

一种数据中心三维场景 模型描述语言规范的设计和应用

Design and Application of A Three-Dimensional Scene Model
Description Language Specification for Data Center

田阿康,张 勇,崔秋然,徐玉龙,齐少安,党万川(中讯邮电咨询设计院有限公司,北京 100048)

Tian Akang,Zhang Yong,Cui Qiuran,Xu Yulong,Qi Shaoan,Dang Wanchuan(China Information Technology Designing & Consulting Institute Co.,Ltd.,Beijing 100048,China)

摘 要:

提出了一种数据中心的三维场景模型描述语言规范,旨在提高数据中心数字化平台的资源复用率,减少跨平台应用的开发工作量。首先探讨了三维模型语言的设计原理,通过构建模型库、场景模型和监控大屏,并将它们整合为一套三维场景模型语言,以实现资源的高效利用和系统兼容性的扩展。接着介绍了基于三维场景语言开发的三维平台和跨平台应用。最后,指出了三维场景语言当前的不足与潜力,为三维场景语言的进一步完善指明了方向。

关键词:

模型库;三维场景语言;数字孪生;跨平台应用

doi:10.12045/j.issn.1007-3043.2025.07.012

文章编号:1007-3043(2025)07-0070-07

中图分类号:TN919

文献标识码:A

开放科学(资源服务)标识码(OSID):



Abstract:

It proposes a three-dimensional scene model language specification for data center, aiming at enhancing the reuse rate of model resources on digital platforms within data centers and reducing the workload of cross-platform application development. It delves into the design principles of the three-dimensional model language, achieving efficient utilization of resources and extension of system compatibility by constructing a model library, rebuilding scene models, and monitoring dashboards, which are integrated into a set of three-dimensional scene language. Then it introduces a three-dimensional platform and cross-platform applications. Finally, it points out the deficiencies and potential of the three-dimensional scene language and clarifies the direction for its further improvement.

Keywords:

Model library; Three-dimensional scene model language; Digital twin; Cross-platform application

引用格式:田阿康,张勇,崔秋然,等. 一种数据中心三维场景模型描述语言规范的设计和应用[J]. 邮电设计技术,2025(7): 70-76.

0 引言

近2年,实景三维中国建设、新基建下的BIM+GIS^[1]、CIM^[2]、数字孪生、自然资源三维立体“一张图”^[3]等应用需求的出现,共同推动着三维可视化、引擎、GIS等技术不断创新发展,从而打造出多行业、新型的三维应用和生态,三维可视化也逐步在数据中心中普

及。随着技术的不断发展和应用场景的拓展,其三维引擎也不只拘泥于Three.js、Cesium等轻量化引擎,虚幻引擎(UE)和Unity3D等以渲染效果见长的游戏引擎在通信领域的市场需求量也在持续增加^[4]。

数据中心的三维可视化技术是现代数据中心管理的重要组成部分,它通过先进的可视化和3D仿真技术,提升了机房管理的智能化和数字化水平。国内大部分数据中心的三维可视化系统采用WebGL、WebGPU网页三维引擎(如Cesium、Three.js)开发^[5-6],也有

收稿日期:2025-05-06

部分使用 Unity3D 和虚幻引擎开发^[7-9]。采用网页三维引擎开发时,虽然开发和部署方便,但视觉效果不佳^[10],遇到有一定数据规模的应用时系统性能很差。采用游戏三维引擎开发时,虽然场景效果优秀,但是硬件资源消耗大,前端部署(如大屏)只能做成 C/S 方式,维护更新困难,而 B/S 网页应用只能采用低分辨率视频流云渲染模式,极占带宽,使用效果不稳定。2 种技术产品各有优缺点,很多开发商采用双引擎方式^[11-13],即大屏展示使用游戏引擎,桌面业务使用网页引擎^[14],但这样项目就需要在多平台重复开发,极不合理。

为了实现更高效的资源利用和管理,提升模型资源在不同平台上的复用率,减少平台场景搭建的业务逻辑的投入,需要制定统一的三维场景模型描述语言规范,包括数据中心可视化的模型标准、数据中心三维场景组织以及数据中心设备与模型的关联映射的定义。通过制定这些规范,可以确保不同来源和类型的模型资源能够采用统一的模型要素来组织场景,并实现该场景的跨平台应用,同时也为业务系统在未来升级更换平台和功能扩展奠定基础。

