

# OpenCASCADE BRep vs. OpenNURBS BRep

[eryar@163.com](mailto:eryar@163.com)

**Abstract.** BRep short for Boundary Representation. First give the definition of the BRep, then compare the BRep mode between OpenCASCADE and OpenNURBS. There are 3 main representation method: use face/edge/vertex to keep track all information in the solid model. The winged-edge data structure uses edges to keep track all information in the solid model. The paper focus on the BRep of OpenCASCADE and OpenNURBS.

**Key Words.** OpenCASCADE, BRep, OpenNURBS, The Winged-edge Structure,

## 1. Introduction

计算机内对形状信息的描述方法简称为造型（Modeling）技术。实体造型的本质是要解决如何在计算机内部表示一个实体模型的问题。以产品实体造型所建立的三维几何模型为基础，可以进一步进行运动力学分析、模拟装配、动画仿真、干涉检查、数控编程以及加工模拟等。造型技术主要由形状表达和形状操作两个部分组成。形状表达的任务是将形状的结构用数据结构来表示。这种描述形状的数据结构称为模型（Model）。形状操作的任务是实现对模型的生成、修改、综合、分析、计算、显示等等操作，以便完成设计过程中的各种造型任务。造型技术是 CAD 的核心技术之一。目前常用的造型技术有：实体造型技术（Solid Modeling）：它将对象分解为一组有限的三维元素的集合，以及可施加在这组集合元素上的一组操作、曲面造型技术（Surface Modeling）：它用数学函数如 B 样条等来描述曲线曲面，并提供曲面的修改，连接，求交和显示等一组操作、非几何形体的造型技术：大多数自然物体如山水，树木等。

工程和制造行业的设计对象主要是三维实体，因此实体模型是目前 CAD 系统的主要造型方法。实体造型系统的结构如下图所示：

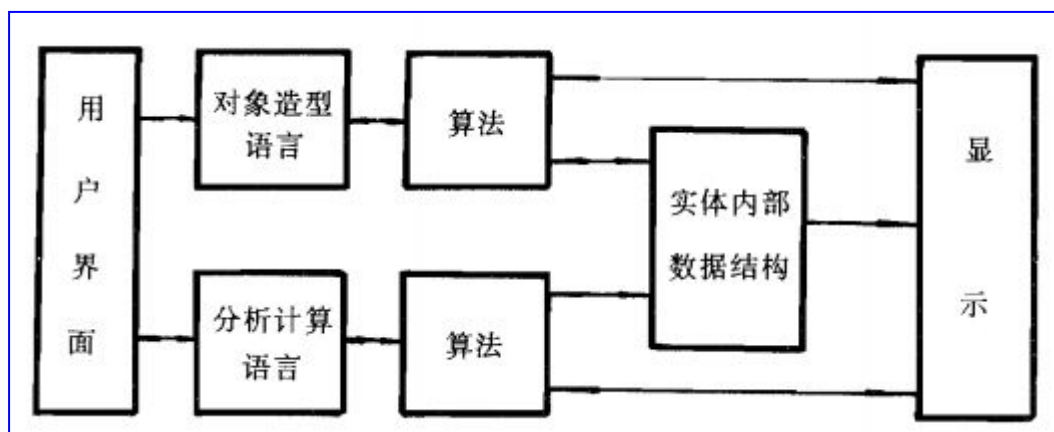


Figure 1.1 Solid Modeling System Structure

实体在该系统中具有双层表达：内部表达和在用户接口中的表达。在内部表达中，实体通常被表达为顶点、边和面的数据结构，既要满足各种操作的要求，又要满足良好的时空效率的要求。在用户接口中，实体按照可理解性和操作的方便，使得设计人员易于掌握和操作。关于实体的操作通常可以分为两类：一类是关于形体的操作，如生成、修改、显示等；另一类

是关于形体的分析计算，如计算实体的体积、面积、检测运动的实体间是否接触等。这些操作各由算法支持，施加于实体内部数据结构之上。算法往往是组成成模块，分层次实现。显示软件通常要求有三维显示、二维投影显示和真实感觉显示等模式。实体造型视所用模型的不同，通常分为以下三种：

