[opencascade造型引擎功能介绍](https://www.cnblogs.com/amtf/articles/2285775.html)

现今的CAD 系统大多通常都基于CAD 系统提供的二次开发包，用户根据要求定制符合自己要求的功能。AutoCAD就提供了AutoLISP、ADS 等都是比较通用的开发工具包。UG 也提供了多种二次开发工具：UG/Open MenuScript、UG/Open API 等。然而此类开发有着自身的缺点：

1．各个CAD 系统提供的二次开发工作各不相同，他们只能提供对特定CAD 系统的支持，甚至不支持相同CAD 系统的不同版本。

2．二次开发构建的协同设计平台，开放性能不够理想，很难充分利用企业现有的软、硬件条件，也很难利用将来增加的各类设施。因为在开发期间，很多功能已和具体的CAD系统静态绑定了。

3．一般CAD 系统功能强大，操作复杂，二次开发只能开发其中的一部分功能。如果要开发一个功能强大的平台，其工作量非常大，从而成本相当高。

Open CASCADE，一个开放源码CAD 内核，可以定制和扩展（添加新的功能组件，类的进一步继承），面向CAD/CAM ， 对主流CAD 数据格式提供支持（STEP/STL/IGES 等，可自行开发转换程序提供特定数据格式的支持），提供高级建模函数（拟合，有理样条曲线，拉

伸、旋转、扫出、层叠拉伸、圆角、倒角、薄壳、修剪、偏移等），参数化模型，提供几何模型的特征提取，对Visual C++/MFC 有很好的支持。

Open CASCADE是一套开放源代码的CAD/CAM/CAE几何模型核心，源自于法国的Matra Datavision公司，这一套函式库系统原来是著名的CADCAM软体EUCLID的开发平台，但是在1998年， Matra Datavision改变了经营策略，从以销售软体为主改变为提供CAD/CAM/CAE软体服务为主的获利模式，并且在1999年公布Open CASCADE的程式原始码，任何人都可以在该公司的网站上下载并且使用这些程式码，而Matra Datavision则以此函式库为基础位客户建立客制化的数值模拟软体，国内CADCAM产业所需的套装软体常常动辄数百万甚至上千万，如此高昂的软体 费用对于中小企业来说确实是非常沉重的负担，而且套装软体中的功能有些其实根本用不到，但是软体使用者却必须花钱购买这些用不到的功能，如果能够针对中小 企业的需求建构一个精简化的CAD/CAM软体，并且在生产需求改变的时候才增加新的功能，将可以有效节省软体成本，而Open CASCADE的成功经验或许可以提供产业界甚至学术界一些新的思考方向。

Open CASCADE是一个功能强大的三维建模工具，提供了点、线、面、体和复杂形体的显示和交互操作，经过深度开发后可实现纹理、光照、图元填充、渲染等图形操作和放大、缩小、旋转、漫游、模拟飞行、模拟穿越等动态操作。

   开源Open CASCADE从底层构建的CAD平台，所以在做机械仿真方面会好用很多，比如数控加工中模拟切屑的去除过程，对于干涉的检查等，使用CASCADE的实体布尔运算即可都可以实现。

OpenCascade提供二维和三维几何体的生成、显示和分析。主要功能有：

1．创建锥、柱、环等基本几何体；

2．对几何体进行布尔操作（相加，相减，相交运算）；

3．倒角，斜切，镂空，偏移，扫视；

4．几何空间关系计算（法线，点积，叉积，投影，拟合等）；

5．几何体分析（质心，体积，曲率等）；

6．空间变换（平移，缩放，旋转）。

高级功能：

1．应用框架服务；

2．数据交换服务。

      OCC是基于OO概念的C++类库，用于精密设备等设计应用程序。典型的应用是CAD（2D/3D）应用程序的开发，制造业，仿真，或者一些其他的图形工具。OCC让你能够很快的开发这些程序。

OCC提供以下模块:

n   提供2D/3D几何模型库，用以创建任意模型: 创建基本图元包含：棱柱，圆柱，棱锥，圆环等实现了布尔运算（交并差）

l   对实体进行倒角，圆角等操作

l   用偏移，壳，镂空等方式创建实体

l   计算属性，例如：surface, volume, center of gravity, curvature

l   计算几何 projection, interpolation, approximation

n   模型的显示，和视图操作等这些可视化模块:

l   3D 旋转

l   缩放

l   阴影

n   一个应用程序框架设计:

l   让没有几何信息的应用程序联合几何信息

l   参数化模型

l   Java Application Desktop (JAD), a framework for creating your Graphical User Interfaces (GUI)

n   数据交换模块提供了将OCC模型导入，导出为IGES和STEP等这些标准格式的功能。

Open CASCADE 模型库是Open CASCADE公司基于 CAS.CADE 技术 开发和销售的。它的设计是组件化的。 As such, they separate C++ classes for:

l   定义数据结构 (geometric modeling, display 和 graphic selection)

l   实现复杂算法

l   提供应用程序接口 (APIs)

让有关系的或者相似的类在同一个包内可以防止类的冲突;

C++ 类名的前面加上了包的名字做前缀。比如所有用来定义3D几何对象的类都在前面加上 Geom前缀。在 Geom中, 实现 Bezier surfaces 的类被命名为 BezierSurface, 它的全名是Geom\_BezierSurface。