三维场景模型描述语言规范的制定将有助于推动数据中心生命周期中各阶段三维可视化技术的发展,实现更高效、更智能的机房管理,确保数据中心的稳定运行和业务连续性。

1 面向通信行业的场景模型描述语言架构

1.1 三维场景模型描述语言的应用系统架构

三维场景模型描述语言基于模型资源、数据中台

和模型管理平台产生,其中数据中台负责提供实时数据;模型资源平台和建模软件负责提供基础模型或场景的裁剪和资源供给;模型管理平台负责模型、场景、业务数据、用户自定义数据的采集管理及三维场景模型描述语言的组装。应用则通过解析还原三维场景模型描述语言来快速构建可视化场景,进行更多业务开发。基于三维场景模型描述语言的应用系统架构如图 1 所示。

1.2 三维场景模型描述语言的数据流

系统采用了一套清晰定义的数据规范体系,以确保模型数据、业务数据和用户自定义数据在系统中得到有序整合。通过设计的数据编辑和解析流程,用户能够迅速搭建起直观的可视化场景,并实现场景数据在不同平台间的无缝应用。

三维场景模型描述语言的数据流程示意如图 2 所示。三维场景模型描述语言从数据中台、模型设计软件和资源平台中获取设备实时数据和设备模型。通过在模型管理平台中编辑配置,生成包含模型、交互、大屏 UI 定义的三维场景模型描述语言并在业务平台上直接引用或解析,以减少系统的开发工作量,提高业务平台的交付速度。

2 面向通信行业的三维场景语言构建

数据中心的可视化功能主要涵盖以下几个核心业务模块:三维模型的渲染展示、大屏幕数据可视化、三维模型的交互操作以及对设备实时数据的监控。针对这些模块,提炼出以下关键数据,并为其制定规

