XML Representation and Process Algorithm for Layered Manufacturing of Heterogeneous Objects

Wing Lau[1,2], Tom Bradbury, Adolphe Youssef, Chris Gaylo
[1]Therics Inc., 115 Campus Drive, Princeton, NJ 08540

And

Wei Sun, Alan Lau
[2]Department of Mechanical Engineering and Mechanics
Drexel University, Philadelphia, PA 19104

Abstract

In the fast developing technology of solid freeform fabrication, it remains a challenge to fabricate a heterogeneous object by layered manufacturing because conventional CAD systems and STL databases do not contain material and manufacturing information. This paper presents a novel approach to layered manufacturing of heterogeneous objects. A new processing algorithm based on the Extensible Markup Language (XML) format is being developed to process heterogeneous objects for layered manufacturing. The model database containing geometry, topology, material and manufacturing information is described by XML structural elements with the hierarchy analogized to the TREE data structure of the Boundary Representation (B-Rep) and/or the Constructive Solid Geometry (CSG) model. A process model for layered manufacturing is formulated based on the XML format for the fabrication of heterogeneous objects. A case study is presented to demonstrate the process algorithm for representing a multi-material B-Rep model in the context of Three Dimensional Printing (3DP) technology. Specifically, the case study will show the conversion of a solid model to XML representation, the material assignment, the slicing, and the machine toolpath generation. The paper also presents a brief survey of technological application of XML and associated technologies.

1. Introduction

In a layered manufacturing (LM) or rapid prototyping (RP) process, the STL file format, generated from a CAD system, is commonly used to approximate the shape of a solid model using small triangles called facets. A STL file lists the X, Y and Z coordinate triplets of the vertices forming the triangular facet and its normal vector. The next step in the RP process is to slice the STL file into layers of 2-D raster segments from which tool path instructions are generated for the RP machine. Since the STL file only contains geometry information, inherently it is good for representing homogeneous solids of one material. Various approaches have been studied by researchers attempting to model or process heterogeneous objects [1–6].

The objective of this paper is to present a novel approach to layered manufacturing of heterogeneous objects by means of applying the emerging XML technology. Section 2 gives a brief introduction to the structure of a B-Rep solid and XML. The benefit of the match between the two is also discussed. Section 3 gives an overview of the Fabrication Planning Program being developed in Therics for processing heterogeneous objects using XML structure for the 3-D printing. A case study of processing a simple heterogeneous CAD model is presented in Section 4. The current use of XML in other science and technology areas is discussed in Section 5. A summary and potential future work is given in Section 6.

2. Boundary Representation (B-Rep) Solids and XML

Boundary schemes are the most widely used representations for solids [7]. The basic concept in boundary representations is discussed in this section followed by a simple example showing the data structure for its topology and geometry.

Basically a solid in B-Rep is described by its bounding faces, while each face is defined by edges, and in turn, each edge is defined by vertices. Essentially a B-Rep can be represented by a tree structure with nodes corresponding to faces, edges and vertices. The relationship between the nodes is expressed by links connecting them. An example of a solid 2-inch cube is given below. Figure 1b shows the data structure in tabulated format while Figure 1c partially displays the data in tree structure format.

Face#	Description	Edges
1	Rear	1,2,3,4
2	Front	5,6,7,8
3	Left	12,4,11,8
4	Right	9,2,10,6
5	Bottom	5,9,1,12
6	Top	7,10,3,11

Edge #	Vertex Start	Vertex End
1	1	2
2	2	3
3	3	4
4	4	1
5	5	6
6	6	7
7	7	8
8	8	5
9	6	2
10	7	3
11	8	4
12	5	1

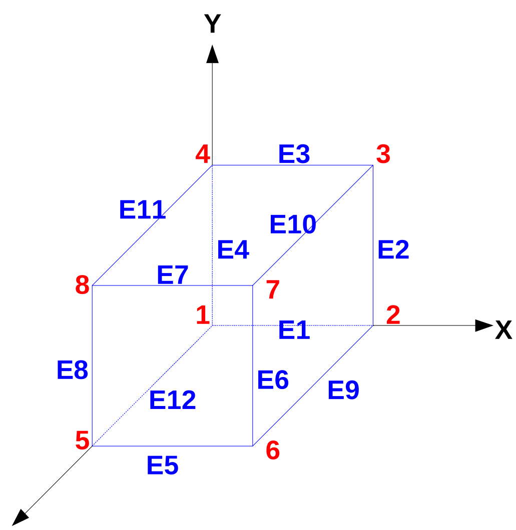


Figure 1a: 2-inch cube with edge#/vertex#

Vertex #	X	Y	Z
1	**0**	**0**	**0**
2	**2**	**0**	**0**
3	**2**	**2**	**0**
4	**0**	**2**	**0**
5	**0**	**0**	**2**
6	**2**	**0**	**2**
7	**2**	**2**	**0**
8	**0**	**2**	**2**

