A PROPOSED DIGITAL THREAD FOR ADDITIVE MANUFACTURING

A. R. Nassar* and E. W. Reutzel*

*Applied Research Laboratory, The Pennsylvania State University, University Park, PA 16804

Accepted August 16th 2013

Abstract

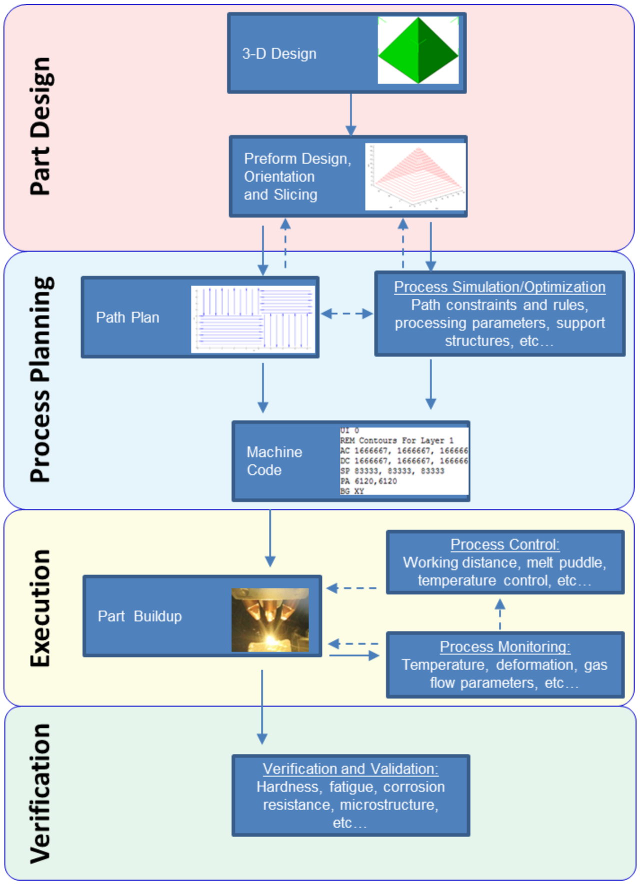
Additive manufacturing (AM) has been explored by the automotive, aerospace and medical industries for many years but has yet to achieve wide-spread acceptance. This is partially due to the lack of standard frameworks for the exchange of data related to design, modeling, build plan, monitoring, control, and verification. Here, a unified paradigm, built on Extensible Markup language (XML) -based file formats and influenced by the ASTM F291 standard, is proposed, to record and transmit data at every stage of the AM process. This digital thread contains all essential parameters, from design to testing of metal-based AM parts while remaining accessible, traceable and extensible.

1. Introduction

Additive manufacturing (AM) has been explored by the automotive, aerospace and medical industries for many years. The primary advantage of AM over conventional processes is the ability to produce complex and customized objects for low-volume or high-end use at a fraction of the cost and time [1, 2]. Within the aerospace industry in particular, AM of metals has garnered interest and investment, as illustrated by the acquisition of two additive manufacturing companies by GE Aviation in late 2012 [3] and the membership of Lockheed Martin, Boeing and others in the National Additive Manufacturing Innovation Institute [4]. Of primary interests are the fabrication and repair of rib-web structural components, for aircraft sub-structures, and engine components [5].

A recent national emphasis on this technology in the U.S. has highlighted the need to have a unified paradigm for sharing of digital data associated with the process: from design, to simulation, to build plan, to process monitoring and control, to verification [6]. Standards organizations, such as ASTM, have already begun to establish file formats that address some of these data links [7], but additional data formats must be established to realize the greatest potential of cyber-enabled manufacturing. Ideally, data necessary for part design, manufacturing, qualification and testing should be part of a single “digital thread” [8]. That is, essential parameters, from design to testing should be easily accessible, traceable and interoperable with all machines along the process chain.

In order to address data needs at various stages of AM, it is useful to view the entire process holistically. The additive manufacturing process can be simplified and considered as consisting of four phases: part design; path planning; execution of the part buildup; and, testing and verification. This simplified process description is illustrated in figure 1. Each phase of the process requires the generation and storage of a wide assortment of data.



```
UI 0
REM Contours For Layer 1
AC 1666667, 1666667, 166666
DC 1666667, 1666667, 166666
SP 83333, 83333, 83333
PA 6120,6120
BG XY
```

Figure 1: Simplified additive manufacturing engineering process. Feedbacks are illustrated by dashed arrows.

The first step within the part design phase is the construction of a 3-D object. This is typically generated as a solid model using 3-D Computer-Aided Drafting (CAD) software. The surfaces of the constructed object can be described using various file formats including STL or the, recently developed, Additive Manufacturing File (AMF) file [7]. While the STL file only describes geometry, the AMF file also allows materials and textures to be specified. ASTM International, which publishes the AMF standard, has noted that future versions of the standard may also include dimensional and geometric tolerances and provisions for surface roughness, support structures and surface textures. It may be noted that some manufacturers design a 3-D object "preform" to account for expected deformation, shrinkage and final machining of the part. Another factor that may influence part design is build orientation and support structure generation. Part orientation may not only determine the feasibility of the process and the required supports, but also build time and total cost.

Layer by layer slices are next generated using the object surfaces (typically using an STL file). Each slice along the buildup direction is of a defined thickness, dictated by the process conditions of the selected AM system, and is described as being two-and-a-half dimensional (2 ½ D). Within the slice file, the inner and outer perimeters (or contours) of each slice are described. Additional information regarding hatching (filling) of each slice may also be specified. For simple or heavily process-dependent geometries, process planning may be fully or partially manual, rather than automated. In these cases, machine code is directly generated, by an experienced operator, based on desired part geometry and prior process development. That is, rather than slicing a 3-D object, an operator devises a path plan, which may or may not be layer-by-layer, to build the object.

Following the generation of slices, a path plan is generated along each slice for part buildup. At this stage, path optimization—for example, via thermomechanical simulation—can be considered by taking constraints, e.g. build time, fixtures, and deformation, into account. Optimization of thermal history to produce a desired microstructure or to minimize residual stresses may also be considered here. Machine-specific processing parameters must also be specified for path planning. Although critical to defining the final geometry, microstructure, material properties, residual stresses and distortion of the final part, path planning and process data are typically not saved at this point. Instead, machine code is directly generated. While machine code is sufficient for replication of a process on a specific vendor's machine (assuming that essential process variables during part buildup are identical), it cannot be directly used to numerically simulate the process or to reproduce the process using a different vendor's technology. Additionally, interruptions of the buildup process, for example to heat-treat the part, clear a clogged powder nozzle or correct a problem with a wire feed, cannot typically be accounted for within the machine code. As previously noted, a path plan may also be partially or fully manually encoded. Inability to compare machine-to-machine build plans has been identified as a key challenge to structural design and qualification and certification of AM parts [2].

During the execution phase, the part is built-up contour by contour and hatch by hatch according to the machine code. Note, parts need not necessarily be built-up layer by layer. Each hatch and contour can be thought of an individual clad or weld. When thought of in this way, the need for recording of "essential variables", a term borrowed from the welding industry, along each clad or weld becomes readily apparent. While essential variables, such as processing power, translation speeds, part temperature, flow conditions, processing pressure, oxygen concentration, etc.., may be monitored for quality control, continuous recording of these variables is generally not done—no standard format exists for saving this data. Additional data may also include digital video or still images, e.g. melt pool shape or temperature, or measurement of real-time part deformation. Such data can also be fed back for real time process control, e.g. height or melt puddle control.

An additional step, not included in the simplified additive manufacturing engineering process shown in figure 1, is post processing of the part. Specifications for post processing of the part may be conceived within the design and planning phase and implemented at the end of the execution phase or following the testing and verification phase. Heat treatment of AM parts has been shown to have a significant effect on their mechanical properties [9]. A natural extension of the work presented here is the formulation of a standard format describing post processing and heat treatment.