大的模块划分为6个：

基础类：Foundation Classes

模型数据：Modeling Data

建模算法：Modeling Algorithms

可视化：Visualization

数据交换：Data Exchange

程序框架：Application Framework

OpenCasCade 6.3目录结构：

l   *data - 这里面包含各种不同格式的模型文件。都是occ支持的格式。*

l   *doc –occ的文档。*

l   *ros/adm - occ团队工具，可以用来重新生成occ。*

l   *ros/adm/win32 - 包含Visual C++ 6.0., 7.1. 和 8.0.的工程文件, 用来生成occ的win32版本。*

l   *ros/adm/win64 - 包含 Visual C++ 8.0.工程,用来生成win平台64位版本。*

l   *ros/drv - 此文件夹包含WOK生成的源码文件。*

l   *ros/inc - 包含所有的occ头文件。*

l   *ros/src - 此文件夹包含occ源码，根据开发模块组织存放。*

l   *ros/win32 - 此文件夹包含了occ的可执行程序（dll）和lib，有debug，release。*

l   *Samples - 此文件夹包含了帮助文档中的标准例程。*

l   *tools - 包含OCAF 浏览器 和 MFC 应用程序向导工具;*

l   *wok – 此文件夹包含 WOK (Workshop Organization Kit) 工具包，用以组织大型项目和开发团队。*

l   *3rdparty - 此文件夹包含用以支持occ工作的第三方库(Java, QT and Tcl) 和例程。*

l   *uninstall - 卸载要用到的文件。*

VS 下 Open Cascade Source Code 编译及自定义工程设置：

源码编译--工程编译顺序：

l   Foundation Classes (file FoundationClasses.\*)

l   Modeling Data (file ModelingData.\*)

l   Modeling Algorithms (file ModelingAlgorithms.\*)

l   Visualization (file Visualization.\*)

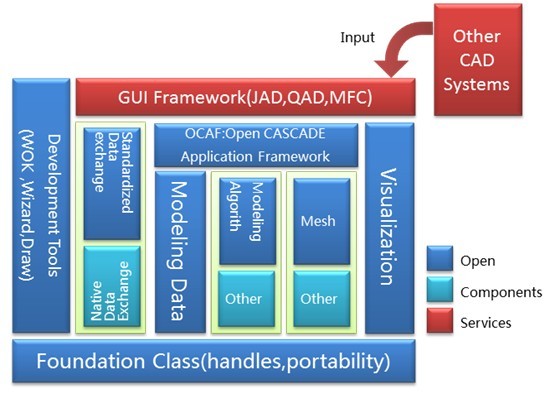
l   Application Framework (file ApplicationFramework.\*)

l   Data Exchange (file DataExchange.\*)

l   Draw (file Draw.\*)

l   WOK (file WOK.\*)：***WOK (Workshop Organization Kit) is Open CASCADE development environment。***

l   Wrappers (file Wrappers.\*)



OCC的结构框图

1.1   *Foundation Classes*基础类提供了一些通用的服务：

n   基本数据类型，字符串等

n   堆内存的自动化管理

n   异常处理

n   局部支持多线程

n   数据集合的处理

n   向量、矩阵及原始几何类型的数据计算工具

n   用ASCLL文件保存数据的基本服务

以上服务组织在以下三个类库中：

n   Kernel Classes（内核- TKernel）

n   Math Utilities（数据- TKMath）

n   Basic Persistence（基础持久存储-TKAdvTools）

|  |  |  |
| --- | --- | --- |
| **项目** | **目录名** | **功能描述** |
| **TKernel** | Dico |  |
| FSD |  |
| Message |  |
| MMgt |  |
| NCollection |  |
| OSD |  |
| Quantity | 数据及物理数量，是一个REAL 值；包括了所有您所需要的物理量。 |
| Resource |  |
| SortTools |  |
| Standard |  |
| StdFail |  |
| Storage |  |
| TCollection |  |
| TColStd |  |
| TShort |  |
| Units |  |
| UnitsAPI |  |
| **TKMath** | Bnd |  |
| BSplCLib |  |
| BSplSLib |  |
| Convert |  |
| CSLib |  |
| ElCLib |  |
| ElSLib |  |
| gp | STEP-compliant，实现基本几何及代数实体，用于定义和处理基本数据结构。  当在更为复杂的数据结构中时，GP实体不可以共享。 |
| math |  |
| PLib |  |
| Poly |  |
| Precision |  |
| TColgp |  |
| TopLoc | OCC中使用的拓扑数据结构包；提供了处理三维局部坐标系的资源， |
| **TKAdvTools** | Dynamic |  |
| Expr |  |
| ExprIntrp |  |
| GraphDS |  |
| GraphTools |  |
| Materials |  |

1. **1.**Root Classes是OCC的基本内核，是其它所有类创建的基础，包括：

l   基本数据类型，如BOOLEAN、CHARACTER、INTEGER、REAL(实型)；

l   可优化配置和回收大量的C++对象的，基于计数的内存管理器；

l   基类（Standard\_Transient）,通过智能指针-OCC句柄进行自动化内存管理，大多数的OCC类从该类继承；

1. **2.**Quantities类库提供以下服务***Standard\_Real class***：

l   常用***数学及物理量***的基本类型（如长度、面积、体积、质量、密度、重量、温度和压力）；

l   单位的转换

l   日期时间信息

l   颜色的管理

1. **3.**Exceptions类库能通过任何OCC功能引发，且兼容ANSI C++标准，允许你处理异常情况；OCC中的每一个异常类都直接或间接的继承于Standard\_Failure。

1. **4.**Strings类库提供以下服务处理字符串：

l   用内置的String管理器，编辑操作string对象；

l   处理动态的字符串序列

OCC支持ASCII码(8bit)、Unicode字符UTF-8(8bit)、utf-16-UCS-2 (16bit)；内存管理使字符串对象使用起来比字符数组更简单，可以通过名柄处理，可以共享的。串类在TCollection包和NCollection中实现。

