

面向 Web 的 BIM 三维浏览与信息管理

王珩玮 胡振中 林佳瑞 张建平

(清华大学土木工程系, 北京 100084)

【摘要】 BIM 服务器为 BIM 对建设全生命周期的支持提供了解决方案。目前 BIM 服务器主要为 C/S 架构, 但 B/S 架构的优势也渐渐凸显。HTML5 标准的发布带来了支持浏览器硬件图形加速的 WebGL 技术, 更带来了一系列划时代的 Web 技术。从而为实现 B/S 架构下的 BIM 信息管理与模型三维浏览奠定了基础。本文分析了 BIM 的模型信息范围与表达形式, 研究了基于 WebGL 框架 Three.js 的 BIM 模型显示技术以及面向 Web 的 BIM 信息管理技术, 开发了 BIM 模型三维浏览与信息管理系统。应用表明, 系统的图形平台性能稳定, 渲染效果较好, 能够支持较大的 BIM 模型。同时, 系统为 Web 客户端提供了丰富的功能支持, 更体现出良好的交互特性, 因而具有较大的应用价值。

【关键词】 建筑信息模型(BIM); WebGL; HTML5; B/S; Three.js

【中图分类号】 TU17; TP391 **【文献标识码】** A **【文章编号】** 1674-7461(2013)03-0001-07

1 引言

上世纪 90 年代, 国内设计企业逐步抛弃图板^[1], 取而代之的是计算机辅助设计(CAD)技术的普及。这是建筑业信息化与数字化的第一波潮流, 图纸的形式从纸变为数字信息, 从而有利于降低绘图强度、提高绘图精度与设计效率, 并便于资料管理与传输。然而, 绘图的数字化并不能代表整个建设过程的数字化。仅通过 CAD 技术并不能完全将数字化的优势引入建设的整个流程。

建筑业是复杂的, 建设项目的全生命周期是多阶段的, 而整个项目过程中涉及的人员更是数量庞大、跨各类学科领域的。他们之间的信息传递与共享的低效率是制约项目进度、消耗项目的一大因素。信息化与数字化为我们提供了解决此问题的空间, 而目前出现的技术便是建筑信息模型(BIM, Building Information Model)。

理论中的 BIM 应能够贯穿建设项目的整个生命周期, 并支持项目全部信息的储存与交互。然而仍需要一个以实际技术搭建的系统将 BIM 的这些特性实现, BIM 服务器应运而生。BIM 服务器使用

服务器作为 BIM 信息的存储载体, 通过服务器与客户端之间的通信完成 BIM 信息的交互。

根据客户端的形式不同, 网络结构模式可分为 C/S 架构与 B/S 架构。其中, C/S 架构的客户端为桌面应用程序, 而 B/S 架构的客户端为浏览器。B/S 架构的处理过程与结果呈现之间的分离更为明晰, 主要运算集中在服务器中, 因而相对于 C/S 架构, 其对网络前端的软硬件要求更低。并且 B/S 架构的系统更新仅仅发生在服务器端, 因而维护与升级更便利、灵活^[2]。而随着移动通信与移动终端技术的进步^[3], 针对移动终端的浏览器也日趋成熟并得到普及。因而 B/S 架构对于移动平台的兼容性也成为了其优势之一。

目前, BIM 服务器的主要形式为 C/S 架构, 很少为 B/S 架构。这主要是 Web 环境的功能限制所造成的, 而其中最关键的便是基于浏览器的三维图形加速技术。

BIM 必须包含建筑模型的三维几何信息, 合格的 BIM 浏览应用程序必须拥有一个支持三维交互显示的图形平台。90 年代初期, 基于 Web 的三维图形交互技术已经出现, 但由于效果不佳, 故它们的

【基金项目】 国家自然科学基金(No. 51278274, No. 51008168), 清华大学自主科研计划(No. 2011THZ03)。

【作者简介】 张建平(1953-), 女, 教授, 博士生导师。主要从事土木工程 CAD/CAE、4D-CAD、BIM、建设领域信息化、数字减灾及智能决策技术等方面的研究。