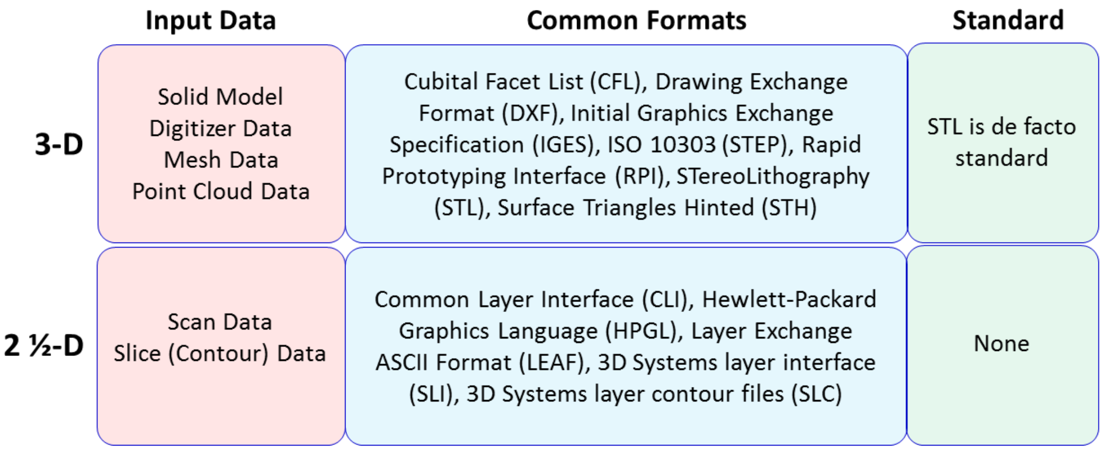
2 Data Standards

To date, standards development has focused on the early stages of the additive manufacturing engineering process, specifically defining the part geometry that can be fed to proprietary slicing and process-planning code. Several of the leading 3-D and 2 ½-D standards are described in this section and illustrated in figure 2.

2.1 Part Design—3-D standards

Part design, as shown in figure 1, is defined here as encompassing the 3-D design along with slice data. Common 3-D and slice file formats are shown in

figure **2** (adapted from [10]). As indicated in figure 2, the STL format is the de facto standard for transmission of 3-D design data. One challenger to the STL format is the ISO 10303 standard [11]. The 10303 standard, also known as STEP, has been championed by some [12] as the standard which ought to replace the STL standard within the AM community. Another, newly introduced competitor is the ASTM F2915 (AMF file) [7] standard. As will be discussed below, the ASTM F2915 standard has a foothold within the AM community and is rapidly gaining popularity.



	Input Data	Common Formats	Standard
3-D	Solid Model Digitizer Data Mesh Data Point Cloud Data	Cubital Facet List (CFL), Drawing Exchange Format (DXF), Initial Graphics Exchange Specification (IGES), ISO 10303 (STEP), Rapid Prototyping Interface (RPI), STereoLithography (STL), Surface Triangles Hinted (STH)	STL is de facto standard
2 ½-D	Scan Data Slice (Contour) Data	Common Layer Interface (CLI), Hewlett-Packard Graphics Language (HPGL), Layer Exchange ASCII Format (LEAF), 3D Systems layer interface (SLI), 3D Systems layer contour files (SLC)	None

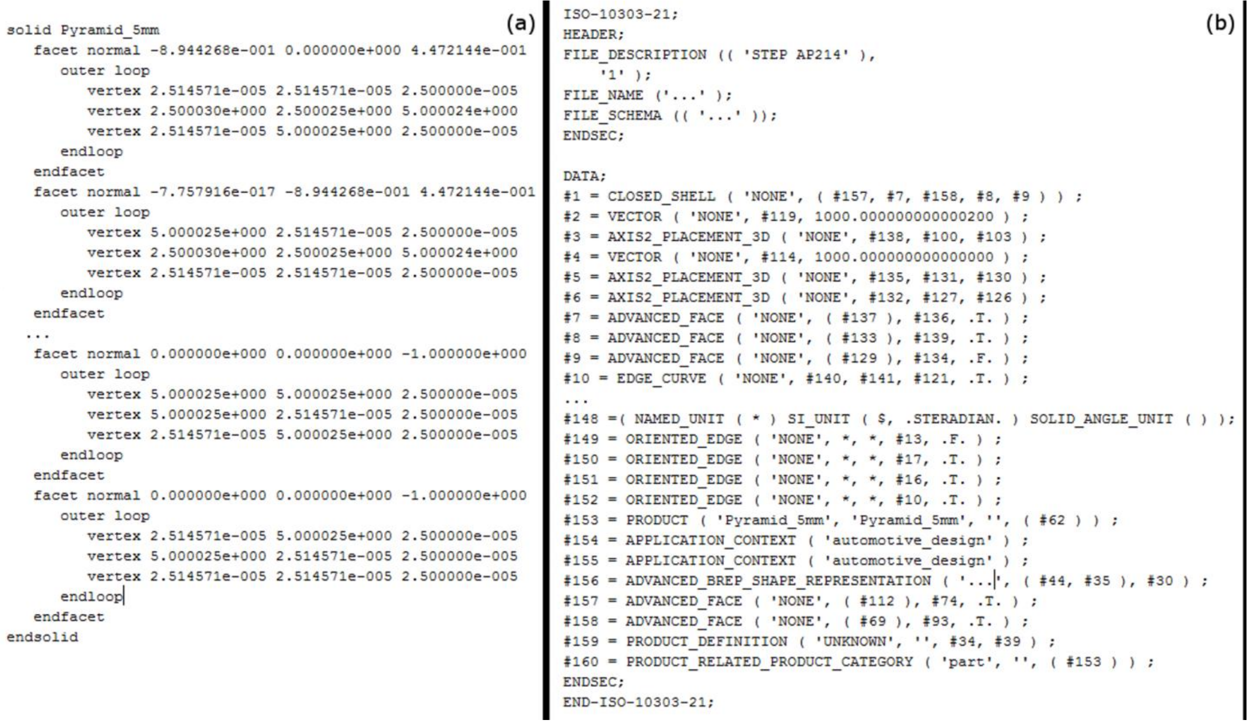
Figure 2 Common file formats of 3-D and slice files. Adapted from **[10]**.

2.1.1 Standard Tessellation Language Standard

The de facto standard representation of 3D part geometries is the Standard Tessellation Language (STL) format. This representation describes object surfaces using a triangular mesh. Each triangular facet is specified by a unit normal and three vertices. Facets are ordered arbitrarily while the order of vertices, following the right-hand rule, indicates the exterior of the object. An excerpt from an STL file, encoded in ASCII, is shown in figure 3(a).

In addition to its simplicity, the STL format has endured for over two decades due to several advantages. Expressing solids using triangular facets is "simple, robust and reliable" [12]. STL files are also accurate and, when saved in binary format, small [13]. Additionally, the format can accommodate a wide range of 3D representations [12].

The triangular facet approximation and the STL file schema do however have several drawbacks. While the triangular facet approximation is generally accurate, representation of curved surfaces can require a very large number of facets resulting in large file size. Redundancy of information contained within the file format, such as multiple vertices belonging to different facets and explicit inclusion of surface normals, which can be inferred from the order of vertices, also contributes to large file sizes. Another problem is that the format contains no scale information; AM slicing software is often left to guess the units based on part dimensions. The lack of topological information in STL files also contributes to gaps, degenerate and overlapping facets, and non-manifold topologies [12]. Other information, such as material, texture and life-cycle data are also not included within the STL format.



```
solid Pyramid_5mm
   facet normal -8.944268e-001 0.000000e+000 4.472144e-001
      outer loop
         vertex 2.514571e-005 2.514571e-005 2.500000e-005
         vertex 2.500030e+000 2.500025e+000 5.000024e+000
         vertex 2.514571e-005 5.000025e+000 2.500000e-005
      endloop
   endfacet
   facet normal -7.757916e-017 -8.944268e-001 4.472144e-001
      outer loop
         vertex 5.000025e+000 2.514571e-005 2.500000e-005
         vertex 2.500030e+000 2.500025e+000 5.000024e+000
         vertex 2.514571e-005 2.514571e-005 2.500000e-005
      endloop
   endfacet
  ...
   facet normal 0.000000e+000 0.000000e+000 -1.000000e+000
      outer loop
         vertex 5.000025e+000 5.000025e+000 2.500000e-005
         vertex 5.000025e+000 2.514571e-005 2.500000e-005
         vertex 2.514571e-005 5.000025e+000 2.500000e-005
      endloop
   endfacet
   facet normal 0.000000e+000 0.000000e+000 -1.000000e+000
      outer loop
         vertex 2.514571e-005 5.000025e+000 2.500000e-005
         vertex 5.000025e+000 2.514571e-005 2.500000e-005
         vertex 2.514571e-005 2.514571e-005 2.500000e-005
      endloop
   endfacet
endsolid
```

