Overview:
Encapsulation of OBJ Models for 3D Manufacturing and Virtual Reality

Allan Noordvyk & Justin Ryan
Co-Chairs of WG17: 3D Manufacturing
Contents

• Background
• Direction & Current Challenges
• Main Components
• Expected Use
• Specific Changes
Background
• Allow store/query/retrieve 3D models, intended for 3D manufacturing (and virtual reality), as DICOM objects
• Addressed by Work Item 1
• Leverage
  a) Existing and growing ecosystem of DICOM-capable systems in use in healthcare institutions and
  b) Standards and conventions already in use in the 3D printing industry
In 2018 WG17 focused on getting **DICOM Encapsulated STL** was added to the standard

- This provides a lowest common denominator for use cases
- It was recognized that while everyone can utilize STL, there are more advanced options
• Approached by members of medical **Virtual Reality (VR)**, **Augmented Reality (AR)**, and **Mixed Reality (MR)**

• This community also uses non-medical 3D models and have **overlapping use cases** with 3D manufacturing

• WG17 is now including **their input** into selection of formats for encapsulation and other needs
  
  • Primary format in current VR/AR/MR use is **OBJ**
  • Also concerned with multi-part assemblies and color
Direction & Current Challenges
Direction

• Both the 3D printing and AR/VR/MR communities (together known as XR) have provided the following direction to WG17...
  • Address limitations of STL by allowing option for encapsulation of a more advanced format
    • Select based on current ubiquity of use in both communities
  • Address model management challenges related to...
    • Multi-part assemblies
    • Persistent component color
Challenge 1: Beyond STL

• Limitations of STL format
  • No ability to indicate color/texture individual polygons in model
    • Important for replicating real-world appearance of modeled anatomy/pathology or delineate between structures
  • Poor adoption in virtual/augmented/mixed reality applications

• Many other 3D model file formats address these deficits (OBJ, X3D, AMF, 3MF)
  • OBJ format has high current adoption among both 3D printing and VR/AR/MR applications & users
Challenge 2: Assemblies

• Many 3D models are meant to be assembled together, example:
  • Multi-part implants
  • Training simulators requiring different materials
  • Explorable anatomic models
• May be multiple assemblies in the same DICOM study, example:
  • Left and right versions of multi-part implants
• Any convention using study and series can be ambiguous and inconsistent
• Desire to explicitly leverage DICOM identify which subset of models belong to the same assembly
Challenge 3: Persistent Color

• Many situations where **specific preferred color** should be used for a specific model
  • Example: Color-coded assemblies of multiple models (bone, venous, arterial, …)

• No good solution inside STL or OBJ models…
  • STLs **completely lacks** standard ability to indicate the **color** of the model
  • OBJ can indicate color, but it must be done on the **polygon-by-polygon or vertex-by-vertex** level (overkill)

• Desire to leverage DICOM to persistently indicate **desired color** for a specific model
Main Components of Supplement
This Supplement

• The second output of work item 1 is 
  Supplement 208: DICOM Encapsulation of OBJ Models for 3D Manufacturing and Virtual Reality

• Enable encapsulation of OBJ in a pathway similar to STL encapsulation

• Augment current encapsulation approach for assemblies and color
New Information Object Definition (IOD)s:
  • Encapsulated OBJ (and supporting files) for Creation, Review, Update, and Printing (manufacturing)
  • Encapsulated MTL

New Attributes:
  • Reference Name
  • Model Group
  • Recommended Display CIELab Value
  • Recommended Presentation Opacity
  • Referenced Instance Sequence
  • Referenced Image Sequence

New IOD Devices:
  • Texture Map
Expected Use
The new 3D Model encapsulation attributes is expected to address these real world use cases

- Model Group
- Component Color/Opacity
Expected Use: Model Group

- Medical reconstruction software queries Image manager system

- **User creates patient-specific 3D model** (reconstruction and modeling)

- User segments different regions into discrete manifolds (e.g., aorta, pulmonary artery, and airway)

- Modeler system creates 3 DICOM objects containing the 3D models

- Specifying **same Model Group UID in each object** enables modeler or subsequent DICOM-enabled software to identify group for joint printing / presentation

