

基于Catia Composer的三维模型Web浏览平台开发

沈 刚, 官雄明, 冯卫星, 张杰铭

(红塔烟草(集团)有限责任公司, 云南 玉溪 653100)

摘 要: Composer player是法国达索公司开发的一款轻量化三维模型播放软件, 轻量化模型的展示在产品验证、推广、销售等环节起着至关重要的作用。本文结合控件提供的开发接口, 简要介绍了Web网页内调用该控件的方法, 并辅以实例讲解控件所提供方法、事件、属性的调用步骤。

关键词: Composer API; 二次开发; 轻量化模型; 网页开发

中图分类号: TP311.1 **文献标识码:** A

Development of 3D Model Web Browsing Platform Based on Catia Composer

SHEN Gang, GUAN Xiongming, FENG Weixing, ZHANG Jieming

(Hongta Tobacco (Group) CO., LTD, Yuxi 653100, China)

Abstract: Composer player is a 3D model desktop application developed by Dassault Systèmes, which plays significant role in the fields of marketing, sales, customer service, etc. In combination with the API in the ActiveX control, this paper introduces how to use the control and how to call the methods, events, and properties provided by the ActiveX control in a web page.

Keywords: Composer API; secondary development; simplified models; web development

1 引言(Introduction)

Composer是法国Dassault Systèmes研发的一款桌面应用程序, 旨在为非CAD用户提供一个可以直接从3D源文档创建相关的2D、3D轻量化产品文档的平台, 被广泛应用于销售、市场推广、客户服务、培训、支持等领域^[1]。Composer支持直接打开大部分3D CAD设计软件的原始设计文档(Catia、UG、Creo、Solidworks、Inventor), 以及各类中间格式, 并可以在原始设计文档发生变更时, 将变更更新到Composer内, 避免了Composer文档的重建。

目前, 基于Composer的轻量化三维模型可视化系统在各行业得到了广泛的应用, 相关研究人员分别提出了三维模型装配工艺可视化系统^[2-4]; 基于Flash的网络教学虚拟仿真课件平台^[5], 这些成果重点研究了轻量化模型的展示效果, 未对轻量化平台与其他系统集成具体实施过程进行介绍。本文将重点讨论轻量化软件Composer Player与网页集成的详细方法。

2 网页与3D模型的结合(Combination of 3D models and web pages)

Composer最初是达索公司的独立软件3D Via Composer, 后来被分别集成到Catia V6和Solidworks的安装包内供用户选择购买安装, 相应的名称分别变更为Catia Composer和Solidworks Composer, 但其在使用方法与内核上保持一致。

当下, 越来越多的行业将产品3D模型置于WEB网站供用户浏览, 以此提高客户对产品的感知度, 帮助客户制定精准可靠的采购计划。WEB作为企业宣传的有力保障, 相对于传统的Word、PPT, 其适用面更广泛, 用户群体更大, 更方便用户通过WEB与产品进行互动交流操作, 更有利于企业产品的宣传推广。

在工业领域, 3D模型零件数量庞大, 产品结构复杂。为降低企业设计图纸泄密风险, 同时提高用户体验, 3D模型发布至网页端前的精简轻量化必不可少。Composer为企业提供了可靠的模型轻量化解决方案, 同时开放众多的API接口, 用

户可以灵活地定制符合工作需求的各类命令及功能^[6]。

3 二次开发准备(Secondary development preparation)

本文将结合Catia Composer Player Pro的ActiveX控件所提供的API,使用Visual Studio 2015作为开发工具,介绍如何在HTML网页内调用接口内的事件、方法和属性。

在进行网页开发前,需在电脑上安装Composer Player软件并取得相应许可。其中,Player ActiveX 内的部分功能需取得Pro许可证后方可访问。控件会在软件安装过程中自动注册到计算机,用户也可通过安装路径内的批处理文件“<install_dir>\bin\Register.bat”进行手动注册^[1]。

4 网页框架构建(Web framework construction)