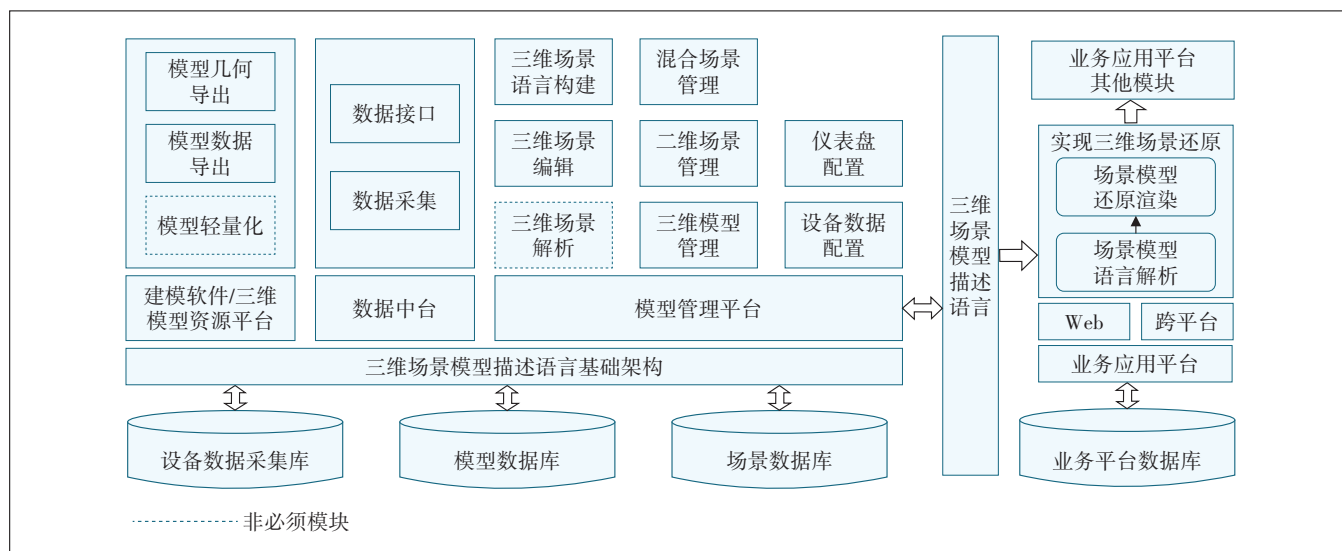


图1 基于三维场景模型描述语言的应用系统架构

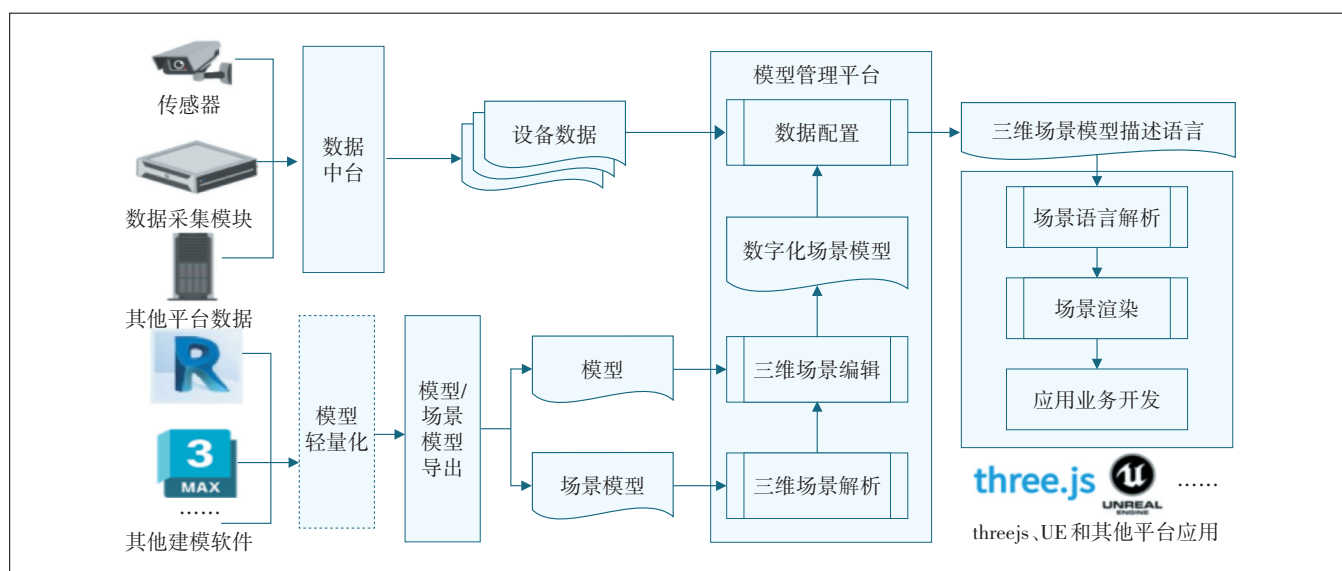


图2 三维场景模型描述语言的数据流程示意

范,以便于精确地描述场景。

a) 三维模型数据。三维模型数据包括了模型的几何数据信息,如几何实体、拓扑结构、度量信息、空间关系和变换等。在本场景定义规范中,最小的几何单元是具有明确业务含义的模型几何体。