- [To 3D printer or XR system]
Expected Use: Model Group (Alternate)

- Medical reconstruction software queries Image manager system
- User queries **DICOM Segmentation** objects and imports into 3D modeler software
- Modeler system creates 3 DICOM objects containing the 3D models
- Specifying **same Model Group UID in each object** enables modeler or subsequent DICOM-enabled software to identify group for joint printing / presentation
- [To 3D printer or XR system]
Expected Use: Component Color

- Medical reconstruction software queries Image manager system
- **User creates patient-specific 3D model** (reconstruction and modeling)
- User segments different regions into discrete manifolds (e.g., aorta, left ventricle, left atrium)
- Modeler system creates 3 DICOM objects containing the 3D models
- **Color each component** (Recommended Display CIELab Value)
- Assign alpha/transparency (Recommended Presentation Opacity Value)
- [To 3D printer or XR system]
The proposed OBJ encapsulation necessitates multiple file encapsulation

1. An OBJ object may actually be comprised of 2 or more files:
   • 1 OBJ main file
   • 0-1 MTL supporting file
   • 0-n Texture Map Image supporting files

2. These files currently refer to each other by filename
Background - OBJ Schema

OBJ

URL: `filename.mtl`
usemtl `referenceMap`

MTL

Newmtl `referenceMap`
URL: `TextureName.jpg/png`

Texture Map 1
PNG/JPG

Texture Map 2
PNG/JPG

Texture Map n
PNG/JPG
### Background: OBJ Referencing MTL

<table>
<thead>
<tr>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>2V1</th>
<th>2V2</th>
<th>2V3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1368658</td>
<td>vn</td>
<td>-0.3449577391148</td>
<td>-0.6821315884590</td>
<td>-0.6447485089302</td>
<td></td>
</tr>
<tr>
<td>1368659</td>
<td>vn</td>
<td>-0.2826717197895</td>
<td>0.0076980688609</td>
<td>-0.959185836993</td>
<td></td>
</tr>
<tr>
<td>1368660</td>
<td>vn</td>
<td>0.6301043033600</td>
<td>0.7272020578384</td>
<td>0.272969949245</td>
<td></td>
</tr>
<tr>
<td>1368661</td>
<td>vn</td>
<td>0.5974761843681</td>
<td>0.7891337871552</td>
<td>-0.1424432843924</td>
<td></td>
</tr>
<tr>
<td>1368662</td>
<td>vn</td>
<td>-0.4890609681606</td>
<td>-0.7319647669792</td>
<td>0.4743911325932</td>
<td></td>
</tr>
<tr>
<td>1368663</td>
<td>vn</td>
<td>-0.6451370120049</td>
<td>-0.6867179870605</td>
<td>-0.3349875509739</td>
<td></td>
</tr>
<tr>
<td>1368664</td>
<td>vn</td>
<td>0.9334340691566</td>
<td>-0.1631953120232</td>
<td>0.3194811046124</td>
<td></td>
</tr>
<tr>
<td>1368665</td>
<td>vn</td>
<td>0.0918825566769</td>
<td>-0.7192198826241</td>
<td>0.6886800527573</td>
<td></td>
</tr>
<tr>
<td>1368666</td>
<td>vn</td>
<td>0.0357546926000</td>
<td>0.8034136295319</td>
<td>0.5943468213081</td>
<td></td>
</tr>
<tr>
<td>1368667</td>
<td>vn</td>
<td>0.9314748644829</td>
<td>0.3603391945362</td>
<td>-0.0501018352807</td>
<td></td>
</tr>
<tr>
<td>1368668</td>
<td>vn</td>
<td>-0.6770498156548</td>
<td>0.6993246078491</td>
<td>0.2292350381613</td>
<td></td>
</tr>
<tr>
<td>1368669</td>
<td>vn</td>
<td>0.621904731685</td>
<td>0.7298039793968</td>
<td>0.2839385569096</td>
<td></td>
</tr>
</tbody>
</table>

