



中华人民共和国国家标准

GB/T 45393.4—2025

信息技术 建筑信息模型(BIM)软件 第4部分:网格模型

Information technology—Building information model(BIM) software—
Part 4: Mesh model

2025-03-28 发布

2025-10-01 实施

国家市场监督管理总局
国家标准化管理委员会 发布

目 次

| | |
|-------------------------------------|-----|
| 前言 | III |
| 引言 | IV |
| 1 范围 | 1 |
| 2 规范性引用文件 | 1 |
| 3 术语和定义 | 1 |
| 4 缩略语 | 2 |
| 5 总体要求 | 2 |
| 5.1 网格模型 | 2 |
| 5.2 数据结构 | 3 |
| 5.3 文件结构 | 5 |
| 5.4 关系信息 | 5 |
| 5.5 属性信息 | 5 |
| 6 网格模型表达要求 | 6 |
| 6.1 模型根节点 | 6 |
| 6.2 项目节点 | 7 |
| 6.3 空间节点 | 8 |
| 6.4 产品节点 | 9 |
| 7 层次细节模型表达要求 | 10 |
| 7.1 层次细节模型 | 10 |
| 7.2 瓦片集节点 | 11 |
| 7.3 瓦片节点 | 11 |
| 附录 A (资料性) 网格模型信息数据结构 JSON 描述 | 12 |
| A.1 基本信息数据结构 | 12 |
| A.2 项目信息数据结构 | 14 |
| A.3 空间信息数据结构 | 16 |
| A.4 产品信息数据结构 | 18 |
| A.5 关系信息数据结构 | 19 |
| A.6 属性信息和属性集信息数据结构 | 20 |
| 附录 B (资料性) 网格模型数据结构 JSON 描述 | 23 |
| B.1 模型根节点数据结构 | 23 |
| B.2 项目节点数据结构 | 23 |
| B.3 空间节点和产品节点统一的数据结构 | 24 |
| 附录 C (资料性) 层次细节模型数据结构 JSON 描述 | 25 |

GB/T 45393.4—2025

| | |
|-------------------------|----|
| C.1 瓦片集数据模式和元数据示例 | 25 |
| C.2 瓦片集节点索引信息数据结构 | 25 |
| C.3 瓦片节点索引信息数据结构 | 26 |
| 参考文献 | 27 |

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是 GB/T 45393《信息技术 建筑信息模型(BIM)软件》的第4部分。GB/T 45393 已经发布了以下部分：

- 第2部分：参数化模型；
- 第4部分：网格模型；
- 第5部分：数据接口。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本文件起草单位：清华大学、中设数字技术有限公司、北京飞渡科技股份有限公司、泰瑞数创科技(北京)股份有限公司、中国建筑科学研究院有限公司、广联达科技股份有限公司、深圳市华阳国际工程设计股份有限公司、奥格科技股份有限公司、中南大学、中国测绘科学研究院、中国电子技术标准化研究院、中交第二公路勘察设计研究院有限公司、中国铁路设计集团有限公司、上海秉匠信息科技有限公司、中建工程产业技术研究院有限公司。

本文件主要起草人：顾明、高歌、柯子翊、文盼、陈亚军、李腾、宋彬、刘俊伟、郑国勤、李宾、刘亚鑫、邓明亮、仇阿根、蒋丽忠、陈中治、范登科、夏海兵、药圣琦、贾晓波、魏标、杜阿安、张强、彭贵堂、韩菲、李德文、李珂、皮凯。

引 言

BIM 软件以三维化、结构化、语义化模型的建立与运用为特征,通过提供 BIM 应用所需的建模、计算、表现、管理、交付、检查、协同功能,在建筑全生命周期不同阶段、不同专业、不同应用中发挥着支撑作用。

GB/T 45393《信息技术 建筑信息模型(BIM)软件》旨在指导从业人员对 BIM 软件形成一致认识,在 BIM 软件开发中,采用本系列标准,实现 BIM 软件的互联互通、BIM 模型的交换共享,提高 BIM 软件的可用性、适用性和成熟度,促进 BIM 软件生态的形成,助力 BIM 模型的资产化。