应用受到了较大限制。虽然之后针对硬件加速进行了一系列技术开发,但始终存在着种类繁多、环境依赖、可操作空间小等问题。随着几年前 HTML5 的发布,其中原生的 WebGL 技术能够有效地解决这些问题,为基于 Web 的三维图形渲染带来福音。然而 HTML5 带来的并不仅仅如此,它的出现为 Web 浏览器带来了各式各样的功能支持,引领我们走向内嵌式富互联网应用的新纪元。于是,面向 Web 的 BIM 服务器有了理想的技术支撑。而本文研究的内容即为实现面向 Web 的 BIM 服务器的一些关键性技术。

2 相关研究

2.1 BIM 服务器研究

目前,国外的 BIM 服务器平台主要有 IFC Model Server, EDM Model Server, BIM Server, Eurostep Model Server 以及各 BIM 软件开发商自行开发的与设计软件配套的协同设计服务器^[4]。其主要特征的比较如表 1 所示。

表 1 国外主流 BIM 服务器对比

名称/类型	B/S 或 C/S 架构	面向全生命周期	是否开源	开发阶段
IFC Model Server	B/S	能	否	完成
EDM Model Server	C/S	能	否	完成
BIM Server	均有	能	是	开发中
Eurostep Model Server	B/S	能	否	完成
BIM 协同设计服务器	基本为 C/S	否	否	完成

国内,张建平提出了基于 IFC 的 BIM 数据平台的基本架构,并应用于 4D 管理系统等成果中,并且提出了面向全生命周期的集成 BIM 平台框架以及基于此框架的 BIMDISP 系统^[4]。此系统基于 C/S 架构。

目前,较大部分 BIM 服务器基于 C/S 架构,而基于 B/S 架构的服务器技术在功能方面并不完善,并且也没有非常成熟的公开的服务器技术可供使用。因而,在这个方向上具有明显的开发价值与意义。

2.2 基于 WebGL 的 BIM 模型显示研究

目前,支持 IFC 的基于 WebGL 的平台有 BIM-Surfer、IfcWebViewer^[5,6]。

其中,BIMSurfer 是 BIMServer 的相关开源模型

浏览平台,基于 WebGL 中的 SceneJS 框架,能够实现对项目模型简单的交互操作以及渲染选项的简单调整。同时,对 BIMServer 项目原生支持,并能够读取本地的 JSON 格式模型文件。支持子模型的精确筛选以及属性查询功能。

IfcWebViewer 是 IFC 模型的浏览平台,能够完成模型的显示与简单交互操作,并有颜色修改、基于类型的子模型筛分等简单的模型辅助识别功能。

同时,也有其他基于 WebGL 显示 IFC 格式模型的相关研究^[7]。该课题成果表现为一款 Web 应用程序,有较好的开发前景。但目前的效果、效率和功能方面均存在较大问题。

目前也有基于 WebGL 显示的 BIM 平台研究^[8],但研究关注点是 BIM 信息的集成处理,对于平台的交互性能没有深入。而 WebGL 技术仅作模型显示使用,并没有进一步的数据交换与跨领域的信息显示。

综上,目前而言,已有的基于 WebGL 的 BIM 显示平台已经能够较好显示模型数据,但也有其局限性:

- (1) 完全基于 IFC,对于模型信息表达方式并没有进行针对性的效率优化;
- (2) 仅支持显示建筑信息,并不能同时显示模型结构信息;
- (3) 并不存在 Web 客户端与服务器之间的模型信息交互;
- (4) 功能开发集中在显示上,并不注重网络传输的效率问题;
- (5) 模型真实感较差,无材质、效果渲染等。

3 BIM 的模型信息表达

3.1 模型信息内容

OmniClass 是一个字典规范,它所涵盖的数据内容贯穿了整个建设环境,包括了从宏观的项目整体结构至微观的构件和材料的众多信息。这些数据跨建设、工业、经济、住宅等领域,形成了包括动作、人员、工具以及在项目设计、施工、运维等阶段产生并使用的所有信息的一系列表^[9]。这些表中的内容理论上均应包含在 BIM 之内,但为了能够在 B/S 架构下实现信息的传输,需要进行一定的筛选,结果如表 2 所示,主要包括建筑单元、空间划分、材料和属性。

表 2 据 OmniClass 表筛选得到的 BIM 信息范围

编号	名称	内容	是否需要
OmniClassTM Table 13	Space by function	按功能划分的空间	是
OmniClassTM Table 14	Space by form	形态划分的空间	是
OmniClassTM Table 21	Elements	建筑单元	是
OmniClassTM Table 41	Materials	材料	是
OmniClassTM Table 49	Properties	属性	是