Figure 1b: Tabulated data structure for a 2-inch cube

XML (Extensible Markup Language) is an open standard managed by World Wide Web Consortium (W3C) for describing structured information. Similar to the tree structure, XML file contains data in hierarchical order. In general, it can be used in any problem domain, which exhibits hierarchical nature. For example, the structure of an invoice and a vertex coordinates can be described by XML as in Listings 1 and 2 respectively.

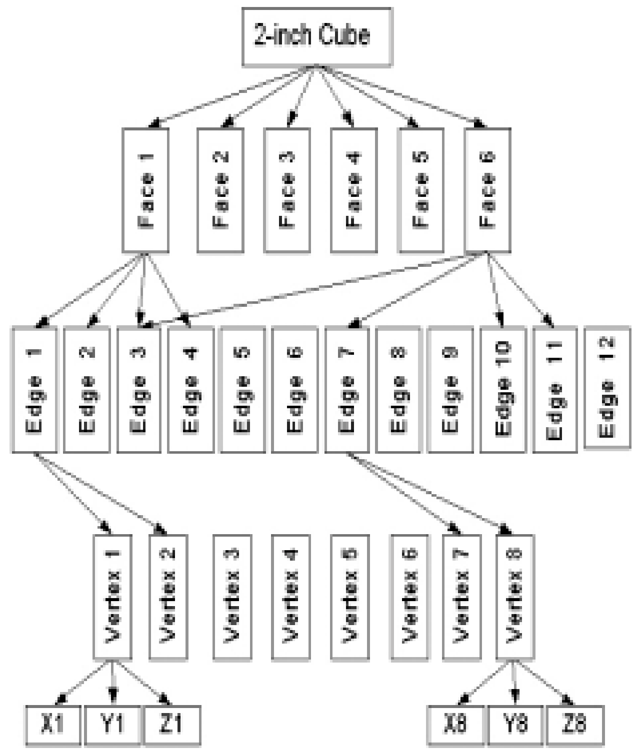


Figure 1c: Tree structure for a 2-inch cube

Listing 1: Business Application

```
<Invoice>
        <Customer>
                <Name>ABC Inc.</Name>
                <Address>1234 Main Street</Address>
                <Phone>609-500-7000</Phone>
        </Customer>
        <Amount>1000.00</Amount>
</Invoice>
```

Listing 2: CAD Application

```
<Vertex ID=”1”>                         Vertex is an element with ID as its attribute
        <X>2.5</X>
        <Y>3.0</Y>
        <Z>3.5</Z>
</Vertex>
```

XML is made up of elements (or tags) and attributes and each element can contain any number of other elements, thus forming a hierarchy like a tree structure. It is flexible that it does not have a predefined set of tags, which allows customization to fit particular problem domain. It is extensible that new elements can be added as needed.

As XML is a machine and human readable cross-platform language, some standard APIs (Application Program Interfaces) have been created to manipulate XML data. These APIs are implemented in JavaScript, Java, Visual Basic, C++, Perl and many other languages. These provide a standard way of manipulation, and developing for, XML documents [9].

XML-related technology continues to evolve and Figure 2 shows a list of the current technologies being built around XML. DOM (Document Object Model) and SAX (Simple APIs for XML) are used programmatically for accessing and manipulating XML data. DTD (Document Type Definition) and Schema are used to define constraints for individual data [10]. The other technologies are more related to the internet and interoperability between processes.

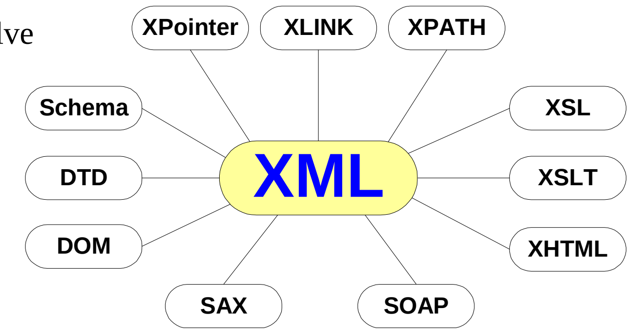


Figure 2: XML-Related Technology

From the discussion just given, it is evident that a B-Rep solid can be expressed in XML format. The XML format for the 2-inch cube given in Figure 1 is shown in Figure 3.