GB/T 45393 拟由 5 部分构成。

- 第 1 部分:通用要求。明确 BIM 软件的基本概念、功能和数据的要求,目的是协助 BIM 软件的需求定义、开发实现。
- 第 2 部分:参数化模型。给出采用标准化参数集表示几何对象的模型格式,接收方通过解析与修改几何对象参数来显示、修改 BIM 模型中的对象,目的是实现 BIM 软件间可编辑的模型交换。
- 第 3 部分:模型视图定义。提供对模型中的内容进行形式化定义的语言,目的是根据特定需求对提交的模型进行完备性检查或模型视图抽取。
- 第 4 部分:网格模型。给出基于三角形网格表示几何对象不可参数化编辑的模型格式,目的是用于 BIM 模型可视化的表现、沟通、管理。
- 第 5 部分:数据接口。给出不同 BIM 软件与平台间的数据调用方式、发送与返回信息,目的是实现不同的 BIM 软件以接口的方式共享 BIM 数据。



信息技术 建筑信息模型(BIM)软件

第4部分:网格模型

1 范围

本文件定义了BIM软件网格模型的数据结构以及扩展记录语义信息的方式,规定了多层次细节模型在结构、实现和渲染对象等方面的具体要求。

本文件适用于BIM软件构建模型的轻量化文件或文件片段,也适用于支持开发相关应用软件,对符合本文件的网格模型文件进行读取、创建及修改等操作。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 36456.1 面向工程领域的共享信息模型 第1部分:领域信息模型框架

GB/T 51447 建筑信息模型存储标准

3 术语和定义

下列术语和定义适用于本文件。

3.1

网格模型 **mesh model**

由顶点、边与面定义的网格几何表示的模型。

注:面通常由三角形、四边形组成。

3.2

节点 **node**

信息模型中的基本数据元素,是信息模型中的最小数据单位。

3.3

语义信息 **semantic information**

建筑信息模型中几何信息之外的具有一定意义的信息。

注:如对象类型、属性、关系等信息。

3.4

项目 **project**

以实体产品建造为主要目标,包括设计、施工、运营、维护生命周期的整体工作的集合。

3.5

空间 **space**

表现用于工程在空间上的各分解的结构信息,包含场地、楼房、楼层、房间等空间分解结构信息。

3.6

产品 product

组成建筑信息模型的、在工程领域具有共性意义的事物。

3.7

关系 relationship

描述空间结构、产品之间相互联系的信息单元。

3.8

属性 property

描述空间结构、产品的特定实体实例等特征信息单元。

3.9

属性集 property set

基于相同特征的所有同类型属性的集合,即根据属性的来源、用途或功能等一致性的逻辑,进行归类分组形成的集合。

3.10

层次细节 level of detail

在不影响原始模型视觉效果前提下,通过逐次简化原始模型的表面细节,建立多个表示原始模型不同程度的细节的简化模型,来降低场景的几何复杂性,从而提高渲染效率的技术。

3.11

瓦片 tile

按照缩放级别或比例尺,以分层、分块的形式组织的二维地图或三维场景,是层次细节模型最基本的数据单元。

3.12

瓦片集 tileset

同一二维地图或三维场景的多层、多块瓦片的集合,是层次细节模型最大的数据单元。

4 缩略语

下列缩略语适用于本文件。

BIM:建筑信息模型(Building Information Model)

GLTF:图形语言传输格式(Graphical Language Transmission Format)

ID:索引号(Index)

JSON:JavaScript 对象简谱(JavaScript Object Notation)

PDV:产品数据可视化格式(Product Data Visualization format)

URI:统一资源标识符(Uniform Resource Identifier)

5 总体要求

5.1 网格模型

网格模型符合下列要求。

- a) 网格模型的内容应同时包含图形信息和语义信息,图形信息和语义信息通过唯一 ID 关联。网格模型中承载图形信息和语义信息的模型资源文件宜采用满足本文件模型表达要求的通用图形格式(如 GLTF、PDV 等)。
- b) 图形信息应采用离散化几何表达,即通过有限精度的几何逼近物体的形状,宜采用适当的压缩