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (( 'STEP AP214' ),
    '1' );
FILE_NAME ('...' );
FILE_SCHEMA (( '...' ));
ENDSEC;

DATA;
#1 = CLOSED_SHELL ( 'NONE', ( #157, #7, #158, #8, #9 ) ) ;
#2 = VECTOR ( 'NONE', #119, 1000.000000000000200 ) ;
#3 = AXIS2_PLACEMENT_3D ( 'NONE', #138, #100, #103 ) ;
#4 = VECTOR ( 'NONE', #114, 1000.000000000000000 ) ;
#5 = AXIS2_PLACEMENT_3D ( 'NONE', #135, #131, #130 ) ;
#6 = AXIS2_PLACEMENT_3D ( 'NONE', #132, #127, #126 ) ;
#7 = ADVANCED_FACE ( 'NONE', ( #137 ), #136, .T. ) ;
#8 = ADVANCED_FACE ( 'NONE', ( #133 ), #139, .T. ) ;
#9 = ADVANCED_FACE ( 'NONE', ( #129 ), #134, .F. ) ;
#10 = EDGE_CURVE ( 'NONE', #140, #141, #121, .T. ) ;
...
#148 =( NAMED_UNIT ( * ) SI_UNIT ( $, .STERADIAN. ) SOLID_ANGLE_UNIT ( ) );
#149 = ORIENTED_EDGE ( 'NONE', *, *, #13, .F. ) ;
#150 = ORIENTED_EDGE ( 'NONE', *, *, #17, .T. ) ;
#151 = ORIENTED_EDGE ( 'NONE', *, *, #16, .T. ) ;
#152 = ORIENTED_EDGE ( 'NONE', *, *, #10, .T. ) ;
#153 = PRODUCT ( 'Pyramid_5mm', 'Pyramid_5mm', '', ( #62 ) ) ;
#154 = APPLICATION_CONTEXT ( 'automotive_design' ) ;
#155 = APPLICATION_CONTEXT ( 'automotive_design' ) ;
#156 = ADVANCED_BREP_SHAPE_REPRESENTATION ( '...', ( #44, #35 ), #30 ) ;
#157 = ADVANCED_FACE ( 'NONE', ( #112 ), #74, .T. ) ;
#158 = ADVANCED_FACE ( 'NONE', ( #69 ), #93, .T. ) ;
#159 = PRODUCT_DEFINITION ( 'UNKNOWN', '', #34, #39 ) ;
#160 = PRODUCT_RELATED_PRODUCT_CATEGORY ( 'part', '', ( #153 ) ) ;
ENDSEC;
END-ISO-10303-21;
```

Figure 3: Comparison of an (a) STL file with a (b) STEP file, both in ASCII format.

2.1.2 ISO 10303 Standard

The ISO 10303 standard, informally known as STandard for the Exchange of Product model data (STEP), was developed by the International Standards Organization (ISO) with the aim of establishing a single standard for product life-cycle data [11]. Life-cycle data extends beyond geometric data, such as that included in an STL file, and was envisioned to include all information regarding a product—from initial design to disposal.

STEP data are transferred between systems using a neutral 10303 format. An excerpt of a STEP file is included in figure 3(b). Schemas, descriptions of the structure and restraints on contents, of entries within the 10303 format are specified in the EXPRESS language. The EXPRESS language was initially developed for 3D geometrical modeling but, much like XML, can be extended to include any type of entity [14]. Data exchange standards defining the neutral 10303 format are called Application Protocols (APs). A large number of Application Protocols have been written, including those defining neutral file formats for exchanging drafting, 3D designs, structural analysis, electronic assembly, dimensional inspection, plant spatial configuration, material, verification, product life cycle and numerical control process plan data [15].

The use of STEP for additive manufacturing processes has been championed by several authors—see [12] and references therein. A primary advantage of using STEP is its ability to transfer not only process parameters and planning data but also “the results of build simulation and analyses on how different scan strategies will affect the final part” [12]. However, APs specific to additive manufacturing have yet to be developed. Other barriers facing the adoption of STEP for AM applications include the required familiarity with the, very complex, EXPRESS language and slow development of APs, compared with XML standards [14].

2.1.3 ASTM F2915 Standard

The ASTM F2915 standard specifies an XML (Extensible Markup Language) -based file format for additive manufacturing files [7]. XML hold several advantages over the EXPRESS language and other formats: it is self-describing and human-readable; it is simple to understand and use; and, it is ubiquitous and commonly used for online data exchange [16]. In addition to this, XML is extensible by nature—users can create and define tags, as they wish, to meet their specific needs. Files which conform to the ASTM F2915 standard are known as Additive Manufacturing File (AMF) files.

Within an AMF file, object surfaces are described by a triangular mesh. As in the STL format, each triangle is specified by a unit normal and three vertices. However, unlike the STL format, the AMF standard allows definition of curved triangles, in order to better model curved geometries using fewer triangular facets, and includes material, texture and constellation descriptions. Additional information can also be included as metadata. Metadata elements can be used to specify attributes of the overall file, such as authorship and part description, or attributes of surfaces, materials, textures or constellations. Moreover, the AMF format is such that STL files can be converted to AMF files directly and without loss of information [7].

A complete description of the AMF file is available in ASTM F2915 standard [7]. Here, portions of the file structure will be summarized for the reader’s convenience. Five top-level elements are specified within the AMF file format [7]: <object>, <material>, <texture>, <constellation> and <metadata>. All five elements are children of an <amf> element, which is the root element and are specified as follows.

- Surfaces of one or more build objects are defined within <object> elements.
- The <object> element requires the definition of a child mesh element, <mesh>.
- The <vertices> element is a child of the <mesh> element and contains a list of implicitly numbered vertex coordinates, contained within <vertex> element.
- Cartesian coordinates of each vertex are defined within a <coordinates> element, which is a child of the <vertex> element and contains the <x>, <y> and <z> elements as children.
- The object element must also contain at least one volume element, <volume>, which is a sibling of the <mesh> element.

- The <volume> element defines a closed surface based on the mesh of <triangle> elements.
- Each <triangle> elements is a child of the <volume> element and contains three vertices within <v1>, <v2> and <v3> elements.
- Contained in each of the <v1>, <v2> and <v3> elements is an index of a previously defined vertex.

Materials specified within <material> elements are referenced to within each <volume> element. Multiple materials, as well as mixed and graded materials, can be specified. Additionally, 2-D and 3-D texture maps can be encoded within the AMF file as a string of bytes (Base64) within the <texture> element. The <constellation> element can be used to specify the position and orientation of multiple objects. Any other attributes of the overall file or of surfaces, materials, textures or constellations can be included within <metadata> elements. An excerpt of an AMF file is shown in figure 4.