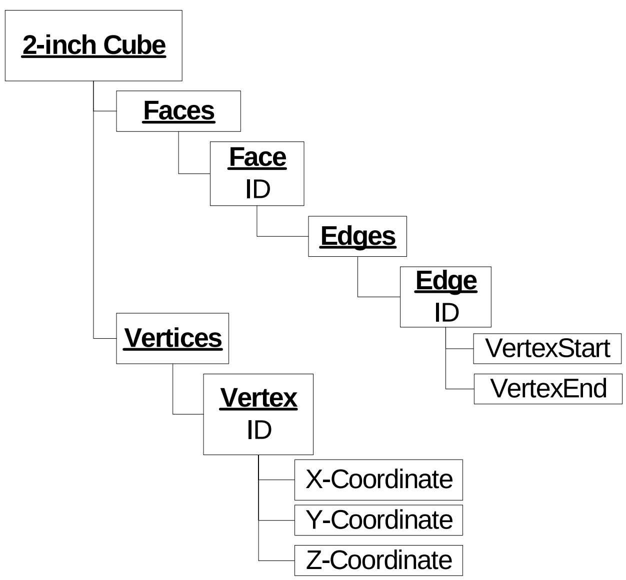


Figure 3a: XML Structure of a 2-inch Cube

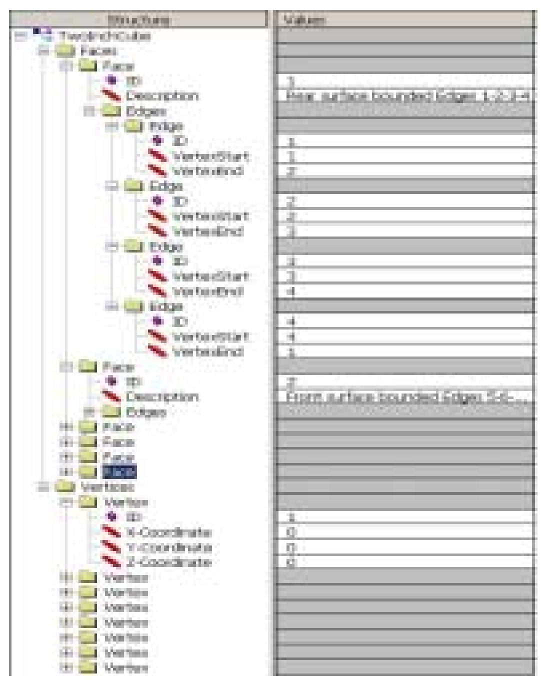


Figure 3b: XML Structure & Data of a 2-inch Cube displayed in XML Notepad

The benefit of this representation is flexibility and extensibility in which additional attributes such as material and color, specific manufacturing info can be included. In terms of the layered manufacturing of a heterogeneous model, a XML structure containing shape information (topology and geometry), attributes (material and color), and related manufacturing information (slicing and layout) can be employed. More detail on the implementation is given in the next two sections.

Being embraced by the major players in the industry, there are many commercial tools currently available for manipulating XML files. A number of new technologies building around XML continue to emerge.

3. Processing B-Rep Heterogeneous Model with XML

To facilitate the assignment of attributes (material and color) and specification of manufacturing information, a customized software application, called the Fabrication Planning Program, is being developed. The overall process algorithm is shown schematically in Figure 4. The essential functional components are:

(1) Read in and parse a B-Rep model (in STEP or Parasolid XT format) generated by a CAD system.
(2) Convert the topology and geometry to XML format.
(3) Provide an option to assign material and color attributes to each geometric body.
(4) Slice the multi-material model into layers of raster segments with assigned material being carried over.
(5) Based on the sliced data, generate the machine toolpath instructions with associated material information.

