

# BIM 引擎及其应用研究

张金辉<sup>1,2</sup> 张其林<sup>1,2,3</sup> 常治国<sup>1,2</sup>

- (1. 上海同磊土木工程技术有限公司, 上海 200433;
2. 上海土木工程结构健康监测工程技术研究中心, 上海 200092;
3. 同济大学 土木工程学院, 上海 200092)

**【摘要】** BIM 难以在全过程管理中得到应用。本文分析探讨了常见的原因, 而后在 IFC 模型服务器的基础上, 进一步改进优化, 构建了可嵌入其它常用管理系统使用的 BIM 引擎, 并在实践中与装配式施工管理系统和 OA 流程管理系统作了结合, 实现了基于 BIM 的模块化建造信息系统, 较好地解决了 BIM 信息系统常见的问题。

**【关键词】** BIM; IFC 模型服务器; 轻量化; JSON; 数据库; 版本迭代

**【中图分类号】** TU17; TP311.5 **【文献标识码】** A

**【版权声明】** 本文被《土木建筑工程信息技术》、中国知网重要会议论文全文数据库(CPCD)收录上网, 未经授权严禁登载。

## 引言

目前在土木工程领域, Revit Server、CATIA、以及国内的 PKPM BIM<sup>[1]</sup> 等已经可以实现设计的协作, 虽然在设计管理<sup>[2]</sup> 和信息技术方面还有提升和优化的空间。但后续的施工阶段, 乃至运营阶段, 尤其在管理中应用 BIM, 还是有不少的困难, 或者说是有其特定的需求尚未得到满足, 妨碍着 BIM 的应用。总的来看, 是建筑行业本身的复杂性造成的, 比如不同的项目情况有各种差异, 周期长、不可预见问题多, 工作界限不恒定, 分工时有变化, 有些环节要权宜处理等; 而相比的工厂生产线生产, 则其步骤、环节、要求、标准等会相当清晰和一致。进一步分析 BIM, 可以梳理出以下几方面的问题或需求:

(1) 管理工作总是充满调整、变化, 建筑行业尤甚, 因此管理信息系统很多功能被设计成可配置式的, 以适应调整变化; 为降低发布调试的复杂度, 通常还会采用 Browser/Server 的架构。当前 BIM 的图形桌面端“重型”应用难以适应, 即便可二次开发也

会严重依赖原平台而迁移不易, 如 BIM-QR<sup>[3]</sup>。

(2) 管理要把各种细节信息组织成为有机的整体, 才能有效地工作。因此管理信息需要与模型的分部、分项、构件、乃至零件等细部建立相互关系, 并令之随管理信息变动而有序变化。而管理信息自身又有复杂的相互关系和结构特点, 因而不适合将管理信息分散存储到模型数据结构中去, 只适合将管理信息与模型及其各细部作信息的关联, 而后根据管理需要交互动。

(3) 建筑工程大多是特例, 不是批量标准产品, 没有技术冻结概念<sup>[4]</sup>, 进入施工后还会根据实际情况作调整修改, 因此模型在施工过程中还会有版本迭代, 并动态地与管理信息匹配关联。另外模型版本迭代的频次通常远小于管理信息的变化, 适合把模型缓存在客户端, 只在版本迭代时作局部更新, 减少模型显示操作时的数据通信, 提高操作响应性能。

(4) 实际管理中, 大都是对局部信息的操作, 相应的 BIM 只需要作局部的显示联动, 因此很适合采

**【基金项目】** 国家重点研发计划(National Key R&D Program of China)“乡村住宅装配式快速建造体系与被动式节能集成研究”(编号:2018YFD1100903)

**【作者简介】** 张金辉(1974 -), 男, 硕士, 高级工程师, 主要研究方向: 建筑信息模型 BIM、建筑物联网; 张其林(1962 -), 男, 博士, 教授, 主要研究方向: 空间结构、建筑信息模型 BIM、结构健康监测等; 常治国(1977 -), 男, 博士, 高级工程师, 主要研究方向: 建筑信息模型 BIM。

用局部策略做轻量化<sup>[4]</sup>,其它如遮挡剔除和视域剔除<sup>[5]</sup>等可同时应用,不过模型显示的时候,需要用合适的方式,体现出该局部与整体的关系,以方便用户理解。