- mtllib reference.mtl
- g Polygonal_Model_1 Ao
- usetl _image

```c
# Number of triangles: 58932
```

```
/1 1161/1/25034 1172/2/25035 1149/3/25036
/2 1174/4/25042 1175/5/25043 1176/6/25044
/3 1175/5/25045 1174/4/25046 1177/7/25047
/4 1175/5/25144 1177/7/25145 1194/6/25146
/5 1176/6/25147 1184/9/25148 1174/4/25149
/6 1229/10/25300 1194/8/25298 1230/11/25299
/7 1229/10/25300 1230/11/25299 1231/12/25301
/8 1231/12/25351 1230/11/25349 1243/13/25350
/9 1231/12/25351 1243/13/25350 1244/14/25352
/A 1393/15/25702 1175/5/27097 1394/16/299
/B 1150/17/25880 1406/18/25881 1407/19/25882
/C 1149/3/25036 1172/2/25035 1408/20/301
/D 1149/3/25036 1408/20/301 1406/18/25037
/E 1415/21/25283 1408/20/301 1172/2/25035
/F 1419/22/25892 1415/21/25890 1172/2/25891
/G 1229/10/25300 1423/23/303 1424/24/25906
```
Background: MTL Referencing Texture Maps

```
8  newmtl  _image
  Kd 1.0 1.0 1.0
9     illum 1
10    d 1
11   Ns 0.125
12  sharpness 60
13     Ni 1
14  map_Kd_wrapU Clamp
15  map_Kd_wrapV Clamp
16  map_Kd_magFilter Linear
17  map_Kd reference_image.png
18  newmtl  _image 2
19
20
```
Solution Part 1: Multiple Objects

The encapsulation strategy for OBJ will introduce 2 new DICOM IODs:

- **Encapsulated OBJ**
  - Stores the main OBJ byte stream
- **Encapsulated MTL**
  - Stores the MTL byte stream

And make minor extensions to 2 existing DICOM modules to support texture maps:

- **Multi-frame True Color SC Image**
- **General Image**
Solution Part 2: Preserve File Name

- The URI used in the encapsulated document to reference the SOP Instance in this Item:
  - Relative URI Within Encapsulated Document
- Stores the file name under which the object may be referenced in encapsulated objects
  - From earlier examples
    - "matlist.mtl"
    - “ntissue.png”
- It is important to note that when de-encapsulating MTL file, the texture map images must be restored to both their original file name and file format.
Solution Part 3: Link the Objects

• A new attribute is added to the Encapsulated OBJ object
  • Referenced Instance Sequence

• A new attribute is added to the Encapsulated MTL object
  • Referenced Image Sequence

• This is a sequence of UIDs for the:
  • Encapsulated MTL
  • Texture Map Images

• This allows a simple DICOM query to easily retrieve all of the supporting objects for a given Encapsulated OBJ
Use Cases for OBJ Encapsulation
The new IOD/SOP is expected to address these real world use cases:

- Creation
- Review / XR Display
- Update
- Print
Use Case 1: Creation

- Medical reconstruction software queries Image manager system
- **User creates patient-specific 3D model** (reconstruction and modeling)
- Alternatively, user queries a **DICOM Segmented Object**
- Modeler system creates the new type DICOM object containing the 3D model along with color information, populating all required metadata
- **User saves 3D model back to the patient’s record in DICOM format** as either (a) an addition to an existing study or (b) a new study
- The **Modeler system stores the new DICOM object** in the Image Manager system
Use Case 2: Review

- At a later time to Use Case 1, a user indicates desire to **visually review a 3D model**, prior to 3D printing
- The Display system queries the Image Manager for the DICOM objects of new type
- The Display system retrieves the indicated object
- The 2D or XR **Display system extracts the 3D model** from the object and displays it to the user, potentially registered for simultaneous display with source images
Use Case 3: Update

- At a later time to Use Case 1, a user indicates desire to modify a 3D model for a particular patient
- The Modeler system queries the Image Manager for the DICOM objects of new type
- If necessary, the Modeler system retrieves any source images (s1 to sN) required for this modification to occur
- User interacts with the Modeler system to adjust the 3D printable model as desired
Use Case 3: Update (cont’d)

- User saves back to the patient’s record in DICOM format as either (a) an addition to an existing study, or (b) a new study.
- The Modeler system creates the new type DICOM object containing the new version 3D model, populating all required metadata and including a unique identifier reference to the supplanted earlier 3D print model object.
- The Modeler system stores the new DICOM object in the Image Manager system.