b) 通用业务数据。通用业务数据为几何模型赋予了具体的业务含义,涵盖了模型的分类、制造商信息、型号和设计细节等。

c) 设备实例数据。设备实例数据确保了模型的唯一性,包括模型实例的唯一标识符(uuid)和模型点

位数据的配置信息等。

d) 模型交互数据。模型交互数据涉及3D模型、系统界面中的BI面板以及按钮交互的信息。

通过组织这些数据,能够为用户提供一个强大而灵活的可视化平台,不仅能够展示复杂的三维模型,还能实现实时数据的动态交互和分析。

2.1 三维模型数据的组织与管理

在三维模型数据的组织过程中,主要关注2个核心环节:三维场景语言的构建与还原三维场景(见图3)。

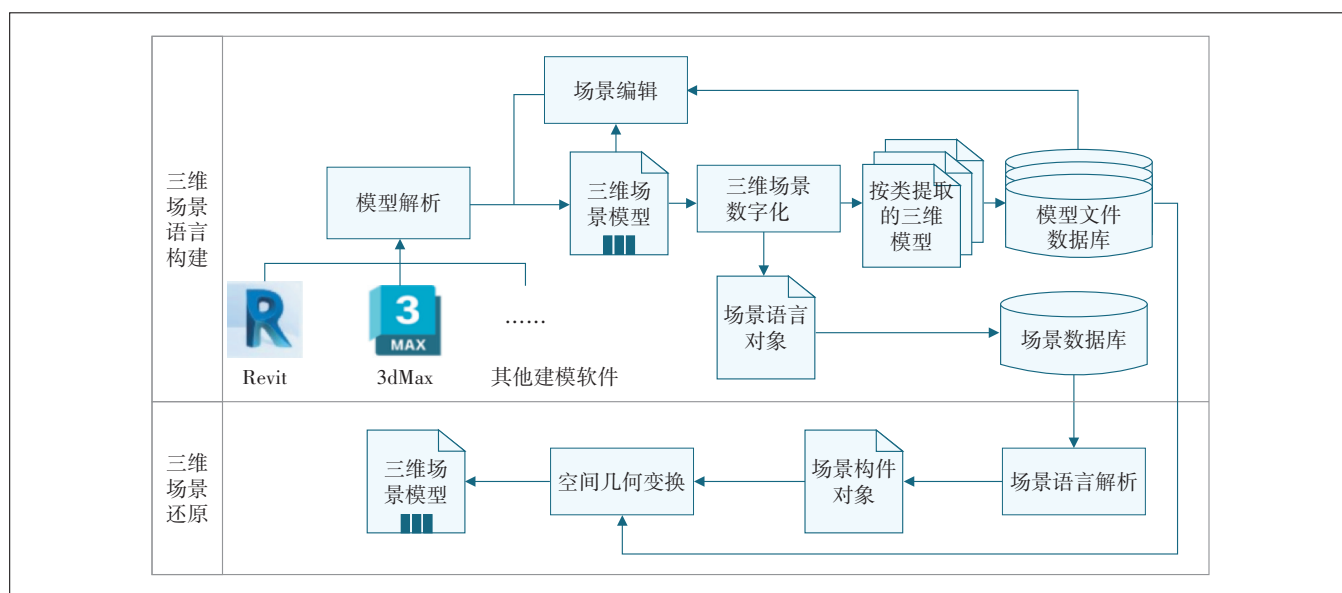


图3 三维模型数据组织与管理示意

首先,三维场景语言的构建环节涉及对不同平台三维模型的兼容性解析。通过这一过程,能够将独立的三维模型构件以及由这些构件组合而成的场景语言进行整合,并分别存储至相应的数据库中。在这一阶段,通过对场景语言的修改,用户可以进行场景的自定义编辑、场景的嵌套等高级操作,从而创造出全新的三维场景。

其次,在三维场景还原环节中,通过解析场景语言中所描述的层级结构和空间关系,可获取所需的模型文件。通过空间几何变换,可将这些模型准确地还原到三维场景中,这个过程与三维场景语言构建的过程不存在技术上的耦合,因此不要求与构建过程使用同一个平台。

通过这2个流程的紧密配合,能够确保三维模型数据的高效组织和精准还原,为用户提供一个强大而灵活的三维场景构建与展示平台。

2.1.1 三维模型场景数字化

在进行数据中心三维场景的数字化前需要赋予场景中的模型构件业务意义。在三维场景语言中,按构件功能,通过“type”字段划分为建筑(building)、场景(scene)、空间(space)、模型(model)4类。

a) 建筑。从建模软件导出未作处理的构件,默认为建筑类别,通常为数据中心建成后不产生变化且不关注的结构、设备等模型。

b) 场景。已存入模型库的三维场景模型,具备完整的场景语言和相应的构件。

c) 空间。模型中用户定义的抽象空间范围。空间为非实体模型,空间内的构件位置均为相对空间的局部坐标系,可以为房间、楼层等范围。

d) 模型。场景中会频繁变化或者需要关注的设备模型构件,通常为各类园区设施和机房设备。

场景构件对象如表1所示。

经用户手动录入或自动解析模型中的业务参数后,结合以上场景语言的定义,通过三维引擎加载场景,解析提取模型文件及其几何参数,按照场景模型中的层级组织成场景语言并保存相应的模型文件到数据库,即可完成三维场景语言的构建。