算法优化几何数据的存储,宜通过节点引用加空间变换矩阵的方式重复利用图形数据。

- c) 语义信息应包含 BIM 的基本信息、项目信息、空间信息、产品信息、关系信息以及属性信息等,应符合 GB/T 36456.1 中的通用类和属性定义。
- d) 在满足对本文件定义的语义信息表达的基础上,可根据工程建设项目的具体业务场景和实施阶段,添加额外的关系信息和属性信息。
- e) 描述特定类型的语义信息时,应使用以下形式予以定义:
 - 通过表格描述语义信息所具有的“特性”“特性名称”“特性类型”“基数”以及“说明”来定义;
 - “特性”应采用英文单词进行定义,采取下驼峰命名法,即除第一个单词外,后续每个单词的首字母大写,其余字母小写;
 - “特性名称”应采用中文名词进行定义,意义与“特性”保持一致;
 - “特性类型”应采用本文件的数据类型约定进行定义;
 - “基数”应定义“特性”可出现的次数;
 - “说明”应对“特性”做出清晰明确的解释,在“特性”取值范围确定的情况下,列举出所有的取值。

5.2 数据结构

数据结构符合下列要求。

- a) 应采用模型根节点表达基本信息,应采用项目节点表达项目信息,应采用空间节点和产品节点分别定义空间信息和产品信息,应按照 BIM 的空间分解结构以树形数据结构建立网格模型,合理组织模型根节点、项目节点、空间节点和产品节点的层次关系。网格模型的数据结构定义见图 1。



图 1 网络模型 UML 类图

- b) 应在模型根节点存储整个 BIM 模型的所有关系信息、属性信息和属性集信息，并在项目节点、空间节点和产品节点中通过数组下标引用模型根节点存储的上述信息。
- c) 宜使用层次细节技术处理网格模型，并构造满足本文件层次细节模型表达要求的模型文件。

5.3 文件结构

文件结构符合下列要求。

- a) 网格模型文件应是仅包含一个文件夹的 ZIP 文件包, 应使用“CBM”作为文件扩展名。ZIP 文件包宜压缩并归档, 可仅归档, 归档的文件夹和文件路径应使用左斜杠(“/”)作为路径分隔符。
- b) 网格模型的文件夹中宜存储一个 tileset.json 文件, 此时可存储多项模型资源文件, 访问网格模型文件的软件应通过 tileset.json 定义的瓦片集和瓦片信息索引并访问各项模型资源文件。
- c) 网格模型文件夹可不存储 tileset.json 文件, 此时应仅存储一项模型资源文件, 访问网格模型文件的软件应直接访问该模型资源文件。

5.4 关系信息

应使用关系信息描述一项关联对象与一项或多项被关联对象之间的交互逻辑, 在部分情况下, 可使用关系信息描述关联对象与被关联对象之间的联结对象。

关系信息应符合 GB/T 36456.1 中关系的通用类和属性, 定义见表 1, 相应数据结构描述代码见附录 A 的 A.5。

表 1 关系信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|---|--------------|----------|----------------------------|--------|--|
| 1 | id | 标识符 | string | [1,1] | 空间的标识符, 其取值全局唯一 |
| 2 | relationName | 关系类型名称 | oneOf <integer, string> | [1,1] | 该关系的名称: 记录通用的关系类型名称时, 宜存储在表 5 模型基本信息定义的 relationshipTypes 字段中, 并采用整型数据记录索引下标; 记录特殊的关系类型名称时, 可直接采用字符串数据记录 |
| 3 | relating | 关联对象 ID | integer | [1,1] | 关联的元素、对象或具体产品的 ID |
| 4 | related | 被关联对象 ID | integer | [1, *] | 被关联的元素、对象或具体产品的 ID 列表 |
| 5 | connection | 联结对象 ID | integer | [0,1] | 部分关系会在关联与被关联对象间加入一个用于联结的中间件, 需相应记录其 ID |
| 注: “oneOf”为 JSON 数据模式支持的定义形式, 表示该字段为两种数据类型中的任意一种。 | | | | | |

5.5 属性信息

应使用属性信息描述 BIM 中产品的物理、工艺方面的信息。属性信息可分为简单属性和复杂属性两大类。

- a) 简单属性表达单一取值表示的属性信息, 分为“单值”“枚举值”“有界值”“序列值”“表格值”“引用值”。
- b) 复杂属性表达多重取值的属性信息, 其属性类型是另一项完整的属性集, 以此可实现复杂的嵌套的属性信息描述。