1. **5.**Collections类库除了字符串，该包还包含动态大小数据集合类：

*【****TCollection****】、【****TColStd package (Collections of Standard Objects component)****】*

*【****The TColStd and TShort packages provide frequently used instantiations of generic classes****】*

l   阵列：一维和纯二维，一般用于快速访问的一个项目，且是固定大小的数组；

l   序列及列表可排序的非唯一对象集合；

l   散列图：提供对成员项的快速存取；

l   队列、堆栈；

该类库也提供特定的针对序列、散列、堆栈的迭代器；

集合容器是通用的，即每一种集合容器定义了一种结构和一些算法，可持有许多对象——通常这些对象不必从根类继承。这与C++模板相似。如果需要使用一个给定对象类型的集合容器，则必须对这个元素的指定类型进行实例化。一旦这个实例声明被编译，所有基于这个通用集合容器的函数都可以在集合容器对象中实现。

如果在工程项目中你不需要CDL（在WOK下编辑CDL需要实例化所有通用集合（TCollection包）），应该好好利用NCollection中定义的集合，它包含以上描述的同样通用的集合定义，队在其种形式的C++模板中；因此实例化所有无额外支持的集合类型需要越过ANSI C++编译器。

大多数集合是值语义的，意味着实例化一个集合后，它是真实存在的，并不是对它的一个句柄。

标准对象的集合容器：TCollection:标准对象集合类库提供频繁使用的来自TCollection包的通用类的实例，字符串来自于TCollection包和基本几何；存在一些局限性：OCC通用类需要在CDL语言中编译，因此仅能在WOK中实例化。

NCollection

1. **6.**Math Utilities：Vectors and Matrices 向量与矩阵类库提供常用的数据算法【***既有大小又有方向的量叫做向量***】，如下：

l   向量与矩阵的基本运算；

l   计算特征值与特征向量的方矩阵；

l   线性代数方程组的求解器；

l   找到一个非线性方程组的根；

l   找到一个或多个自变量函数的最小值算法；

同时也提供相应的数据结构，以表示任何表达式、关系或函数用于数学计算，包括变量的分配。

1. **7.**Primitive Geometric Types，【***gp package is a STEP-compliant implementation of basic geometric and algebraic entities---gp包是基于STEP实现的一个基础几何和代数实体***】这基本几何类型定义在基础Geometry包中，提供基本几何形状的描述如：点，向量、直线、圆、圆锥、平面及基本曲面；这些类型都是通过值来处理，非引用；通过坐标轴或坐标系，使几何图形在空间或者平面中进行定位以及几何变换的定义、应用（平移、旋转、对称、缩放、复合变换）。

1.2   *Modeling Data模型数据*

表现2D、3D几何模型的数据结构主要包含在四个类库中：2D几何（2D Geometry）、3D几何（3D Geometry）、几何工具（Geometry Utilities）、几何拓扑（Topology）。

|  |  |  |
| --- | --- | --- |
| **项目** | **目录名** | **功能描述** |
| **TKBRep** | BRep |  |
| BRepAdaptor |  |
| BRepLprop |  |
| BRepTools | 访问与处理抽像**拓扑**数据结构的方法；explore,   manipulate, 读、写BRep数据结构； |
| TopExp | 访问与处理抽像**拓扑**数据结构的方法；explore   and manipulate（操作）TopoDS中所描述的拓扑数据结构。 |
| TopoDS | OCC中使用的**拓扑**数据结构包；描述模型的类集合，建立纯拓扑的数据结构。 |
| TopTools | 访问与处理抽像**拓扑**数据结构的方法；用在拓扑数据结构上的工具库 |
| **TKG2d** | Adaptor2d |  |
| Geom2d | 描述简单几何对象的基本数据结构。用STEP实现二维几何对象；依据继承关系来组织类库；如：Geom2d\_Ellipse inherit Geom2d\_Conic  Geom2d\_BezierCurve inherit Geom2d\_BoundedCurve  Curve inherit Geom2d\_Curve  Curves, points and vectors inherit Geom2d\_Geometry |
| Geom2dAdaptor |  |
| Geom2dLProp | 二维曲线的本地属性； |
| GProp |  |
| LProp | 提供一个枚举用于标记二维曲线上的特征点 |
| TColGeom2d |  |
| TopAbs | OCC中使用的**拓扑**数据结构包；为拓扑驱动程序提供一些资源，包含描述基本拓扑概念的枚举，如形状、方向、状态等；同时也提供方法，管理这些枚举。 |
| **TKG3d** | Adaptor3d |  |
| AdvApprox |  |
| Geom | 描述简单几何对象的基本数据结构。用STEP实现三维几何对象；依据继承关系来组织类库；如：Geom2d\_Geometry inherit Geom\_ElementarySurface  Geom\_BezierSurface inherit Geom\_BoundedSurface  surfaces inherit Geom\_Surface  Curves, points and vectors inherit Geom\_Geometry |
| GeomAdaptor |  |
| GeomLProp | 三维曲线及曲面的本地属性； |
| LProp3d |  |
| TColGeom |  |
| **TKGeomBase** | AdvApp2Var |  |
| AppCont |  |
| AppDef |  |
| AppParCurves |  |
| Approx |  |
| BndLib |  |
| CPnts |  |
| Extrema | 两个二维曲线、两个三维曲线、一个三维曲线与一个曲面以及两个曲面的极值计算； |
| FEmtool |  |
| GC | 通过Direct Construction组件构造诸如（gp\Geom2d\Geom）的对象类型。但是前者实现了几何构造算法，用于转换构造器参数到每个对象的特殊数据结构。  它们提供了一组构造算法类；  gce\_MakeCire:定义了八个，在圆的几何构造中遇到的问题。实现了八个相关的构造算法； |
| gce |
| GCE2d |
| GCPnts |  |
| Geom2dConvert | Geom2d包相关；  transformation Bezier->BSpline；  transformation BSpline ->Bezier； |
| GeomConvert | Geom包相关；  BSpline、Bezier曲线，曲面的构造、转换等与之相关的算法； |
| GeomLid |  |
| GeomProjLib |  |
| GeomTools |  |
| Hermit |  |
| IntAna |  |
| IntAna2d |  |
| ProjLib |  |