4.1 初始化控件

首先,新建一个ASP.NET空网站,本文以Visual C#语言为例。在新建的空网站项目上添加一个WEB窗体,并命名为MyWebPlayer。接下来,在该WEB窗体内声明Composer Player ActiveX.ocx控件,OCX控件存放在Composer安装路径下。要声明OCX控件,首先需要获得控件的ClassID,ClassID是ActiveX控件的唯一标识符,ClassID可以从注册表编辑器中获得。”\HKEY_CLASSES_ROOT\CLSID\{410B702D-FCFC-46B7-A954-E876C84AE4C0}”,花括号内的字符串即为所需ClassID。

在MyWebPlayer.aspx文件的<body> </body>间使用<Object>标签声明该控件。

```
<body>
<form id="composer_Ocx_Form" runat="server">
  <object id="_ComposerPlayerActiveX"
height="600" width="100%"
classid="CLSID:410B702D-FCFC-46B7-A954-E876C84AE4C0">
  <param name="FileName" value="HTML玉溪红塔山.smg" />
  <param name="AutoPlay" value="0" />
  </object>
</form>
</body>
```

声明语句中,“id”是用户为该控件定义的身份标识,用户通过该id在WEB应用程序内调用控件的属性和方法,“height”和“width”定义了控件在WEB窗口中的大小,WEB应用程序通过“classid”调用ActiveX控件。

同时,在“<Object>”标签内通过“<param>”定义控件的一系列属性值,如模型渲染模式、灯光效果、大地栅格、自动播放、循环播放等模型加载后的显示效果,也可以定义控件加载时是否显示视图框、标准工具条、协同窗口等用户操作菜单栏。需要说明的是,禁用不必要的工具栏能够极大地提高控件加载速度、改善用户体验。文中仅定义了控

件初始化时打开的文档路径及名称,自动播放“AutoPlay”处于关闭状态。用户可以查阅编程手册定制更多的属性值。

至此,WEB网页完成了控件的声明,用户可在以IE为内核的浏览器中查看效果。

4.2 控件交互

网页内的按钮、文本框、进度条是实现用户与操作界面交互的基本手段,交互界面的设计应力求简洁。简单易懂的界面设计既可以提升用户使用过程中的易用性,又能够降低使用者的培训成本^[7]。

在WEB页面内设计三个按钮,分别用于改变所选零件的颜色、位置,重载默认属性;插入一个文本框,用于显示信息。

```
<body>
<input id="Button1" type="button" value="改变颜色"
onclick="Javascript:SetColor()"/>
<input id="Button2" type="button" value="零件平移"
onclick="Javascript:MovePart()"/>
<input id="Button3" type="button" value="恢复默认"
onclick="Javascript:RstDft()"/>
<textarea id="MsgBox" cols="40" rows="1" aria-
multiline="True"></textarea>
</body>
```

在<head>标签内声明鼠标左键单击事件,当用户单击3D零件时,在文本框内显示所选零件名称;同时声明“EndLoadModel”事件,当模型加载完毕后,触发该事件,获取并备份所有几何模型的默认属性值,默认属性值用于在用户误操作后将零件属性重置为初始值。

```
<head>
<script event="Click" for="_ComposerPlayerActiveX"
language="javascript">
  ShowName(); //鼠标左键单击事件
</script>
<script event="EndLoadModel" for=
"_ComposerPlayerActiveX" language="javascript">
  GetNeutralProps(); //模型加载完成触发事件:获取模型默认属性
</script>
</head>
```

5 功能实现(Function implementation)

为实现用户的定制化需求,控件内提供了大量事件与方法供用户访问。本文将控件提供的方法封装到JavaScript文件内,简化代码编写和调用过程。

为项目添加一个Js文件,命名为“MyFunctions.js”。在文件内添加函数以实现按钮功能,添加事件响应函数以处理鼠标单击事件。

文中主要使用了API提供的以下接口函数,如表1所示。

表1 文中引用的接口列表
Tab.1 API lists used in this paper

API方法	API方法简要解析
Selection()	返回鼠标所选零件的名称或GUID, 支持多选
SetPropertySet()	修改、设置所选对象的单个或多个属性
SetPropertyMap()	修改、设置属性页
RefreshScene()	刷新屏幕, 将更改刷新到屏幕显示
TranslateInWCS()	在世界坐标系内移动指定的对象
GetAllActorsByType()	通过对象类型选择对象, 对象可以是几何体、BOM表、视图等
GetActorsProperties()	返回所选对象的所有属性, 支持多选