2.1.2 空间局部坐标系

三维场景语言空间的引入极大地便利了与其他数据的对接。在这一语言框架中,子模型的空间定位是基于局部坐标系的参数设定。这种设计带来了显著的优势。

表1 场景构件对象

字段名	数据类型	字段含义	空间构件参数
family	String	构件的族,如机柜、电源柜	否
familySymbol	String	构件的族类型,例如精密空调 2 700×1 000×2 000、精密空调 1 850×1 000×2 000	否
fileId	String	模型构件在数据库中的文件唯一识别编码	否
type	String	场景构件的类别	是
position	Object	构件在场景中的位置参数,单位为 cm	否
rotation	Object	构件在场景中围绕空间中垂直方向的旋转弧度	否
scale	Object	构件在场景中的缩放参数	否
children	Array	场景构件下的子构件信息	是
uuid	String	场景构件在场景中的唯一识别编码	是
name	String	场景构件的显示名称	否
bounding-BoxXYZ	Object	空间构件的包围盒	是

a) 便于新构件的空间定位。在空间内添加或调整模型构件的位置时,无需精确知晓世界坐标原点的具体位置,只需提供相对于局部空间的坐标即可。

b) 便于房间级和楼层级模型数据的直接采集。例如,在仿真过程中提取房间的几何数据时,可以直接使用空间包围盒作为墙体,无需通过世界坐标系重新计算设备相对于房间的具体坐标。

c) 为场景语言提供了与不同平台数据兼容的可能性。其他应用平台的数据,经过适当的整理和组织,只需补充相应的空间位置信息,即可实现在特定空间下所有数据的无缝接入。

通过这种方式,三维场景语言不仅提高了数据操作的灵活性和效率,还为多平台数据的整合提供了强有力的支持。

2.1.3 三维场景嵌套

场景嵌套,顾名思义是可以在三维场景中支持嵌套已经在库的场景,这一特性不仅将三维场景的定义从园区、楼宇、楼层等空间范围延伸到机柜、通道等设备级范围,还提升了三维场景语言的灵活性和场景资源的利用效率。通过子场景的嵌套使用,能够构建更为复杂且层次分明的三维环境,同时避免了重复劳动,使三维场景构建过程更加高效和系统化。

2.1.4 空间几何变换

三维空间中的点可以从一个空间下转换到另一

个空间下,这种转换可以通过齐次坐标和变换矩阵完成。对于一个三维空间中的点 (x,y,z) ,先给它增加一个值为1的分量,形成齐次坐标,再与变换矩阵相乘,就可以得到这个点在另外一个空间的坐标。当需要进行多次空间变换时只需要进行多次变换矩阵的相乘即可,如式(1)所示^[15]。

$$\begin{pmatrix} x_t \\ y_t \\ z_t \\ \omega_t \end{pmatrix} = M_n \cdots M_3 M_2 M_1 \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

对于围绕坐标轴的旋转变换,可直接使用三维变换矩阵与坐标相乘。

2.1.4.1 左右手坐标系的空间几何变换

不同的引擎使用的坐标系会有区别,例如Three.js使用的是右手坐标系,虚幻引擎(UE)使用的是左手坐标系。如果在不同的坐标系下构建三维场景语言和还原三维场景,在完成还原时,还需要进行左右手坐标系的转换,场景的姿态才能完全还原。

在进行左右手坐标系变换前,首先对比2个坐标系的区别。下文以Three.js和UE 2个平台为例进行分析(见图4)。

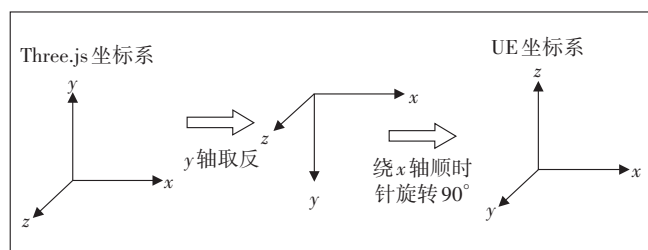


图4 Three.js平台与UE平台坐标系转换

经过y轴取反和旋转变换后可以从Three.js坐标系下的坐标转为UE坐标系下的坐标:

$$P_{ue} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-90^\circ) & -\sin(-90^\circ) \\ 0 & \sin(-90^\circ) & \cos(-90^\circ) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} P_{threejs} \quad (2)$$

即:

$$P_{ue} = S_T P_{threejs} \quad (3)$$

其中, S_T 为从Three.js坐标系转换到UE坐标系的转换矩阵:

$$S_T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4)$$

2.1.4.2 空间几何变换的简化

在三维场景语言中,模型数据并未提供包含模型空间转换的矩阵,而是分开提供坐标、旋转和缩放。这主要是因为通信行业场景的特点:第一,建筑、设备安装方式固定,只会绕着垂直地面的方向旋转;第二,标准设备下不会进行缩放。因此,从Three.js平台转换到UE平台下的坐标变换可简化为:

$$\begin{pmatrix} x_{ue} \\ y_{ue} \\ z_{ue} \end{pmatrix} = S_T \begin{pmatrix} x_{threejs} \\ y_{threejs} \\ z_{threejs} \end{pmatrix} = \begin{pmatrix} x_{threejs} \\ z_{threejs} \\ y_{threejs} \end{pmatrix} \quad (5)$$