❖ 结构模型(Constructive Models)

结构模型又称为构造型实体几何（Constructive Solid Geometry），简称 CSG。CSG 包含一组简单的几何实体类型，如 Box、Sphere、Cylinder、Cone、Torus 等。它们称为基元实体类型（Primitive Solid Type）。CSG 还包含一组施加于基元实体类型上的操作，这些操作包含：几何变换、集合运算及剖割等其他造型操作。CSG 可以通过这些操作将基元实体组成复杂的形状，满足设计师要求。

❖ 分解模型（Decomposition Models）

分解模型的基元实体只有一种类型，即立方体。和二维图像由像素（Pixel）组成一样，分解模型中的形状由空间网格中体素的有无而组合构成。体素通常被组织为八叉树（Octree Representation）的形式，以求得更好的时空效率。

❖ 边界模型（Boundary Models）

边界模型采用体、面、边、点的空间坐标值和拓扑关系来表达实体模型。边界模型可以十分精确地描述实体，又易与图形软件接合，但是数据结构比较复杂。

CSG、分解模型、边界模型都是通用模型，它们适用于一个较大范围的对象且达。各种模型各有优缺点，在实用中往往要对各种模型进行评价，然后选择合适的造型系统，评价一个造型系统常常从以下几个方面来比较：

- ❖ 表达对象的能力（Expressive Power）；
- ❖ 表达的有效性（Validity）；
- ❖ 模型的紧凑性（Conciseness）：即占内存空间的大小；
- ❖ 操作的闭包（Closure of Operations）；
- ❖ 操作的简易性（Ease of Operations）；

## 2. Boundary Representation

边界模型也称为边界表示法 (Boundary Representation)，使用实体的表面来表示实体的形状。不管实体的表面形状是多面体还是自由曲面，边界模型通常都统一为多面体。即用一组三维多边形所围成的形体表示实体。这种表达方法是图形学基元表达的一种推广，因此边界表示法与图形学有很好的衔接关系。组成边界模型的元素是面 Face、边 Edge、顶点 Vertex。

面可以包括平面、二次曲面和参数曲面等，面中可以有孔，但是面上的无孔部分必须是连通的。一个无孔的面只带有一个边界。一个有孔的面带有一个以上的边界。如下图所示：

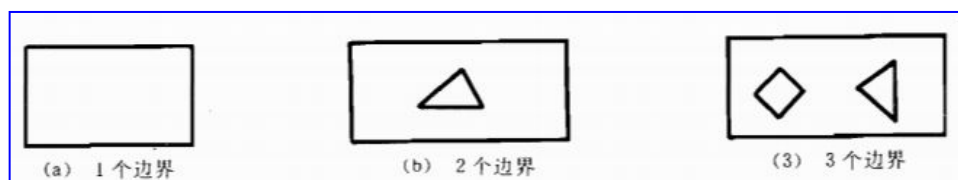


Figure 2.1 Face and its loops

形体由面构成，而面是由边来定义的，边则由点来定义，点通过三个坐标值来定义。如下图所示为边界表示的数据组织方式：

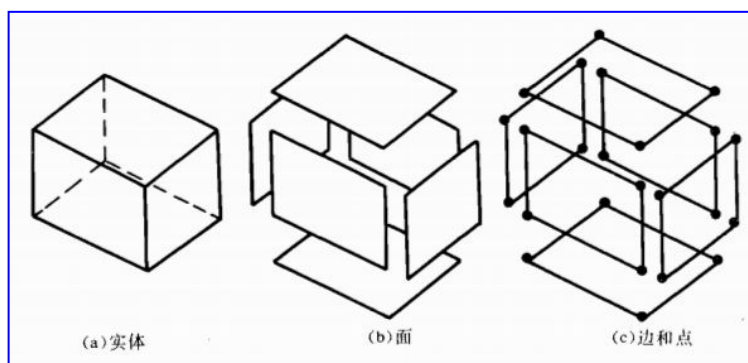


Figure 2.2 Solid, Face, Edge and Vertex of the Brep Model

边界模型的关系型数据结构可以按照三类方法进行组织：以面为基础的方法，以顶点为基础的方法及以边为基础的方法。以面为基础的方法可以通过面来追踪实体的所有信息；以顶点为基础的方法则可以通过点来追踪实体的其他信息；以边为基础的方法则可以通过边来追踪实体的信息。