```
<?xml version="1.0" encoding="utf-8"?>
<amf unit="millimeter" version="1.0" xml:lang="en">
  <material id="1">
    <metadata type="Name">Ti64</metadata>
  </material>
  <object id="0">
    <mesh>
      <vertices>
      <vertex>
        <coordinates>
          <x>0.000</x><y>0.000</y><z>0.000</z>
        </coordinates>
      </vertex>
     ...
      <vertex>
        <coordinates>
          <x>5.000</x><y>5.000</y><z>0.000</z>
        </coordinates>
      </vertex>
      </vertices>
      <volume materialid="1">
      <triangle>
        <v1>0</v1><v2>1</v2><v3>2</v3>
      </triangle>
     ...
      <triangle>
        <v1>2</v1><v2>3</v2><v3>0</v3>
      </triangle>
      </volume>
    </mesh>
  </object>
</amf>
```

Figure 4: AMF file.

Although many have tried to modify, extend or replace the de facto standard of the STL file, the AMF file and other standards developed by the ASTM F42 committee have a promising chance of gaining hold within the additive manufacturing community. Reasons for this include the participation of a number of influential leaders in the AM field, including research laboratories, software developers, machine manufacturers, part fabricators, and the Society of Manufacturing Engineers' (SME) Rapid Technologies and Additive Manufacturing (RTAM) community [17]. Furthermore, the signing of a cooperation agreement between the ASTM F42 and ISO Technical Committee 261 allows for fast-tracking of ASTM standards as ISO standards [18]. Several popular commercial slicing packages, including Materialise® Magics 17 and netfabb, already support the AMF file format.

As a result of its origins in rapid-prototyping with non-metallic materials, a primary limitation of the ASTM F2915 standard is that not all essential data pertinent to laser- and electron-beam-based AM processes can be easily specified within an AMF file. For instance, information regarding the sequence and timing of deposition paths has been shown to influence the microstructure, residual stresses, and deformation of laser-deposited components [5] [19] [20]. The standard does allow for inclusion of non-standard data as metadata or as a new, unofficial element. According to the ASTM F291 standard [7], unofficial elements can be ignored by readers until officially accepted into the standard. However despite this advantage, the standard does not provide a clear framework for the inclusion of essential parameters required for replication or numerical modeling of a 3D part build. Specification of the laser or electron-beam deposition path would require the generation of a complex arrangement of <metadata> elements or the creation of unofficial elements. This however undermines several advantages of the XML Schema defined by ASTM F291—that it is intuitively structure, simple and standard.

2.2 Part Design—2 1/2-D Standards

Whereas 3-D data currently has a de facto, technology-independent, standard, in the form of an STL file, no de facto standard exists for slice data. Rather, several open-source and proprietary file formats are used, including Common Layer Interface (CLI), 3D Systems layer interface (SLI), 3D Systems layer contour files (SLC) and Layer Exchange ASCII Format (LEAF). Vendor-specific file formats are also often referred to as SLC or SLI files [3]. Among these formats, the open-source CLI is notable for its simplicity—slices are defined using polyline contours which specify both external and internal boundaries. Additionally, the CLI files format standard is freely available, both in American Standard for Information Exchange (ASCII) and binary encoding, on the internet (http://www.forwiss.uni-passau.de/~welisch/papers/cli_format.html).

CLI files begin with a header section which contains information regarding the file type (binary or ASCII), the units and the file version. The header may also contain the date, the coordinates of a bounding box, which contains the part, the number of layers used to construct the part as well as comments regarding the software used to produce the slice or the author. Contained within the body of a CLI file are geometric commands specifying coordinates of polylines used to construct outer and inner contours as well as hatches. Outer contours are specified using a counter-clockwise ordering of points while inner contours are specified using a clockwise ordering, when viewed in the negative build-up direction. A direction parameter is also included to reaffirm the ordering and indicate open lines, which can be used to indicate hatching or support structures [21]. A full description of the file format is available online [22].

2.3 Process planning standards

Processing paths are typically specified based on hatches constructed within slicing software, taking into account processing parameters such as hatch spacing and layer thickness, or directly by the AM machine's internal software. In most cases, planning and process data are not saved. Rather, machine code is directly generated. No standard specifications exist for transmitting technology-neutral processing paths and parameters. It may however be noted that the Hewlett-Packard Graphics Language (HPGL) [23], originally developed as a command set for pen plotters, and G-code [24] are commonly used with AM machines.

2.4 Execution and Verification Standards

No standards currently exist for data produced and recorded during the execution and verification phases of the AM process, e.g. temperature history, deformation, gas flow parameters, microstructure, tensile properties, etc… The ASTM F42 committee has to date released four standards related to additive manufacturing: Standard Terminology for Additive Manufacturing—Coordinate Systems and Test Methodologies (F2921), Standard Specification for Additive Manufacturing Titanium-6 Aluminum-4 Vanadium with Powder Bed Fusion (F2924), Standard Terminology for Additive Manufacturing Technologies (F2792) and Standard Specification for Additive Manufacturing File Format (F2915). Many more proposed standards are currently being developed; including, New Practice for Reporting Results of Testing of Specimens Prepared by Additive Manufacturing (WK30107), New Guide for Conditioning of machines and performance metrics of metal laser sintering systems (WK25479) and New Practice for Machine Operation for Directed Energy Deposition of Metals (WK37654).

Some of the proposed standards currently under review by the ASTM F42 committee may meet the need for standardized data formats at some phases of the AM process. For example, perhaps the New Practice for Reporting Results of Testing of Specimens Prepared by Additive Manufacturing (WK30107) may provide a standard way for reporting data recorded during the verification and validation phase. In the meantime, however, there is a clear and imminent need for standardized, or at the least open and easily-understood, formatting to enable definition of essential process data necessary for numerical simulation, replication and validation of numerical simulations, as well as for recording specimen test data.

3 A Proposed Digital Thread for AM

Rather than attempt to modify the AMF file format to include all the data required at every stage of the AM process, it is proposed that additional file formats be produced, each containing data, or a subset of data, pertinent to a specific phase of the process. Each format will mirror the example set by the ASTM F291 standard; the file formats will aim to be technology-independent (where possible), easy to understand, scalable, require reasonable computer resources to read and write, backwards compatible with existing file formats (where possible) and allow for easy extensibility to accommodate advances in technology. Like the ASTM F291 standard, data will be encoded in XML [16]. The proposed files are summarized in table 1 and will be discussed in the next sections. Together, the files will form a common digital thread. This digital thread will enable designers, manufacturers, end-users and modelers to easily transfer information and speak a common language with the ability to access only information which is of interest or all data at every phase of the AM process. Work is currently underway at Applied Research Laboratory at Penn State University to utilize the formats proposed here. This is viewed as key to cross-linking ongoing experimental work with simulation capabilities and verification efforts. This publication will only deal with the part design and process planning phases. The execution and testing phases are more complex and will be the topic of a future publication.

Table 1: Proposed file formats to contain data necessary at every phase of the additive manufacturing engineering process

Phase	Data Type	File Format
Part Design	3-D Design	AMF
	Slice	AMSF
Process Planning	Path Plan (and processing parameters)	AMPF
Execution	Sensor Data and Qualification Record	AMQF
Testing	Verification and Validation Data	AMVF

3.1 Slice

In addition to the AMF file, used to specify the 3-D design, four additional file formats are believed necessary and will be specified. The Additive Manufacturing Slice File (AMSF) will contain data regarding the slicing of the 3-D object and will be backwards compatible with the CLI file format. The AMSF will form part of the digital thread connecting all AM files. As such, information regarding the material, texture and color and constellation may be inferred from the AMF file. For example, the “materialid” attribute may be used within the AMSF to refer to a material defined within the AMF file. Alternatively material, texture, color and constellation data may also be specified within AMSF, following the AMF file standard. For cases in which both AMF and AMSF specify this data, data in the AMSF shall take precedence. An excerpt of the proposed AMSF file format is shown in figure 5.

As with the CLI format, the AMSF format will represent slices along the z-axis (the build-up direction as defined by ASTM F2921 [25]) using a polyline representation of the slice contours (boundaries). The definition of inner and outer contours as well as open lines and hatches will be identical to the CLI format [22]. Outer contours will be specified using a counter-clockwise ordering of points while inner contours will be specified using a clockwise ordering, when viewed in the negative build-up direction. As with the CLI format, a direction parameter will also be included to reaffirm the ordering and indicate open lines. Hatches will also be specified as in the CLI format, start and end (x,y) coordinates will be specified for each hatch.