3.2 模型信息表达

作为 BIM 的通用数据信息表达方式,IFC 对于 BIM 中模型信息的内容细节与存储结构有很大的参考价值。但由于 IFC 中覆盖范围较广,而且其中的表现形式可能并非适合于 B/S 架构下的数据传输与 Web 客户端显示,因此在确定模型信息范围的基础上,需要对 IFC 整体内容进行简化与调整。该过程包括基本属性筛选、根本关系分析、表达方式简化。

(1) 基本属性筛选

基于已确定模型信息范围,需要对所有相关 IFC 类的基本属性进行一定的筛选。主要的 IFC 类包括 IfcProject、IfcSite、IfcBuilding、IfcStorey、IfcBuildingElement、IfcPropertySet 与 IfcQuantity 等。

(2) 根本关系分析

基于 IFC 的根本关系分析主要包括的信息内容有空间结构、属性集与单元量。其中,空间结构为树状结构,根节点为项目节点,叶节点为建筑单元节点,之间的所有层级均为空间划分,而各层级之间的关系为从属关系。属性集亦为树状结构,其中叶节点为简单属性、根节点为属性集,而其他节点均为复杂属性。单元量为特殊的属性集,其中包含的是并列的量属性。

(3) 表达方式简化

考虑到 Web 环境以及 Web 客户端的性能有限,需要将 IFC 中的表达方式进行一定的简化。根本关系分析是一个方面,而抽象类枚举化和三维表现统一化是最主要的两个简化方式。

在 IFC 中,有众多的抽象类,而其中一部分衍生而出的子类的属性几乎没有差别,例如 IfcBuildingElement。若将其所有子类均进行定义,则无论是传输还是 Web 客户端的处理,均效率较低。因此,需要将部分抽象类枚举化,即在抽象类中加入类别属性判断其究竟为何种子类。

在三维表现方面,IFC 中定义了多种表达方式,其主要目的是支持对模型形状的修改。而在 Web 客户端的模型显示并不需要该功能,能够进行三维的模型浏览即可。因此,为了降低 Web 客户端解析显示与服务器端生成的复杂度、提高效率,将三维表现的表达方式全部简化为三角面片的方式。

在结构信息方面,IFC 中的建筑结构信息的相关表达方式较为复杂,而 Web 客户端对结构信息的需求非常简单,仅包括梁、板、柱、约束、荷载信息,因而结构信息表达方式的确定仅基于该范围。

3.3 数据接口定义

JSON 格式的轻量性、文本性、语言的非依赖性、便携性与结构性令其成为了理想的信息交换格式^[10]。而 Web 客户端的 JavaScript 脚本对 JSON 有着原生支持,因而用于服务器端与 Web 客户端模型数据传输的数据接口建议采用 JSON 格式。

基于 JSON 格式,数据接口的定义主要使用的数据结构是对象、对象数组、数值数组,基本的数值类型包括数字、字符串与布尔值。在接口定义过程中,这些数据结构或单值之间的相互关系如图 1 所示。

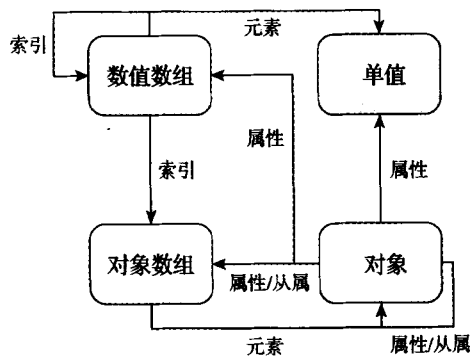


图 1 基于 JSON 的数据接口中主要数据结构以及相互关系

4 基于 WebGL 的 BIM 模型显示

4.1 三维模型显示

Web 客户端的三维模型显示基于 HTML5 标准中的 WebGL 技术,该技术为 OpenGL ES 在 Web 环境下的使用提供了相应接口^[11]。为了更为快速、高效地实现基于 WebGL 的图形三维浏览,可使用 WebGL 的上层框架。目前较为成熟的 WebGL 框架有 Three.js、PhiloGL、GLGE、SceneJS、C3DL 等。