(5)第一线的管理,考虑运作便利,会引入各种操作终端设备:扫描枪、手持机、平板电脑、触摸屏机等,有 Windows、Android 等多种操作系统平台,还有些设备硬件资源有限、性能弱。这些因素要求 BIM 须有足够的可移植性和可轻量化的潜能。

当前在 BIM 系统研究中出现了两点新的技术趋势:其一是 JSON 格式运用,由于 JSON 格式在云计算中的广泛应用和高效性能<sup>[6]</sup>, BuildingSMART 以及有关团队正在酝酿 IFCJSON 标准<sup>[7-10]</sup>。其二是数据库运用,以数据库为基础存储和管理 BIM 模型,来展开 BIM 的应用工作。基于数据库的 BIM 同样可以遵循 IFC 的标准<sup>[11-15]</sup>,同时 IFC 的格式文本作为序列化的模型信息,在数据的备份、传递、协同等方面发挥关键的桥梁作用。实现了这样功能的应用服务程序被定义为 IFC 模型服务器(IFC model server)<sup>[16]</sup>,目前较为成熟的有: IMSvr、EMS、BIM-Server 等等。

本文在 JSON 格式文本和层次型数据库基础上,进一步构想更为合理、有效的模型服务器及其运用方法,并进行了开发实践,以期觅得更为轻量、便捷、实用的 BIM 应用技术路线。

### 1 技术构想

针对上述问题和需求,总的构想是将 BIM 的部分摘离出来,包括服务端程序和客户端程序,构建独立运作的可嵌入其它程序(比如 OA)共同工作的程序,其机制类似于微信公众号,当然也可独立工作(比如更新或查看模型),不妨称之为 BIM 引擎(BIM Engine)。BIM 引擎只负责 BIM 部分的功能,

包括服务端模型的存取、计算、以及版本迭代等等,以及客户端对模型的操控,比如:旋转、选取、消隐、变色等等,这部分功能相对独立稳定,受管理功能变化的影响小,当不够满足管理功能需要时,可以扩充 BIM 引擎功能,升级 BIM 引擎版本。BIM 引擎及服务的软件系统整体结构如图 1 BIM 引擎与其服务之软件系统的整体结构,图中中心虚线的左侧是 BIM 引擎,右侧是其服务的软件系统,上部是各自的服务端软件部分,下部是各自的客户端软件部分。

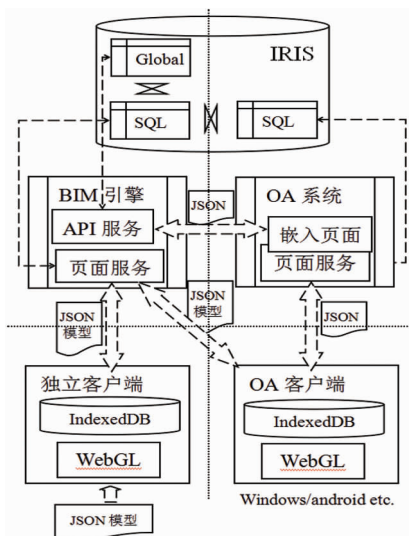


图 1 BIM 引擎与其服务之软件系统的整体结构

BIM 引擎主要由以下几部分组成:

(1)BIM 引擎服务端数据库,考虑到模型数据和与管理配合数据的存储及操作,数据库选用了兼容层次型、对像型和关系型 SQL 的 IRIS 数据库。层次型的数据存储适用于模型数据,如图 2 层次型数据库构件数据结构与 JSON 构件数据结构对比所示;关系型 SQL 的数据存储适用于与管理配合数据,便于采用 SQL 检索和查找,比如图 3 与管理配合的构件数据结构中用 componentId 指向了图 2 中