```
<?xml version="1.0" encoding="utf-8"?>
<amsf angle="degree" unit="millimeter" version="1.0"
xml:lang="en">
  <object id="0">
    <transformation>
      <translation>0</translation>
      <rotation>0</rotation>
      <scaling>1</scaling>
    </transformation>
    <dimension>
      <x1>0</x1>
      <y1>0</y1>
      <z1>0</z1>
      <x2>5</x2>
      <y2>5</y2>
      <z2>5</z2>
    </dimension>
    <slices materialid="1">
      <numberOfSlices>19</numberOfSlices>
      <slice>
        <z>0.254</z>
        <polyline>
          <direction>1</direction>
          <points>
            <numberOfPoints>5</numberOfPoints>
            <pointCoordinates>
              <p1x>0.12701</p1x>
              <p1y>4.873</p1y>
              <p2x>0.12701</p2x>
              <p2y>0.12701</p2y>
              <p3x>4.873</p3x>
              <p3y>0.12701</p3y>

              <p4x>4.873</p4x>
              <p4y>4.873</p4y>
              <p5x>0.12701</p5x>
              <p5y>4.873</p5y>
            </pointCoordinates>
          </points>
        </polyline>
      </slice>
                        ...
      <slice>
        <z>4.826</z>
        <polyline>
          <direction>1</direction>
          <points>
            <numberOfPoints>5</numberOfPoints>
            <pointCoordinates>
              <p1x>2.413</p1x>
              <p1y>2.587</p1y>
              <p2x>2.413</p2x>
              <p2y>2.413</p2y>
              <p3x>2.587</p3x>
              <p3y>2.413</p3y>
              <p4x>2.587</p4x>
              <p4y>2.587</p4y>
              <p5x>2.413</p5x>
              <p5y>2.587</p5y>
            </pointCoordinates>
          </points>
        </polyline>
      </slice>
    </slices>
  </object>
</amsf>
```

Figure 5: Structure of proposed AMSF file format.

Ideally, all axes coordinates should conform to the ASTM F2921 standard [11]. This standard specifies an absolute, right-handed coordinate system having an origin at the center of the build volume, a Z-axis pointing in the build-up direction, an X-axis parallel to the front of the machine (pointing from left to right) and a Y-axis perpendicular to the X and Z axes. There is however a potential conflict between the ASTM F2921, coordinate system, standard and the ASTM F2915, AMF file, standards, as well as with the standard proposed here. The AMF file standard was designed for backwards compatibility with STL files. However, STL files typically require positive coordinates. While some software, such as SolidWorks 2012, do allow export of STL files with negative coordinates, others, like AutoCAD 2012, do not allow this. Therefore, vertex coordinates for files directly converted from STL to AMF will not conform to the ASTM F2921, coordinate system, standard. Moreover, some slicing formats, such as CLI and SLI, do not typically support negative coordinates—unsigned integers are typically used to encode point coordinates. Therefore, files directly converted from the CLI format to the proposed AMSF format may also not adhere to the ASTM F2921, coordinate system, standard. This seemingly trivial conflict may result in significant confusion at later stages in the digital thread—for example, in identifying the location where sensor data was recorded. Therefore, to maintain backwards compatibility to the STL and CLI formats, maintain compatibility with existing drafting and slicing software and reduce confusion, it is recommended that all coordinates be specified in the positive X-Y-Z octant within the initial AMF file. Additionally, any translation, rotation or scaling from the original AMF file should be specified within the AMSF.

One more point should be made with respect to the Z-axis coordinates of Slice files. The first slice (z=0) is typically empty. Some slicing software specify an empty first slice (at z=0), others begin with the first slice for which contours can be defined (at z= slice thickness). That is, since there is nothing to be sliced through at the bottom of the part, contours are not specified. Therefore, the first slice for which contour coordinates are specified will be where the z-axis equals the slice (layer) thickness. This may cause some confusion as some deposition systems define the starting position of the process at z=0. Slices can be thought of as defining the top z-coordinate of each layer.

The file declaration will be identical to that specified within the AMF file. The <amsf> element will be the root element. Within the <amsf> element, the version and unit attributes will be specified as in the AMF file but an additional attribute, angle, will be added. Possible values for the angle attribute will be "degree" and "radian." In its absence, "degree" will be assumed as the value of the angle attribute. Also, as in the AMF file, <metadata> elements will be used to specify file name information as well as any additional element or object information. Other elements will be specified as follows:

- The <object> element will be a top-level element. Within it, a unique identification number, id, attribute will be contained beginning with "0," for the part. The id number should equal the objectid, specified within the AMF file, of the sliced object.
- The <transformation> element will be a child of the <object> element. The <translation>, <rotation> and <scaling> elements will be children of the <transformation> element.

Their element contents will specify translation distance, rotation angle and scaling factor, respectively, applied to the part, specified within the AMF file, prior to slicing. Transformation order shall be implied by the order of the <translation>, <rotation> and <scaling> elements.

- The <dimension> element will be a sibling of the <transformations> element and will be used to specify the boundaries of a bounding box which contains the part. Note that the bounding box should enclose the object, described in the AMF file, to be sliced, not just the slices. Coordinates of the boundary box will be specified within the contents of the <x1>,<y1>,<z1>,<x2>,<y2> and <z2> elements, which will be children of the <dimension> element. Coordinates should be specified such that the contents of <x1> are less than <x2>, etc…
- The <slices> element will be a child of the <object> element. It will contain the <numberOfSlices> element and the <slice> element. An optional materialid attribute may be specified within the <slices> element. This attribute should refer to a material specified within the AMF file or within the AMSF.
- The number of slices specified for the object will be contained within the <numberOfSlices> element. If included, an empty first layer counts towards the total number of layers.
- The <slice> element will be a child of the <slices> element and a sibling of the <numberOfSlices> element. In the case of graded materials, an optional materialid attribute may be specified within the <slice> element instead of the <slices> element. This attribute, which allows for graded structures, should refer to a material specified within the AMF file or within the AMSF.
- Children of the <slice> element will include the <z> element along with the <polyline> element.
- The contents of the <z> element will define the z-axis coordinate at which a slice through the object, specified in the AMF file, was made. That is, the z-coordinate specifies the top of each layer.
- The <polyline> will have the <direction> and <points> elements as children.
- The contents of the <direction> element will be a redundant specification of the polyline orientation, where 0 indicates an internal contour, 1 an external contour and 2 an open line. To maintain compatibility with CLI files, open lines may be used to specify hatches or support structures. Orientation will be defined by the <direction> element in case of a discrepancy between the order of points and the <direction> element.
- The <numberOfPoints> element will be a child of the <points> element and will contain the number of points used to construct the polyline.
- The <pointCoordinates> element will be a child of the <points> element and will contain <p1x>,<p1y>,<p2x>,<p2y>,…,<pnx>,<pny> as children elements—the x and y coordinates of points along the polyline will be the contents of these elements. A clockwise ordering of points, when viewed in the negative z (build) direction, will

indicate an internal contour while a counter-clockwise ordering will indicate an external contour. The first and last coordinates along each polyline must match for closed contours.

Hatches can also be optionally included within the AMSF within a <hatch> element. In the absence of a path specification within the path plan file, which will be described in Section 3.2, the hatching contained within the AMSF shall be assumed to describe the processing path. An excerpt of the hatch contained within an AMSF is shown in figure 6. Note that, though hatches provide a machine or scan path, alone they do not provide enough information to perform machine programming or simulation, i.e. information such as the processing speed, laser/E-beam parameters and material feed rate are not included.

- The <hatch> element will be the child of the <slice> element.
- The <numberOfHatches> element and < hatchCoordinates > element will be children of the <hatch> element.
- The contents of the <numberOfHatches> element will be the number of hatches.
- The <hatchCoordinates> element will contain <hp1sx>, <hp1sy>, <hp1ex>, <hp1ey>,…, <hpnsx>, <hpnsy>, <hpnex>, <hpney> as child elements. The x and y coordinates of the start and end points of each hatch will be the contents of these elements.

