TGeoCAD: an interface between ROOT and CAD systems

ROOT Users Workshop
Cinzia Luzzi

University of Ferrara - CERN

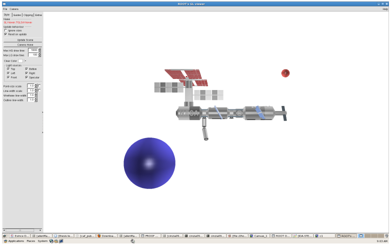
11 March 2013

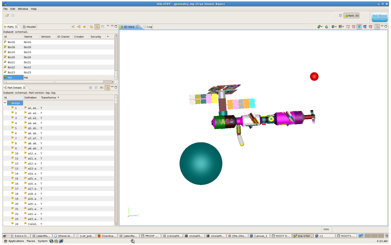
TGeoCAD

Why?

Often, a very high precision in the description of the detector geometry is essential to achieve excellent results of the experiment. Therefore, the necessity to make easier the collaboration between physicists and engineers led to the implementation of an interface between the ROOT simulation software and the CAD systems.

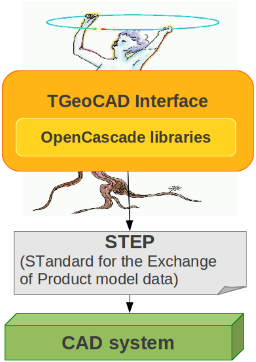
The TGeoCAD interface enables the use of ROOT files in several CAD systems.





TGeoCAD: Components

- ROOT
- Open CASCADE Technology (OCCT):
  - Open source software development platform.
  - C++ components for 3D surface and solid modeling, visualization, data exchange and rapid application development.
- STEP Standard and CAD systems.

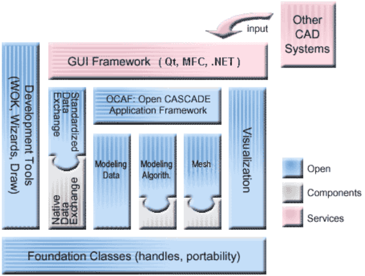


STEP format (ISO 10303)

- ISO Standard for the Exchange of Product model data.
- Represents 3D objects in Computer-aided design (CAD) and related information.
- The Application Protocols (AP) are the top parts of the STEP Standard:
  - Grouped into three main areas:design, manufacturing and life cycle support.
- OCCT creates files according to STEP AP203 / AP214 parts:
  - Part 203:configuration controlled 3D designs of mechanical parts and assemblies.
  - Part 214:core data for automotive mechanical design processes. It is the default format used by OCCT.
- EXPRESS data modeling language used to describe data models.
- Step-File (Part 21): encoding mechanism on how to represent data according to their EXPRESS schema.

OpenCascade Foundation Data Toolkit

- Math utilities (Primitive Geometric Types) provides:
    - Description of elementary geometric shapes as points, vectors, lines, circles and conics, planes and elementary surfaces;
    - Means for positioning geometry in space or on a plane using an axis or a coordinate system;
    - Definition of geometric transformations as translation, rotation and symmetries;

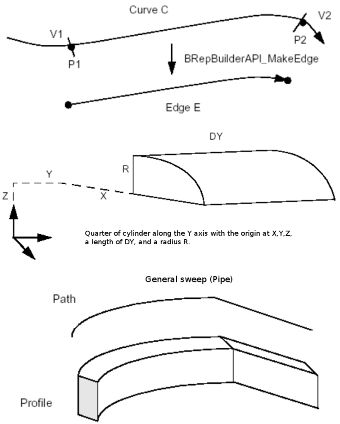


OpenCascade Modeling Data Toolkit

- Data structures to represent 2D and 3D geometric and topological models.
- The topological library allows to build pure topological data structures and to defines relationships between simple geometric entities.
- The abstract topological data structure describes the shape which can be divided into the following component topologies:
  - VERTEX: zero dimensional element (point).
  - EDGE: element created from a curve and vertices (1D).
  - WIRE: set of edges connected by their vertices.
  - FACE: part of a surface bounded by a number of closed wires. It is two dimensional.
  - SHELL: set of faces connected by some of the edges of their wire boundaries.
  - SOLID: part of 3D space limited by closed shells.
  - COMPSOLID: set of solids connected by their faces.
  - COMPOUND: group of any type of topological object.