翼边结构就是以边为基础的结构，通过边来追踪实体的所有信息。在翼边结构中，点、边、面之间的拓扑关系可以通过推理求得。如下图所示：

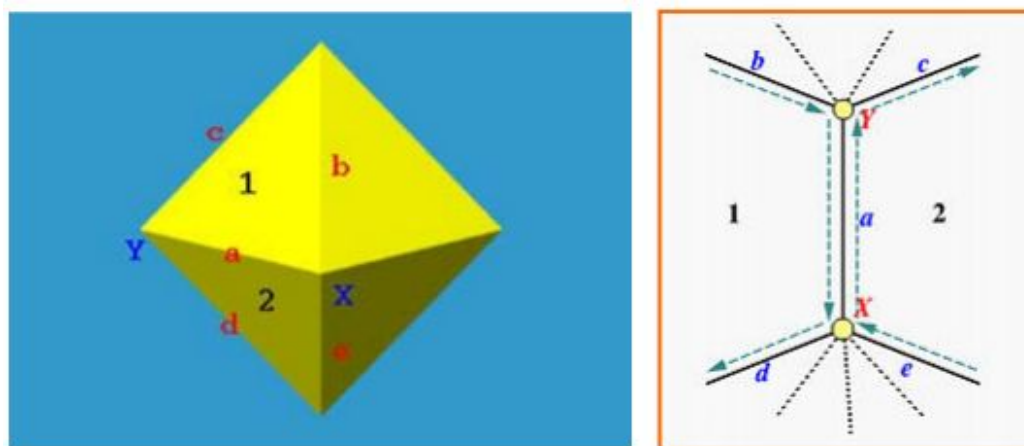


Figure 2.3 The Winged-Edge data structure

边 Edge 是整个数据结构的核心，每个 Edge 代表一条边，含有九个域：

1. Edge 的编号 ID: a;
2. Edge 的起始顶点指针: X;
3. Edge 的终止顶点指针: Y;
4. Edge 左方向相邻的面的指针: 1;
5. Edge 右方向相邻的面的指针: 2;
6. Edge 上方向前邻边指针: b;
7. Edge 上方向后邻边指针: c;
8. Edge 下方向前邻边指针: d;
9. Edge 下方向后邻边指针: e;

Edge	Vertices		Faces		Left Traverse		Right Traverse	
Name	Start	End	Left	Right	Pred	Succ	Pred	Succ
a	X	Y	1	2	b	d	e	c

