vtkMRMLSegmentationNode: Use cases to define structure and behavior

Contents

[Outline 1](#_Toc394318457)

[Terminology 1](#_Toc394318458)

[Use Cases 2](#_Toc394318459)

[SlicerRt requirements, Queen’s University, Canada 2](#_Toc394318460)

[Steve Pieper, Isonomics, United States 2](#_Toc394318461)

[Andrey Federov, Brigham and Womens Hospital, United States 2](#_Toc394318462)

[Proposed Structure 2](#_Toc394318463)

[Proposed Behavior 3](#_Toc394318464)

[Discussion 3](#_Toc394318465)

# Outline

This document is to be used to collect use cases that will help to shape the behavior of the segmentation node. This node is aimed to support the Segmentation Information Object Definition (IOD) in section C.8.20 (please see <http://dicom.nema.org/medical/dicom/2014a/output/pdf/part03.pdf>) as well as the Surface Segmentation IOD in section C.8.23.

The node’s overall aim is to provide a structure that can support the C.8.20 and C.8.23 definitions but also support DICOM’s RT Structure Set (see terminology) contour encoding as well. In the first phase of implementation, read-only support is proposed due to complications with maintaining loss-less encoding of contours due to edits and conversions during SlicerRt runtime operations.

Many of the decisions taken in the design of the proposed node are directly affected by the discussions held in Boston, MA during the 2014 NAMIC Summer Project Week (see <http://www.na-mic.org/Wiki/index.php/2014_Project_Week_Breakout_Session:_Contours>).

Active development and discussion is available at: <https://www.assembla.com/spaces/slicerrt/tickets/626-enable-support-of-dicom-segmentation-objects#/activity/ticket>:

# Terminology

* vtkMRMLSegmentationNode: The mrml node that is the container for one or many vtkSegments, it also contains all meta data about the segmentation
* vtkSegment: A class containing a pair of image data and mesh data as well as the meta data for a segment
* vtkMRMLSegmentationDisplayNode: A mrml display node used to render a segmentation node
* RTSS: RT Structure set, a DICOM standard for representing contours as contour points
  + See <https://www.dabsoft.ch/dicom/3/A.19/>

# Use Cases

Presented here are some of the use cases collected to help shape this document.

## SlicerRt requirements, Queen’s University, Canada

The segmentation node must accomplish all features currently supported by the vtkMRMLContourNode. Some requirements:

* Hide and show individual image data and mesh data representing a feature
* Edit the image data of a segmentation, possibly edit mesh data in the future
* Perform morphological operations on the image data of a segmentation
* Compare segmentations (currently image data only, possibly extend to mesh data)
* Use as inputs to dose volume histogram calculations (representing the volume definition portion)
* Convert from mesh data to image data and vice versa
* Import DICOM RTSS objects and represent them as segmentations
* Export segmentations as DICOM RTSS objects

## Steve Pieper, Isonomics, United States

Section to be completed.

## Andrey Federov, Brigham and Womens Hospital, United States

Section to be completed.

# Proposed Structure

The general structure is as follows:

* A segmentation node contains one or many vtkSegment objects (referred to as segments)
* A segmentation node contains meta-data relating to the standard
  + An identifier to identify this segmentation
* A segment object contains a pair of image data and mesh data objects
  + Only one of the pair is considered the “source” representation as it is loaded directly from DICOM
  + The other is converted via an algorithmic process
* A segment contains meta-data relating to the standard
  + A UID to uniquely identify the segment
  + A feature UID to uniquely identify the feature that the segment labels (some features might include: liver, lung, heart, left ventricle, etc..)
  + A segment algorithm type to identify the state of the segment. Values:
    - AUTOMATIC
    - SEMIAUTOMATIC
    - MANUAL

# Proposed Behavior

For each proposed behavior, the responsibilities of the user are identified. It is assumed that the developer is responsible in each instance to provide an interface that supports these responsibilities.

Some of the proposed behavior:

* A segmentation node can be split and many segmentation nodes can be merged
  + The split action takes a single segmentation node, removes the segments identified by the user to be split, and places the removed segments in a new segmentation node
    - The user is responsible for identifying the features
  + The merge action takes the segments contained in the identified segmentation nodes and places them into a new segmentation node
    - The user is responsible for identifying the feature that the merged segmentation identifies (prostate, liver, etc…)
* A segmentation node can be edited
  + If the source representation of a segmentation is a mesh, all segments in the segmentation will be converted to image data
  + The editor module will be reused to enable editing of image data based segmentations
    - See discussion question 1
* A segmentation can be rendered
  + Each segmentation will have its own display node
    - See <https://www.assembla.com/spaces/slicerrt/tickets/628-create-a-display-node-and-displayable-manager-for-the-upcoming-segmentation-node#/activity/ticket>:
  + A display manager will be created that is aware of the internal structure of the display node to enable rendering of segmentations in the 3d viewer and slice viewer
    - See <https://www.assembla.com/spaces/slicerrt/tickets/628-create-a-display-node-and-displayable-manager-for-the-upcoming-segmentation-node#/activity/ticket>:
  + A segmentation can be hidden and shown
    - In order to support extremely large numbers of segments without crippling performance due to high numbers of MRML nodes, we will only enable the hiding and showing of entire segmentation nodes, and not individual segments within
      * This is accomplished by providing a display node at the segmentation level only

# Discussion

Outstanding questions/issues:

1. The editor can only operate on one image data at a time, if a segmentation contains N segments, each with their own image data, how can we enable editing?