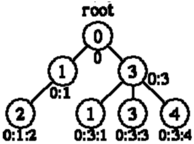
OpenCascade Modeling Algorithm toolkit

- Geometric and topological algorithms used in modeling:
    - allow to create vertices, edges, faces, solids;
    - allow to build primitive objects (boxes, wedges and rotational objects);
    - provides sweeping operations (Prism - linear sweep, Revolution - rotational sweep and Pipe - general sweep) and boolean operations;



Open Cascade Application Framework (OCAF)

- Based on an application/document architecture;
- Provides an infrastructure to attach any data to any topological element;
- Document: container class for an application data;
  - Data structure are reference-key(labels) driven;
  - Data are implemented as attributes (shape, general, relationship etc) attached to reference-keys label;
  - Set of labels organized in a tree structure. Each label has a tag expressed as an integer value.
  - A label is identified by a string build by concatenation of tags from the root of the tree, for example [0:1:2].



Data Exchange and Extended Data Exchange (XDE) module

- The basis of XDE (XCAF) is a framework based on OCAF.
- Shapes are OCAF object and their locations are separated entities.
- XDE allows to work with assemblies and their attributes. An assembly consists of several components (shape, subshape), each of these components refers to a specified shape with different locations.
- A label can be attached to a main part or to a sub-part. To a simple shape or can be a located reference to another one.
- STEPCAFControl Package (Data Exchange module) provides tools to write STEP file from XDE format;
- Allows software based on Open Cascade to exchange data with various CAD software;

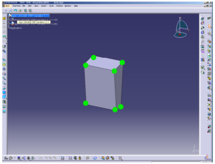
TGeoCAD geomety conversion

- Shapes created step by step starting from points (edge, wire, face, shell and solid) such as:
  - Box;
  - Parallelepiped;
  - Trapezoid;
- Shapes created using OCCT capabilities for solid primitives creation and boolean operations such as:
  - Tube;
  - Cone;
  - Sphere;
- Shapes created by using modeling algorithm (extrusions, revolutions, lofts) applied to basic geometries such as:
  - Hyperboloid;
- TGeoCompositeShape created using OCCT boolean operations between two or more shapes;

TGeoCAD geomety conversion

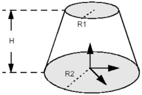
TGeoTrd1

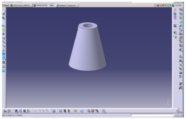
Creates edges from points, wire from edges, faces (planar surfaces) from wires, shells from faces and solid from shells;



TGeoCone

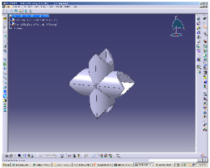
Starting from radius, creates inner and outer cones;
Substract the inner cone from the outer cone.





TGeoCompositeShape

Boolean operations between a box, a tube and a pgon.



TGeoCAD classes

TGeoToStep

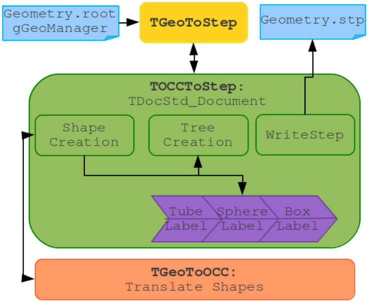
Takes a gGeoManager pointer and gives back a STEP file;

TGeoToOCC

For each ROOT shape creates the corresponding OCCT shape.

TOCCToStep

Reproduces the ROOT tree (mother-children relationship) on the XDE document and writes it to the STEP file;



TGeoToOCC class

Converts ROOT shapes to OCCT shapes.

Each ROOT shape is translated in the correspondent OCC shape using the following methods:

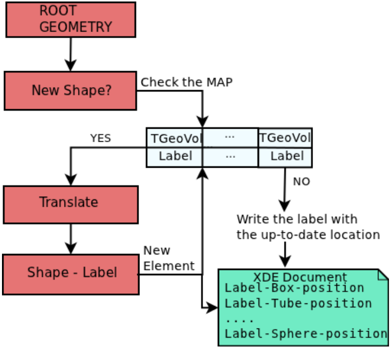
ROOT	TGeoToOCC	ROOT	TGeoToOCC
TGeoBBox	OCC_Box	TGeoTrd2	OCC_Trd
TGeoSphere	OCC_Sphere	TGeoTubeSeg	OCC_Tube
TGeoArb8	OCC_Arb8	TGeoCtub	OCC_Cuttub
TGeoConeSeg	OCC_Cones	TGeoTube	OCC_TubeSeg
TGeoCone	OCC_Cones	TGeoPcon	OCC_Pcon
TGeoPara	OCC_ParaTrap	TGeoTorus	OCC_Torus
TGeoGtra	OCC_ParaTrap	TGeoPgon	OCC_Pgon
TGeoTrd1	OCC_Trd	TGeoEltu	OCC_Eltu
TGeoHype	OCC_Hype	TGeoXtru	OCC_Xtru
TGeoComposite..	OCC_Composite		

TOCCToStep class

TOCCToStep::OCCShapeCreation

Starting from the top of the ROOT geometry tree translates each ROOT shape in the OCCT version.