Three.js 的渲染结构如图 2 所示,基本要素为场景、相机和渲染器。三维对象被放置在场景中,而将场景与相机载入渲染器的渲染方法便可进行三维渲染。

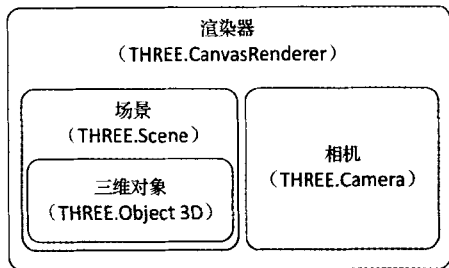


图 2 基于 JSON 的数据接口中主要数据结构以及相互关系

一个三维显示对象的基本信息包括几何与材质,在数据接口中均有相关定义。这些信息在 Web 客户端被读取之后,将被载入新生成的 Three.js 三维对象中。之后将三维对象添加进入场景,等待其被渲染器渲染。

4.2 三维动态交互

三维动态交互是基于鼠标事件、动画循环实现的。JavaScript 支持各类鼠标事件,并在事件参数中能够提供鼠标位移,而动画循环可通过 requestAnimationFrame() 方法实现。它是 JavaScript 专门为动画显示提供的框架,通过 requestAnimationFrame() 方法,能够令某一方法进行持续循环,并且由于采用了一定的监听机制,从而令该框架的资源占用较低、效率较高。

最基本的三维动态交互包括转动、平移与缩放,它们的实现通过对相机的参数操作完成。其中,转动的过程即将相机绕过目标点的两个轴旋转相应的鼠标位移,两轴的方向分别为相机平面的两个坐标轴方向;平移即将相机目标点与自身坐标均

平移鼠标位移量,而平移量可按相机平面内横纵坐标分解考虑;缩放与相机的视图类型相关,当相机视图为正交视图时,缩放通过改变相机的视口大小实现,而当相机的视图为透视视图时,缩放通过将相机在其与目标点连线所在的直线上移动实现。

4.3 真实感显示

基于 OpenGL 的真实感显示包括光学真实感、材质真实感^[12]。其中,光学真实感主要包括法向量的定义以及光照模型的选择。而材质真实感主要包括材料光学属性、纹理贴图。而基于 WebGL 的真实感显示同样如此。

(1) 法向量

通过法向量能够计算光源反射的方向,从而得到合理的光学效果。法向量包括面法向量与顶点法向量。其中面法向量用于计算各个面片的明暗效果;顶点法向量用于计算各面片交界处的明暗效果,使相邻面片的色彩在边界上有平滑的过渡,从而能够对曲面进行光学近似。在 Three.js 中,面法向量可以自行计算,而顶点法向量能够直接赋值。

(2) 光照模型

光照模型的选择是影响模型的光学渲染效果的决定性因素。无光照模型,则三维形体的材质无明显变化,仅有一种不变的颜色。较为经典的光照模型有 Lambert 模型与 Phong 模型。其中,Lambert 模型仅包含材质的漫反射,适合于表现粗糙的木材、布匹等漫反射材质;而 Phong 模型中加入了高光系数与镜面反射,适合于表现金属、玻璃等高光材质。Three.js 中三维面片形体的材质对象按光照模型被划分为基本材质、Lambert 材质与 Phong 材质等,可直接生成使用。

(3) 材料光学属性

材料的光学属性包括透明度、环境光颜色、漫射光颜色、反射光颜色、辐射光颜色、镜面指数等。不同的光照模型,均需要一部分光学属性作为计算量,因而材料的光学属性是影响材质真实感的根本因素。

(4) 纹理贴图

材质的纹理需要通过纹理贴图的方式实现。每个平面多边形的材质贴图需要定义图片,并定义各点对应的贴图坐标。从而,渲染时便能够将图片中贴图坐标点围成的区域映射至平面多边形上。在 Three.js 中,贴图的资源被定义于材质对象中,而贴图坐标以及索引被定义于几何对象中。

5 面向 Web 的 BIM 信息管理

5.1 Web 客户端模型修改

为实现 Web 端对服务器端模型数据的修改功能,在前端定义双模型对象。其中之一为原始的未修改模型,另一个为修改后模型。当用户确定提交修改时,将根据上述两个对象生成包括所有修改属性的对象。在修改属性对象中仅包括已修改的属性信息与必要的关联结构。具体的模型修改流程如图 3 所示。