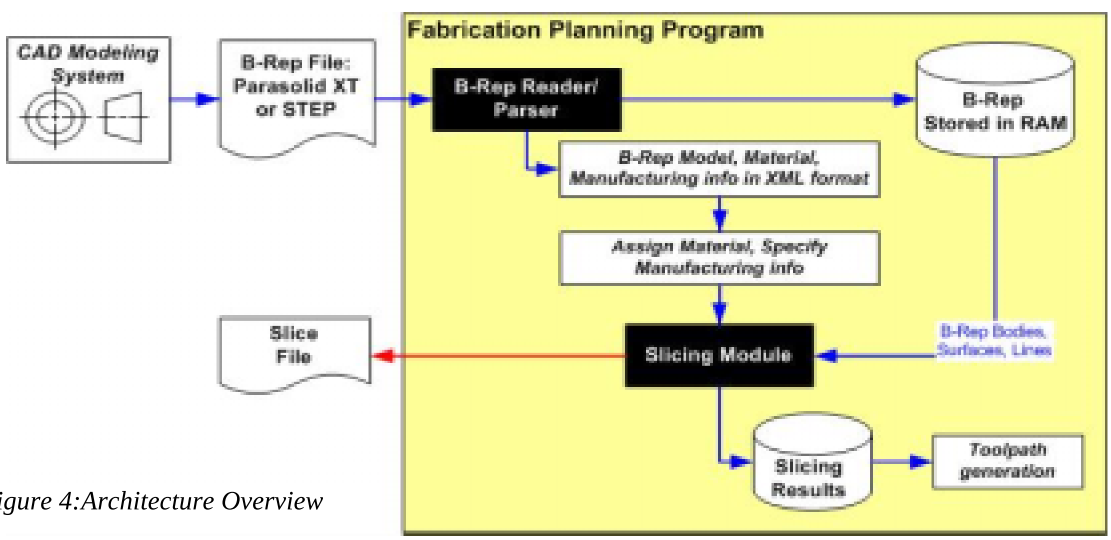


Figure 4:Architecture Overview

In terms of implementation, the step (2) above can be handled in either one of the following methods:

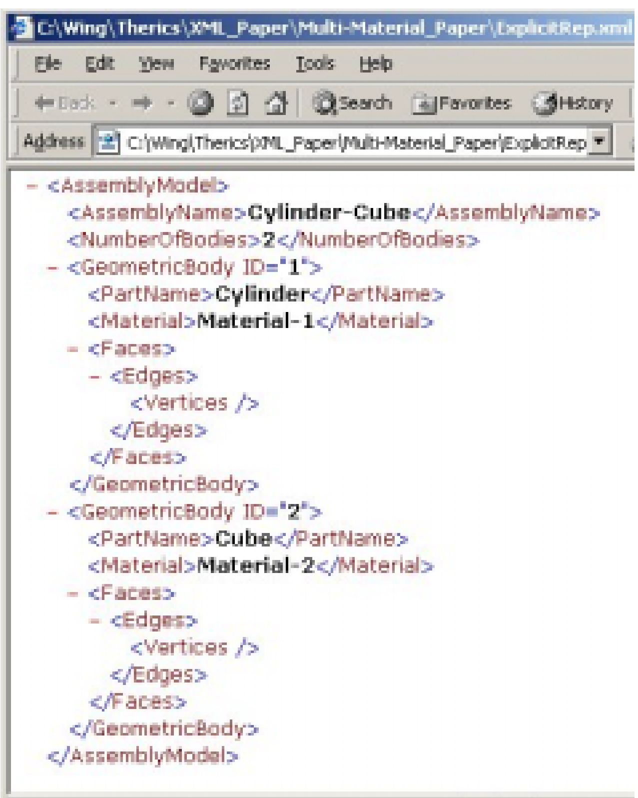
Method (1): Explicit Representation - Convert the model's topology and geometry in terms of XML nodes as described in Section 2. Significant effort may be required for the conversion. This will explicitly express the model's topology/geometry, material and related manufacturing information in terms of XML tree nodes. Customized slicing algorithm may also be required to work with topology/geometry in XML format.

Method (2): Implicit Representation - This is similar to (1) except that a node is created with link to external B-Rep file instead of converting the B-Rep model into explicit XML nodes. This approach is efficient if third-party modeling toolkit or light-weight modeling kernel is employed because the model's topology and geometry can be maintained in the native format of the toolkit or kernel.

Figures 5a and 5b show the possible XML structure, rendered in Internet Explorer, for an assembly model having two geometric bodies with different materials in explicit and implicit representations respectively.

The Fabrication Planning Program being developed is using the implicit representation in conjunction with a light-weight solid modeling kernel. XML nodes, specific to Therics' 3DP manufacturing process, are created for best fit of the problem domain. This highly flexible and extensible characteristics of XML structure make the application as light-weighted as possible.

STEPml (XML for STEP, ISO 10303) is a standard that has been developed using the explicit scheme. In addition to embedded shape information, STEPml contains a lot of product information. The overall hierarchy appears fairy deep and complex, and therefore may require a specialized commercial toolkit to handle. More detail on STEPml can be found in Section 5.