1. 2D Gemoetry类库提供2D几何数据结构及拓扑方向。

Gemo2D几何包提供通过引用及遵循STEP标准处理2D几何数据结构处理，这些对象被参数化，面向对象的，包括Bezier贝塞尔曲线、B样条曲线、offset curves偏移曲线，提供从基本几何类型（非面向对象，非参数化）与Gemo 2D 对象的转换功能。Geom2dLProp几何属性包允许诸如此的计算：二维曲线基于一个参数点的Derivative向量、二维曲线（正常或曲率）上基于一个参数点的切向量

1. 3D Gemoetry类库提供3D几何数据结构及拓扑方向。

Gemo几何包提供通过引用及遵循STEP标准处理3D几何数据结构处理，这些对象被参数化，面向对象的，包括Bezier贝塞尔曲线、B样条曲面、offset curves偏移曲线及曲面，提供从基本几何类型（非面向对象，非参数化）与Gemo对象的转换功能。GeomLProp几何属性包允许诸如此的计算：二维曲线或曲面基于一个参数点的Derivative向量、二维曲线或曲面（正常或曲率）上基于一个参数点的切向量。

1. Geometry Utilities  该类库提供标准的在二维及三维几何上的高级功能：

l   几何算法的直接构造；

l   一组点插值形成一条曲线；

l   初等几何转换B样条曲线及曲面；

l   2D或3D曲线上点的计算；

l   两种几何的极值计算；

1. Topology库可以建立抽象拓扑的数据结构。

它定义了简单几何实体之间的关系，通过该方法可以利用简单实体的装配形成复杂的几何模型。由于一个内置的*非流形*non-manifold (or mixed-dimensional)*（混合维）*特性，可以建立混合模型（0维实体如点、1维实体如线、2维实体如面，3维实体如体积）。

抽象的拓扑数据结构描述了一个基本的实体 -一个形状，可分为以下组件拓扑结构：

* l   Vertex顶点： 一种0维形状对应于几何中的点；
* l   edge边： 一种形状对应一条曲线,并在每个末端通过一个顶点来约束；
* l   wire线框：一个通过顶点互连的边的序列；
* l   face面：部分（二维）平面或者（三维）表面，被一个闭合线框约束；
* l   shell壳(qiao)：线框边界线的边连接的面的集合；
* l   solid实体：与一个壳绑定的三维空间的一部分；
* l   Compound solid复合实体：实体的集合；

线框与实体可以是无限的也可以是关闭的；

三角网格模型（*STL文件格式模型*）就是用一系列的小三角形平面来逼近原来的模型，该模型就类似于一个多面体。

STL(stereolithography)文件格式是由美国3D Systems公司于1988年开发的，目前各类商用CAD软件均带有STL文件的输出功能，STL模型就是CAD模型进行三角网格化处理的结果。

1.3   *Modeling Algorithms 一系列在模型中使用拓扑算法形成建模算法模块组，通过这些，可以找到几何算法，服务封装在以下类库中：*

n   Geometric Tools ：几何工具

n   Topological Tools ：拓扑工具

n   Construction of Primitives ：图元构造

n   Boolean Operations ：布尔运算

n   Fillets and Chamfers ：倒圆与倒角

n   Offsets and Drafts ：偏移与拔模

n   Features ：特征

n   Hidden Line Removal：隐藏线去除

|  |  |  |
| --- | --- | --- |
| **项目** | **目录名** | **功能描述** |
| **TKBO** | BooleanOperations |  |
| BOP |  |
| BOPTColStd |  |
| BOPTools |  |
| BRepAlgoAPI |  |
| IntTools |  |
| **TKBool** | BRepAlgo |  |
| BRepFill |  |
| BRepProj |  |
| TopOpeBRep |  |
| TopOpeBRepBuild |  |
| TopOpeBRepDS |  |
| TopOpeBRepTool |  |
| **TKFeat** | BRepFeat |  |
| LocOpe |  |
| **TKFillet** | Blend |  |
| BlendFunc |  |
| BRepBlend |  |
| BRepFilletAPI |  |
| ChFi2d |  |
| ChFi3d |  |
| ChFiDS |  |
| ChFiKPart |  |
| FilletSurf |  |
| TKGeomAlgo | AppBlend |  |
| ApproxInt |  |
| FairCurve |  |
| GccAna |  |
| GccEnt |  |
| GccInt |  |
| GccIter |  |
| Geom2DAPI |  |
| Geom2dGcc |  |
| Geom2dHatch |  |
| Geom2dInt |  |
| GeomAPI |  |
| GeomFill |  |
| GeomInt |  |
| GeomPlate |  |
| Hatch |  |
| HatchGen |  |
| IntCurve |  |
| IntCurveSurface |  |
| Intf |  |
| IntImp |  |
| IntImpParGen |  |
| IntPatch |  |
| IntPolyh |  |
| IntRes2d |  |
| IntStart |  |
| IntSurf |  |
| Law |  |
| LocalAnalysis |  |
| NLPlate |  |
| Plate |  |
| TopClass |  |
| TopTrans |  |
| TKHLR | Contap |  |
| HLRAlgo |  |
| HLRBRep |  |
| HLRTopoBRep |  |
| Intrv |  |
| TopBas |  |
| TopCnx |  |
| TKMesh | BRepMesh |  |
| IntPoly |  |
| MeshAlgo |  |
| MeshDS |  |
| MeshShape |  |
| TKOffset | BiTgte |  |
| BRepOffset |  |
| BRepOffsetAPI |  |
| Draft |  |
| TKPrim | BRepPrim |  |
| BRepPrimAPI |  |
| BRepSweep |  |
| Sweep |  |
| TKShHealing | ShapeAlgo |  |
| ShapeAnalysis |  |
| ShapeBuild |  |
| ShapeConstruct |  |
| ShapeCustom |  |
| ShapeExtend |  |
| ShapeFix |  |
| ShapeProcess |  |
| ShapeProcessAPI |  |
| ShapeUpgrade |  |
| TKTopAlgo | Bisector |  |
| BRepApprox |  |
| BRepBndLib |  |
| BRepBuilderAPI |  |
| BRepCheck |  |
| BRepClass |  |
| BRepClass3d |  |
| BRepExtrema |  |
| BRepGProp |  |
| BRepIntCurveSurface |  |
| BRepLib |  |
| BRepMAT2d |  |
| BRepTopAdaptor |  |
| IntCurvesFace |  |
| MAT |  |
| MAT2d |  |
| TKXMesh | XBRepMesh |  |