属性信息应符合 GB/T 51447 中属性的通用类和属性,定义见表 2,相应数据结构描述代码见 A.6。应使用属性集信息描述存在一定关联或共性的属性信息。

属性集信息应符合 GB/T 51447 中属性集的通用类和属性,定义见表 3,相应数据结构描述代码见 A.6。

表 2 属性信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|-------------|------|--------|-------|--|
| 1 | name | 属性名称 | string | [1,1] | 该属性的标识名称 |
| 2 | valCategory | 值分类 | string | [1,1] | 值的分类,可选择“单值”“枚举值”“有界值”“序列值”“表格值”“引用值”“复杂值” |
| 3 | valType | 值类型 | string | [0,1] | 值的数据类型 |
| 4 | value | 值 | string | [0,*] | 所有值的列表 |

表 3 属性集信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|-------------|-------|--------|-------|----------|
| 1 | name | 属性集名称 | string | [1,1] | 属性集名称 |
| 2 | description | 描述 | string | [1,1] | 属性集描述 |
| 3 | valIndex | 值索引 | string | [1,*] | 所有值的索引列表 |

6 网格模型表达要求

6.1 模型根节点

网格模型中应定义模型根节点,即树状数据结构中最顶层的节点。模型根节点是其他所有节点的祖先,不应存在父节点。

模型根节点应使用模型基本信息描述网格模型关联的 BIM 语义信息元数据,语义信息符合下列要求。

- a) 宜使用另外的文件单独存储语义信息,此时应以字符串形式记录指向该文件的 URI。
- b) 可将单独存储语义信息的文件以二进制模式访问并应用 BASE64 编码,此时应以字符串形式记录编码的数据 URI。
- c) 可使用 JSON 对象直接嵌入语义信息。

模型根节点定义见表 4,相应数据结构描述代码见附录 B 的 B.1。

表 4 模型根节点定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|---|----------|------|-------------------------|-------|---------|
| 1 | metadata | 元数据 | oneOf <string, 见表 5> | [0,1] | 语义信息元数据 |
| 注:“oneOf”为 JSON 数据模式支持的定义形式,表示该字段为两种数据类型中的任意一种。 | | | | | |



模型基本信息应记录 BIM 基本信息、项目信息、空间信息、产品信息、关系信息、属性信息等内容。模型基本信息定义见表 5，相应数据结构描述代码见 A.1。

表 5 模型基本信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|-------------------|----------|--------|--------|-----------------------|
| 1 | version | 版本 | string | [0,1] | 文件版本 |
| 2 | description | 描述 | string | [0,1] | 项目描述,包含该项目的背景和目标等重要信息 |
| 3 | vendorName | 创建软件名称 | string | [0,1] | 创建该网格模型的软件名称 |
| 4 | vendorVersion | 创建软件版本 | string | [0,1] | 创建该网格模型的软件版本 |
| 5 | coordinateSystem | 坐标参考系 | string | [0,1] | 模型采用的坐标参考系 |
| 6 | projects | 项目信息序列 | 见表 7 | [1, *] | 模型对应的项目语义信息 |
| 7 | spatialStructures | 空间信息序列 | 见表 9 | [1, *] | 模型中所有的空间语义信息 |
| 8 | products | 产品信息序列 | 见表 10 | [1, *] | 模型中所有的产品语义信息 |
| 9 | relationships | 关系信息序列 | 见表 1 | [0, *] | 所有关系信息,以支持重用 |
| 10 | relationshipTypes | 关系类型名称序列 | string | [0, *] | 通用的关系类型名称,以支持重用 |
| 11 | properties | 属性信息序列 | 见表 2 | [0, *] | 所有属性信息,以支持重用 |
| 12 | propertySets | 属性集信息序列 | 见表 3 | [0, *] | 所有属性集信息,以支持重用 |

6.2 项目节点



网格模型中应定义项目节点。项目节点应作为模型根节点的子节点。根节点下可包含若干项目节点。根节点应指定默认的项目节点编号。

项目节点应以字符串形式记录项目编号,访问网格模型文件的软件应使用项目编号在模型基本信息的项目信息序列中索引关联的项目信息。

项目节点定义见表 6,相应数据结构描述代码见 B.2。

表 6 项目节点定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|----|------|--------|-------|----------------------------------|
| 1 | id | 项目编号 | string | [1,1] | 项目的标识符,用以索引当前场景关联的项目,及确认模型文件的一致性 |