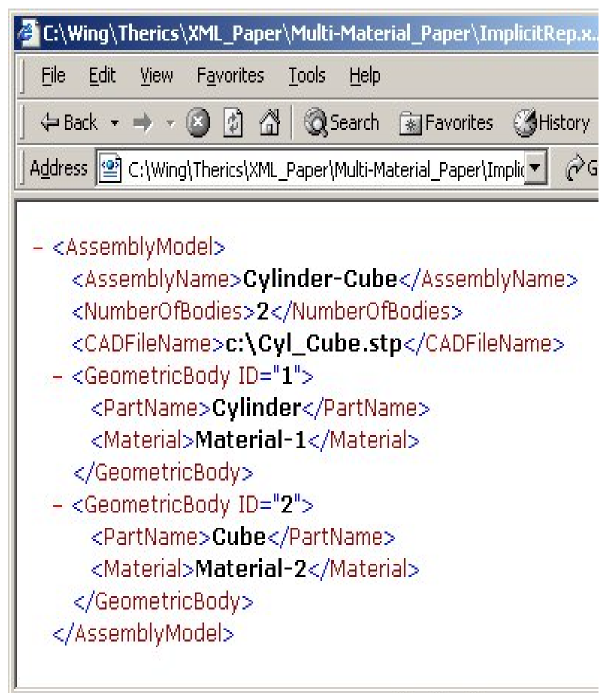


Figure 5a: Explicit Representation *Fig*

4. Case Study

This section presents a case study in which a simple 2-body component (or assembly) has been processed by the Fabrication Planning Program for TheriForm™ Fabrication, a 3-dimensional printer developed and built by Therics Inc.

A STEP file is used in this case study and the five steps outlined in Section 3 are followed as shown in the following figures.

- Read CAD file
- Express in XML format using implicit scheme
- Assign materials and specify manufacturing information
- Slicing and toolpath generation

