3MF Boolean Operations Extension

Specification & Reference Guide

Version	**0.6.1**
Status	Draft

Table of Contents

Preface

About this Specification

This 3MF Boolean Operations Extension is an extension to the core 3MF specification. This document cannot stand alone and only applies as an addendum to the core 3MF specification. Usage of this and any other 3MF extensions follow an a la carte model, defined in the core 3MF specification.

Part I, “3MF Documents,” presents the details of the primarily XML-based 3MF Document format. This section describes the XML markup that defines the composition of 3D documents and the appearance of each model within the document.

Part II, “Appendices,” contains additional technical details and schemas too extensive to include in the main body of the text as well as convenient reference information.

The information contained in this specification is subject to change. Every effort has been made to ensure its accuracy at the time of publication.

This extension MUST be used only with Core specification 1.x.

Document Conventions

See the 3MF Core Specification conventions.

In this extension specification, as an example, the prefix "o" maps to the xml-namespace "http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/02". See Appendix C. Standard Namespace.

Language Notes

See the 3MF Core Specification language notes.

Software Conformance

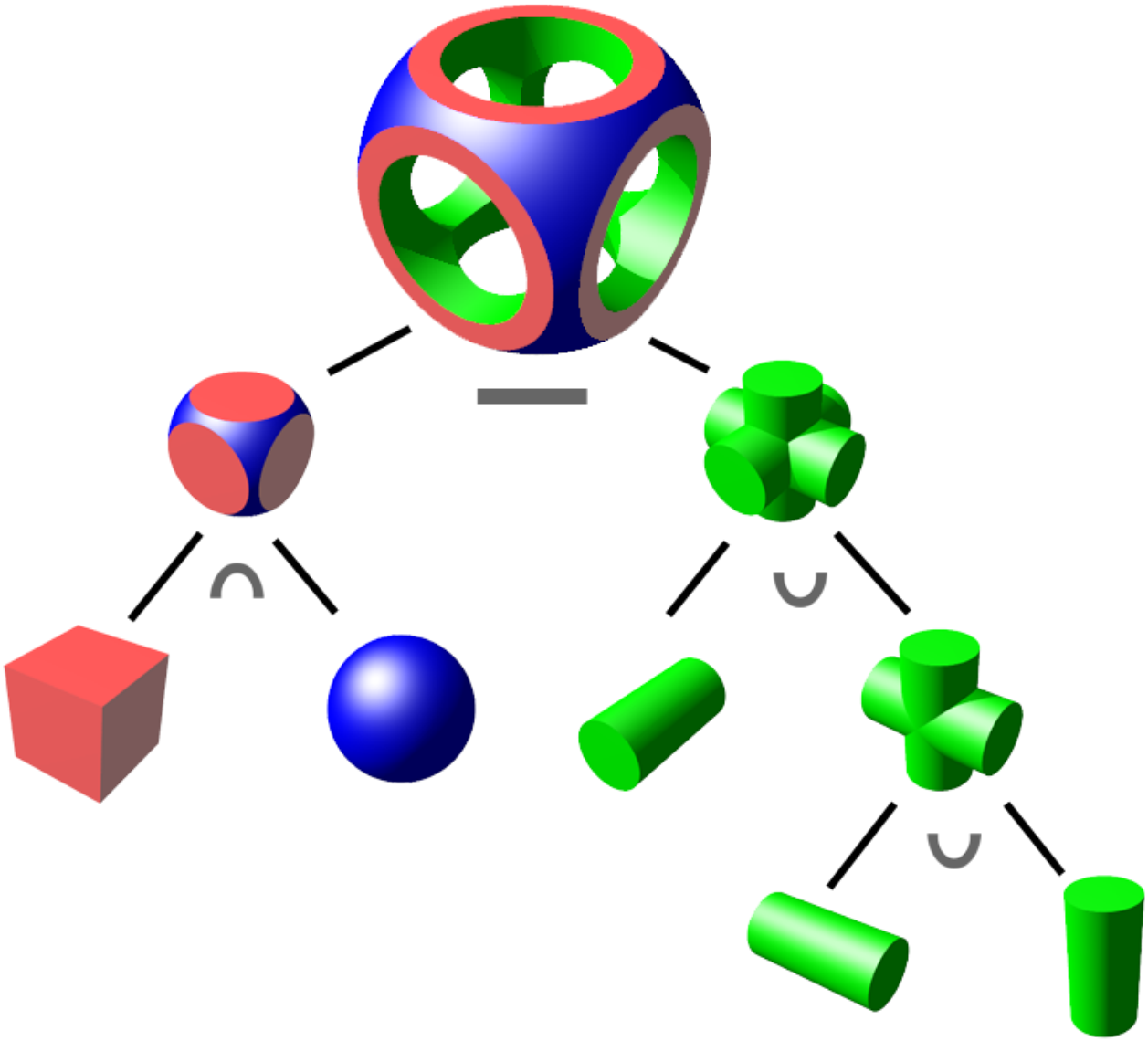
See the 3MF Core Specification software conformance.

