

OCCT OCAF

Chen Chen

January 13, 2024

Contents

1	OCAF (OpenCascade Application Framework)	1
1.1	Purpose of OCAF	1
1.2	OCAF 依赖与 OCCT 的其他 modules	2
1.3	OCAF Architecture	2
1.3.1	Application	3
1.3.2	Document	3
1.3.3	Attribute	3
1.4	Reference-key model	4
1.4.1	Ref-keys 创建途径: 程序创建和用户创建	5
2	The Data Framework	5
2.1	学习步骤	7
2.2	TDF package	8
2.3	TDocStd package	8
2.4	XCAF package	8
2.5	TFunction package	8
2.6	TPrsStd package	8

1 OCAF (OpenCascade Application Framework)

OCAF 模块提供了一种在 CAD 应用中组织和管理数据的框架。OCAF 的设计目标是提供一个灵活的、可扩展的、支持版本控制的数据模型，以便更容易地创建和管理复杂的 CAD 数据。

1.1 Purpose of OCAF

一般来说，要开发一个设计软件，我们需要提前想清楚如下一些问题：

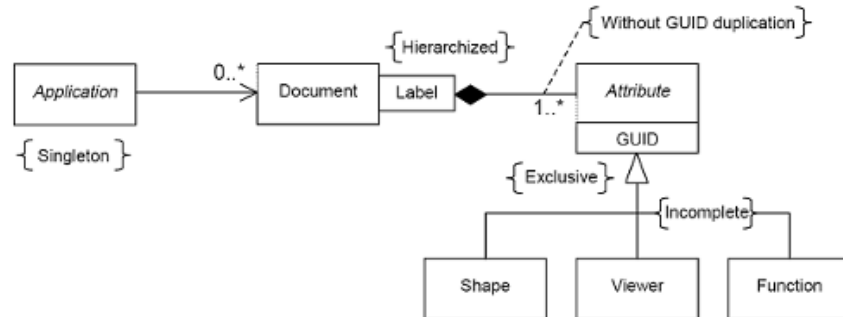
- 软件的架构是怎样的？有哪些 components，组件间如何交互
- 软件的数据模型是怎样的？如何组织软件中的各种数据
- 如何保持 data model 与 viewer 之间的同步
- 如何支持 undo/redo 功能
- 如何持久化软件数据

OCAF 模块为上面这些设计问题，提供了解决方案。

1.2 OCAF 依赖与 OCCT 的其他 modules

- Shape from Modeling Data module
- Viewer from Visualization module
- Modeling functions from Modeling Algorithms module

1.3 OCAF Architecture



- 一个设计软件只有一个 Application 实例
- Application 实例可以若干个 Document
- 一个 Document 是一个 Label tree，可以包含多个 Labels
- 一个 Label 包含了若干个 Attribute
- 一个 Attribute 实例可以是 Shape, Viewer, Function 中的一种

1.3.1 Application

Application 是负责管理文档的抽象类。其主要职责有：

- 创建文档
- 保存文档
- 打开文档
- 初始化文档视图

1.3.2 Document

Document 是承载 App 数据的容器，主要职责有：

- 管理数据的变更通知
- Update external link # 不太懂
- 管理数据的保存与加载
- store the names of software extensions # 不太懂
- 管理 command transactions (command 一般指修改操作)
- 管理 Undo/Redo 选项

Document 以一定的 format 保存为 ASCII 文件。通过 reference key，一个文档内可以引用另一个文档中的 label。

1.3.3 Attribute

All Attributes are organized by the OCAF Data Framework. The Data Framework references all attributes using persistent identifiers in a single hierarchy. (感觉 Data Framework 像一个 Allocator)

软件数据实际上存储在 Attributes 上。Attributes 包含多种继承子 Attribute abstract class 的类型：

1. Standard attributes

Standard attributes 涵盖了

- Simple common data-types: integer, real, string, arrays

- 辅助功能的 types: tag sources attribute for the children of the label counter.
- 创建依赖的 types: reference, tree node

2. Shape attributes

Shape attributes 包含了建模的几何数据，以及 shape reference, shape evolution tracking 等数据。

还有一些其他的几何数据:

- **Datums (基准)** (points, axis, plane)
- **Constraints** (tangent-to, at-a-given-distance, from-a-given-angle, concentric, etc.)

3. Visualization attributes

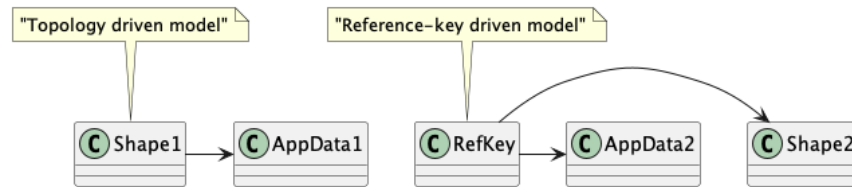
Visualization attributes 包含了一些用于 graphical data representation 的数据类型，如 viewer information, visual representation of objects, 以及 auxiliary visual info.

4. Function services

- The purpose of these attributes is to rebuild objects after they have been modified (parameterization of models).
- While the *document* manages the notification of changes, a *function* manages propagation of these changes.
- The function mechanism provides links between functions and calls to various algorithms.