使用基于 HTTP 协议的 POST 方式将修改属性对象提交至服务器中特定方法后,由服务器进行反序列化。但由于该对象结构不确定,因而并不能够使用普通的基于已定义类的反序列化方式,需要自定义 JSON 转换器将其转换为由名值对和数组构成的动态对象。之后,在服务器端即可根据该对象对数据库进行相应修改。

5.2 Web 传输压缩

目前文本压缩算法有很多,最为基本的包括有霍夫曼编码、算数编码、游程编码以及以 LZ77 为代表的字典编码等^[13]。其中,字典编码方式以其优异的性能成为了划时代的文本压缩方式。在这些早期压缩

技术的基础上,或发展或组合,形成了一系列的应用性压缩技术,其中便包括 DEFLATE。DEFLATE 组合自 LZ77 与霍夫曼编码,被应用于 ZIP、GZIP 与 PNG 格式图片中。其中的 GZIP 是目前 Web 数据传输中最有效的数据压缩方式,大部分网页的 GZIP 压缩率均超过 60%。IIS、Tomcat 和 Apache 等服务器环境均支持开启 GZIP 压缩,配置非常便捷。

目前已有的各类压缩算法的应用前提是在文本压缩之前并不知晓文本信息。但由于数据接口有明确定义,因而为固定字典的针对性压缩提供了可能。顾名思义,固定字典即意味着压缩字典可以存储在任意位置,例如分别储存在服务器端与 Web 页面脚本中。因此,字典中内容原本的多次传输将直接变为单次传输(在页面生命周期内),从而达到提高信息传输效率的目的。面向 Web 的固定字典压缩原理如图 4。

固定字典内容可基于已有的数据接口定义确定,其中所有的既定名称是理想的固定字典条目,因为对象名在数据接口中很可能重复出现且占较大的存储空间。为此,首先需要获取接口文件中所有名称列表,剔除重复项以及长度较短项后形成固定字典。

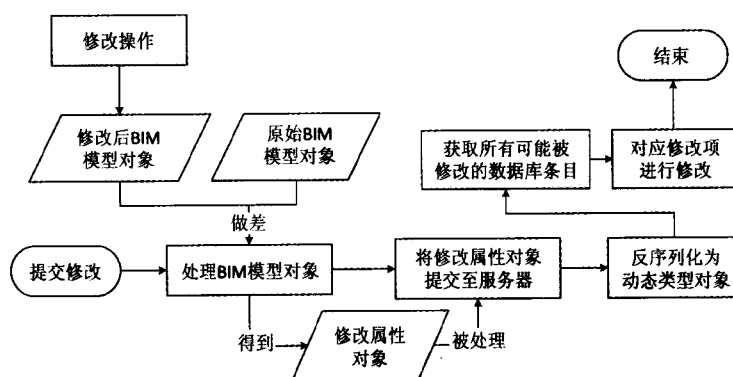


图 3 Web 端对模型数据修改流程

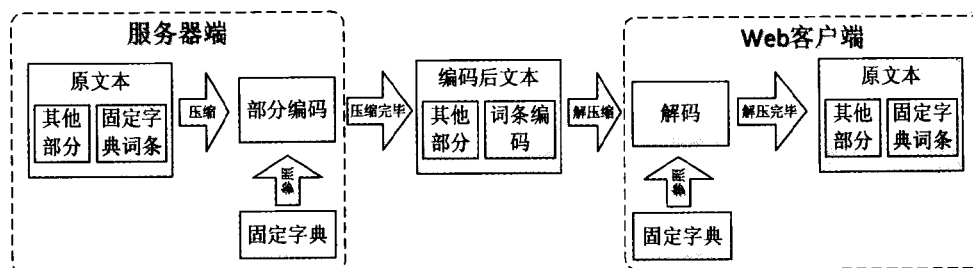


图 4 面向 Web 的固定字典压缩

通过先针对性压缩后 GZIP 的方式,能够显著压缩传输的 JSON 数据。本文对该方式进行了测试,得到的结果如表 3。从而可见其压缩效果显著。

表 3 压缩算法测试数据大小

原数据	针对性压缩	GZIP 压缩	针对性压缩后进行 GZIP 压缩
307 KB	211 KB(68.7%)	49 KB(16.0%)	30KB(9.8%)
10147 KB	7551 KB(74.4%)	1641 KB(16.2%)	1559 KB(15.4%)