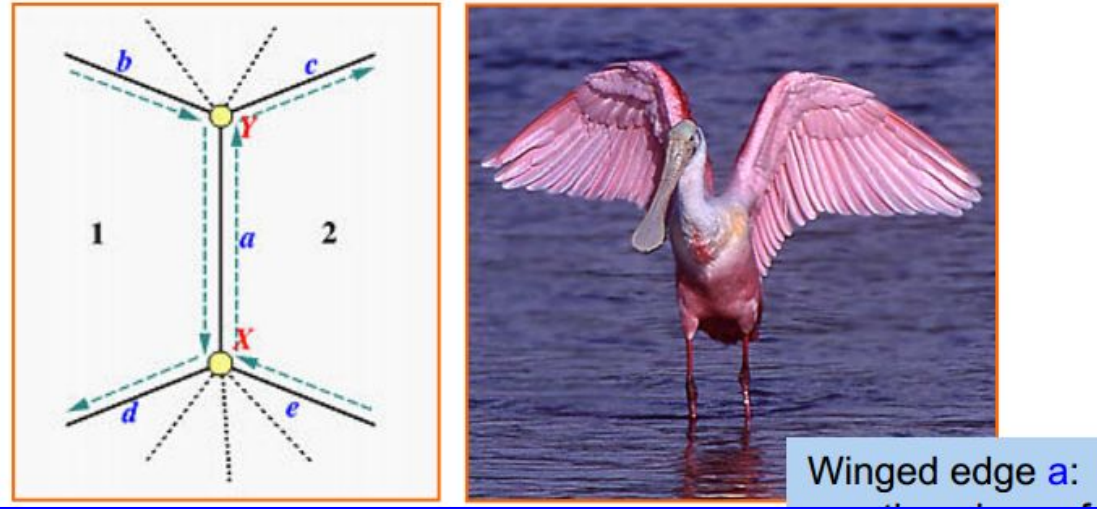


Figure 2.4 The Winged Edge a

由上图可知，以边为基础的结构很像一只鸟张开的双翼，故而得名：The Winged-Edge data Structure。

### 3. OpenCASCADE BRep

根据 OpenCASCADE 中的类的继承关系绘制其 Brep 表示主要的类图如下所示：

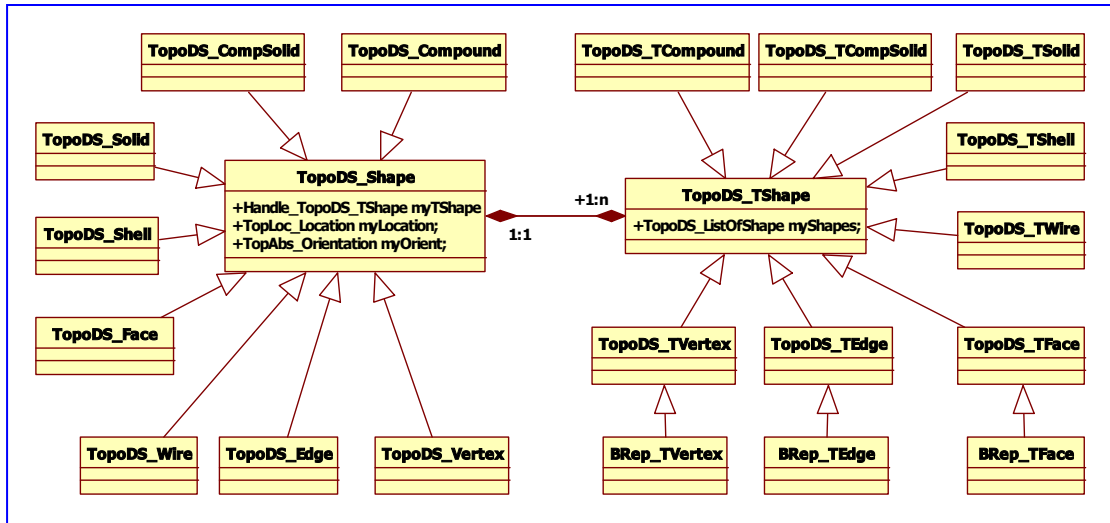


Figure 3.1 OpenCASCADE Brep Class Diagram

由上图可知，**TopoDS\_Shape** 中有一个 **TopoDS\_TShape** 的成员变量，即 **TopoDS\_Shape** 包含 **TopoDS\_TShape**，而一个 **TopoDS\_TShape** 中又包含 N 个 **TopoDS\_Shape**。这种关系感觉有些绕，不过和结构型模式 Composite 模式有几分相似，但不是 Composite 模式，因为 **TopoDS\_Shape** 与 **TopoDS\_TShape** 之间只有包含关系，没有继承关系，如下图所示的 Composite 模式中的 **Composite** 类是派生自并且包含 **Component** 类的：