1.4 Reference-key model

Reference-key driven model V.S. Topology driven model



In most existing geometric modeling systems, the data are topology driven. They usually use a BRep, where geometric models are defined by a collection of faces, edges and vertices, to which app-data are attached.

In OCAF, the data are reference-key driven. It is a uniform model in which reference-keys are the persistent identification of data. All data are implemented as attributes attached to ref-keys. 通过一个 ref-key, 多个 attributes 被关联在一起。

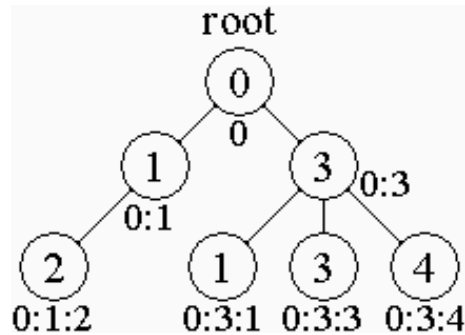
1.4.1 Ref-keys 创建途径: 程序创建和用户创建

- As an application developer, you generate reference-keys in order to give semantics to the data.
 - For example, a function building a prism may create three ref-keys:
 - * one for the base of the prism,
 - * a second for the lateral faces
 - * a third for the top face.
 - * these ref-keys makes up a semantic for the prism feature.
- The end-user can set a texture to a face he selects, and a ref-key to the face will be created.
 - 当你为所选 face/edge/vertex 创建 ref-key 时, OCAF 会自动附加一个 Naming attribute, 用于标注所选 topo.

在 parametrized modeling 中, 当你修改了一个 parameter value, 一个 topo element 可能会发生平移。这时通过 ref-key 关联的 data 也会相应更新。

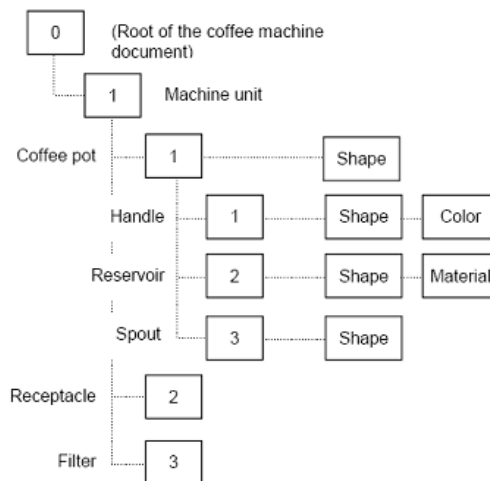
2 The Data Framework

OCAF Data Framework (TDF) 是一个 ref-key model 的实现, 基于 tree structure。



TDF 的基本构建元素有: Tag, Label, Attribute.

- 每个 Label 都有一个整数型的 Tag 作为其 local ID。
 - TDF 中的第一个 Label 为 root node of tree structure
 - * 一个 label 的 sub-labels 被称为其 **children**
 - * 除了 root label, 每个 label 都有一个 **father**
 - Label 的语义 (以一个水壶模型为例):
 - * root label 代表水壶模型整体
 - * 其 sub-labels 分别代表了手柄, 壶嘴, 盖子, 壶体等
 - * 以盖子 label 为例, 可以进一步按照盖子的构造进行细分
 - * 下图为水壶模型的 TDF Tree



- 每个 Label 都可以由一个 tag-sequence 在文档内被唯一标识, 如 0:1:2:1, 即从 root label 到当前 label 的路径。

- 每个 Label 可以包含一个 Attribute 列表。
- Attribute 用于存储真正的数据。
 - 每个 Attribute 被一个 GUID 唯一标识。

OCAF 中的一些关键 packages 有：

- **TDF (Data Framework)**

提供了一种用于存储和管理通用数据的框架，是 OCAF 的核心包，定义了文档、标签、数据存储等概念。提供了基本的数据存储和版本控制机制。

它定义了数据模型的基本概念，如标签、数据类型等。

- **TDocStd (Technology Document Standard)**

TDocStd 提供了文档和存储的概念。它定义了如何组织和管理存储在文档中的数据。

- **XCAF (Extended Computer-Aided Facilities)**

XCAF 提供了对扩展数据的支持，包括属性、引用和组件的概念。它使用户能够为几何模型添加附加信息。

- **TFunction**

TFunction 允许用户将功能与数据关联，使得模型的操作和行为更加动态和可定制。

- **TPrsStd (Technology Presentation Standard)**

TPrsStd 提供了用于在用户界面中显示和呈现几何模型的工具。

2.1 学习步骤

- 先了解 TDF 的基本概念，包括标签、数据类型和版本控制。为后续学习奠定基础。
- 学习使用 TDocStd 创建、管理和保存文档。了解如何在文档中存储数据，并理解版本控制的原理。
- 学习使用 XCAF 扩展数据，包括属性、引用和组件。让你能够为几何模型添加附加信息，使其更丰富和灵活。
- 学习使用 TFunction 将功能和操作关联到模型。让你更灵活地处理和操作几何模型。

- 学习使用 `TPrsStd` 在用户界面中呈现和显示几何模型，包括创建图形视图以及将模型呈现到用户界面中。

2.2 TDF package

2.3 TDocStd package

2.4 XCAF package

2.5 TFunction package

2.6 TPrsStd package