CUSTOMER: HOMAG **PROJECT NAME:** WOODWOP

PROJECT REF.: WOODWOP **DELIVERABLE:** JA

TITLE

TOPOLOGICAL NAMING

TYPE

WHITE PAPER

OCC DOC. NUMBER: OCC-2015-04-27-06-44

DOC. FILE NAME: TopNaming

History

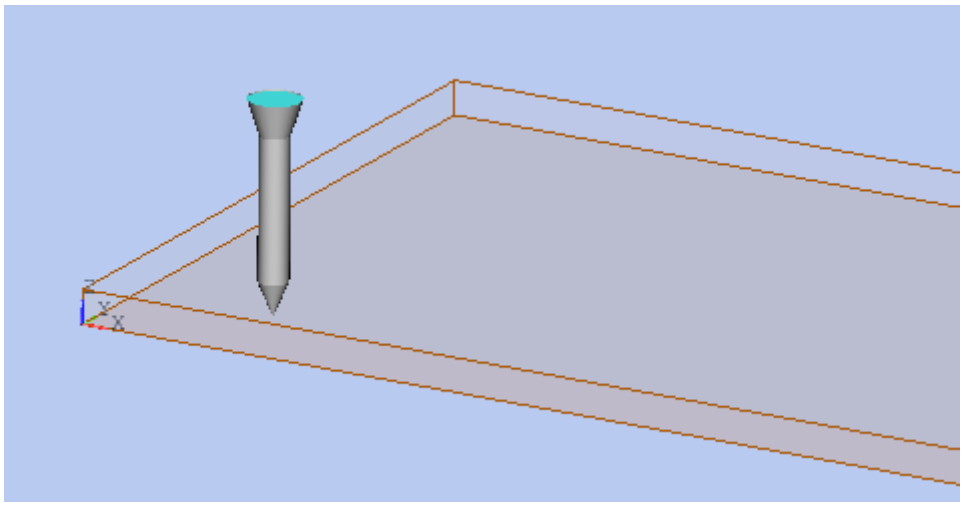
Version	Date	Author	Description	Status
0.1	27.04.2015	Vlad Romashko	From scratch	D

D: Draft, **P**: Proposed for approval, **A**: approved

This article describes basics of *topological naming* of Open CASCADE. We will consider a simple short example, where the topological naming helps the user to make a job. Then, we will describe how it works.

Some introductory words about topological naming – it is a mechanism of Open CASCADE aimed to keep reference to the selected shape. If, for example, we select a vertex of a solid shape and "ask" the topological naming to keep reference to this vertex, it will refer to the vertex whatever happens with the shape (translations, scaling, fusion with another shape, etc.).

Now let us consider an example: imagine a wooden plate. The job is to drive several nails in it:



A nail driven in a wooden plate

There may be several nails with different size and position. A Hammer should push each Nail exactly in the center point of the top surface. For this the user does the following:

1. Makes several Nails of different height and diameter (according to the need),
2. Chooses (selects) the upper surface of each Nail for the Hammer.

The job is done. The application should do the rest – the Hammer calculates a center point for each selected surface of the Nail and "strikes" each Nail driving it into the wooden plate.

What happens if the user changes the position of some Nails? How will the Hammer know about it? It keeps reference to the surface of each Nail. However, if a Nail is relocated, the Hammer should know the new position of the selected surface. Otherwise, it will "strike" at the old position (keep the fingers away!)…

Topological naming mechanism should help the Hammer to obtain the relocated surfaces. The Hammer "asks" the mechanism to "re-solve" the selected shapes (it calls method `TNaming_Selection::Solve()`) and the mechanism "returns" the modified surfaces located at the new position (by means of a call to `TNaming_Selector::NamedShape()`).

The topological naming is represented as a "black box" in the example above. Now it is time to make the box a little more "transparent":
The application contains 3 functions:

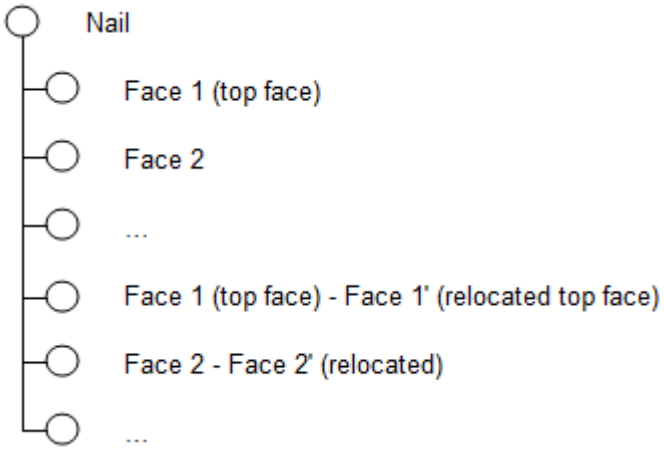
- Nail (it produces a shape representing a nail),
- Translator (it translates a shape along the wooden plate)
- Hammer (it drives the nail in the wooden plate).

Each function gives the topological naming some hints how to "re-solve" the selected sub-shapes:

- The Nail constructs a solid shape and puts each face of the shape into sub-labels:

Distribution of faces through sub-labels of the Nail

- The Translator moves a shape and registers modification for each face: it puts a pair: “old” shape – “new” shape at a sub-label of each moving Nail. The “old” shape represents a face of the Nail at the initial position. The “new” shape – is the same face, but at a new position:



Registration of relocation of faces of a Nail

How does it work?

- The Hammer selects a face of a Nail calling `TNaming_Selector::Select()`. This call makes a unique name for the selected shape. In our example, it will be a direct reference to the label of the top face of the Nail (Face 1 (top face), see the picture above).
- When the user moves a Nail along the wooden plate, the Translator registers this modification by means of putting into sub-labels of the Nail the following pairs: “old” face of the Nail – new face of the Nail (see the picture above).
- When the Hammer calls `TNaming::Solve()`, the topological naming “looks” at the unique name of the selected shape and tries to re-solve it:
  - It finds the 1st appearance of the selected shape in the data tree – it is a label under the Nail function (Face 1 (top face)).
  - It follows the evolution of this face. In our case, there is only one evolution – the translation: Face 1 (top face) – Face 1’ (relocated top face). So, the last evolution is the relocated top face.
- Calling the method `TNaming_Selector::NamedShape()` the Hammer obtains the last evolution of the selected face – the relocated top face.

The job is done.

P.S. Let us say a few words about a little more complicated case – selection of a wire of the top face. Its topological name is an “intersection” of two faces. We remember that the Nail puts only faces under its label. So, the selected wire will represent an “intersection” of the top face and the conic face keeping the “head” of the nail (see the 1st picture). Another example is a selected vertex. Its unique name may be represented as an “intersection” of three or even more faces (depends on the shape).