystem, and CameraSubsystem. The System class also includes meta information about the machine's capabilities, classes, and attributes about the software installed on the machines, and information about the maintenance, configuration, and calibration of both the hardware and software.

Overall, the System module of the CDM is used to model information about the hardware and software used in the end-to-end additive manufacturing process.

6.4. Build Module

The Build module models all the information generated during a single, AM process cycle during which one or more components are 'built up' in layers inside the process chamber of the additive manufacturing system (ASTM 52900). The main Build class acts as a central reference point for all data related to a build, as shown in Figure 8. The definition of the Build class not only has all the metadata attributes that describe a build, such as buildID, buildType, and buildTime, but also contains all the attributes linking to other modules, e.g., feedstockMaterial and AMSystem. Other attributes, including part, buildPlatform, buildParameters, amInsituData, buildSimulation and buildSoftware, are defined by the classes in the Build module.

Part refers to instances of 3D designs in the as-built state made in this build cycle. One Build can have multiple build Parts of the same partDesign. A partDesign can be realized by many Builds. In this sense, PartDesign can be defined independent of Build. However, for this data model, PartDesign is not used by any other modules, hence the class is defined in the Build module. buildParameters extends processParameters which can have attributes defined by multiple process parameters or just a build command file. The amInsituData extends the datasetMetadata class defined in the Base module with additional attributes associated with the in-situ monitoring device, the reference framework of the device, the configurations, and the data acquisition information. BuildSimulation and SynthesizedData classes defined the metadata that describe a simulation of the build, and the data analysis results for the build.

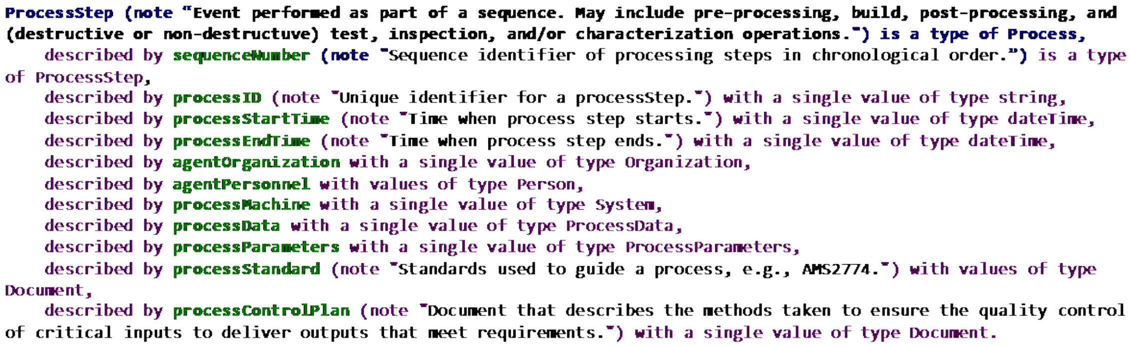


```
Build (note “A build is an action to make one or more objects in a single action by an additive manufacturing
system. A Build record encompasses all of the data for the build cycle; and single process cycle in which one or
more components are built up in layers in the process chamber of the additive manufacturing system.”) is a type of
ProcessStep,
    described by buildID (note "An identifier for an AM build.") with a single value of type string,
    described by buildType (note "Description of the primary purpose of the Build (end use component, specimens,
    research, etc.).") with a single value of type string,
    described by buildSimulation (note "Reference to Simulation records associated with this build.") with a
    single value of type BuildSimulation,
    described by feedstockMaterial (note "Material consumed during build.") with a single value of type Material,
    described by quantityOfFeedstockMaterialUsed (note "Mass of material used in solidified parts.") with a single
    value of type Measurement,
    described by processingMaterial (note "Ancillary material used for build, e.g., environmental gas.") with a
    single value of type Material,
    described by buildPlatform with a single value of type BuildPlatform,
    described by part (note "Parts built from this build. One build can have multiple built parts.") with values
    of type Part,
    described by buildPlan (note "Document that describes the build specifc design intent and requirements. This
    may include plate layout, witness coupons, object quantity, position/orientation.") with values of type
    Document,
    described by buildSoftware (note "Software used in build.") with values of type Software,
    described by buildOrganization (note "Production organization for this build.") with a single value of type
    Organization,
    described by buildLayout with a single value of type Document,
    described by buildJobFile with a single value of type Document,
    described by amSystem (note "AM system used for this build.") with a single value of type AMSystem,
    described by amInSituData (note "Advanced AM in-process monitoring BIG data, is differentiated from
    processData.") with values of type AMInsituData,
    described by buildDuration (note "Recorded duration of build cycle.") with a single value of type Measurement,
    described by buildLocation (note "Place of an AM process.") with a single value of type Address,
    described by buildCooldownDuration (note "Time from build completion to cooling to 100 degrees C.") with a
    single value of type Measurement.
```