```
<hatch>
          <numberOfHatches>9</numberOfHatches>
          <hatchCoordinates>
            <hp1sx>0.12701</hp1sx>
            <hp1sy>0.40201</hp1sy>
            <hp1ex>4.873</hp1ex>
            <hp1ey>0.40201</hp1ey>
            <hp2sx>0.12701</hp2sx>
            <hp2sy>0.95201</hp2sy>
            <hp2ex>4.873</hp2ex>
            <hp2ey>0.95201</hp2ey>
             . . .
            <hp8sx>0.12701</hp8sx>
            <hp8sy>4.252</hp8sy>
            <hp8ex>4.873</hp8ex>
            <hp8ey>4.252</hp8ey>
            <hp9sx>0.12701</hp9sx>
            <hp9sy>4.802</hp9sy>
            <hp9ex>4.873</hp9ex>
            <hp9ey>4.802</hp9ey>
          </hatchCoordinates>
        </hatch>
```

Figure 6: Example of hatches contained within an AMSF

3.2 Path Plan

Data regarding the path plans and processing parameters, such as power, speed and time will be contained within an Additive Manufacturing Path File (AMPF). This file captures the information necessary to generate machine code to drive a scanner or linear stages and to control the energy source, or to perform a thermomechanical simulation. While the path (AMPF) file contains sufficient data for reconstruction of slices (AMSF), it will be distinct in that it contains points and vectors describing the path of deposition as well as essential processing parameters required for replication and modeling. In contrast, the slice file exclusively contains a slice-based representation of the part. In other words, although geometric data contained within the AMF and AMSF files can, in principle, be reconstructed using the AMPF, they will be kept separate to ensure compatibility and ease of comparison with STL and CLI file formats, respectively, while still remaining part of the continuous digital thread. For cases in which AMSF contains hatch information and AMPF specify paths, data in the AMPF shall take precedence, with respect to the actual deposition path.

The path plan file (AMPF) will be structured similar to the AMF and AMSF files. An excerpt of an AMPF is shown in figure 7. The file declaration will be identical to that specified within the AMF and AMSF files. The <ampf> element will be the root element. Within the <ampf> element, the version unit and angle attributes will be specified as in the AMF file but additional attributes, time, mass, temperature, pressure, energy, power, voltage, current and flow will be added. Possible values for each attribute, along with default values, are given in table 2.

Table 2: Attributes contained within the <ampf> element specifying units.

Attributes	Possible Values	Default Value
unit	“millimeter”, “inch”, “foot”, “meter”, “micrometer”	“millimeter”
angle	“degree”, “radian”	“degree”
time	“second”, “millisecond”, “hour”	“second”
mass	“gram”, “kilogram”, “pound”	“gram”
temperature	“celsius”, “fahrenheit”, “kelvin”	“celsius”
pressure	“pascal”, “bar”, “atm”, “torr”, “psi”	“pascal”
energy	“joule”	“joule”
power	“watt”, “kilowatt”	“watt”
voltage	“volt”, “kilovolt”	“kilovolt”
current	“ampere”,	“ampere”
volume	“liter”, “gallon”, “cubicCentimeter”, “cubicMeter”, cubicInch”, “cubicFoot”	“liter”

Variables essential for modeling and reproducing the process will be contained at the beginning of the AMPF file. In determining which process variables ought to be specified, Weld Process Specification (WPS) standards adopted by the American Welding Society (AWS C7/C7.4M [26]) as well as by the American Society of Mechanical Engineering (ASME Boiler and Pressure Vessel Code, Section IX [27]) provide a starting point for specification of essential variables in AM processes. Both codes specify similar variables. It may be noted that equivalent or analogous variables are also used in electron beam welding WPS [28]. It should also be noted that many processes may not require specification of the all parameters and variables discussed in the following paragraphs. For those processes, users may wish to specify only those parameters which are essential to their process.

Many of the variables contained within WPS specifications for laser and electron-beam welding are directly applicable to additive manufacturing processes using lasers and electron beams. The recently adopted ASTM F2924 [29] standard also provides guidance as to which variable ought to be specified in AM processes. The XML language is especially well-suited for recording and transmitting such structured data. Within the AMPF, general data, such as the company information, date of production and a tracking or part number will be contained within a <general> element. All other data regarding the process will be contained within a <process> element which also contains an "id" parameter. This process id may be referenced further down the digital thread, within the AMVF. The reader is referred to figure 7 for an example of how process variables will be specified. In addition to specification of process variables, the option for including technical drawings or diagrams is also included. Within the <drawing> element, metadata describing the file contents, formatting, size and location will be included—the schema for this is loosely based on the Dublin Core Metadata Element Set (DCMES) vocabulary [30]. In this sense, "virtual datasets" of technical drawings can be constructed without incurring the high costs of data reformatting and transfer [31].

To be clear, the AMPF is not intended to be a qualification record. Variables recorded during processing will be recorded separately in a verification file (AMVF) which may then be used to qualify parts. A list of variables to be specified within the AMPF is shown in table 3. The structure and sample content of elements associated with these variables is illustrated in figure 7.

Table 3: Variables included in the proposed AMPF

Variable Category	Parent Element Type	Variable(s)	Child Element Type
General	<general>	Company Name	<company>
		Date	<date>
		Process/Part Number	<number>
Process	<process>	Laser/E-beam Settings, optics, environment configuration, materials, drawings	<laser>/<eBeam>, <optics>, <environment>, <configuration >, <base>, <filler>, <drawing>
Laser/E-beam Settings	<laser>/<eBeam>	Process category as specified by ASTM F2792	<category>
		Wavelength/Voltage	<wavelength>/<voltage>
		Current, Filament Type	<current>, <filament>
		Nominal beam profile at work piece	<profile>
		Laser Beam Quality/ E-beam raster	<quality>/<raster>
		Operating Mode	<mode>
		Power	<power>
		Pulse parameters: Energy, rate, length	<energy>, <rate>, <length>
Beam delivery optics	<optics>	Laser Polarization	<polarization>

		Spot size at work piece	<spotAtWorkpiece>
Environment	<environment>	Chamber pressure (absolute not gauge)	<chamberPressure>
		Gas compositions	<gas>
		Flow rates, Gas pressures	<flowRate>, <gasPressure>
		Flow orientation	<flowOrientation>
Process configuration	<configuration>	Angle of beam relative to part normal vector	<beamAngle>
		Controlled substrate temperature, cooling or heating	<substrateTemperature>, <substrateCooling>, <substrateHeating>
		Preheating, interpass and post heat treatment	<preHeating>, <interpassHeating>, <postHeating>
		Process Interruptions	<Interruption>
Base Material	<base>	Type	<type>
		Standard classification: M-number, UNS,ASTM Grade	<mNumber>, <uns>, <astmGrade>
		Shape	<baseShape>
		Geometry: thickness, length along x-axis, length along y-axis	<baseThickness >, <baseXDimension>, <baseYDimension>, <baseXCurvature>, <baseYCurvature>
		Description	<description>
Filler material	<filler>	Type	<type>
		Standard classifaction: UNS, ASTM Grade	<uns>, <astmGrade>
		Shape (wire or powder)	<shape>
		Dimensions: size, distribution, tap density	<size>, <distribution>, <tapDensity>
		Total mass feed rate	<massFeedRate>
		Method of delivery	<deliveryMethod>
		Number of feeders	<numberOfFeeders>
		Position (Feeder to workpiece distance)	<feederWorkingDistance>
		Description	<description>
Technical drawings	<drawing>	Title	<title>
		Creator	<creator>
		Description	<description>
		Date	<date>
		Format	<format>
		Identifier (link to file)	<identifier>
		File size	<fileSize>

Metadata	<metadata>

The AMPF will also define the process path used to construct the part and will include the power <powern>, speed <speedn>, beginning time <timen> and the start (<pnsx>,<pnsy>) and end (<pnex>,<pney>) coordinates for each (nth) path. A constant power for each layer, rather than a power for each path, can be specified using a <power> element and speed within a <speed> element. A materialid parameter can also be contained within <path> or <layer> elements as was done for <layer> and <polyline> elements within the AMSF. The materialid can refer to a material specified within the AMPF file or in one of the upstream files along the digital thread. Materials defined within the AMPF shall take precedence, over those specified in the AMSF or AMF files.

Only paths used for part build up will be included within the AMPF file. Dwell times and time used to move to the beginning of a path will be taken into account by defining the beginning time of each laser path. These elements will be specified as follows:

- The <object> element will be defined as in the AMSF format.
- The <layers> element will be a child of the <object> element and will have the <numberOfLayers> element and the <layer> element as children.
- The <numberOfLayers> element will contain the number of layers to be deposited. The number of layers may be one less than the number of layers specified in the AMSF file since an empty first layer can be used to indicate the first layer, as in a CLI file. This should however be avoided. If an empty first layer is specified in the AMSF file, an empty first layer should be specified within the AMPF file.
- The <layer> element will be a child of the <layers> element and have the <z> element as its child along with the <path> element.
- The <z> element will define the z-coordinate though which the slice was made. The first z-coordinate on which paths are specified should equal the layer thickness. Working distances are with respect to the first z-coordinate on which paths are specified.
- The <path> element will have <numberOfPaths>, <powers>, <times>, <speeds> and <points> as children elements.
- The < numberOfPaths > element will define the number of paths used for part construction.
- The <powers> element will have either the <power> element, for constant power along the entire layer, or <powern> elements, for a defined power along each (nth) path, as children.
- The <speeds> element will have the <speed> element, for constant speed along the entire layer, or <speedn> elements, for a defined speed along each (nth) path, as children.
- The <times> element will have <timen> elements, defining the time at the beginning of each (nth) path. All times are with respect to the first time on the first processing path, typically equal to zero.

- The <points> elements will have <pCoordinates>. Children of the <pCoordinates> element, <pnsx>,<pnsy>,<pnex> and <pney> will define the start and end (x,y) coordinates of each (n^{th}) process path.