- For each shape a new label is written in the XDE document. The correspondance shape-label is stored in a map of volumes and labels;
- If the shape-label correspondance is present in the map, add the label to the document updating the location;
- A shape positioned several times in a ROOT volume is translated only once;

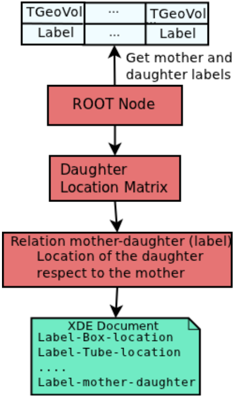


TOCCToStep class

TOCCToStep::OCCTreeCreation

For each node from the end to the top of the ROOT physical tree:

- Gets mother and daughter label reference from the map;
- Takes the daughter location matrix;
- Connects the daughter label to the mother label with its location resulting in a new label;
- Adds the new label to the document wich reproduce the relationship mother-children.



TOCCToStep class

TOCCToStep::OCCWriteStep

Writes the XDE document on the step file using STEPCAFControl_Writer OCC class;

```
3103,
  'distance_accuracy_value','confusion accuracy');
#3107 = SHAPE_DEFINITION_REPRESENTATION(#3108,#2783);
#3108 = PRODUCT_DEFINITION_SHAPE('','',#3109);
#3109 = PRODUCT_DEFINITION('design','',#3110,#3113);
#3110 = PRODUCT_DEFINITION_FORMATION('','',#3111);
#3111 = PRODUCT('bar1','bar1','',(#3112));
#3112 = MECHANICAL_CONTEXT('',#2,'mechanical');
#3113 = PRODUCT_DEFINITION_CONTEXT('part definition',#
2,'design');
#3114 = CONTEXT_DEPENDENT_SHAPE_REPRESENTATION(#3115,#3117);
#3115 = ( REPRESENTATION_RELATIONSHIP('','',#2783,#1551)
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION(#3116)
SHAPE_REPRESENTATION_RELATIONSHIP() );
#3116 = ITEM_DEFINED_TRANSFORMATION('','',#11,#1568);
#3117 = PRODUCT_DEFINITION_SHAPE('Placement','Placement of an
item',
  #3118);
#3118 = NEXT_ASSEMBLY_USAGE_OCCURRENCE('14','=>[0:1:1:5]','',#
1546,#3109
  ,$);
#3119 = PRODUCT_TYPE('part',$,(#3111));
#3120 = CONTEXT_DEPENDENT_SHAPE_REPRESENTATION(#3121,#3123);
#3121 = ( REPRESENTATION_RELATIONSHIP('','',#1551,#81)
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION(#3122)
SHAPE_REPRESENTATION_RELATIONSHIP() );
#3122 = ITEM_DEFINED_TRANSFORMATION('','',#11,#94);
#3123 = PRODUCT_DEFINITION_SHAPE('Placement','Placement of an
```

Relationship definition between shape label 2783 and 1551

TOCCToStep summary

- Create an OCAF application;
- Create an XDE Document;
- Add the OCCT shapes to an assembly entity:
  - Each of its subshapes define a sublabel;
  - Each node of the assembly therefore refers to its subshapes;
  - Identical root shapes present several times in the geometry can be reproduced just copying their references in the step file with up-to-date locations labels;
- Write the XDE document on the step file;

TGeoCAD requirements

- OCCT must be installed
  (export CASROOT=path-to-OCCT);
- ROOT must be compiled using the configuration options:

```
svn co http://root.cern.ch/svn/root/trunk root
./configure --enable-geocad;
--with-occ-incdir:  location of OpenCascade inc files
($CASROOT/inc);
--with-occ-libdir:  location of OpenCascade lib files
($CASROOT/lib);
```

- A ROOT geometry myst be loaded in the memory.

```
root[0] gSystem->Load("libGeoCad.so");
root[1] .x roottest.C
root[2] TGeoToStep *myStep = new TGeoToStep (gGeoManager);
root[3] myStep->CreateGeometry();
```

Thank you for your attention!