“MyFunctions.js”文件中, 各函数的实现代码如下所示:

```
function SetColor() { //改变所选零件的颜色
    var strSelection;
    var bValue;
    var strPropertySet;
    strPropertySet="<CLitPropertySet><Actor.Color
R='153' G='51' B='0' /></CLitPropertySet>"
    strSelection=document._ComposerPlayerActiveX.
Selection();
    bValue=document._ComposerPlayerActiveX.SetPr
opertySet(strSelection,strPropertySet);
    document._ComposerPlayerActiveX.
RefreshScene(1);
}
function MovePart() { //零件平移
    var strSelection;
    var bValue;
    strSelection=document._ComposerPlayerActiveX.
Selection();
    bValue=document._ComposerPlayerActiveX.Tran
slationInWCS(strSelection,30,0,25);
    document._ComposerPlayerActiveX.
RefreshScene(1);
}
function ShowName() { //鼠标单击事件响应
    var strSelection;
    var TextBox;
    TextBox=document.getElementById("MsgBox");
    TextBox.value=document._ComposerPlayerActiveX.
Selection();
}
function RstDft() { //恢复默认属性
    var bRtn;
    bRtn=document._ComposerPlayerActiveX.
SetPropertyMap(strProps);
    document._ComposerPlayerActiveX.
RefreshScene(1);
}
```

```
function GetNeutralProps() { //获取默认属性,在
模型加载完毕后触发事件
    var strSelection;
    strSelection = document._ComposerPlayerActiveX.
GetAllActorsByType("CLitActorMesh");
    strProps = document._ComposerPlayerActiveX.
GetActorsProperties(strSelection);
}
```

在浏览器内打开网页, 验证各命令按钮功能是否与设计相符, 如图1所示。

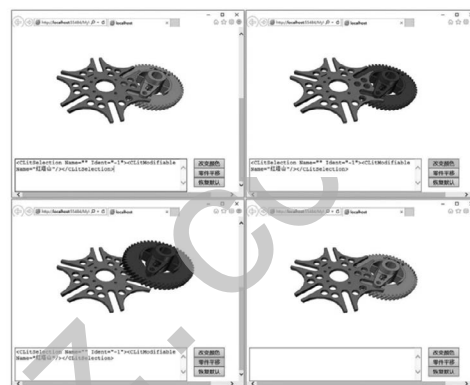


图1 在浏览器内验证各按钮功能是否与设计相符

Fig.1 Verification of the function and design of each button in the web browser

6 结论(Conclusion)

本文以Visual Studio 2015为开发平台, 简单介绍了Catia Composer Player ActiveX在WEB端二次开发的基本流程, 并结合实例讲解了实际应用过程中的关键步骤。定制化的浏览器端产品三维展示平台的搭建, 能够极大地增强企业产品的展示效果, 助力销售、研发、制造等环节。

参考文献(References)

- [1] Dassault Systèmes.Composer Programming Guide R2015x.D Systèmes,2014.
- [2] 屈力刚,魏佩振,苑俊超.基于三维工艺模型的装配现场可视化[J].制造业自动化,2013,35(20):83-87.
- [3] 周亚丽,江征风,王立伟,等.基于3 DVIA Composer 的金刚石线切割机装配工艺可视化研究[J].现代制造工程,2015(3):71-74.
- [4] 彭煜,盛步云,潘阳芬,等.基于3DVIA Composer的汽轮机装配工艺可视化研究[J].智能制造,2012(4):87-89.
- [5] 谷艳华,朱艳萍,杨得军,等.用于网络教学的虚拟仿真交互式课件研究[J].图学报,2016,37(4):545-549.
- [6] DS SOLIDWORKS公司.SolidWorks Composer使用指南[M].北京:机械工业出版社,2014.
- [7] 马跃.SolidWorksComposerAPI应用小窥[J].智能制造,2015(11):56-59.

作者简介:

沈 刚(1990—),男,硕士,助理工程师.研究领域:软件开发.
官雄明(1988—),男,硕士,助理工程师.研究领域:虚拟仿真.
冯卫星(1972—),男,本科,助理工程师.研究领域:机械设计.
张杰铭(1987—),男,本科,工程技术人员.研究领域:前端设计.