5.3 实时交互

服务器与 Web 客户端之间基本的信息传输方式是基于 HTTP 协议的,最原始的数据交互方式为 Web 客户端发起请求,之后服务器立即处理、传输信息。在这种模式下,服务器并无信息发送的主动权,因而并不能够实时地向 Web 客户端发送数据信息,从而实时交互无从说起。为了实现服务器的主动信息传输而诞生的技术被称为服务器推送技术,传统的服务器推送技术基于 HTTP 协议,包括 long-

polling、iframe 等方式。

随着 HTML5 的发布,基于全新协议的 WebSocket 技术为服务器推送带来了更加专业的解决方案。WebSocket 通过服务器与客户端的握手便可建立二者之间的双向通信通道,从而实现数据的相互传输。整个过程如图 5 所示。与 HTTP 协议冗长的信息头不同,WebSocket 中数据相互传输的信息头很小,占用资源更少。

6 系统应用

实例模型采用一栋约 42 层高层建筑的主体结构部分,为框架核心筒结构。其中包括约 6000 个建筑单元。该模型数据信息储存在数据库中,在将其转换为 JSON 接口对象并序列化后,不进行压缩的数据大小为 9.90MB。经测试,模型在与 JSON 的形式传输至 Web 前端,页面加载模型信息共花费约 12 秒时间。模型在系统的 Web 客户端页面中的显示如图 6 所示。

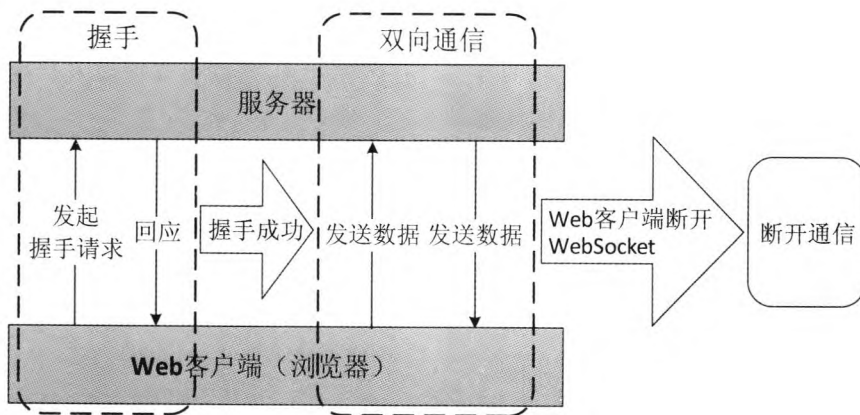


图 5 WebSocket 通信步骤

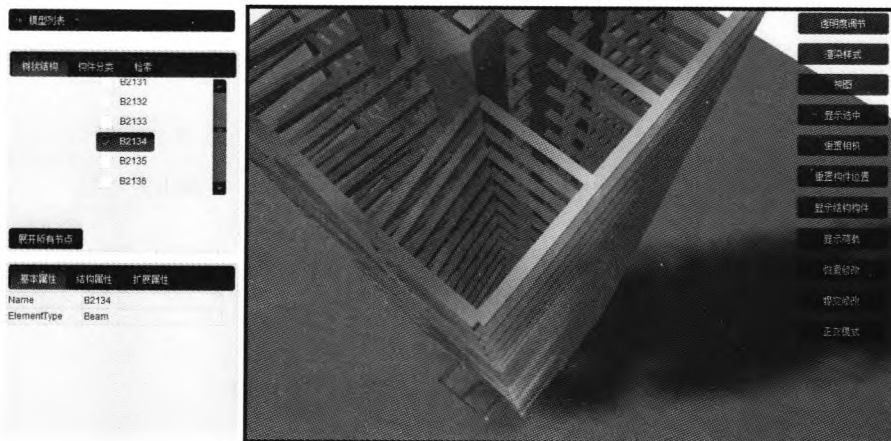


图 6 Web 客户端页面

系统支持模型选择、模型信息查看、模型三维浏览与交互、模型信息管理等方面的功能。功能丰富,并且应用表明这些功能的完成均较为流畅。应用表明,该系统已经满足一定的 BIM 服务器需求。