即模型构件从Three.js转到UE坐标系只需要把场景语言中的y轴和z轴的坐标值对调。假设在Three.js中模型绕垂直地面的方向(y轴)旋转 θ 度,旋转的变换可简化为:

$$\begin{pmatrix} x_{ue} \\ y_{ue} \\ z_{ue} \end{pmatrix} = S_T^{-1} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} S_T \begin{pmatrix} x_{threejs} \\ y_{threejs} \\ z_{threejs} \end{pmatrix} \quad (6)$$

计算可得:

$$\begin{pmatrix} x_{ue} \\ y_{ue} \\ z_{ue} \end{pmatrix} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{threejs} \\ y_{threejs} \\ z_{threejs} \end{pmatrix} \quad (7)$$

即在UE坐标系中模型绕z轴旋转 $-\theta$ 度。

综上,Three.js平台构建的场景语言在UE坐标系中进行还原时只需要将模型按照计算结果调整空间姿态,即y轴和z轴的坐标值对调,绕y轴的旋转角度取负值赋值给绕z轴的旋转即可。因此,在场景语言中不使用矩阵表达,一方面减少了场景复原时大量的计算量,使数据量较大的场景的还原性能明显提升;另一方面使得数据更直观,可以从数据轻易分辨出设备的相对位置和旋转。

2.2 交互数据组织与管理

交互面板数据多由用户自定义数据组织而成。通过系统与用户的交互,将面板样式、数据定义完成后,提取关键信息组织成可复原的交互语言数据。

2.2.1 三维POI面板

在场景语言中,POI(Point of Interest)面板数据的集成是通过将特定模型构件的面板数据与模型绑定来实现的,这使得模型在场景运行时能够展示POI面板并提供交互功能。该过程首先要求系统为用户提供一个交互式窗口,以便在定义面板显示数据内容、交互方式以及模型的移动等参数。随后,这些参数将通过唯一识别码与三维模型数据中的相应构件关联,

并整合自定义数据到三维场景描述语言中。

POI 面板数据的引入增强了模型的信息表达能力,通过交互界面和数据关联,用户可以直观地获取模型的详细信息,同时也为模型的动态展示提供了数据支持。三维 POI 面板用例图如图 5 所示。

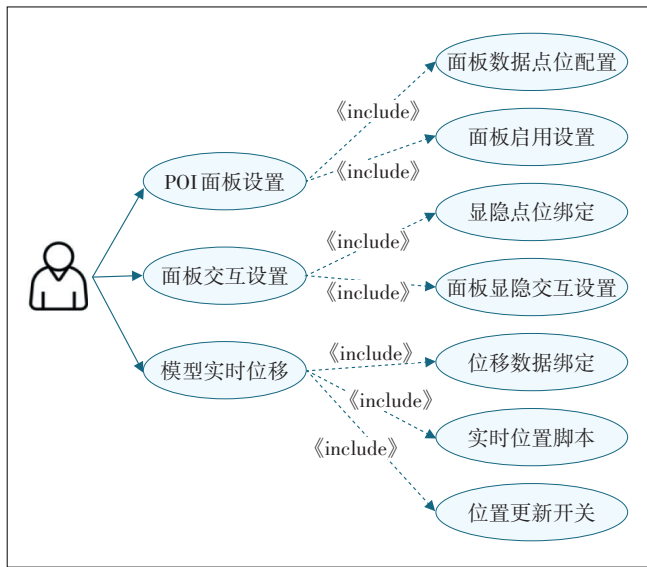


图5 三维 POI 面板用例图

2.2.2 设备实时数据

设备的实时数据主要用于三维 POI 面板,以动态展示实时信息。通过数据接口和特定的 ID 信息,系统能够获取并呈现实时数据。此外,这些数据还可以通过三维 POI 面板的自定义脚本进行进一步处理和展示,从而提供更加灵活和用户友好的展示形式。