```xml
<?xml version="1.0" encoding="utf-8"?>
<ampf unit="millimeter" time="second"  mass="gram" temperature="celsius" pressure="pascal" power="watt"
volume="liter" version="1.0" xml:lang="en">
        <general>
                <company>AM Corp</company>
                <date>11-12-13</date>
                <number>A1B2C3</number>
        </general>
        <process id="0">
                <category>direct energy deposition</category>
                <laser>
                        <wavelength>1070e-6</wavelength>
                        <profile>TEM00</profile>
                        <quality>1.1</quality>
                        <mode>CW</mode>
                        <power>450</power>
                </laser>
                <optics>
                        <polarization>random</polarization>
                        <spotAtWorkpiece>1042e3</spotAtWorkpiece>
                </optics>
                <environment>
                        <chamberPressure>101325</chamberPressure>
                        <gas>Argon</gas>
                        <flowRate>40</flowRate>
                        <flowOrientation>coxaial</flowOrientation>
                </environment>
                <configuration>
                        <beamAngle>0</beamAngle>
                        <preHeating>one laser scan prior to deposition</preHeating>
                        <postHeating>heat treatment at 700 C for 100 h</postHeating>
                        <interruption>pause process for 30 seconds after layer 5 </interruption>
                </configuration>
                <base>
                        <type>Ti-6AL-4V</type>
                        <mNumber>54</mNumber>
                        <uns>R56400</uns>
                        <astmGrade>5<astmGrade>
                        <baseShape>rectangular plate<baseShape>
                        <baseThickness>3.175</baseThickness>
                        <baseXDimension>76.2</baseXDimension>
                        <baseYDimensions>50.8</baseYDimensions>
                        <description>Flat plate purchased from ABCD corp<description>
                </base>
                <filler>
                        <type>Ti-6AL-4V</type>
                        <astmGrade>5</astmGrade>
                        <fillerShape>Powder</fillerShape>
                        <shape>spherical powder</shape>
                        <size>325 mesh</size>
                        <massFeedRate>0.05</massFeedRate>
                        <deliveryMethod>Coaxial Nozzle</deliveryMethod>
                        <numberOfFeeders>4</numberOfFeeders>
                        <feederWorkingDistance>9.27</feederWorkingDistance>
                        <description>virgin PREP powder purchased from ABCD corp delivered coaxially
                       by four nozzles. Nozzles are located at a working distance of 9.27 mm from work
                       piece.
                        </description>
                </filler>
                <drawing name="PowderNozzles">
                        <title>Orientation of Powder Nozzles</title>
                        <creator>ARL at PSU</creator>
```