项目信息应符合 GB/T 36456.1 中项目的通用类和属性定义,定义见表 7,相应数据结构描述代码见 A.2。

表 7 项目信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|-------------|------|---------------|-------|--------------------------------|
| 1 | id | 项目编号 | string | [1,1] | 项目的标识符,用以索引当前场景关联的项目 |
| 2 | projectName | 名称 | string | [1,1] | 项目名称 |
| 3 | description | 描述 | string | [0,1] | 项目描述,包含该项目的背景和目标等重要信息 |
| 4 | version | 版本 | string | [0,1] | 项目信息的版本 |
| 5 | phase | 阶段 | string | [0,1] | 项目所处的阶段 |
| 6 | lengthUnit | 长度单位 | string | [0,1] | 项目中几何数据的基本单位;应在千米、米、毫米中选填,默认是米 |
| 7 | angleUnit | 角度单位 | string | [0,1] | 项目中几何数据的基本单位;应在弧度、角度中选填,默认是弧度 |
| 8 | basePoint | 项目基点 | array<number> | [0,1] | 项目基点坐标 |
| 9 | address | 地址 | string | [0,1] | 项目地址信息 |

6.3 空间节点

网格模型中应定义空间节点。空间节点应作为项目节点的子节点或另一个空间节点的子节点。

应使用不同的空间节点描述项目中的场地、楼房、楼层和房间等,场地空间节点宜作为项目节点的子节点,楼房空间节点宜作为场地空间节点的子节点,楼层空间节点宜作为楼房空间节点的子节点,房间空间节点宜作为楼层空间节点的子节点。在满足对本文件定义的语义信息表达的基础上,可根据工程建设项目的具体业务场景和实施阶段,添加额外的空间节点。

空间节点应以字符串形式记录空间编号,访问网格模型文件的软件应使用空间编号在模型基本信息的空间信息序列中索引关联的空间信息。

空间节点定义见表 8,相应数据结构描述代码见 B.3。

表 8 空间节点定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|------------|-------|---------|-------|-------------------------|
| 1 | id | 空间编号 | string | [1,1] | 空间的标识符,其取值全局唯一 |
| 2 | isAssembly | 是/否集合 | boolean | [1,1] | 表示该节点是否涵盖了多个子空间 |
| 3 | isProduct | 是/否产品 | boolean | [1,1] | 值为“false”,即否定时,表示该节点为空间 |

空间信息应符合 GB/T 36456.1 中空间结构的通用类和属性,定义见表 9,相应数据结构描述代码见 A.3。

表 9 空间信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|----------------------|----------|---------------|---------|-------------------------------------|
| 1 | id | 空间编号 | string | [1,1] | 空间的标识符,其取值全局唯一 |
| 2 | code | 分类编码 | string | [1,1] | 数据表中的分类编码 |
| 3 | name | 名称 | string | [1,1] | 空间的名称 |
| 4 | alias | 别名 | string | [0,1] | 该类描述的空间的其他名称 |
| 5 | phase | 阶段 | string | [0,1] | 该空间范围所处的生命周期阶段 |
| 6 | spaceType | 空间类型 | string | [1,1] | 应在“场地”“建筑”“楼层”“空间”“其他”中选择 |
| 7 | compositionType | 空间组织类型 | string | [1,1] | 当前空间结构是代表它本身,或是多个其他结构的聚合,或是某个结构的一部分 |
| 8 | refLongitude | 相对经度 | integer | [3,4] | 场地的相对参照点的经度,单位:度分秒 |
| 9 | refLatitude | 相对纬度 | integer | [3,4] | 场地的相对参照点的纬度,单位:度分秒 |
| 10 | refPosition | 相对位置 | array<number> | [0,1] | 相对参照点的位置信息 |
| 11 | elevation | 相对海平面高程 | number | [0,1] | 场地的相对海平面的高程或楼层的相对给定项目参照高程的高程 |
| 12 | elevationOfRefHeight | 给定项目参照高程 | number | [0,1] | 给定项目零点的海平面高程,通常是地面 1 层 |
| 13 | elevationOfTerrain | 相对地形低点高程 | number | [0,1] | 相对建筑范围内地形低点的高程 |
| 14 | landTitleNumber | 地区编码 | string | [0,1] | 区域系统内的编码 |
| 15 | address | 地址 | string | [0,1] | 场地地址或建筑地址 |
| 16 | description | 描述 | string | [0,1] | 对空间的描述 |
| 17 | relationships | 关系下标列表 | integer | [0, *] | 当前空间结构的交互关系列表 |
| 18 | propertySets | 属性集下标列表 | integer | [0, *] | 当前空间结构的属性集列表 |