Part I: 3MF Documents

Chapter 1. Overview of Additions

The 3MF Core Specification defines the element in the resource as definition of a logical association of different objects to form an assembly, with the intent to allow reuse of model definitions for an efficient encoding. The resultant shape of a element is the aggregation (union) of each object element.

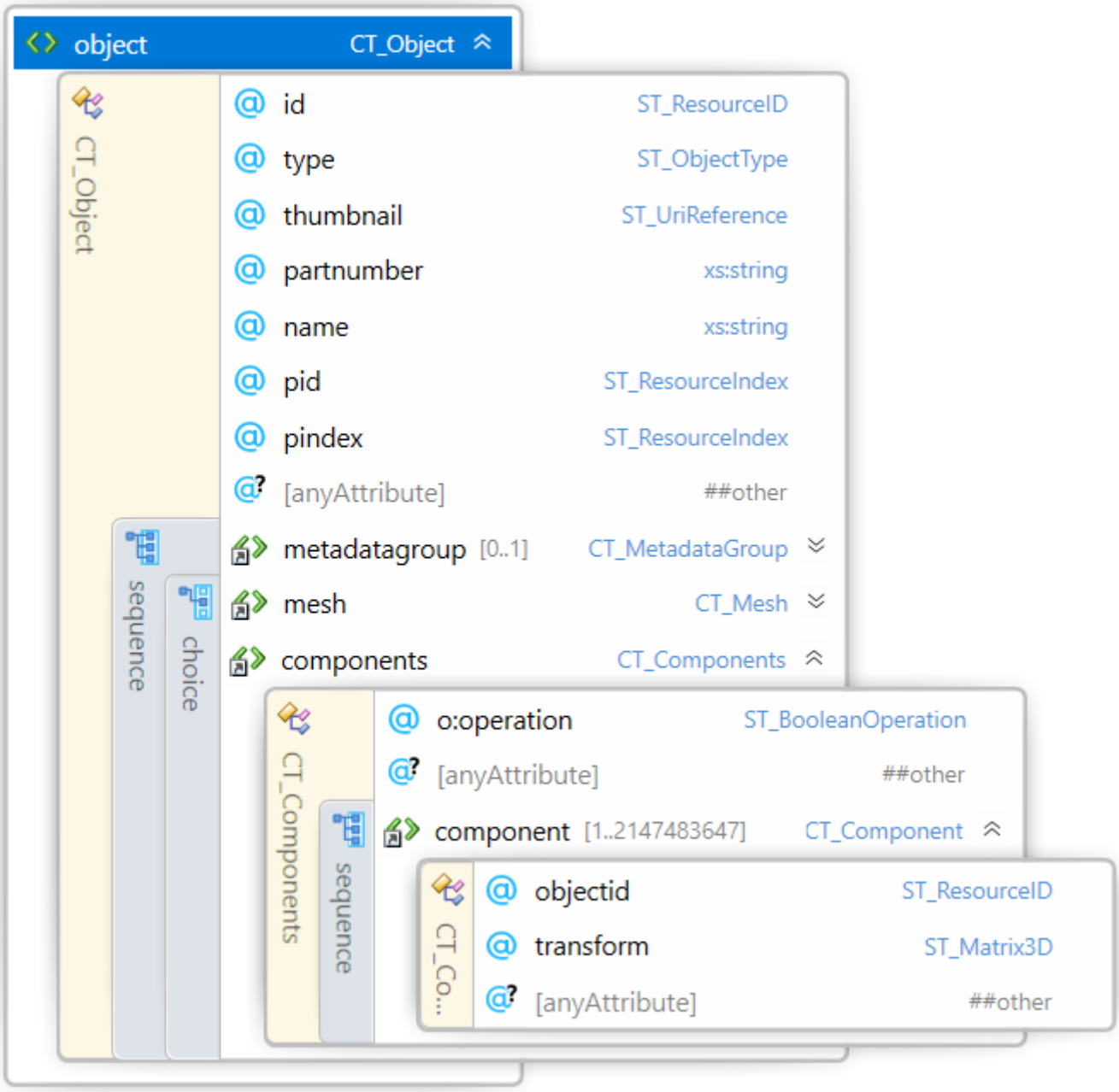
This extension is based in a simplified Constructive Solid Geometry (CSG) which defines a binary tree of operations, and it extends this concept to non-binary tree. To reduce complexity, the scope of the subtracting boolean operations (difference and intersect) is limited.