Figure 8 Build class in Build module written in the Semantic Application Design Language

6.5. Process Module

The Process module defines all the classes that are necessary to describe an additive manufacturing process and the process sequence associated with a part or specimen. The ProcessStep class refers to a manufacturing activity performed as a component of an ordered sequence, which include pre-processing, build, post-processing, and (destructive or non-destructive) test, inspection, and/or characterization operations. A base class of ProcessStep is defined as shown in Figure 9.



```
ProcessStep (note “Event performed as part of a sequence. May include pre-processing, build, post-processing, and
(destructive or non-destructuve) test, inspection, and/or characterization operations.") is a type of Process,
    described by sequenceNumber (note "Sequence identifier of processing steps in chronological order.”) is a type
of ProcessStep,
    described by processID (note "Unique identifier for a processStep.") with a single value of type string,
    described by processStartTime (note "Time when process step starts.") with a single value of type dateTime,
    described by processEndTime (note "Time when process step ends.") with a single value of type dateTime,
    described by agentOrganization with a single value of type Organization,
    described by agentPersonnel with values of type Person,
    described by processMachine with a single value of type System,
    described by processData with a single value of type ProcessData,
    described by processParameters with a single value of type ProcessParameters,
    described by processStandard (note "Standards used to guide a process, e.g., AMS2774.") with values of type
Document,
    described by processControlPlan (note "Document that describes the methods taken to ensure the quality control
of critical inputs to deliver outputs that meet requirements.") with a single value of type Document.
```

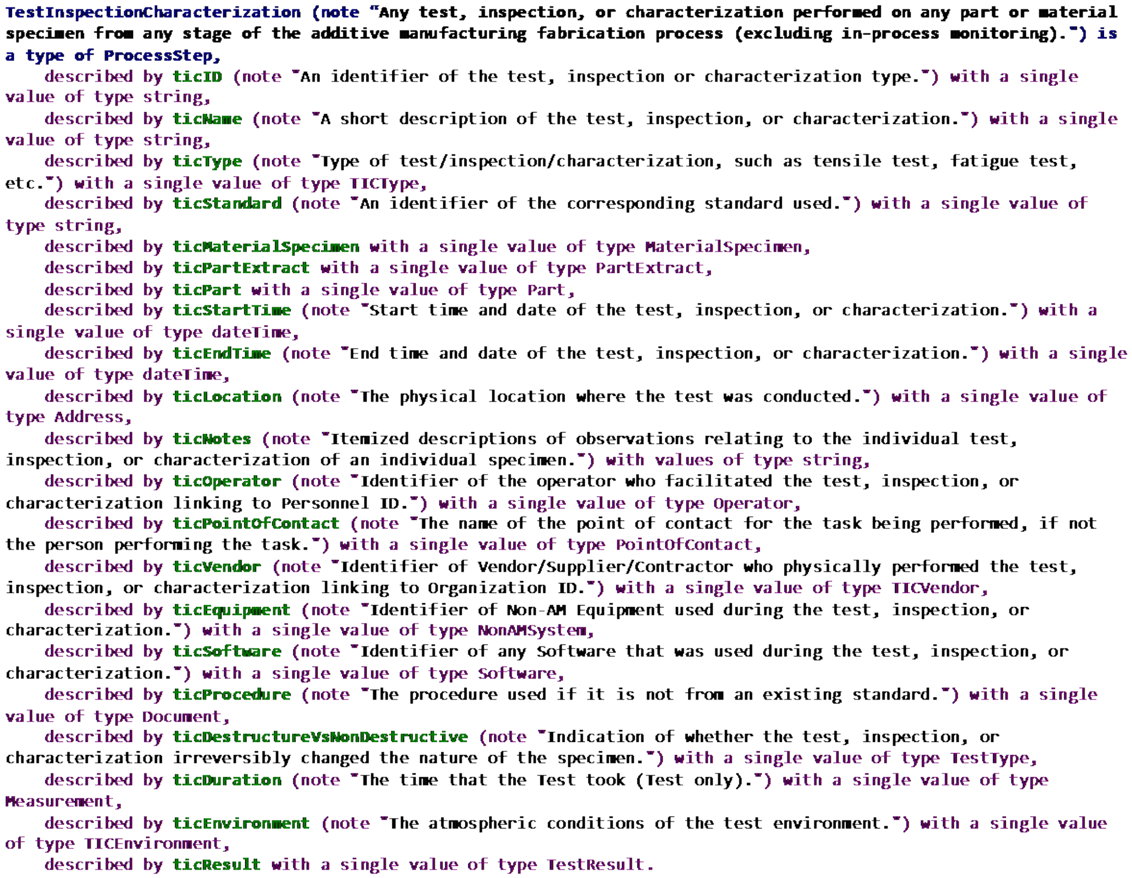
Figure 9 The ProcessStep class from the Process module.