6.4 产品节点

网格模型中应定义产品节点。产品节点应作为空间节点的子节点或另一个产品节点的子节点。

应使用产品节点描述单个构件或构件集合,当产品节点表示单个构件时,则表示该节点作为当前空间节点下的最底层节点,当产品节点表示构件集时,则表示该节点包含子节点。

产品节点应以字符串形式记录产品编号,访问网格模型文件的软件应使用产品编号在模型基本信息的产品信息序列中索引关联的产品信息。

产品节点定义见表 10,相应数据结构描述代码见 B.3。

表 10 产品构件节点定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|------------|-------|---------|-------|------------------------|
| 1 | id | 产品编号 | string | [1,1] | 产品的标识符,其取值全局唯一 |
| 2 | isAssembly | 是/否集合 | boolean | [1,1] | 表示该节点是否是多个产品的集合,即构件集合 |
| 3 | isProduct | 是/否产品 | boolean | [1,1] | 值为“true”,即肯定时,表示该节点为产品 |

产品信息应符合 GB/T 36456.1 中产品和构件的通用类和属性,定义见表 11,相应数据结构描述代码见 A.4。

表 11 产品信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|---------------|---------|---------|-------|---|
| 1 | id | 产品编号 | string | [1,1] | 产品的标识符,其取值全局唯一 |
| 2 | name | 名称 | string | [1,1] | 产品的名称 |
| 3 | description | 描述 | string | [1,1] | 产品的描述 |
| 4 | typeCode | 分类编码 | string | [0,1] | 产品的分类编码,通常引用外部语意资源的标识 |
| 5 | typeDesc | 类型描述 | string | [0,1] | 产品的类型描述,通常引用外部语意资源的标识 |
| 6 | relationships | 关系下标列表 | integer | [0,*] | 当前产品的交互关系列表 |
| 7 | propertySets | 属性集下标列表 | integer | [0,*] | 当前产品的属性集列表 |
| 8 | geometries | 几何节点列表 | integer | [0,*] | 当前产品的不同精度的几何表达,为空时即只有 node 本身的 meshPrimitive 作为几何表达 |
| 9 | domain | 产品所属专业 | string | [0,1] | 建筑、结构、机电等 |
| 10 | stage | 产品所在阶段 | string | [0,1] | 设计、施工、运维、报废等 |

7 层次细节模型表达要求

7.1 层次细节模型

层次细节模型符合下列要求。

- a) 应在不损坏网格模型信息完整性的情况下应用层次细节技术,将网格模型按照一定的几何误差测度依次划分为不同层次的瓦片结构,划分方式宜采用四叉树结构进行分层,可采用八叉树结构进行分块。
- b) 层次细节模型中应定义模型根节点,应使用最顶层的瓦片集描述模型根节点。
- c) 描述模型根节点的瓦片集应使用一个 JSON 文件存储,文件名应使用“tileset.json”。

7.2 瓦片集节点

瓦片集节点符合下列要求。

- a) 瓦片集节点应通过数据模式和元数据关联 BIM 语义信息,应使用索引信息记录瓦片集对应的网格模型节点类型和节点编号。
- b) 瓦片集节点数据模式应使用“BIM_semantic_data”,数据模式示例见附录 C 中 C.1。
- c) 瓦片集节点元数据宜使用另外的文件单独存储语义信息,此时应以字符串形式记录指向该文件的 URI;可将单独存储语义信息的文件以二进制模式访问并应用 BASE64 编码,此时应以字符串形式记录编码的数据 URI;可使用 JSON 对象直接嵌入语义信息。使用另外的文件单独存储语义信息的元数据示例见 C.1。
- d) 瓦片集中上下层级的瓦片应定义细化方式,宜使用“REPLACE”方式表示下一层级的瓦片在渲染时将完全替换上一层级的瓦片,可使用“ADD”方式表示下一层级的瓦片在渲染时仅在上一层级的瓦片基础上增加内容。