This document extends the <components> element by defining a boolean operation performed on the components.

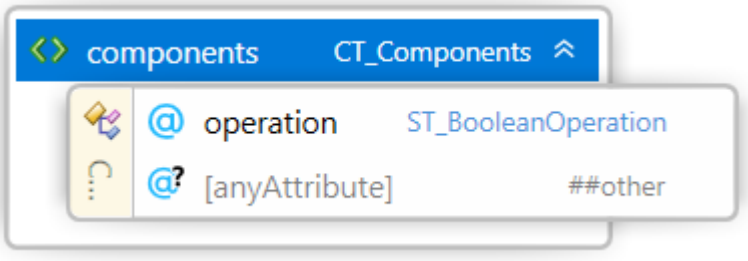
To avoid data loss while parsing, a 3MF package which uses referenced objects MUST enlist the 3MF Boolean Operations Extension as “required extension”, as defined in the core specification. However if the 3MF Boolean Operations Extension is not enlisted a required, any consumer non-supporting the 3MF Boolean Operations Extension may be able to process the rest of the document.

Figure 1-1: Overview of 3MF Boolean Operations Extension XML structure



Chapter 2. Components

Element <components>



The <components> element in the the 3MF Core Specification is enhanced by new attributes to define the compenents association.

Name	Type	Use	Default	Annotation

Name	Type	Use	Default	Annotation
operation	**ST_BooleanOperation**		union	Boolean operation: union, difference or intersection operations.
@anyAttribute				

The <components> element is enhanced the the following attributes:

operation - The boolean operation to perform.

It starts by performing the boolean operation to the first components with the second component, if available. If more components are available, it iteratively performs the boolean operation from previous result and next component.

The options for the operations are the following:

1. *union*. The new object shape is defined as the merger of the shapes. The new object surface property is defined by the property of the surface property defining the outer surface. If material and the volumetric property, if available, in the overlapped volume is defined by the added object.

   union(a,b,c,d) = ((a U b) U c) U d

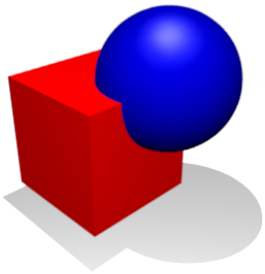
2. *difference*. The new object shape is defined by the shape in the first object shape that is not in any other object shape. The new object surface property, where overlaps, is defined by the object surface property of the substracting object(s).

   difference(a,b,c,d) = ((a - b) - c) - d = a - union(b,c,d)

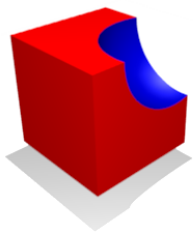
3. *intersection*. The new object shape is defined as the common (clipping) shape in all objects. The new object surface property is defined as the object surface property of the object clipping that surface.

   intersection(a,b,c,d) = ((a ∩ b) ∩ c) ∩ d

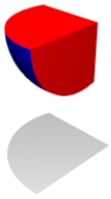
When specifying a *difference* or a *intersection*, those subtracting or intersecting objects MUST only contain an object or a tree of components with shapes defined in the 3MF Core Specification and no other extension, and in turn MUST NOT contain any other *difference* or a *intersection* operation.



union: merger of objects into one



difference: subtraction of object from another one



intersection: portion common to objects

The boolean operations follow the fill rule conversion defined by See the 3MF Core Specification fill rule.