PACS/VNA
Use Case 4: Print

- At a later time to Use Case 1, a user indicates desire to print a 3D model for a particular patient.
- The Print Manager system queries the Image Manager for the DICOM objects of new type belonging to the patient.
- The Print Manager system retrieves the indicated 3D print model object.
- The Print Manager access the 3D model information within the object, using this to create non-DICOM print instructions for a specific 3d printer (e.g. *.obj)
**Use Case 4: Print** (cont’d)

- The Print Manager prompts the user for any necessary additional print parameters (e.g. support, bed placement, material parameters, etc.)
- The Print Manager submits the print job to the printer
- Optionally, the Print Manager may save an updated 3d print object back to the Image Manager in order to preserve exact print parameters used (per Use Case 3, steps 7+).
Specific Changes to Standard
ISSUE TO RESOLVE

• Encapsulated models lack uniform inherent component color

ADDRESS VIA

• New Attribute: Recommended Display CIELab Value
  • Specifies color for DICOM component
  • Modeled on similar attributes in the standard

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Tag</th>
<th>Type</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended Display CIELab Value</td>
<td>(0062,000D)</td>
<td>3</td>
<td>Specifies the color recommended to be used for the model when visually representing and selecting material for manufacturing. This would typically be used to visually distinguish between models that are part of the same assembly and/or provide best analog to real world appearance. The CIELab triplet value is specified in PCS-Value units. See section 10.7.1.1. This value should be ignored if individual colors have been specified inside the encapsulated model (when encapsulated format allows this).</td>
</tr>
</tbody>
</table>
**ISSUE TO RESOLVE**

- Encapsulated models lack uniform inherent component opacity

**ADDRESS VIA**

- Extended Attribute: Recommended Presentation Opacity
  - Specifies opacity for DICOM component
  - A fraction between 0.0 and 1.0

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Tag</th>
<th>Type</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended Presentation Opacity</td>
<td>(0066,000C)</td>
<td>3</td>
<td>Specifies the opacity in which it is recommended to be used for the model when visually representing and selecting material for manufacturing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A value of zero is interpreted as complete transparency, while a value of 1 is interpreted as fully opaque.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A non-opaque value would typically be specified when either (a) another model grouped in the same assembly needs to be visible behind or inside this model,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or (b) the model represents anatomy that is not fully opaque.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If not present, then it is assumed the model should be presented and manufactured as opaque.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>See Section C.27.1.1.3.</td>
</tr>
</tbody>
</table>
Assembly Group Information

ISSUE TO RESOLVE

• Encapsulated models that are part of same assembly have no inherent grouping
  • Relying on humans to guess grouping of numerous DICOM encapsulated models is problematic

ADRESSED VIA

• Optional *Model Group* UID
  • Explicitly allows model grouping if part of same assembly

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Tag</th>
<th>Type</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Group</td>
<td>(aaa1,bbb1)</td>
<td>3</td>
<td>UID shared by manufacturing models that are considered distinct parts within the same assembly.</td>
</tr>
</tbody>
</table>
Payload - OBJ

- Builds on approach used for encapsulation of STL
  - Encode OBJ and MTL files via Encapsulated Document (0042,0011) attribute
  - Store texture map Images as Multi-frame True Color Secondary Capture
  - Registration of 2 new MIME types model/obj and model/mtl to be completed with IANA
**ISSUE TO RESOLVE**

- How does someone locate & retrieve all of the DICOM objects that an OBJ model directly and indirection references?