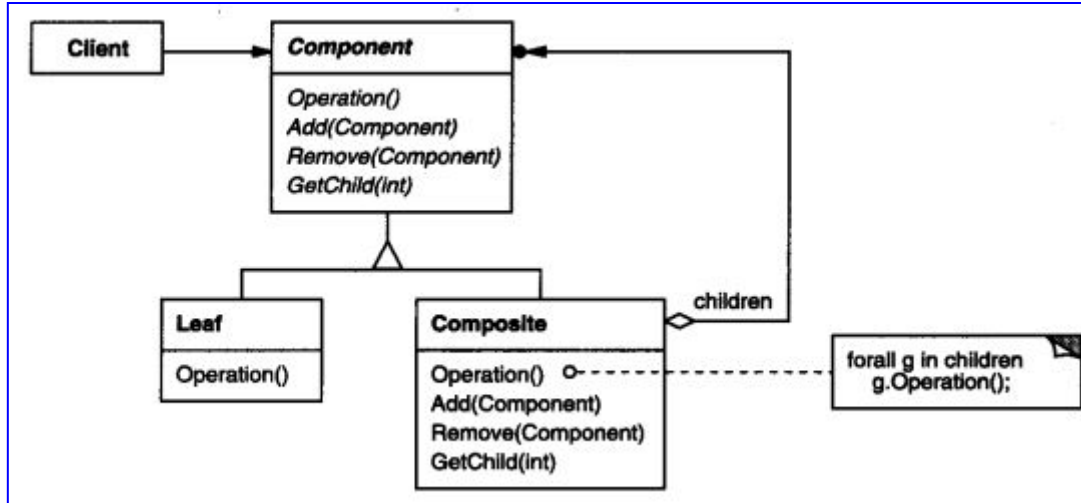


Figure 3.2 Composite Pattern Class Diagram

由图 3.1 可知，根据一个 **TopoDS\_Shape** 来访问其子结构很方便，因为其成员变量 **myTShape** 中就包含了其子结构的一个列表。而访问其父结构是不方便的，如要访问 **TopoDS\_Edge** 的父节点数据就很不方便，需要遍历包含 **Edge** 的整个 **TopoDS\_Shape** 模型。由于这种结构只有包含关系，所以可以方便地找到父结构到子结构的关系，如一个 **Face** 包含哪几个 **Wire** 或 **Edge**，而没有直接从子结构找到父结构的拓朴关系，如与一条 **Edge** 相关的所有面是哪些不能直接确定。

OpenCASCADE 的 Brep 中几何信息都是保存在 **BRep\_TFace**、**BRep\_TEdge** 和 **BRep\_TVertex** 中。画出类图如下所示：