Part II. Appendices

Appendix A. Glossary

See the 3MF Core Specification glossary.

Appendix B. 3MF XSD Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns="http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/02"
xmlns:xs="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://schemas.microsoft.com/3dmanufacturing/booleanoperations/20
21/02"
  elementFormDefault="unqualified" attributeFormDefault="unqualified"
blockDefault="#all">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:annotation>
    <xs:documentation><![CDATA[   Schema notes:

  Items within this schema follow a simple naming convention of appending a prefix
indicating the type of element for references:

  Unprefixed: Element names
  CT_: Complex types
  ST_: Simple types

  ]]></xs:documentation>
  </xs:annotation>
  <!-- Complex Types -->
  <xs:complexType name="CT_Components">
    <xs:attribute name="operation" type="ST_BooleanOperation" default="union"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <xs:simpleType name="ST_BooleanOperation">
    <xs:restriction base="xs:string">
      <xs:enumeration value="union"/>
      <xs:enumeration value="difference"/>
      <xs:enumeration value="intersection"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- Elements -->
  <xs:element name="components" type="CT_Components"/>
</xs:schema>
```

Appendix C. Standard Namespace

BooleanOperation	http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/02

Appendix D: Example file

3D model

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<model xmlns="http://schemas.microsoft.com/3dmanufacturing/core/2015/02"
    xmlns:o="http://schemas.microsoft.com/3dmanufacturing/booleanoperations/2021/0
2"
    requiredextensions="o" unit="millimeter" xml:lang="en-US">
    <resources>
        <basematerials id="2">
          <base name="Red" displaycolor="#FF0000" />
          <base name="Green" displaycolor="#00FF00" />
          <base name="Blue" displaycolor="#0000FF" />
        <basematerials>
        <object id="3" type="model" name="Cylinder" pid="2" pindex="1">
            <mesh>
                <vertices>...</vertices>
                <triangles>...</triangles>
            </mesh>
        </object>
        <object id="4" type="model" name="Cube" pid="2" pindex="0">
            <mesh>
                <vertices>...</vertices>
                <triangles>...</triangles>
            </mesh>
        </object>
        <object id="5" type="model" name="Sphere" pid="2" pindex="2">
            <mesh>
                <vertices>...</vertices>
                <triangles>...</triangles>
           </mesh>
        </object>
        <object id="6" type="model" name="Union">
            <components o:operation="union">
                <component objectid="3" transform="0.0271726 0 0 0 0 0.0271726 0
-0.0680034 0 4.15442 3.58836 5.23705" />
                <component objectid="3" transform="0.0272014 0 0 0 0.0272012 0 0 0
0.0680035 4.05357 6.33412 3.71548" />
                <component objectid="3" transform="0 0 -0.0272013 0 0.0272013 0
0.0680032 0 0 5.05103 6.32914 3.35287" />
            </components>
        </object>
        <object id="7" type="model" name="Insersection">
            <components o:operation="intersection">
```

```
                <component objectid="4" transform="0.0741111 0 0 0 0.0741111 0 0 0
0.0741111 2.91124 -0.400453 1.60607" />
                <component objectid="5" transform="0.0921218 0 0 0 0.0893995 0 0 0
0.0873415 2.52016 2.37774 2.21481" />
            </components>
        </object>
        <object id="11" type="model" name="Difference">
           <components o:operation="difference">
              <component objectid="7" />
              <component objectid="6"/>
            </components>
        </object>
    </resources>
    <build>
        <item objectid="11" transform="25.4 0 0 0 25.4 0 0 0 25.4 0 0 0" />
    </build>
</model>
```

References

CSG

From Wikipedia, the free encyclopedia. "Constructive solid geometry". https://en.wikipedia.org/wiki/Constructive_solid_geometry

Cornelia Haslinger, Universität Salzburg. "Constructive Solid Geometry in Education". https://www.uni-salzburg.at/fileadmin/multimedia/Mathematik/images/EMMA/Workshop_Turkey/education_days/CSG_Haslinger_low_quality.pdf.

See the 3MF Core Specification references for additional references.