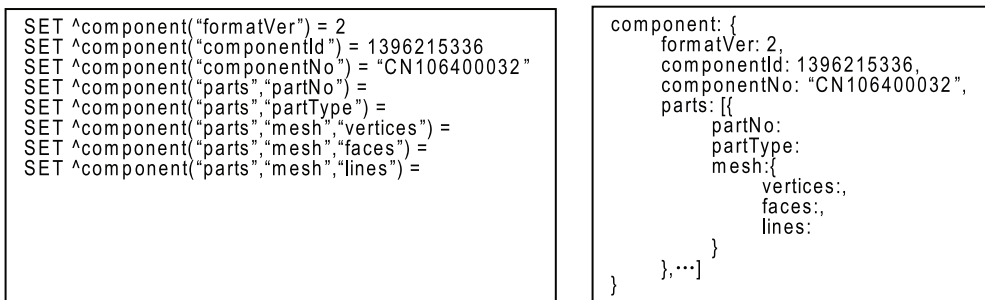


图 2 层次型数据库数据结构与 JSON 数据结构对比

模型数据,并列出多项重要字段用于查询。与管理配合数据和其服务的软件系统的管理数据需要关联查询,因此管理数据也被配置在 IRIS 数据库中。

若条件不允许,管理数据需要配置在另外的关系型数据库中,则与管理配合的数据应跟随被配置在另外的关系型数据库中,模型数据可配置在 IRIS 或其它 NoSQL 数据库中,如 MongoDB、HBase<sup>[13-14]</sup>等,只是这样分开存储、管理、查询会损失模型操作效率和系统整体性能。

TableName		bizprojectcomponent				biz
SizeInformation		Size			InitRows	
SEQ	COLUMN	TYPE	LEN	PK	NOTNUL	DESC
1	componentId	BIGINT	16	○	○	构件ID (unique)
2	projectNo	VARCHAR	32		○	项目编号
3	buildingNo	VARCHAR	16		○	建筑编号
4	componentNo	INTEGER	10		○	构件编号
5	componentVersion	INTEGER	10		○	构件版本
6	delflg	INTEGER	1		○	删除标记

图3 与管理配合的构件数据结构

(2) BIM 引擎服务端的功能与 IFC 模型服务器类似,根据需要将模型分解为分部、分项、构件、乃至零件等细部保存在数据库中,并实现模型的增、删、改、查询、版本迭代、及特定的操作组合等。其中版本迭代采用分层次版本迭代的方式<sup>[4]</sup>,简单说就是整体版本由局部版本组合而得,因此版本更新时只需要更新有新版本的局部即可;特定的操作组合有很多种,比如分解模型到最基础的原材料作材料用量统计,还有根据进度各时间段来统计已完成内容的模型等。

(3) BIM 引擎中实现模型的操控功能,主要是客户端用的嵌入式页面,页面内模型可翻转、漫游、选择、过滤、以及模型版本更新等等。这些在 BIM 引擎服务端被封装进 API 内,供其服务的软件系统(通常是可配置或二次开发的施工管理或 OA 系统)访问调用,BIM 引擎自身独立的客户端也通过调用此 API 实现功能。

(4) BIM 引擎独立客户端采用桌面和页面混合的模式,也就是在桌面客户端里嵌入浏览器,桌面客户端在后台实现一些浏览器没有的功能供浏览器调用,浏览器访问 API 从服务端获得嵌入式页面,嵌入式页面采用 WebGL 技术显示模型、及操控模

型,模型的显示与操控主要采用局部策略来轻量化<sup>[4]</sup>。

具体的轻量化做法是将所需版本的完整模型,保存在本地的数据库 IndexedDB 里,只把工作中用到的模型部分加载入内存,显示与操控,其它部分不加载入内存,或只加载显示粗略的外形(低精细度 LOD),因此即便是大模型也可在较少硬件资源的设备上显示操控。同时 IndexedDB 内模型有版本信息,根据业务逻辑的需要来决定,是否从服务器获取模型新版本作更新,大多数情况下模型无需更新,因此一般的页面切换不需要传输大量的模型数据,提高了响应性能。

另被 BIM 引擎服务的软件系统,其客户端也转用桌面和页面混合的模式,还可很方便地扩展到其它的平台,如:Android、iOS、或某些定制的终端等。

(5) 采用 JSON 格式传递模型数据,包括服务端客户端之间 HTTP 协议通信中的数据流,或上传下载的一类 IFC 文件。JSON 格式数据体积小、紧凑,JavaScript 对其转换快、效率高,尤为适合页面使用。但有个缺点,需要把整个闭包对象加载到内存,才能解析转换,目前整个闭包通常是一个文件或一个响应流,内存负荷较大。

考虑到在较少硬件资源的设备上显示操控,也采用局部策略,将一个文件或一个响应流的大闭包,拆分成每行一个小闭包,第一行是总的闭包,后面是分闭包(需要时还可多级再拆分),见图 4 多闭包 JSON。解析程序读一行,解析一行,缓存进数据库一行,大大降低了内存负荷。