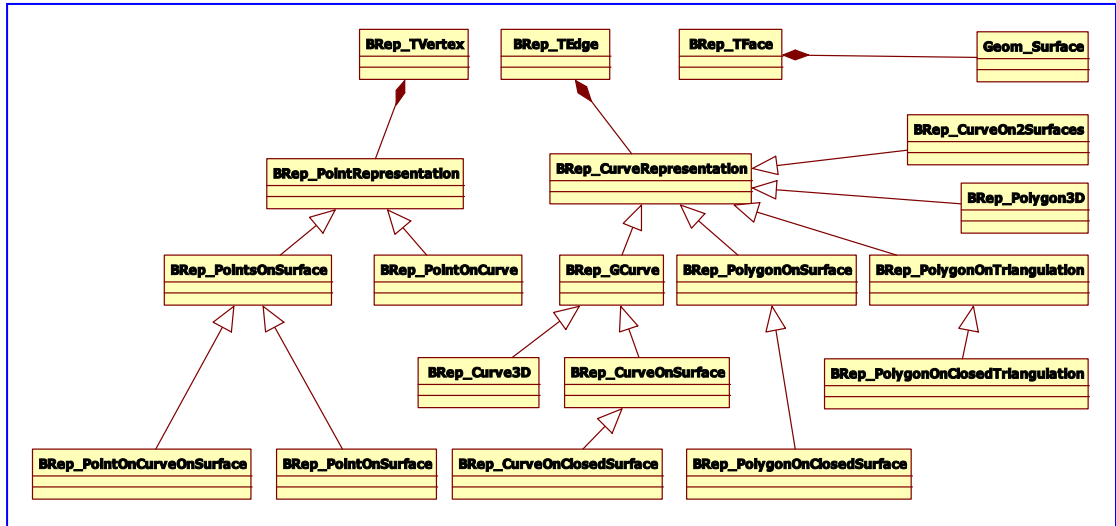


Figure 3.3 Geometry in OpenCASCADE Brep

由上图可知，BRep\_TVertex, BRep\_TEdge, BRep\_TFace 这三个类中都包含了几何信息，其中顶点和边包含了几何信息的一个列表，列表中可以保存多种不同类型的几何信息，每种类型都表示了一种几何之间的关系，如顶点的几何信息中 BRep\_PointOnSurface 就定义了点与面的关系，边中的 BRep\_CurveOnSurface 也定义了边与面的关系。即他们的每一个派生类就描述了一种相互关系。OpenCASCADE 中的顶点的几何数据包含点、线和面的几何之间的一些关系，如面上的点、线上的点、PCurve 上的点等。边的几何数据包含了线与面的关系，如面上的线、闭合面上的线、三角网格上的多段线等。而面中包含的几何数据只有面的参数表示的数据。

如果硬要往翼边结构这种方式上来靠，可以看出 OpenCASCADE 中从顶点的几何数据出发，可以找到与之有关的线和面，所以从这方面来说是以顶点为基础的边界表示方法。而从边的几何数据出发只能找到与之相关的面的信息，而找不到点的信息，所以不是翼边结构。也不是以面为基础的表达方法，因为面中只包含面的几何信息，没有与其他几何数据的关系。但是 OpenCASCADE 并不推荐直接访问 TopoDS\_TShape 的数据。由类 TopoDS\_TShape 的说明：Users have no direct access to the classes derived from TopoDS\_TShape。即这种关系也不能直接用来访问。