瓦片集节点索引信息定义见表 12,相应数据结构描述代码见 C.2。

表 12 瓦片集节点索引信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|-------------|------|--------|-------|---------------------|
| 1 | tilesetType | 节点类型 | string | [1,1] | 根节点、项目节点、空间节点、产品节点等 |
| 2 | tilesetId | 节点编号 | string | [1,1] | 网格模型节点的标识符,其取值全局唯一 |

7.3 瓦片节点

瓦片节点符合下列要求。

- a) 瓦片节点应包含可渲染内容、零个或多个子瓦片,以及访问层次细节模型的软件在运行时判断瓦片是否渲染的条件因素,通常使用包围体和几何误差测度作为条件因素。
- b) 瓦片节点的内容宜使用满足本文件模型表达要求的通用图形格式(如,GLTF 格式、PDV 格式等),可使用另一个瓦片集。
- c) 瓦片节点的内容应使用扩展元数据的方式定义索引信息(如,在 GLTF 格式内容中应使用 EXT_structural_metadata 扩展元数据),访问层次细节模型的软件应使用该索引在瓦片集元数据记录的语义信息中查找相应的项目信息、空间信息和产品信息。
- d) 同一网格模型节点在不同瓦片中的各个部分,应记录相同的节点编号,通过任一瓦片中的部分应均可查询到所属构件的完整语义信息。

瓦片节点索引信息定义见表 13,相应数据结构描述代码见 C.3。

表 13 瓦片节点索引信息定义

| 序号 | 特性 | 特性名称 | 特性类型 | 基数 | 说明 |
|----|----|------|--------|-------|--------------------|
| 1 | id | 节点编号 | string | [1,1] | 网格模型节点的标识符,其取值全局唯一 |

附录 A

(资料性)

网格模型信息数据结构 JSON 描述

A.1 基本信息数据结构

以下是使用 JSON 描述的模型基本信息的数据结构。

| |
|--|
| name: BIM_semantic.schema.json |
| <pre> { "\$schema": "http://json-schema.org/draft-04/schema", "title": "基本信息定义", "type": "object", "description": "基本信息数据结构定义", "properties": { "version": { "type": "string" }, "description": { "type": "string" }, "vendorName": { "type": "string" }, "vendorVersion": { "type": "string" }, "coordinateSystem": { "type": "string" }, "projects": { "type": "array", "items": { "type": "object", "allOf": [{ "\$ref": "BIM_semantic.project.schema.json" }] } }, "spatialStructures": { "type": "array", "items": { </pre> |

```

    "type": "object",
    "allOf": [
      {
        "$ref": "BIM_semantic.spatialStructure.schema.json"
      }
    ]
  },
  "minItems": 1
},
"products": {
  "type": "array",
  "items": {
    "type": "object",
    "allOf": [
      {
        "$ref": "BIM_semantic.product.schema.json"
      }
    ]
  },
  "minItems": 1
},
"relationships": {
  "type": "array",
  "items": {
    "type": "object",
    "allOf": [
      {
        "$ref": "BIM_semantic.relationship.schema.json"
      }
    ]
  }
},
"relationshipTypes": {
  "type": "array",
  "items": {
    "type": "string"
  }
},
"properties": {
  "type": "array",
  "items": {
    "type": "object",
    "allOf": [
      {
        "$ref": "BIM_semantic.property.schema.json"
      }
    ]
  }
}

```

```

    }
  },
  "propertySets": {
    "type": "array",
    "items": {
      "type": "object",
      "allOf": [
        {
          "$ ref": "BIM_semantic.propertySet.schema.json"
        }
      ]
    }
  },
  "extensions": {},
  "extras": {}
},
"additionalProperties": true,
"required": [
  "project",
  "spatialStructures",
  "products"
]
}

```

A.2 项目信息数据结构

以下是使用 JSON 描述的模型项目信息的数据结构。

```

name: BIM_semantic.project.schema.json

{
  "$ schema": "http://json-schema.org/draft-04/schema",
  "title": "项目信息定义",
  "type": "object",
  "description": "项目信息数据结构定义",
  "properties": {
    "id": {
      "type": "string"
    },
    "projectName": {
      "type": "string"
    },
    "description": {
      "type": "string",
      "maxLength": 255
    },
    "version": {