2.2.3 监控大屏

监控大屏由用户自定义设置的 BI(Business Intelligence)面板和大屏标题栏组成。其中 BI 面板由用户平台中配置的仪表盘和样式、标题数据组成。最终通过集成三维 POI 面板数据和三维模型数据形成完整的三维场景语言。监控大屏 BI 面板数据生产流程如图 6 所示。

监控大屏按功能通过“type”字段分为图表(chart)、按钮(button)和模型(3D)3 种元素。

- 图表。包含 BI 面板和大屏标题栏,通常会有一个大屏标题栏和若干个 BI 面板。
- 按钮。用于触发跳转到其他场景。
- 模型。大屏监控中需要加载展示的三维模型。

2.3 三维场景语言的兼容性与安全性

2.3.1 兼容性

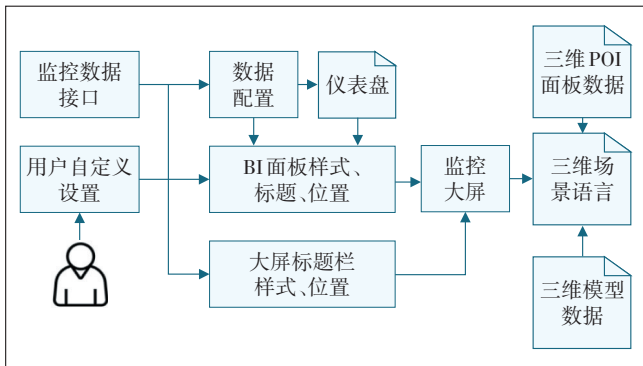


图6 监控大屏 BI 面板数据生产流程

在三维场景语言的设计阶段,本文充分考虑了从数据中心园区级到设备级的模型数据需求,以及大屏下常用图表和面板的展示需求,确保了对这些功能的支持。三维场景语言采用 JSON 格式进行数据存储,在使用不同的模型设计平台和三维引擎平台进行开发时都能保持良好的兼容性。此外,推荐使用*.glb 或*.gltf 格式的三维模型文件,作为开源标准,该格式不仅在多个平台上具有较好的兼容性,而且在开发过程中提供了丰富且便捷的工具支持。然而,针对通信行业的多数场景,三维场景语言在支持不规则形状的房间和设备方面未作兼容。

2.3.2 安全性

安全性是三维场景语言中的一个重要方面,特别是在处理敏感数据或在网络环境中运行时。三维场景语言使用文本方式,开发者可以选择适合其特定需求的加密算法和协议,以确保数据在传输和存储过程中的安全性。

3 面向通信行业三维可视化场景规范的应用

在自研三维平台上和虚幻引擎的应用上对三维场景模型描述语言的预期效果进行了验证,结果显示其达到了数据中心可视化平台快速满足业务需求的目的。

3.1 自研三维平台

平台实现了模型的上传、解析、构件分离、空间编辑、性能优化和场景编辑。此外,平台支持 POI 面板和监控大屏的编辑,能够通过与用户交互,迅速构建三维场景和监控大屏,并按照既定规范生成三维场景语言供其他平台使用。自研三维平台示意如图 7 所示。

3.2 虚幻引擎的应用

得益于三维场景语言良好的兼容性,利用在自研

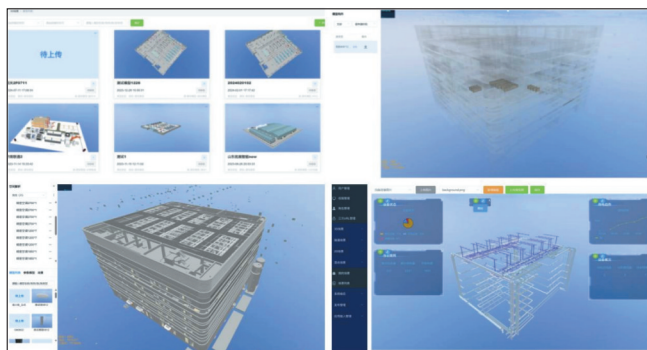


图7 自研三维平台示意

三维平台生成的三维场景规范描述数据,在虚幻引擎中实现了一键加载Web端场景数据、自动创建关卡、导入场景和大屏监控,可将Web端数据在虚幻引擎端进行还原,在获得更好的渲染效果的同时减少了大部分的开发工作。虚幻引擎应用示意如图8所示。