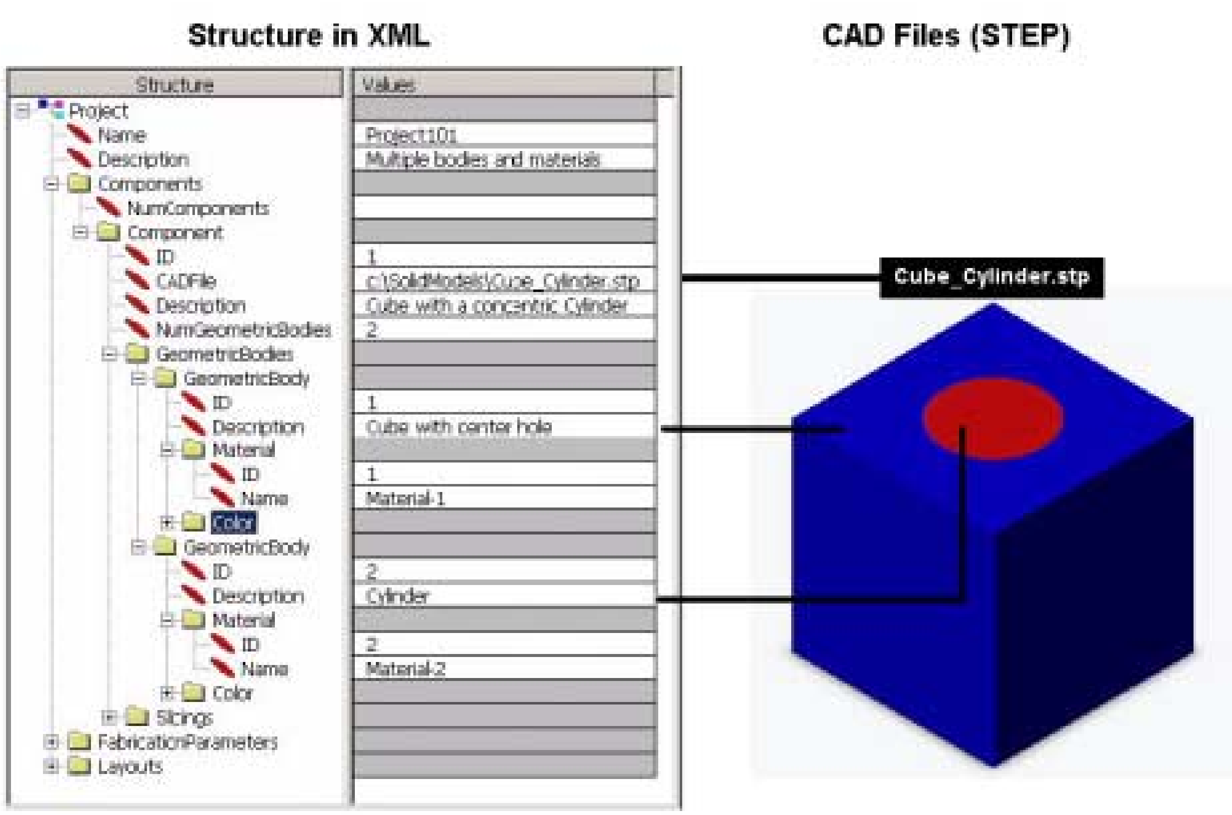


Figure 6: XML structure for fabrication of 2-body CAD model

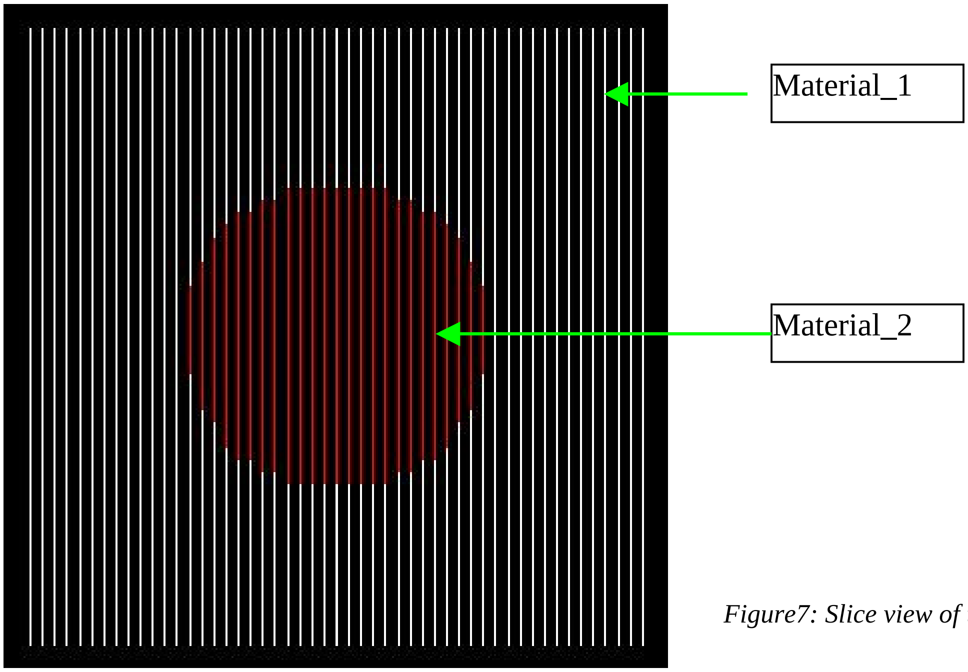


Figure7: Slice view of the first slice

Figure 7 is a slice view displaying all the sweeps and raster segments composing the structure of the first slice. All the slicing results are stored in XML format. For example, the structure for the first sweep in the first slice can be expressed as given below. The material assigned for each geometric body will be carried over through the slicing process to the slices' data structure, from which multi-material processing instructions are generated for manufacturing.

Listing 3: XML Structure for (Slice, Sweep, RasterSegment) = (1, 1, 1)

```
<Slice ID=”1”>
        <Z> 0 </Z>
        <Sweep ID=”1”>
                <RasterSegment ID=”1”>
                        <X> 0 </X>
                        <StartY> 0 </StartY>
                        <EndY> 10 </EndY>
                        <Material> Material_1 </Material>
                </RasterSegment>
        </Sweep>
</Slice>
```

5. Current use of XML in Science and Technology

There are increasing number of technological areas utilizing the power of XML. A partial list of these applications is given in this section. Detailed information can be found in respective reference.

- **<u>eXT</u>** - Parasolid XT is a format published by Unigraphics Solutions (UGS) for storing MCAD data. Since XT only contains part data, it does not satisfy the requirements of product data exchange. Parasolid Extended XT (eXT), implemented using XML, is intended to provide a mechanism for exchanging product, part and process data. eXT can be used to represent product structure, shape information and process information [7]. However, the shape information (ie. Parts), can be represented using external references or internally stored within the product definition itself.

- **<u>STEPml</u>** – STEP standard, ISO 10303, has become the predominant standard for the definition, management, and interchange of "product" data, being used in a wide variety of industries [12]. STEPml takes the data models from STEP and publishes them as XML specifications. This brings together the rich semantics of STEP and the widespread adoption of XML technology [13]. Figure 8 shows a simple STEP model and its representation in STEPml.