Important attributes of ProcessStep include processParameters, processControlPlan and processData. The first two attributes define the controls of a manufacturing process parametrically and with a document, respectively. processData captures the information about the measured data, or derived outputs from that data, obtained from the AM system and in-situ monitoring equipment during a build process. Type-specific information about a manufacturing process is defined in the extensions of the ProcessStep class, such as Build, HeatTreatment, TestInspectionCharacterization, etc.

6.6. Test-Inspection-Characterization Module

The Test-Inspection-Characterization (TIC) module can be thought of as an information space that includes measurement methods, structural features, and material properties all of which are based on the materials data and the processing history. AM projects need TIC for designing and assessing the building strategies, as well as the post-building treatments. Each material system, measurement method, and targeted application have unique domain knowledge and requirements. Developing a thorough ontology to cover all possible aspects can be very challenging and so we focus on a high-level structure to convey the concepts of the model development for TIC.

Starting from the top level in Figure 10, the metadata for TestInspectionCharacterization includes testing facility, testing conditions, operator, and environmental conditions. Testing facility refers to the hardware model, software version, and calibration schedule. Testing conditions indicates the shape and size of the coupon and testing variables such as strain rate and temperature for tensile tests. Operator and environmental conditions may cause variability in the outcomes, and hence the need to capture such metadata. Because the results of TIC highly depend on the status of raw material and processing history, appropriate links or a handler system is also required to create a comprehensive dataset.



```
TestInspectionCharacterization (note "Any test, inspection, or characterization performed on any part or material
specimen from any stage of the additive manufacturing fabrication process (excluding in-process monitoring).") is
a type of ProcessStep,
    described by ticID (note "An identifier of the test, inspection or characterization type.") with a single
value of type string,
    described by ticName (note "A short description of the test, inspection, or characterization.") with a single
value of type string,
    described by ticType (note "Type of test/inspection/characterization, such as tensile test, fatigue test,
etc.") with a single value of type TICType,
    described by ticStandard (note "An identifier of the corresponding standard used.") with a single value of
type string,
    described by ticMaterialSpecimen with a single value of type MaterialSpecimen,
    described by ticPartExtract with a single value of type PartExtract,
    described by ticPart with a single value of type Part,
    described by ticStartTime (note "Start time and date of the test, inspection, or characterization.") with a
single value of type dateTime,
    described by ticEndTime (note "End time and date of the test, inspection, or characterization.") with a single
value of type dateTime,
    described by ticLocation (note "The physical location where the test was conducted.") with a single value of
type Address,
    described by ticNotes (note "Itemized descriptions of observations relating to the individual test,
inspection, or characterization of an individual specimen.") with values of type string,
    described by ticOperator (note "Identifier of the operator who facilitated the test, inspection, or
characterization linking to Personnel ID.") with a single value of type Operator,
    described by ticPointOfContact (note "The name of the point of contact for the task being performed, if not
the person performing the task.") with a single value of type PointOfContact,
    described by ticVendor (note "Identifier of Vendor/Supplier/Contractor who physically performed the test,
inspection, or characterization linking to Organization ID.") with a single value of type TICVendor,
    described by ticEquipment (note "Identifier of Non-AM Equipment used during the test, inspection, or
characterization.") with a single value of type NonAMSystem,
    described by ticSoftware (note "Identifier of any Software that was used during the test, inspection, or
characterization.") with a single value of type Software,
    described by ticProcedure (note "The procedure used if it is not from an existing standard.") with a single
value of type Document,
    described by ticDestructureVsNonDestructive (note "Indication of whether the test, inspection, or
characterization irreversibly changed the nature of the specimen.") with a single value of type TestType,
    described by ticDuration (note "The time that the Test took (Test only).") with a single value of type
Measurement,
    described by ticEnvironment (note "The atmospheric conditions of the test environment.") with a single value
of type TICEnvironment,
    described by ticResult with a single value of type TestResult.
```

Figure 10 TestInspectionCharacterization class from the TIC module

7. CONCLUSIONS AND FUTURE WORK

This work provides the design philosophy of an AM common data model under development to make AM data FAIR, which enables the future development of unique data models for specific applications. The AM CDD, which provides the core attribute definitions for the data model, has been adopted by industry consortia and was used in early 2020 to demonstrate that the CDD can be used to facilitate common queries to search data from multiple proprietary databases. These experiences highlighted that implementing and exercising a common data model required additional communication between AM practitioners and data engineers to better create a data pedigree for a specific working environment. This work mitigates that technical barrier and provides the foundations of such communications. Therefore, the next release of the AM CDD will be implemented in different file formats, such as RDF and XML, to enhance the interoperability for data engineers.