## 4. OpenNURBS BRep

OpenNURBS 主要用于 NURBS 相关的计算，代码清晰简明，其中也有关于 Brep 的数据结构，结合其代码，将其类图列出如下所示：

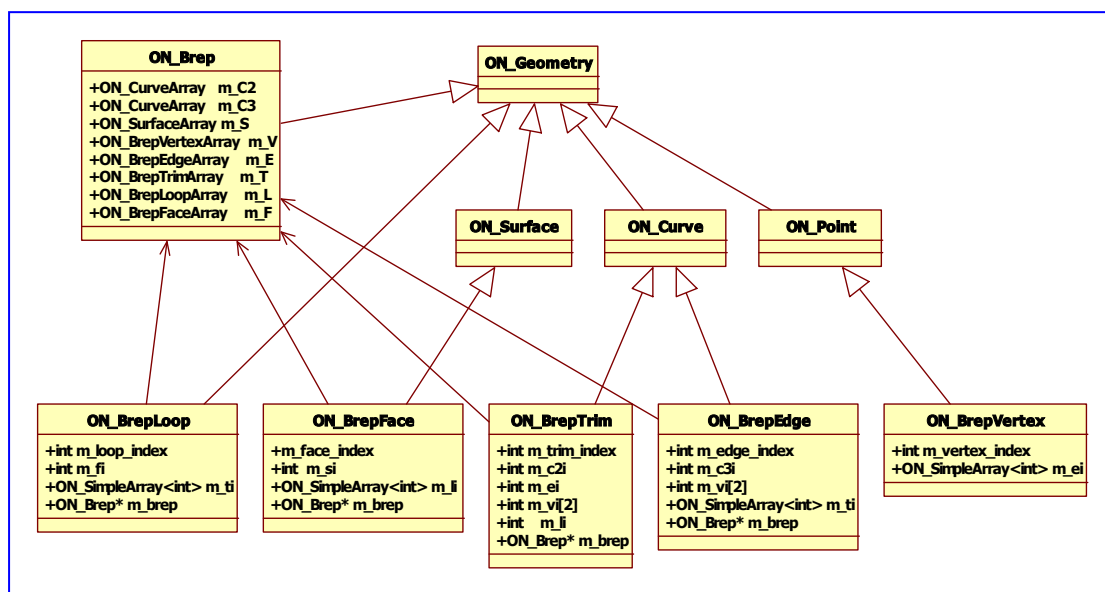


Figure 4.1 OpenNURBS Brep Class Diagram

由上图可知，类 ON\_Brep 中包含了 Brep 体的所有几何及拓扑信息。几何信息包含二维参数曲线 m\_C2、三维曲线 m\_C3、曲面 m\_S；拓扑信息包括顶点 m\_V、边 m\_E、Trim(m\_T)、Loop(m\_L)、面 m\_F。其它几种拓扑结构中分别包含信息描述如下：

ON_BrepFace	
成员变量	变量说明
m_face_index	Face 在 ON_Brep 中的编号；
m_li	Face 中包含的所有 Loop 的编号；
m_si	Face 的几何信息编号；
m_brep	包含所有几何及拓扑信息的 ON_Brep 指针；
ON_BrepLoop	
成员变量	变量说明
m_loop_index	Loop 在 ON_Brep 中的编号；
m_ti	Loop 中所有 Trim 的编号；
m_fi	使用这个 Loop 的 Face 的编号；
m_brep	包含所有几何及拓扑信息的 ON_Brep 指针；
ON_BrepTrim	
成员变量	变量说明
m_trim_index	Trim 在 ON_Brep 中编号；
m_c2i	Trim 的二维参数曲线；
m_ei	Trim 中边的编号；
m_vi[2]	Trim 端部的两个顶点编号；
m_li	与这个 Trim 相关的所有 ON_BrepLoop 的编号；
m_brep	包含所有几何及拓扑信息的 ON_Brep 指针；

ON_BrepEdge	
成员变量	变量说明
m_edge_index	边在 ON_Brep 中的编号;
m_v[2]	边两端顶点编号;
m_ti	与这条边相关的 ON_BrepTrim 的编号;
m_brep	包含所有几何及拓朴信息的 ON_Brep 指针;

ON_BrepVertex	
m_vertex_index	顶点在 ON_Brep 中的编号;
m_ei	从这个顶点出发或者回到这个顶点的边的编号;

从上面的类及表格可知，OpenNURBS 的 Brep 与 The Winged-Edge 结构很像了，在类 ON\_BrepTrim 中包含了与其他结构之间的关系。这个 ON\_BrepTrim 与 ON\_BrepEdge 的意义类似。