```
            <description>Orientation of powder nozzles relative to substrate and laser
            beam.</description>
            <date>01-01-2013</date>
            <format mimeType="application/pdf"></format>
            <identifier>Drawing_Nozzles.pdf</identifier>
            <fileSize>147663</fileSize>
        </drawing>
    </process>
    <object id="0">
        <layers>
            <numberOfLayers>19</numberOfLayers>
            <layer>
                <z>0.254</z>
                <path>
                    < numberOfPaths >9</ numberOfPaths >
                    <powers>
                        <power1>450</power1>
                        <power2>0</power2>
                        <power3>450</power3>
                        <power4>0</power4>
                        <power5>450</power5>
                        ...
                        <power86>0</power86>
                    </powers>
                    <times>
                        <time1>0.00000</time1>
                        <time2>0.89988</time2>
                        <time3>1.79977</time3>
                        <time4>2.69966</time4>
                        <time5>3.59955</time5>
                        ...
                        <time9>37.19910</time9>
                    </times>
                    <speeds>
                        <speed1>10.58333</speed1>
                        <speed2>10.58333</speed2>
                        <speed3>10.58333</speed3>
                        <speed4>10.58333</speed4>
                        <speed5>10.58333</speed5>
                        ...
                        <speed9>10.58333</speed9>
                    </speeds>
                    <points>
                        <pCoordinates>
                            <p1sx>1.27010</p1sx>
                            <p1sy>0.40201</p1sy>
                            <p1ex>4.8730</p1ex>
                            <p1ey>0.40201</p1ey>
                            <p2sx>0.12701</p2x>
                            <p2sy>0.95201</p2y>
                            <p2ex>4.8730</p3x>
                            <p2ey>0.95201</p3y>
                            ...
                        </pCoordinates>
                    </points>
                </path>
            </layer>
            ...
        </layers>
    </object>
</ampf>
```

Figure 7: Structure of proposed AMPF file format.

5. Discussion, Concluding Remarks and Ongoing Work

The digital thread for additive manufacturing files proposed here hold many advantages over the status quo—a de facto STL standard for 3D data which is disconnected from a myriad of open source and proprietary slice formats which are disconnected from the actual path plan used for part build up. Adoption of a single digital thread, in the form of AMF, AMSF, AMPF, AMQF and AMVF files, will enable designers, manufactures, modelers and end-users to have complete access to the variables and parameters they need to better understand and document AM processes and to enable well-informed decision making. The formats presented here are flexible and will continue to evolve to the needs of users are every level of the AM engineering process.

This work demonstrates the importance of having an ability to input processing and path-plan data, using a neutral format. The proposed AMSF and AMPF formats address this need. These file formats will enable users to easily compare the performance of different AM software and hardware systems. It will also reduce the time required to learn vendor-specific software. While operators must still have a thorough knowledge of the AM system's operational capabilities and limitations, they will not have to learn a specific machine code or reverse-engineer a vendor's software and hardware to customize processing parameters and path plans. In the opinion of the authors, empowering operators with the ability to simulate, tune and validate processing parameters to obtain desired microstructures, stresses, and properties is critical to the further development and adoption of AM technologies. Such "open" formats will also drive competition amount numerical simulation software developers and enable users to readily compare and contrast different AM simulation software.

A key challenge to the adoption of the strategy proposed here may be the reluctance of AM machine manufacturers to adopt a non-proprietary format for transmission and input of process parameter data. In fact, AM systems manufactures have been known to charge hefty prices simply to enable operators to modify and develop new processing parameters. Operators may also be charged for material-specific processing parameters, which may be considered proprietary by systems manufacturers. Data encryption, together with the proposed formats, can be used to safeguard this intellectual property while allowing end users to easily accesses processing data. Standards organizations can play an important role with respect to this challenge.

Also critical to the wide-spread adoption of AM technologies are the recording and transmission of sensor data. Recording and transmitting time-dependent sensor data, such as time and/or spatially-dependent deflection or temperatures, within an XML format can however be problematic [31]: the format and encoding of multimedia data associated with a sensor is designated by its manufacturer, or chosen by the end user, and cannot be reasonably expected to adhere to a single standard; data may require proprietary software or algorithms to interpret; the data size may be enormous, especially for video data; and, end users may only be interested in small subsets of the data. One solution is to point to the data along with a description of the data (metadata) within the AMQF file. In this sense, "virtual datasets" can be constructed without incurring the high costs of data reformatting and transfer [31].

Several standards already exist which aim to describe the meaning and format of stored data. Among these standards are Dublin Core Metadata Element Set (DCMES) [30], MPEG-7 [32] and METS [33]. In many cases however, these standards require detailed knowledge regarding their encoding schema. Since users of the AQMF file are likely more interested in accessing and understanding sensor data rather than details concerning the data encoding schemas, a simplified schema is being developed at the Applied Research Laboratory at Penn State, built partly on the DCMES vocabulary [30] and METS [33] standards.

After execution of the part buildup, verification and validation of part parameters and properties is often necessary. Verification and validation data will be recorded within an Additive Manufacturing Verification File (AMVF). A wide variety of Non-Destructive evaluation (NDE) as well as destructive evaluations techniques can be used to evaluate and verify the properties of a part. The techniques used are largely dependent on a part's intended application. Therefore, as with sensor data, it is envisioned that the locations and format of verification and validation data be specified within the AMVF file along with any information necessary for analysis. The format and contents of the AMVF are also under development at the Applied Research Laboratory at Penn State.

Ideally, all five files as well as any multimedia data, such as images or video should be stored within the same directory or folder. Filenames and descriptions of multimedia data will be included within the AMSF and AMVF files. Together, all five files will provide all the data necessary to reproduce, numerically model and validate a part produced using additive manufacturing process.

Acknowledgements

This work was supported by the Office of Naval Research , under Contract No. N00014-11-1-0668. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

[1] I. Campbell, D. Bourell and I. Gibson, "Additive manufacturing: rapid prototyping comes of age," *Rapid Prototyping Journal,* vol. 18, no. 4, pp. 255-258, 2012.

[2] W. E. Frazier, "Direct digital manufacturing of metallic components: vision and roadmap," in *Twenty-First Solid Freeform Fabrication Symposium*, Austin, TX, 2010.

[3] K. Linebaugh, "GE Brings Engine Work Back," *The Wall Street Journal,* p. B2, 7 February 2013.

[4] National Center for Defense Manufacturing and Machining, "National Additive Manufacturing Innovation Institute," [Online]. Available: http://namii.org/. [Accessed 11 February 2013].

[5] P. A. Kobryn, N. R. Ontko, L. P. Perkins and J. S. Tiley, "Additive Manufacturing of

Aerospace Alloys for Aircraft Structures," AFRL/MLSC Acquisition Systems Support Branch Materials and Manufacturing Directorate Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio, 2006.

[6] Energetics Incorporated, "Roadmapping Workshop: Measurement Science for Metal-Based Additive Manufacturing," Gaitherburg, MD, 2012.

[7] ASTM International, "F2915-12 Standard Specification for Additive Manufacturing File Format (AMF) Version1.1," ASTM International, 2012.

[8] Society of Manufacturing Engineers, "Connecting the Digital Threads," *Manufacturing Engineering,* vol. 146, no. 6, pp. 33-36, 2011.

[9] M. Thone, S. Leuders, A. Riemer, T. Troster and H. A. Richard, "Influence of heat-treatment on selective laser melting products--e.g. Ti6Al4V," in *Twenty-Third Annual International Solid Freeform Fabrication (SFF) Symposium*, Austin, Texas, 2012.

[10] M. J. Wozny, "CAD and Interfaces," Rapid Prototyping Association of the Society of Manufacturing Engineering, Baltimore, Maryland, 1997.

[11] M. J. Pratt, "Introduction to ISO 10303 - the STEP Standard for Product Data Exchange," *Journal of Computing and Information Science in Engineering,* vol. 1, no. 1, pp. 102-103, 2001.

[12] A. L. Marsan, V. Kumar, D. Dutta and M. J. Pratt, "An assessment of data requirements and data transfer formats for layered manufacturing," U.S. Department of Commerce, 1998.

[13] C. K. Chua, K. F. Leong and C. S. Lim, Rapid Prototyping: Principles and Applications, World Scientific Publishing Co., 2010.

[14] M. Hardwick, "On STEP-NC and Complexities of Product Data Integration," *Transactions of the ASME,* vol. 4, no. 1, pp. 60-67, 2004.

[15] SCRA, "STEP Application Handbook ISO 10303 Version 3," North Charleston, SC, 2006.

[16] World Wide Web Consortium (W3C), "Extensible Markup Language (XML) 1.0 (Fifth Edition)," November 2008. [Online]. Available: http://www.w3.org/TR/xml/.

[17] "ASTM International News Releases," ASTM International, February 2009. [Online]. Available: http://www.astmnewsroom.org/default.aspx?pageid=1674. [Accessed January 2013].

[18] "Standardization News," ASTM International, November/December 2011. [Online]. Available: http://www.astm.org/standardization-news/outreach/astm-and-iso-sign-additive-manufacturing-psdo-agreement-nd11.html. [Accessed January 2013].

[19] S. M. Kelly, E. W. Reutzel and T. A. Palmer, "Impact of Path Planning on Thermal History during Direct Digital Deposition of Ti-6Al-4V," in *Materials Science & Technology 2012*, Pittsburg, PA, October 7-11, 2012.

[20] J. P. Kruth, M. Badrossamay, E. Yasa, J. Deckers, L. Thijs and J. V. Humbeeck, "Part and material properties in selective laser melting of metals," in *Proceedings of the 16th International Symposium on Electromachining (ISEM XVI)*, Shanghai, China, 2010.

[21] P. K. Venuvinod and W. Ma, Rapid Prototyping: Laser-based and Other Technologies, Norwell, Massachusetts: Kluwer Academic Publishers, 2004.

[22] A. Welisch, "COMMON LAYER INTERFACE (CLI) VERISON 2.0," [Online]. Available: http://www.forwiss.uni-passau.de/~welisch/papers/cli_format.html. [Accessed 14 February 2013].

[23] Hewlett-Packard Company, The HP-GL/2 Reference Guide - A Handbook for Program, Barcelona, Spain: Hewlett-Packard Company, 1996.

[24] International Organization for Standardization, ISO 6983-1:2009 Automation systems and integration -- Numerical control of machines -- Program format and definitions of address words -- Part 1: Data format for positioning, line motion and contouring control systems, 2009.

[25] ASTM International, "F2921-11e2 Standard Terminology for Additive Manufacturing—Coordinate Systems and Test Methodologies," ASTM International, 2011.

[26] American Welding Society, "AWS C7.4/C7.4M:2008 Process Specification and Operator Qualification for Laser Beam Welding," 2008.

[27] American Society of Mechanical Engineers, "2013 ASME Boiler and Pressure Vessel Code (BPVC), Section IX: Welding and Brazing Qualifications," 2013.

[28] American Welding Society, "AWS C7.3:1999R Process Specification for Electron Beam Welding," 1999.

[29] ASTM International, "F2924-12a Standard Specification for Additive Manufacturing Titanium-6 Aluminum-4 Vanadium with Powder Bed Fusion," ASTM International, 2012.

[30] Dublin Core Metadata Initiative, "Dublin Core Metadata Element Set, Version 1.1," 2012.

[31] R. E. McGrath, "XML and Scientific File Formats," The HDF Group, Champaign, IL , 2003.

[32] "MPEG-7," The Moving Picture Experts Group website, [Online]. Available: http://mpeg.chiariglione.org/standards/mpeg-7. [Accessed 02 2013].

[33] Digital Library Federation, "METS: Metadata Encoding and Transmission Standard Version 1.9.1," 2012.