- **FemML**: Data exchange between Finite Element Analysis Codes. Proposed by Naval Research Laboratory [17].

- **CML**: Chemical Markup Language (CML) is a new approach to managing molecular information [15].
- **MathML**: It is a low-level specification for describing mathematics as a basis for machine to machine communication. It provides a much needed foundation for the inclusion of mathematical expressions in Web pages. Open standard by W3C [8].

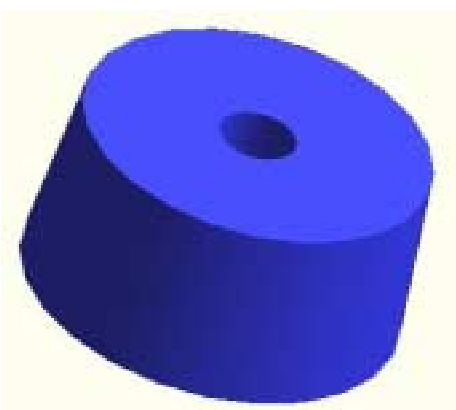


Figure 8a:STEP model

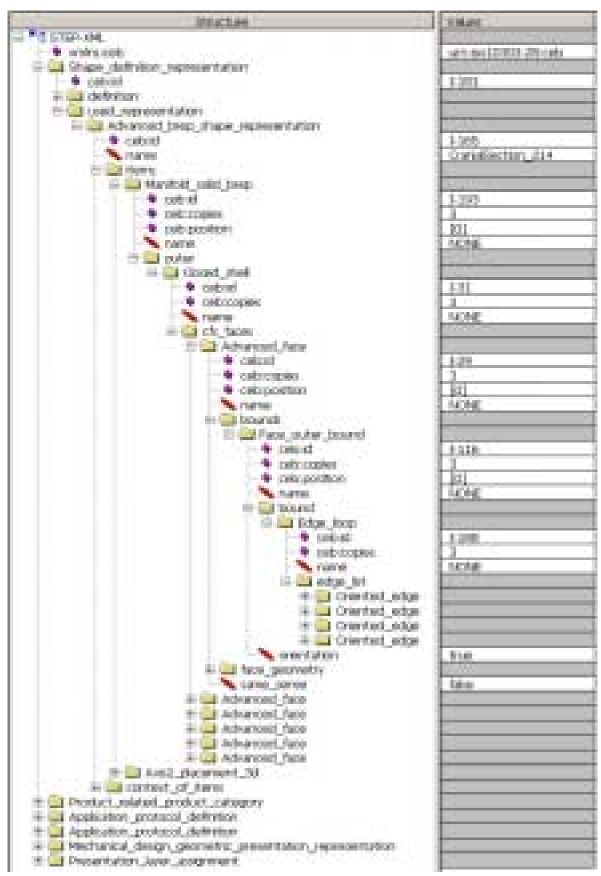


Figure 8b:STEPml structure

- **SVG**: Scalable Vector Graphics (SVG) is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes, images and text [8].

- **MRML**: Medical Reality Modeling Language (MRML) is a format for describing 3D scenes that consist of various types of medical data sets collected in various geometric locations [16].

6. Concluding Remarks

This paper presents a process algorithm by which a multi-material CAD model can be processed for layered manufacturing. XML structure is used to include shape information implicitly or explicitly, material/color attributes for each geometric body and other essential manufacturing information. The shape information along with material attribute is then fed into a slicing module. The slicing results thus obtained will have material attributes being carried over. The last step in the process is to interpret the slicing data with material attributes into machine instructions for manufacturing.

There is a potential to extend the current slice's data structure by including material composition in order to process Functionally Graded Material (FGM). For simple linear varying function, say along the X-axis, the structure shown in Figure 9 can be used for corresponding machine toolpath generation. This can be achieved, at the slices' level, by providing a user-interface for entering the linear function.