图8 虚幻引擎应用示意

3.3 应用拓展

多平台应用的实现解决了渲染效果与性能需求的矛盾。用户可以在演示、汇报、监控时使用UE引擎,获得较好的视觉效果;在维护、执行任务时使用Three.js等轻量化引擎,快速完成工作。多平台的兼容使得产品在增加极少开发工作量的同时,把选择权力交给了用户,可以根据用户使用场景、渲染需求等不同条件选择不同的平台,以达到最佳的业务体验。但是,三维场景描述语言在Cesium、Unity3D等平台的应用还待共同验证。

4 总结

三维场景模型语言目前正处于发展阶段,其成熟度尚需通过不断的优化和完善来提升。在未来的发展中,该语言需要在横向上扩展其兼容性,以满足更多行业在仿真、施工等关键应用场景的可视化和数字孪生方面的需求。同时,三维场景语言也在致力于兼容不同平台间的数据差异,充分发挥各平台(如UE、Unity等引擎)的渲染优势,提升Three.js、Cesium的功能流畅性,从而为用户提供更加丰富和高效的体验。

三维场景语言为多平台模型及场景资源的统一使用提供了强有力的数据支撑。该技术不仅提高了资源的利用率,还显著减少了跨平台应用的开发工作量。它为数据中心数字孪生平台的研发开辟了新的方向,使原本复杂的开发过程变得更加高效和便捷。

参考文献:

- [1] 吕东,王婧,杨璟林,等. 基于WebGL引擎的铁路工程BIM+GIS平台研发及(设计阶段)应用[J]. 铁道标准设计,2023,67(10):62-69.
- [2] 邓资银,杨宁,刘家琛. 基于CIM的智慧城市应用研究综述[J]. 工业控制计算机,2024,37(8):135-137.
- [3] 邓颂平,周俊杰,范延平,等. 自然资源三维立体“一张图”建设思路探讨[J]. 自然资源信息化,2022(2):1-7.
- [4] 佚名. 虚实相映 中国数字孪生行业研究报告[C]//2023艾瑞咨询4月研究报告会论文集. 上海:上海艾瑞市场咨询有限公司,2023:297-341.
- [5] 陈建华,钟瀚霆,侯明才,等. 数字露头实景三维Web平台研究与云端地质考察应用[J]. 地球学报,2024,45(2):232-242.
- [6] 许俊,牛建生,田阿康,等. 基于BIM的数据中心数字孪生云平台技术研究[J]. 邮电设计技术,2023(12):7-11.
- [7] 赵玲,王子贤,王晓晖,等. 数字孪生在智慧生活应用中的关键技术及前景分析[J]. 系统仿真技术,2023,19(4):369-382.
- [8] 王娜. 虚拟制片中数字资产管理探究——以Epic Games和Unity为例[J]. 家庭影院技术,2024(4):107-112.
- [9] 张鹏德. 基于Unity3D的长距离输水管道数字孪生技术研究[D]. 西安:西安理工大学,2024.
- [10] 阮兵,张可心,田芮利,等. 面向工业建筑场景的BIM可视化引擎功能需求研究[J]. 工程建设标准化,2023(8):82-85.
- [11] 朱蕾蕾,张鹏程. 实景三维广州应用服务平台设计与功能实现[J]. 城市勘测,2024(5):62-65.
- [12] 孙斌,刘斌,武健超,等. 可视化场景中的GIS引擎与游戏引擎的融合技术实践[J]. 水利技术监督,2024(2):48-50.
- [13] 张昊. 通用战场态势可视化系统的设计及实现[J]. 计算机工程与应用,2018,54(17):258-265.
- [14] 许玲,汪可可,王克明,等. 宁波市轨道交通数字孪生地铁安全应急保障平台项目[J]. 城市轨道交通,2023(12):43-46.
- [15] DONALD H. Computer graphics with OpenGL [M]. Upper Saddle River:Pearson Prentice Hall,2004.

作者简介:

田阿康,工程师,硕士,主要从事数字孪生、可视化平台的研发工作;张勇,高级工程师,硕士,主要从事数字孪生研究和可视化技术的开发工作;崔秋然,助理工程师,硕士,主要从事建筑数字孪生的研究工作;徐玉龙,工程师,硕士,主要从事XR、机器人应用、平台产品的研发工作;齐少安,高级工程师,硕士,主要从事CAE、三维场景重建产品的研发工作;党万川,工程师,硕士,主要从事产品研发工作。