1. Geometric Tools提供算法有：

l   计算两个交叉的二维曲线、曲面或三维的曲线、曲面；该组件用来计算2D或3D几何对象的交叉（二维曲线的交叉、二维曲线的自交叉、三维曲线与曲面的交叉、两个面个交叉）；

l   二维和三维曲线的投影，表面和表面上的三维曲线的投影（二维曲线上的二维点的投影、三维曲线或曲面上的三维点的投影、曲面上的三维曲线的投影、从三维到二维参数空间的平面曲线变换、二维几何对象在三维几何空间中的位置）；

l   众多二维圆弧或数值型直线或者与其它曲线的几何约束的构造算法（圆的半径、直线与直线的角度、直线或圆与一个曲线相切、点是否在一条直线或一个圆弧上等）；众多曲线与曲面的构造算法；

l   通过插值拟合构建曲线和曲面；

OCC也提供一项服务被叫做SSP（基于离散点的曲面重构） Surfaces from Scattered Points (SSP)..，算法首先构造一个初始的B样曲面，再查找其满足条件的变形（有限元法）；利用优化的算法，可以利用500000点来构造出一个曲面；

1. Topological Tools 拓扑工具包，该类库提供以下算法：

l   几何形状的细化；

l   形状验证；

l   确定形状的局部属性；

l   确定形状的全局属性；

l   进行几何变换；

l   查找平面的边界；

l   转换形状到NURBS（NURBS是非均匀有理样条曲线：NURBS是一种非常优秀的建模方式，在高级三维软件当中都支持这种建模方式。NURBS能够比传统的网格建模方式更好地控制物体表面的曲线度，从而能够创建出更逼真、生动的造型。）几何；

为了在模型数据模块的拓扑库上定义拓扑数据结构，还提供一个完整的BREP数据结构（边界表示[ Boundary Representation]也称为BR表示或者BREP表示，它是几何造型中最成熟、无二义的表示法）。

1. Construction of Primitives，该类库提供建立基本图元的算法（棱、柱、锥、球）；
2. Boolean Operations，该类库提供高质量的算法进行布尔运算，利用交、差、并、补计算来演示布尔操作创建新的实体；
3. Fillets and Chamfers过渡特征的构建

倒圆角特征(Fillets)

倒平角特征(Chamfers)

倒圆角特征和倒平角特征,习惯上被称为"倒圆"和"倒角"，它们建于两个面相交形成的棱边上,或多个面相交形成的顶点上，以修剪的形式而生成特征,即最终要将其参照棱线或顶点删除掉。在创建倒圆和倒角特征时,CAD系统会根据选择的棱线或顶点,在面集合中找出其包含面,在面的线集合或点集合中,找出相应的参照线或顶点,而表面隶属于哪些特征可由系统确定。

1. Offsets，Drafts，Sewing and Sweeps (偏移、拔模、缝合、扫掠)

l   创建偏移形状和他们的变体例如：

Hollowing镂空、Shelling抽空、Lofting放样；

l   利用拔模角来创建锥形；

l   缝合；

l   扫掠；

扫掠：是CAD中的一项建模工具 是将二维图形转为三维图形的建模方法，是将一个二维形体对象作为沿某个路径的剖面，而形成的三维图形。

拔模：顾名思义 就是模具做好之后 把它拔出来，如果拔出的作用力面 与力的方向是平行的，因为摩擦力大而难以拔出，同时会对模具表面造成损伤，所以机械直壁一般都不是垂直的，俗称拔模角度。拔模特征的创建功能,可方便地使体特征的任一表面绕其一棱边,顺时针或逆时针转动某一角度,同时保持拓扑关系不变。