**ADDRESS VIA**

- New Attribute: *Referenced Instanced Sequence*
  - Designation of UIDs of
  - OBJ -> MTL

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Tag</th>
<th>Type</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referenced Instance Sequence</td>
<td>(0008,114A)</td>
<td>3</td>
<td>Sequence of UIDs corresponding to supporting instances directly referenced within the encapsulated model.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In an Encapsulated OBJ, only a single item shall be permitted in this sequence and that item shall be the UID of a Encapsulated MTL instance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>In an Encapsulated MTL, all items shall be images and have the Reference Name attribute defined.</td>
</tr>
</tbody>
</table>

Addition to C.35.1 Manufacturing 3D Model Module
DICOM Retrieval of Linked Instances

**ISSUE TO RESOLVE**
- How does someone locate & retrieve all of the DICOM objects that an OBJ model directly and indirection references?

**ADDRESS VIA**
- **New Attribute: Referenced Image Sequence**
  - Designation of UIDs of
  - MTL -> Texture Maps

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Tag</th>
<th>Type</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referenced Image Sequence</td>
<td>(0008,1140)</td>
<td>3</td>
<td>The set of image instances referenced in the encapsulated document. One or more Items are permitted in this Sequence.</td>
</tr>
</tbody>
</table>
File Name References

ISSUE TO RESOLVE

• OBJ files refer to MTL files by file name
• MTL files refer to texture map images by file name
• This naming must be preserved when recreating the files or linkage broken

ADDRESS VIA

• New Attribute: Reference Name
  • Allows disambiguation and recreation of files when de-encapsulating
  • Stored in the Encapsulated MTL and the Texture Map Image objects

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Tag</th>
<th>Type</th>
<th>Attribute Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference Name</td>
<td>(aaa1,bbb3)</td>
<td>1C</td>
<td>The name that will be used when the data is re-constituted as an unencapsulated file. This preserves referential integrity. This attribute is required if the object is referenced from an encapsulated object.</td>
</tr>
</tbody>
</table>

Addition to C.35.1 Manufacturing 3D Model Module

Addition to General Image Module (for Texture Map Images)
Ex. OBJ, MTL, and TM Encapsulation

- **OBJ**
  - URL: matlist.mtl
  - usemtl referenceMap

- **MTL**
  - Newmtl referenceMap
  - URLs: ntissue.png, ...

- **Texture Map 1**
  - ntissue.png
  - PNG
  - Pixels

- **Texture Map 2**
  - distissue.jpg
  - JPEG
  - Pixels

- **Texture Map n**
  - fluid.png
  - PNG
  - Pixels

- **DICOM Objects**
  - Encapsulated OBJ
    - Referenced Instance Sequence = MTL Object
  - Encapsulated MTL
    - Referenced Image Sequence = Texture Image Objects
    - URI = "matlist.mtl"
  - Image Object 1
    - Modality = TEXTUREMAP
    - Reference Name = file:///fluid.png
    - Pixels
  - Image Object 2
    - Modality = TEXTUREMAP
    - Reference Name = file:///fluid.png
    - Pixels
  - Image Object N
    - Modality = TEXTUREMAP
    - URI = "fluid.png"
    - Pixels

- **Original Files**

DICOM Objects
Ex. Model Group

Original Files

OBJ
CardiacAnatomy.obj

OBJ
ThoracicSkeleton.obj

STL
Thyroid.stl

DICOM Objects

Encapsulated OBJ
Model Group = 2.699.8235.5951.35894.153

OBJ

Encapsulated OBJ
Model Group = 2.699.8235.5951.35894.153

OBJ

Encapsulated STL
Model Group = 2.699.8235.5951.35894.153

STL
Ex. Model Group, Color, and Opacity

Original Files

- OBJ
  - AorticCalcifications.obj

- STL
  - Aorta.stl

DICOM Objects

- Encapsulated OBJ
  - Model Group = 52.19.1235.5951.7794.4
  - Recommended Display CIELab Value = 100:0:0 (white)
  - Recommended Presentation Opacity = 1.0 (Fully Opaque)

- Encapsulated STL
  - Model Group = 52.19.1235.5951.7794.4
  - Recommended Display CIELab Value = 53:80:67 (red)
  - Recommended Presentation Opacity = 0.5 (50% Opaque)
Thank You for your Attention

END