FreeCAD Articles
Topology and Geometry

Carlo Dormeletti

Version: 1.1

FreeCAD reference version: 0.20.1

Contents

1 Geometry and Topology in FreeCAD

1.1 Introduction

In **FreeCAD** it is very important to acquire some correct terminology when speaking about things.

The distinction between Topology and Geometry is one of the earliest principles of CAD modeling.

But this could create some mess as Similar concepts are defined in different ways, as the "point of view is different" so the object is different.

A formal distinction would be:

- Geometry defines the mathematics of shape.
- Topology describes the relationship between the pieces of geometry.

1.1.1 Geometry

Geometry, could be thought of as the most "low level" component of the modeling engine.

It permits to define simple entities.

- **Points** is a coordinate in the 2D or 3D space, it has no dimensions.
- **Curves** the best known example is a a **circle**, it has a precise mathematical representation, that permits supplying only two values: a **center** and a **radius**, the points that define the curve can be calculated using these values and a math formula. Same thing if we want to define a **line** or better a **segment**, it suffices to define two points. One error easily done is however to think of curves only in a 2D way, as we are used to do as example in school geometry, on which surves are defined on XY plane only. **FreeCAD** is a 3d modeler, so there are even curves defined in 3d space, in other word that in his geometric definition take in account a free 3d position in space not lying on a plane.
- **Surfaces** for example a **plane**, but also a more complex **surfaces** like a BSpline **surface**.

1.1.2 Topology

Topology is the way relationship between elements that constitutes a **solids** are described.

A solid could be described using only 3 "base components":

- **Vertex**: A topological element corresponding to a **point**. It has zero dimensions.

- **Edge**: A topological element corresponding to a limited curve. An **edge** is generally limited by **vertices**.
- **Face**: In 2D it could be intended as a part of a plane; in 3D it could be intended to a part of a **surface**. Its geometry is constrained (trimmed) by **edges**.

These **TopoShapes** can be grouped to make more complex things:

- **Wire**: a series of **edges** connected by their **vertices**. It can be an open or closed contour depending on whether the **edges** are linked or not.
- **Shell**: A set of **faces** connected by their **edges**. A shell can be open or closed.
- **Solid**: A part of space limited by a **shell**.

1.1.3 FreeCAD modeling paradigm.

Once acquired some terminology, we could analyze how **FreeCAD** use these concepts.

We could establish a sort of hierarchy between **FreeCAD** objects, from the most "low level" ones to the most "complex":

- **Geometric primitives**
- **TopoShape**
- **Document object**

These things are "hidden" when using the GUI, because every object created using the GUI is created in 3D view, and becomes a **Document object**.

In OpenCascade terminology, that is **FreeCAD** "modeling engine", there is a distinction between "geometric primitives" and "TopoShapes".

Geometric primitives are not meant to be visualized, but rather to be used as building elements of **TopoShape**. For example a **edge** could be a segment (a portion of an infinite line) but also a portion of circle (Arc).

This concept could lead to think that **FreeCAD** is complex and involuted, but this is not true, once acquired the necessary concepts.

The main concepts to acquire an remember are:

- An **edge** is almost always a limited portion of a geometric primitive.
- A **face** is a limited portion of a **surface**.
- Usually is possible to retrieve the underlying "unlimited" entity of which the element is a portion.

It can be illustrated using some examples that can be typed even in the Python Console

We will simply use the print function:

```
circle = Part.Circle()

print(circle.TypeId)

circle.Center = FreeCAD.Vector(0, 0, 0)
circle.Radius = 10

cir1 = circle.toShape()

print(cir1.TypeId)

Part.show(cir1, "Circle")
```

We will see in **Report View**:

```
Part::GeomCircle
Part::TopoShape
```

This style of coding, could even be more compact, but this is more meaningful, as every action is clearly shown:

- Create an empty "geometric primitive" `circle`
- Printed its **TypeId** that is **Part::GeomCircle**.
- Assigned proper values at **Center** and **Radius**.
- Transformed the "geometric primitive" in a **TopoShape** using the method **.toShape()**.
- Printed its **TypeId** to show that has became **Part::TopoShape**.
- Create a **Document object** from a **TopoShape** using **.show()**.

A **Document object** is an entity that is visualized in 3D view and is saved in **.FCStd** file.

1.2 Crossing borders

It is always possible to pass from a object to another.

As example to pass from a document object to a TopoShape is simple as using:

```
myToposhape = aDocumentObject.Shape
```

A shape is composed by some elements, and they are accessible this way:

```
myFace = myToposhape.Faces[0]
```

Code above is simply extracting the face at position 0 in the "list of faces" that is contained in `Faces` property of a TopoShape.

You could even extract as examples `Edges` and `Vertexes` using the same technique.

Once acquired a Topology Object you could extract the underlying Geometry elements, using as example:

```
myVector = myVertex.Point
myCurve = myEdge.Curve
mySurface = myFace.Surface
```

The inverse is possible, but you have to in mind the concepts explained.

When you have a TopoShape and want to obtain a DocumentObject you could use:

```
myDocumentObject = Part.show(myToposhape, "DocObjName")
```

Simple and quick at least in **FreeCAD** 0.20 and above.

When you have a Curve you could obtain an Egde using:

```
myEdge = myCurve.toShape(limit1, limit2)
```

Here the concept of "limited" Curve that appear, obviously the limits have to be chosen based on what Curve you are limiting, and it is not possible to explain in an exhaustive way now.

1.3 Conclusion

This is the starting monograph of a would be series of monographies that will try to explain a matter, to make this possible, I need some interaction with users, so your suggestions are welcomed, you could contact me on **FreeCAD** forum using my nickname **onekk**, or using this forum post:

https://forum.freecadweb.org/viewtopic.php?f=36&t=74950

And I need some encouragement, as the time spent in trying to explain things, and making a decent graphic typeset is not negligible.

You could "help development" using the ways explained in this site:

https://github.com/onekk/freecad-doc

Thanks to all for every help and feedback.

Carlo D. onekk