1. Features（*特征建模？*）类库提供算法用于建立形状上的形态（Form）及机械(Mechanical)特征：

l   形态特征：柱形、棱形、旋转特征（Cylinders 、Draft prisms 、Prisms 、Revolved features 、Pipes ）

l   机械特征：筋（肋骨）（Ribs、Grooves）

1. Hidden Line Removal：提供算法来定义线条构成的形状隐藏在一个给定的投射中。这些线可显示或隐藏的。服务对形状自身或者它的简化多面体起作用。

*1.4   Visualization*

OCC平台中，可视化是建立在建模数据及描述图形的数据结构之上的,主要实现对实体对象的显示和选取操作。显示服务是独立于数据结构的，显示服务的表示由合适的算法产生。

|  |  |  |
| --- | --- | --- |
| **项目** | **目录名** | **功能描述** |
| **TKMeshVS** | MeshVS | 网格显示服务，对OCC三维显示能力的扩展 |
| **TKNIS** | NIS | 新的交互服务（New Interactive Services），它提供的API与AIS类似，但有一些重要的不同或改进；  该包是一个实验性的，有很多AIS的特性，它并不支持，比如：拓扑显示，本地环境（上下文）等。可用于创建高效的基于OPENGL的显示组件，但并不能取代AIS。可以认为它是对传统AIS的一个补充，更好的适合某些特殊任务。 |
| **TKOpenGL** | OpenGL |  |
| **TKService** | AlienImage | 导入图片，从其它格式到OCC所支持的格式。 |
| Aspect |  |
| CGM |  |
| Image | 提供伪彩色与彩色图像的定义，图像的一组关键功能； |
| InterfaceGraphic |  |
| MFT |  |
| PlotMgt |  |
| PS |  |
| SelectBasics | sensitive primitive原型的定义；在视窗中一个可选择的实体；  sensitive primitive原型所有者的定义；这个实体所原始实体与在视窗中可选择的应用程序实体相关系 |
| TColQuantity |  |
| Viewer |  |
| WNT | 提供通用的WINDOWS NT的图像接口 |
| **TKV2d** | AIS2D | 提供一些交互服务用于二维显示；提供选择与计算尺寸的方法；管理可视化的属性、提供方法加强可视化及可选择机制；交互对象依赖于可视化与可选择机制，在二维的VIEWER中来显示对象；控制显示及临时属性； |
| GGraphic2d |  |
| Graphic2d | 在视窗中创建二维图像对象；每一个对象称为一个图像对象，是由一些基本原件组成并在视窗中创建。每一个基本原件是一个类而且包含属性。每一个基本原件有自己的绘制方法。视窗负责管理一组图像对象。绘图器负责构造。 |
| Pre2d |  |
| Select2D | 定义二维原件的基本类库，如：点，线、圆，继承于SensitiveEntity   from SelectBasics，且用于表现二维可选对象。 |
| V2D | 提供了一些服务，需要在窗口系统中建立一个二维的mono-view   viewer视窗 |
| **TKV3d** | AIS | 应用交互服务（Application Interactive   Services (AIS)），提供方法创建应用程序GUI VIEWER之间的连接，用于管理选择与显示。定位于建模环境、VIEW以及选择。必须有几种定义的不同的可选择、可显示的对象。必须也是可交互的，也就是说，连接基本参考几何与图像显示；这些实体被称为交互对象，被分成以下四种类型：the Datum\the Relation\ the Object\ None。  the Datum:元素的构造如：线，圆、点、trihedra, plane trihedra、平面、轴线等（tri---三，hedra—多面体，异面体）。  the Relation：由一个或者多个交互形状与相应的几何关系的约束构成；  the Object：包括拓扑形状以及几何之间的连接  None：为了不排除对象，告诉应用程序，更深入分析，直到在其产生的对象（in its generation）中找到一个可接受的对象定义 |
| DsgPrs | 尺寸、关系及三面体的显示工具。 |
| Graphic3d | 在视窗中创建三维图像对象；这些对象称为结构，由一组原件及属性组成。这个组是一个最小的可编辑的结构元素。这个结构可被显示、擦出、高亮。可以对它应用变换。结构通过变换来构成，并连接形成一个继承体系。VIEWER可执行全局的结构操作处理。 |
| Prs3d | 提供通用几何及拓扑的标准显示工具；  显示对象；  属性管理员控制如何显示对象，如颜色、宽度、线的类型（generic   objects）；  通用算法提供默认对象的基本设置如：点，曲线、曲面、形状等  一个根对象提供抽象的框架，对于DsgPrs包中定义的工作； |
| PrsMgr | 提供一个低级别的服务，仅当您不使用AIS提供的服务时，才使用它。  支持一个显示对象的图像结构；当需要时进行重新计算（移动、改变颜色等）；  定义显示对象的显示模式；就AIS\_Shape而言，确定是否对象以（线框0、阴影1）来显示。  每一个新的交互对象必须由它定义自己的显示模式。 |
| Select3D | 定义标准的三维原件如：点，曲线，面  在二维图像选择空间中the bounding boxes的复原。  三维-二维的投影器 |
| SelectMgr | 提供一个低级服务，仅当您不使用AIS提供的服务时，才使用（SelectMgr\_SelectionManager and SelectMgr\_ViewerSelector）。  它通过以下服务来管理进程的动态选择：  激活与关闭交互对象选择模式；  添加与移除视察选择器；  定义抽象的过滤类； |
| StdPrs | 提供对特殊几何及拓扑的标准显示工具；这些类是特殊几何及拓扑，在务种显示模式下显示的定义，如：线框、阴影、去隐藏线等； |
| StdSelect | 定义拓扑几何的选择方式；定义几种过滤类；二维与三维的视窗选择器（viewer   selectors）。 |
| V3d | 包含了一组三维显示（3D Viewer）的命令与服务。控制场景及显示模式。这个包与Visual3D是互补的。 |
| Visual3d | 提供一组对三维视窗需要实现指令的类，视窗来管理场景和光源。 |
| Voxel |  |

1. Services Common to 2D and 3D通用二维与三维服务：该库提供了通用的二维与三维数据结构并分别通过不同的Viewer去处理。

3D Graphics:

运行在OPENGL图像平台的基础上；

支持对图元（折线、带孔或无孔的平面多边形、文本以及标志控制属性：颜色、透明度、反射、线型、线宽、字体等）进行三维操作，这些图元可以在三维视图中进行显示、绽放、旋转等操作；

2D Graphics:

支持对图元（直线、线段、折线、文本以及标志控制属性：颜色、线型、线宽、字体等）进行二维操作，这些图元可以在二维视图中进行显示、绽放、旋转等操作；

以PostScript, HPGL2, and CGM格式输出，同时也支持xwd,gif,bmp格式的图片。

在二维，OCC提供了完整的国际化，中文与日文的字符文本均可以在绘图仪上输入输出打印。

1. 3D Visualization 三维可视化类库提供了3D选择的数据结构及3D显示的数据结构；通过AIS（Application Interactive Services）提供三维的显示与选择；该包是一个高级接口，封装了底层的显示选择服务；AIS扩展了标准3D选择属性、显示管理以及显示属性的基础功能，并在GUI Viewer中进行管理；主要包括有：

l   交互上下文：

引导实体的选取和可视化的中央控制单元就是交互环境；

AIS有两种可操作环境模式；默认中性点（neutral point）类型很容易可视化及选择实体交互对象，这些对象都装载在AIS的上下文中；打开当前环境的方式是为临时选取作准备的，这样做就不会干扰在Neutral Point中的选取了。

l   交互对象：

在AIS Viewer中可显示、可选择的对象称之为实体对象，是基本几何模型与AIS图形显示的桥梁，可以使用OCC已经实现的预定义的标准交互对象的功能函数，也可以自己实现一套交互对象，我们只需要遵守一定的规定和协议就可以利用它来执行自定义的交互对象类。

l   图形属性管理器：

图形属性管理器，或AIS画图引擎，存储具体交互对象或受交互环境控制的交互对象的图形属性。

交互对象本身来说好像就是具体的图形属性（如可视化模式、色彩、材质等），交互环境中有一个绘图器drawer在默认状态下是有效的，用它来对对象的进行控制。当交互对象被可视化时，图形属性就会要求自身的绘图器来检查在当前环境下是否有效。

l   选择过滤器：

在选取操作时对需要选取的实体进行过滤是非常必要的，可以对选择过滤器的动态检测环境进行扩展、完善，以达到想要实现的效果。也可以自己编写过滤器并装载到交互环境中。

1. 2D Visualization 提供二维显示数据结构及二维选择数据结构。

*1.5   Data Exchange*

使用OCC作为应用程序的基础，同时使用其它的CAD系统，数据交换在其中扮演重要角色。它是具有多层软件环境的开源技术，处理外部数据并有良好的集成。它涵盖了基于OCC应用程序的交换的需要，并不拘于外部数据质量或要求。数据交换是以模块化的方式组织的，如下所述：

OCC提供连接器（工具），在基于OCC的应用软件数据转换成其它各种CAD软件的数据，保持数据有良好的互通性。

交换功能是依据各种不同的CAD或PDM等软件包所使用的标准(IGES, STEP)，或通过专有格式直接连接处理，或调用run-time 库访问外部数据。

利用国际标准，把CAD 系统中的模型通过各自的数据转换接口将其内部的模型数据信息转换为符合所选标准的中性文档描述文件，然后通过系统的数据转换接口把这些中性文档的数据读入系统。因此只要异构的CAD 系统支持同一种数据交换标准，就可以实现异构CAD 系统之间通过本平台进行协同。

                            数据交换模型

Open CASCADE 数据交换模块提供了大多数标准三维CAD 模型格式的读写工具类，包括STEP、IGES、BREP等。

|  |  |  |
| --- | --- | --- |
| **项目** | **目录名** | **功能描述** |
| **BinXCAFPlugin** |  |  |
| **TKBinXCAF** |  |  |
| **TKIGES** |  | IGES组件读取IGES文件并转换到OCC模型。目前支IGES 5.3版本，  转换涉及到一个实体、一组实体或整个文件。  转换之前，需要设置一组参数去管理转换进程。  也可以在转换之前，检查文件的一致性。  同时也支持把OCC模型转换到IGES文件。其它的数据类型如：颜色与名称可以通过独立的工具（IGESCAFControl\_Reader   and IGESCAFControl\_Writer）来进行读写。  一个IGES模型是已经被加载到内存的IGES文件；  一个IGES实体就是在IGES （IGES normal sense）中的一个实体；  一个根实体就是任何已给出类型的最高级别实体（如：type 144   for surfaces and type 186 for solids. Roots are not referenced by other   entities） |
| **TKSTEP** |  | STEP组件可以通过STEP应用协议214（类似与版本）读取STEP文件，转换到OCC模型。同时也支持STEP203。  基础转换器可读写几何、拓扑STEP数据以及组配结构。  转换涉及到一个实体、一组实体或整个文件；  其它数据类型如：颜色、验证属性、层、名称以及结构的组配可通过独立的工具（STEPCAFControl\_Reader and STEPCAFControl\_Writer）进行读写。  STEP组件也可以转换OCC模型到STEP文件。  该组件遵循STEP 应用协议214-2级生成STEP文件。当输出一个STEP格式时，需要选择转换模式，可通过STEPControl\_STEPModelType来实现。  一个STEP模型是已经被加载到内存的STEP文件；  所有对形状的引用都指OCC形状，除非有其它明确的说明；  一个根实体就是任何已给出类型的最高级别实体(如：an entity that is not referenced by any other one) |
| **TKSTEP209** |  |  |
| **TKSTEPAttr** |  |  |
| **TKSTEPBase** |  |  |
| **TKSTL** |  | STL组件转换OCC形状到STL文件，STL广泛使用于快速原型。  STL文件仅包括：通过网格结构描述实体（Solids）、写OCC形状必须是实体（Solids），实体的部分或者一个正确定位的封闭的壳。  面集合或者未封闭的壳可能也可以转换成STL格式，但可视化时或许会有错误的！ |
| **TKVRML** |  |  |
| **TKXCAF** |  |  |
| **TKXCAFSchema** |  |  |
| **TKXDEIGES** |  | Extended data exchange (xde)：扩展数据交换：它允许你修改交换的范围，作为增加了几何（“BREP”）数据，从而提高了与外部软件的互通性。 |
| **TKXDESTEP** |  |
| **TKXmlXCAF** |  |  |
| **TKXSBase** |  |  |
| **XCAFPlugin** |  |  |
| **XmlXCAFPlugin** |  |  |