通过这些编号及 m\_brep 指针，可以很容易地访问到所需要的数据。也正是通过这些编号建立起了 Brep 内部数据之间的关系，这种方式比较清晰简单。



## 5. OpenCASCADE vs. OpenNURBS

综上所述，可以看出 OpenCASCADE 中的 Brep 表示法各有特色，下面对他们的做个简单的比较。

- ❖ OpenCASCADE 的拓扑结构要多于 OpenNURBS。OpenCASCADE 中有 TopoDS\_Solid, TopoDS\_CompSolid, TopoDS\_Shell，而 OpenNURBS 中没有这些。其中 OpenNURBS 没有 ON\_BrepSolid 这个结构，其实类 ON\_Brep 已经包含了所有几何及拓扑数据，相当于这个结构了；
- ❖ OpenCASCADE 的 Brep 表示要比 OpenNURBS 复杂。OpenCASCADE 中的 TopoDS\_Shape 和 TopoDS\_TShape 之间的组合关系及各自类的派生，使得其 Brep 表示法不直观，需要一些面向对象编程的思想才便于理解。而 OpenNURBS 中 Brep 表示要清晰简单明了。几何之间的关系（拓扑）通过编号建立联系，比较直观。
- ❖ OpenCASCADE 中的几何之间的关系（拓扑）要比 OpenNURBS 复杂。OpenCASCADE 中几何之间的拓扑关系通过类的成员变量的聚合关系来表示父结构与子结构之间的关系。通过类来定义几何之间的关系，如曲线上的点，描述了点与边之间的关系，表面上的线，描述了边与面之间的关系等。正是使用了聚合的关系，所以从父结构遍历子结构很方便，但是不能从子结构直接访问父结构了。而 OpenNURBS 中直接通过编号来建立这些关系，并且包含 ON\_Brep 的指针，使得数据的查找比较方便，可以直接访问任何结构，包括父结构的数据。
- ❖ 内存使用情况比较。由于 OpenCASCADE 中的智能指针 Handle 的使用，所以几何之间的关系都是使用了这种方式，即几何引用了相同的内存数据。如面上的点和面上的边这些关系中，所用的面都是同一个面的内存数据。但是需要内存空间来保存描述这些关系的对象。OpenNURBS 通过编号来建立这个拓扑图，只是多了这些编号的数据。
- ❖ 拓扑数据访问比较。OpenCASCADE 中的拓扑关系比较复杂，所以提供了一些访问函数，来便于相关数据的访问。OpenNURBS 中可以直接操作这些成员变量，也可以通过相关的函数。在面向对象的程序中，成员变量一般可以设置成 private，以确保数据安全。
- ❖ 边界表示方式比较。因为一般的书籍上都详细重点描述了边界表示中以为边为核心的翼边数据结构，所以有人想从这方面来给 OpenCASCADE 中的 Brep 表示法来给个说法。结合上面的类图可知，从边出发并不能访问到与这条边相关的其他信息，如顶点的信息。如果硬是想在这里给个名分，倒是从点出发可以找到边及面的几何信息，所以 OpenCASCADE 中的 Brep 表示法应该更像是以点为基础的表达方法。OpenNURBS 中通过 ON\_BrepTrim，可以访问其他的信息，而 ON\_BrepTrim 与边 ON\_BrepEdge 的意义有些相似，所以应该是以边为核心的翼边结构了。
- ❖ 几何数据表示的比较。OpenCASCADE 中几何数据的表示方式很多，即有常见几何数据的定义，如圆、抛物线、球面、圆柱面等，也有 B 样条曲线曲面。而 OpenNURBS 统一使用 B 样条曲线曲面。有人说分的种类多，处理有关算法（如求交算法）时可分情况来处理，性能较高。有人说统一使用 NURBS 曲线曲面来表示，算法统一，内核简单，但是有些算法的性能会稍差，都没有完美的解决方案啊。这应该留给数学家们去追求简单、统一、高效的方式了。

## 6. Conclusion

总之，OpenCASCADE 与 OpenNURBS 中的 Brep 表示法各有特色。OpenCASCADE 使用了更多的类的继承与组合，使 Brep 看起来有点复杂。而 OpenNURBS 中只使用了简单的继承，使用编号来建立拓扑结构，看起来要简单清晰很多。

从 Brep 表示法中三种方式：以点为基础、以边为基础及以面为基础上来看，OpenCASCADE 更像是以点为核心的方式，而 OpenNURBS 更像翼边结构。不管以什么方式感觉并不重要，只要能方便访问其拓扑结构即可。不管白猫黑猫，只看结果。

理解 Brep 表示的数据结构后，可结合构造基本体如立方体、圆柱体、球体等的算法，来理解相关的造型算法，并加深对 Brep 表示法的理解。

最后对 OpenCASCADE 和 OpenNURBS 的 Brep 表示法做的比较都是个人观点，仅供参考。由于水平所限，文中的错误不妥之处在所难免，敬请不吝指教，作者将不胜感激。

## 7. References

1. 孙家广，胡事民. 计算机图形学基础教程. 清华大学出版社. 2009
2. 孙家广. 计算机图形学. 清华大学出版社. 2000
3. 莫蓉，常智勇. 计算机辅助几何造型技术. 科学出版社. 2009
4. Hongxin Zhang, Jieqing Feng. Introduction to Solid Modeling. Zhejiang University.  
<http://www.cad.zju.edu.cn/home/zhx/GM/015/00-ism.pdf>
5. OpenCASCADE: <http://www.opencascade.org/>
6. OpenNURBS: <http://discourse.mcneel.com/category/opennurbs>