The common data model is still under development. A comprehensive list of usage scenarios is being developed to test the current version of the common data model. For example, data from the NIST AM Material Database (ammd.nist.gov) is being migrated to an AM CDM-based graph database which is expected to answer research questions such as, "How does HT and HIP change porosity, grain distribution, and tensile strength?" Preliminary experiments demonstrated that the CDM captures both the attributes and their relationships that are necessary for a smart query to retrieve data results that answer these research questions. Adopting the CDM can avoid ad-hoc efforts and mitigate many of the technical challenges for the integration of data from otherwise siloed systems needed to answer such important questions.

This working group is planning to test and mature these models using different datasets and different computer languages. We highly encourage AM practitioners from different areas to participate in these tasks. These exploratory efforts will help to formulate consensus standards for the developments of AM data models and the experiences will be shared in a public repository to initiate an alliance in a FAIR data environment for the AM community.

References

[1] Gibson I, Rosen DW, Stucker B, Khorasani M, Rosen D, Stucker B, Khorasani M (2021) Additive Manufacturing Technologies. Vol 17. Springer. https://doi.org/10.1007/978-3-030-56127-7

[2] Ali MM, Rai R, Otte JN, Smith B (2019) A product life cycle ontology for additive manufacturing. Comput Ind. 105:191-203. https://doi.org/10.1016/j.compind.2018.12.007

[3] Liu X, Rosen DW (2010) Ontology based knowledge modeling and reuse approach of supporting process planning in layer-based additive manufacturing. In: 2010 International Conference on Manufacturing Automation. IEEE: 261-266. https://10.1109/ICMA.2010.40

[4] Butt J. (2020) Exploring the interrelationship between additive manufacturing and Industry 4.0. Designs (Basel): 4(2):13. https://doi.org/10.3390/designs4020013

[5] Li Z, Huang M, Zhong Y, Qin Y (2022) A Description Logic Based Ontology for Knowledge Representation in Process Planning for Laser Powder Bed Fusion. Applied Sciences 12(9):4612. https://doi.org/10.3390/app12094612

[6] Morris KC, Lu Y, Frechette S (2020) Foundations of information governance for smart manufacturing. Smart Sustain Manuf Syst. 2020;4(2). https://10.1520/ssms20190041

[7] Aggour KS, Kumar VS, Cuddihy P, Williams JW, Gupta V, Dial L, Hanlon T, Gambone J, Vinciquerra J (2019) Federated multimodal big data storage & analytics platform for additive manufacturing. In: IEEE International Conference on Big Data (Big Data). IEEE:1729-1738. https://10.1109/BigData47090.2019.9006495

[8] Kannan R, Knapp GL, Nandwana P, Dehoff R, Plotkowski A, Stump B, Yang Y, Paquit V (2022) Data Mining and Visualization of High-Dimensional ICME Data for Additive Manufacturing. Integr Mater Manuf Innov:11(1):57-70. https://doi.org/10.1007/s40192-021-00243-2

[9] Puchala B, Tarcea G, Marquis EA, Hedstrom M, Jagadish H V, Allison JE (2016) The materials commons: a collaboration platform and information repository for the global materials community. JOM: 68:2035-2044. https://doi.org/10.1007/s11837-016-1998-7

[10] HyperThought. https://www.hyperthought.io/login/.

[11] Hällgren S, Pejryd L, Ekengren J. (2016) 3D data export for additive manufacturing-improving geometric accuracy. Procedia CIRP: 50:518-523. https://doi.org/10.1016/j.procir.2016.05.046

[12] Wilkinson MD, Dumontier M, Aalbersberg IjJ, et al. (2016) The FAIR Guiding Principles for scientific data management and stewardship. Sci Data: 3(1):1-9. https://doi.org/10.1038/sdata.2016.18

[13] Cuddihy P, McHugh J, Williams JW, Mulwad V, Aggour KS (2018) SemTK: A Semantics Toolkit for User-friendly SPARQL Generation and Semantic Data Management. In: ISWC (P&D/Industry/BlueSky).

[14] Crapo A, Moitra A (2013) Toward a unified English-like representation of semantic models, data, and graph patterns for subject matter experts. Int J Semant Comput 7(03):215-236. https://doi.org/10.1142/S1793351X13500025