7 结论

面向 Web 的 BIM 服务器需要针对 Web 环境解决三方面问题:Web 客户端使用的 BIM 模型数据接口;能够在浏览器中运行的图形平台;实现服务器与 Web 客户端之间的实时交互。本文研究了 OmniClass、IFC 等内容,对其进行了简化与改变,形成了可实际应用于 Web 客户端的 BIM 模型数据接口。在该数据接口的基础上,基于 WebGL 技术在保证环境兼容性的基础上实现了 Web 客户端中的三维模型浏览与动态交互。而在实现了客户端对服务器端模型信息修改的同时,基于 WebSocket 技术实现了修改信息的实时提醒。成果系统的顺利应用与其丰富的功能支持表明,目前基于 HTML5 标准的 Web 技术已经能够较好地支持基于 B/S 架构的 BIM 服务器。该系统具有良好的发展前景与应用优势。

参考文献

- [1] 明锦郎,王改政. “甩掉图板”后的建筑设计[J]. 建筑学报, 1996(12):34-36.
[2] 邱方亮. 基于 B/S 的档案资料管理信息系统的研究和

构建[D]. 西北工业大学, 2004.

- [3] 官酩杰. 基于 OpenGL ES 的移动平台图形渲染引擎研究与实现[D]. 北京交通大学, 2010.
[4] 张建平, 余芳强, 李丁. 面向建筑全生命期的集成 BIM 建模技术研究[J]. 土木建筑工程信息技术, 2012,4(1):6-14.
[5] BIMSURFER[EB/OL]. <http://bimsurfer.org/>.
[6] IfcWebViewer[EB/OL]. <http://www.ifcwebserver.org/sgl>.
[7] Ferreira N B T. A WebGL application based on BIM IFC [D]. Porto: Universidade Fernando Pessoa, 2012.
[8] LIN T. Cloud BIM: A Web-Based BIM System with Application of Cloud Computing and WebGL[D]. Taiwan: National Taiwan University of Science and Technology, 2012.
[9] OmniClass™ Introduction and User's Guide [M/OL]. http://www.omniclass.org/tables/OmniClass_Main_Intro_2006-03-28.pdf.
[10] Crockford D. The application/json Media Type for JavaScript Object Notation (JSON) [EB/OL]. <http://tools.ietf.org/html/rfc4627>.
[11] Marrin C. WebGL Specification Version 1.0 [EB/OL]. <http://www.khronos.org/registry/webgl/specs/1.0/>.
[12] 张建平, 余芳强, 吴大鹏. 基于 OpenGL 的建筑施工虚拟仿真平台的研究与开发[J]. 土木建筑工程信息技术, 2009,1(2):55-62.
[13] 方世强, 李远清, 胡刚. 文本压缩技术综述[J]. 工业工程, 2002(02):15-18.

Web-oriented BIM 3D Viewing and Information Management

Wang Hengwei, Hu Zhenzhong, Lin Jiarui, Zhang Jianping

(Department of Civil Engineering, Tsinghua University, Beijing 100084, China)

Abstract: BIM Server provides a solution for BIM to support construction life cycle. Currently, the structure of BIM Server is C/S, but gradually B/S also shows its advantages. The release of the HTML5 standard brings the WebGL technology which supports hardware graphics acceleration in browser and is one of a series of revolutionary Web technologies. Thereby the foundation is laid for achieving Web-oriented BIM information management and 3D display of BIM model. In this paper, the model information scope and its expression of Web-oriented BIM is analyzed, and the corresponding information management technology and the BIM model display based on the Three.js, which is a WebGL framework, are studied. Then, a WebGL-based platform for 3D display and information management of BIM is also developed. Application shows that the graphics platform of the platform performs stable with nice rendering for large BIM. Meanwhile the platform provides rich feature supports for Web client by favorable interactive functions, showing a great application value.

Key Words: Building Information Modeling (BIM); WebGL; HTML5; B/S; Three.js

面向Web的BIM三维浏览与信息管理

作者: [王珩玮](#), [胡振中](#), [林佳瑞](#), [张建平](#), [Wang Hengwei](#), [Hu Zhenzhong](#), [Lin Jiarui](#), [Zhang Jianping](#)

作者单位: [清华大学土木工程系, 北京, 100084](#)

刊名: [土木工程信息技术](#)

英文刊名: [Journal of Information Technology in Civil Engineering and Architecture](#)

年, 卷(期): 2013, 5(3)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_tmjzgcxxjs201303001.aspx