综上,本构想的优势来自于将 BIM 部分剥离,独立存在,专注于 BIM 的功能,减少了易变化的管理功能对其的干扰。再通过数据库技术、JSON 格式、轻量化策略等提高了 BIM 性能,尤其是低配置硬件上的运行性能,有较为广泛地嵌入和移植空间。当与管理系统嵌入融合后,模型与管理信息交互互动,直观明了,大大降低了用户的专业要求,不必具备高深的专业想象能力,去补充文字和二维图纸难以表达的细节<sup>[4]</sup>。

```
{modelId: 11354, modelName: "21st Century Tower", componentIds: [1396215336, 1396215478], ...}
{componentId: 1396215336, componentNo: "CN106400032", parts:...}
{componentId: 1396215337, componentNo: "CN106400035", parts:...}
{componentId: 1396215338, componentNo: "CN106400041", parts:...}
```

图4 多闭包 JSON

## 2 系统开发与实践

根据以上构想的技术路线,开发 BIM 引擎,配合某公司的装配式施工管理系统、OA 流程管理系统,实现基于 BIM 的模块化建造信息系统。

装配式施工管理系统有三个主要功能模块:材料成本管理、进度计划追踪和质量检查控制。材料成本管理模块访问 BIM 引擎服务端,调取模型最基础的原材料用量统计,以此为基础控制采购的节奏;进度计划追踪模块调取模型交互的嵌入式页面,与构件列表一同显示,交互操作,共同编排批次,如图 5 是构件批次编排;质量检查控制模块中的任务安排与进度计划追踪模块中的批次编排类似,但质检审批部分涉及流程管理,做法上有所不同。

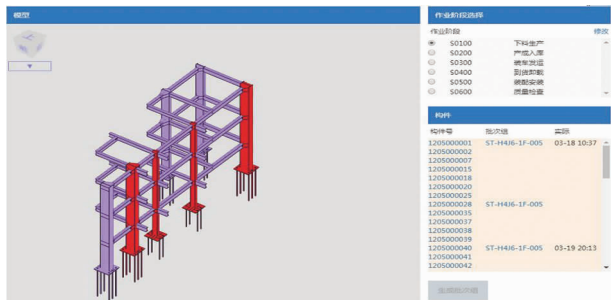


图 5 构件批次编排

OA 流程管理是管理中最易变化的部分,套用现有成熟的流程引擎来管理,同时引用 BIM 引擎的模型嵌入式页面,将质检部分着重显示,方便用户的理解 and 操作。比如图 6(a) 质量检查 OA 审批中,按[查看]按钮获得图 6(b) 质量检查 OA 审批详细,查看嵌入的模型和相关的质检数据。

从图 5 和图 6(b) 中,也可看出只有模型和管理信息有机结合、关联互动才能方便用户理解和操作。

当服务器端的模型有版本更新时,会提醒客户端及时更新模型,见图 7 模型更新提示。更新时,获取 JSON 格式的类 IFC 数据流,内含按版本控制需要更新的模型部分。更新后,删去的构件会在关联管理信息中自动删去,修改的构件关联管理信息不变,增加的构件需要用户去手动关联管理信息。

从图 5 中,还可明显看到模型已作轻量化处理,只显示用到的部分的模型;图 6(b) 中,梁柱都作了简化显示,突出显示节点细节。因此图 8 移动端模型可在手持式小型移动设备上轻松地显示、操控。



图 6(a) 质量检查 OA 审批

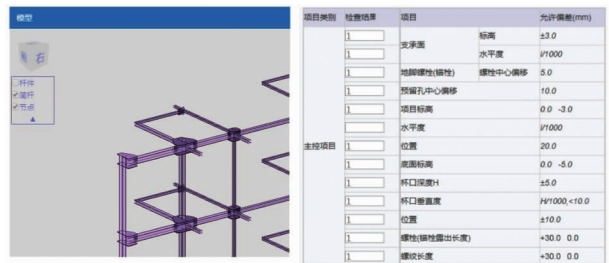


图 6(b) 质量检查 OA 审批详细



图 7 模型更新提示



图 8 移动端模型

## 3 结论与展望

本文在 IFC 模型服务器的基础上,引入了存储模型的层次型数据库、桌面和页面混用的客户端模式、本地数据库 IndexedDB、分行闭包的 JSON 格式文件等关键技术,实现了可嵌入使用的 BIM 引擎,并实践了与已有或另开发的管理系统结合使用的技术路线,能较好地解决目前 BIM 信息系统的常见问题。将来经过更多的工程使用实践,逐步优化改