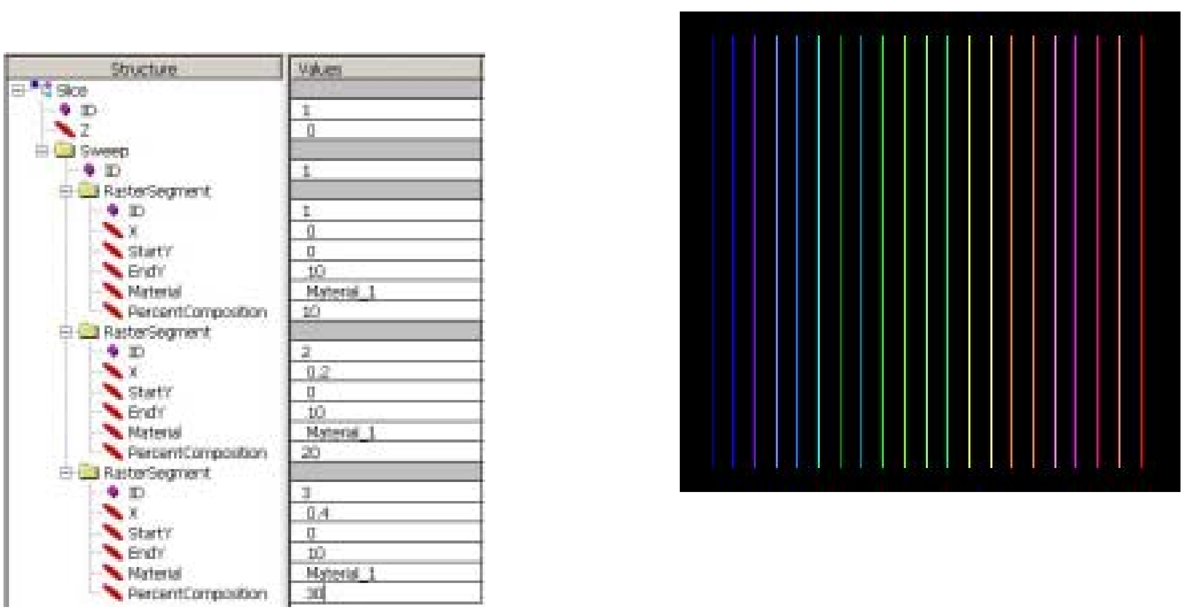


Figure 9: Slice's XML structure for FGM

7. References

[1] Cheng-Tiao Hsieh and Noshir A. Langrana "A System Approach in Extrusion-Based Multi-Material CAD", *Proceedings of the Solid Freeform Fabrication Symposium 2001*, Austin, Texas.

[2] Stephane M. Morvan and Georges M. Fadel, "Heterogeneous Solids: Possible Representation Schemes", *Proceedings of the Solid Freeform Fabrication Symposium 1999*, Austin, Texas. pp. 187-197

[3] Ashok V. Kumar and Aaron Wood, "Representation and design of heterogeneous components", *Proceedings of the Solid Freeform Fabrication Symposium 1999*, Austin, Texas. pp. 179-196

[4] X. Hu, J. Tao, F. Lin and W. Sun, "Reasoning Boolean Operation for Modeling, Simulation and Freeform Fabrication of Heterogeneous Objects", *Proceedings of the Solid Freeform Fabrication Symposium 2000*, Austin, Texas.

[5] Sun W., "Multi-volume CAD Modeling for Heterogeneous Object Design and Fabrication", *Journal of Computer Science and Technolog*, Vol. 15, No. 1, 2000, pp. 27-36.

[6] Sun W. and Lau, A., "A Knowledge-enriched CAD Modeling and Solid Free-form Realization for Heterogeneous Material Structures", *Proceedings of the Seventh International Conference on Rapid Prototyping*, March 31 – April 3, 1997, San Francisco, CA, pp. 79-87.

[7] Geometric Modeling: A First Course, Ari Requicha, http://www-lmr.usc.edu/~requicha/book.html

[8] World Wide Web Consortium (W3C), http://www.w3c.org

[9] Professional XML 2nd Edition, Dinar Dalvi, Wrox Press Ltd.

[10] Using XML Schema, David Gulbransen, QUE

[11] Parasolid eXT Web site: http://www.extxml.org
Parasolid eXT V2.0, eXT Functional Description (available in www.extxml.org)

[12] XML Representation Methods for EXPRESS-Driven Data, NIST GCR 99-781, by W. Eliot Kimber, ISOGEN International Corporation.

[13] STEPml XML Specifications by Robin Cover, http://xml.coverpages.org/stepml.html

[14] STEPml website, http://www.stepml.org/index.html

[15] ChemML website, http://www.xml-cml.org/

[16] Medical Reality Modeling Language, http://slicer.ai.mit.edu/mrml/index.html

[17] J. Michopoulos, P. Mast, T. Chwastyk, L. Gause, R. Badaliance, "FemML for Data Exchange between FEA Codes", *Proceedings of ANSYS Users' Group Conference*, University of Maryland, College Park MD, October 2, 2001.