1. Standardized Data Exchange:标准化的数据交换

标准化的数据包括：STEP、IGES、STL、VRML

1. Proprietary Data Exchange：专用的数据交换

连接器（可选作为独立的组件）提供与其它的CAD软件数据进行直接映射和数据适配（利用几何复原）：

* l   Euclid-3
* l   Catia
* l   ACIS
* l   Parasolid
* l   DXF

此连接器同STEP及IGES的数据交换具有相同的架构。

IGES：基本图形交换规范（The Initial Graphics Exchange Specification），1980年美国国家标准局（NBS）主持成立了由波音公司和通用电气公司参加的技术委员会，制订了基本图形交换规范IGES，并于1981年正式成为美国国家标准。作为较早颁布的标准，IGES被许多CAD/CAM系统接受，成为应用最广泛的数据交换标准。

STEP：产品模型数据交换标准（STandard Exchange of Product data model），1988年ISO制定的描述整个产品生命周期内产品信息的标准，它提供了一种不依赖具体系统的中性机制，旨在实现产品数据的交换和共享。这种描述的性质使得它不仅适合于交换文件，也适合于作为执行和分享产品数据库和存档的基础。发达国家已经把STEP标准推向了工业应用。它的应用显著降低了产品生命周期内的信息交换成本，提高了产品研发效率，成为制造业进行国际合作、参与国际竞争的重要基础标准，是保持企业竞争力的重要工具。

STL ：= STL文件，一种3D模型文件格式STL（STereo Lithography的缩写）

STL 文件格式是由3D SYSTEM 公司于1988 年制定的一个接口协议，是一种为快速原型制造技术服务的三维图形文件格式。STL 文件由多个三角形面片的定义组成，每个三角形面片的定义包括三角形各个定点的三维坐标及三角形面片的法矢量。三角形顶点的排列顺序遵循右手法则。 STL 文件有2 种类型：ASC域格式和二进制格式ASC域格式。 　　STL模型是以三角形集合来表示物体外轮廓形状的几何模型。

VRML：（Virtual Reality Modeling Language）即虚拟现实建模语言。是一种用于建立真实世界的场景模型或人们虚构的三维世界的场景建模语言，也具有平台无关性。是目前Internet上基于 WWW的三维互动网站制作的主流语言。 VRML是虚拟现实造型语言（Virtual Reality Modeling Language）的简称，本质上是一种面向web，面向对象的三维造型语言，而且它是一种解释性语言。VRML的对象称为结点，子结点的集合可以构成复杂的景物。结点可以通过实例得到复用，对它们赋以名字，进行定义后，即可建立动态的VR（虚拟世界）。

1.6   *Application framework应用框架（OCAF）*：

*OCAF（ Open CASCADE Application Framework），Open CASCADE 不仅为用户提供了三维建模、几何体复杂运算、可视化模块等相关类库，还为用户提供了OCAF，OCAF提供了一个结构化的解决方案和应用数据处理，是一种内HE级的管理 ，它是一种基于Open CASCADE 的RAD（Rapid Application Development，快速开发工具）框架，为用户（即开发人员）提供了一个快速开发的手段。*

*OCAF 对于Open CASCADE 是一个的辅助工具，在算法、拓扑或几何建模的过程中，OCAF 能处理这些类库中的任何数据和算法，这个框架还可以用于组织、保存应用数据。通过使用OCAF，应用程序设计者可以专注于功能以及某些特别的算法， 值得一提的是实现“ 撤销/ 重做（Undo-Redo）”命令以及“保存应用程序数据”命令，避免了很多底层建模的问题。*

应用程序框架就是使用一个关联（相关性）引擎，简化对CAD应用系统的开发。

* 通过属性机制处理应用程序数据；
* 依据开发的需要来组织属性；
* 由应用程序来进行多文档管理；
* 对于CAD/CAM应用程序，已实现、可使用的通用的数据结构属性；
* 数据存储服务；
* 已实现撤销/重做功能；

通过使用OCAF，它来处理应用程序结构，应用程序设计者可以专注于特定应用数据的处理及GUIS的开发工作。

OCAF与其它CAD框架不同就在于应用数据的组织。在OCAF中，数据结构不是shape-driven .而是reference-key driven.在这种方法中，如几何数据、颜色、材料等属性都附加于更深的结构不变的模型上。OCAF负责把这些属性组织、嵌入文档。

OCAF文档由OCAF的应用程序来管理。

数据存储服务：

通常来讲，OCC提供的所有数据结构都是一种运行时结构，换句话说，是一种短暂的不稳定的数据，只有在应用程序运行时才存在并不进行永久存储。但，数据存储模块具有存储数据到存储设备上的功能，即持久化数据。数据存储服务提供持久化数据的类库和转换功能，就是为了把短暂不稳定的数据转换成永久的数据。