进,有望走向成熟,成为有效、实用的 BIM 信息系统。

### 参考文献

- [1] 黄立新,马恩成,张晓龙等. PKPM 的“BIM 数据中心及协同设计平台”[J]. 建筑科学, 2018, 34(9): 42-49, 129.
- [2] 焦柯,陈少伟,许志坚等. BIM 正向设计实践中若干关键技术研究[J]. 土木建筑工程信息技术, 2019, 11(5): 19-27.
- [3] 孙亮,张科龙. BIM - QR 系统在钢结构验收中的应用[J]. 土木建筑工程信息技术, 2016, 8(3): 44-47.
- [4] 张金辉,张其林,刘金典. 基于 BIM 的装配式建造信息系统研究[J]. 土木建筑工程信息技术. 2020, 12(3).
- [5] 关峰. 基于 Web3D 的建筑信息模型轻量级可视化研究[D]. 武汉:湖北工业大学, 2018.
- [6] 刘强,张建平,胡振中. 基于键\_值缓存的 IFC 模型 Web 应用技术[J]. 清华大学学报(自然科学版), 2016, 56(4): 348 - 353, 359.
- [7] Kereshmeh Afsari, Charles M. Eastman, Daniel Castro-Lancouture. JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange [J]. Automation in Construction, 77(2017): 24-51.
- [8] Thomas Gilbert, Stuart Barr, Philip James, Jeremy Morley, Qingyuan Ji. Software Systems Approach to Multi-Scale GIS-BIM Utility Infrastructure Network Integration and Resource Flow Simulation[J]. ISPRS International Journal of Geo-Information, 2018, 7, 310.
- [9] H. Butler, M. Daly, A. Doyle, S. Hagen, T. Schaub, "The GeoJSON format", internet engineering task force (IETF), August [Online]. Available: <https://tools.ietf.org/html/rfc7946> (2020. 3 可访问).
- [10] M. Lanthaler, C. Gutl, On using JSON - LD to create evolvable RESTful services, Proceedings of the Third International Workshop on REST Design, New York, NY, 2012. <https://dl.acm.org/doi/10.1145/2307819.2307827> (2020. 3 可访问).
- [11] 陆宁,马智亮. 利用面向对象数据库与关系数据库管理 IFC 数据的比较[J]. 清华大学学报(自然科学版). 2012, 53(6): 836-842.
- [12] 李犁,邓雪原. 基于 IFC 标准 BIM 数据库的构建与应用[J]. 四川建筑科学研究, 2013, 39(3): 296-301.
- [13] 余芳强,张建平,刘强,赵文忠. 基于云计算的半结构化 BIM 数据库研究[J]. 土木建筑工程信息技术, 2013, 5(6): 1-5.
- [14] 陈远,岳石花. 基于 HBase 的 BIM 模型存储技术研究[J]. 土木建筑工程信息技术, 2017, 9(4): 74-81.
- [15] 成于思,李启明,成虎. IFC 数据在关系数据库上的实现研究与应用[J]. 计算机应用与软件, 2014, 31(11): 33 - 34, 44.
- [16] Ghang Lee, Jiyong Jeong, Jongsung Won, et al. Query Performance of the IFC Model Server Using an Object-Relational Database Approach and a Traditional Relational Database Approach. Journal of Computing in Civil Engineering, 2014, 28(2): 210-222.

## Research of BIM Engine and Its Application

Zhang Jinhui<sup>1,2</sup>, Zhang Qilin<sup>1,2,3</sup>, Chang Zhiguo<sup>1,2</sup>

- (1. Shanghai Tonglei Civil Engineering Technology Co., Ltd., Shanghai 200433, China;
2. Shanghai Engineering Research Center of Structural Health Monitoring for Civil Engineering, Shanghai 200092, China;
3. College of Civil Engineering, Tongji University, Shanghai 200092, China)

**Abstract:** BIM is difficult to be applied in the management of whole course. This paper has analyzed and has discussed its common causes, then has made further improvement and optimization based on IFC model server. The BIM engine that can be embedded in other management systems is built, and is combined with assembly construction management system and OA process management system in practice, to achieve the Modular Construction Information System based on BIM. It has addressed those common problems of BIM information system in a better way.

**Key Words:** BIM; IFC Model Server; Lightweight; JSON; Database; Version Iteration