```

```

    "type": "string"
  },
  "phase": {
    "type": "string"
  },
  "lengthUnit": {
    "type": "string",
    "enum": [
      "毫米",
      "米",
      "千米"
    ],
    "maxLength": 255
  },
  "angleUnit": {
    "type": "string",
    "enum": [
      "弧度",
      "角度"
    ],
    "maxLength": 255
  },
  "basePoint": {
    "type": "object",
    "allOf": [
      {
        "$ref": "BIM_semantic.coordinate.schema.json"
      }
    ]
  },
  "address": {
    "type": "string"
  },
  "extensions": {},
  "extras": {}
},
"additionalProperties": true,
"required": [
  "projectID",
  "projectName",
  "version"
]
}

```



A.3 空间信息数据结构

以下是使用 JSON 描述的模型空间信息的数据结构。

```

name: BIM_semantic.spatialStructure.schema.json

{
  "$schema": "http://json-schema.org/draft-04/schema",
  "title": "空间信息定义",
  "type": "object",
  "description": "空间信息数据结构定义",
  "properties": {
    "id": {
      "type": "string"
    },
    "code": {
      "type": "string"
    },
    "name": {
      "type": "string",
      "maxLength": 255
    },
    "alias": {
      "type": "string"
    },
    "phase": {
      "type": "string"
    },
    "spaceType": {
      "type": "string",
      "enum": [
        "场地",
        "建筑",
        "楼层",
        "空间"
      ]
    },
    "compositionType": {
      "type": "string",
      "enum": [
        "自身",
        "聚合",
        "组成要素"
      ]
    },
    "refLongitude": {
      "type": "array",

```

```
"items": [  
  {  
    "type": "integer"  
  }  
],  
"refLatitude": {  
  "type": "array",  
  "items": [  
    {  
      "type": "integer"  
    }  
  ]  
},  
"refPosition": {  
  "type": "object",  
  "allOf": [  
    {  
      "$ref": "coordinate.ref_semantic.schema.json"  
    }  
  ]  
},  
"elevation": {  
  "type": "number"  
},  
"elevationOfRefHeight": {  
  "type": "number"  
},  
"elevationOfTerrain": {  
  "type": "number"  
},  
"landTitleNumber": {  
  "type": "string"  
},  
"address": {  
  "type": "string"  
},  
"description": {  
  "type": "string"  
},  
"relationships": {  
  "type": "array",  
  "items": [  
    {  
      "type": "integer"  
    }  
  ]  
}
```

```

    ]
  },
  "propertySets": {
    "type": "array",
    "items": [
      {
        "type": "integer"
      }
    ]
  },
  "extensions": {},
  "extras": {}
},
"additionalProperties": true,
"required": [
  "id",
  "code",
  "name",
  "spaceType",
  "compositionType"
]
}

```

A.4 产品信息数据结构

以下是使用 JSON 描述的模型产品信息的数据结构。

```

name: BIM_semantic.product.schema.json

{
  "$schema": "http://json-schema.org/draft-04/schema",
  "title": "产品信息定义",
  "type": "object",
  "description": "产品信息数据结构定义",
  "properties": {
    "id": {
      "type": "string"
    },
    "name": {
      "type": "string"
    },
    "description": {
      "type": "string"
    },
    "typeCode": {
      "type": "string"
    },
    "typeDes": {

```

```

    "type": "string"
  },
  "relationships": {
    "type": "array",
    "items": [
      {
        "type": "integer"
      }
    ]
  },
  "propertySets": {
    "type": "array",
    "items": [
      {
        "type": "integer"
      }
    ]
  },
  "geometries": {
    "type": "array",
    "items": {
      "type": "integer"
    }
  },
  "domain": {
    "type": "string"
  },
  "stage": {
    "type": "string"
  },
  "extensions": {},
  "extras": {}
},
"additionalProperties": true,
"required": [
  "id",
  "name",
  "description"
]
}

```

A.5 关系信息数据结构

以下是使用 JSON 描述的模型关系信息的数据结构。

```

name: BIM_semantic.relationship.schema.json

{
  "$ schema": "http://json-schema.org/draft-04/schema",
  "title": "关系信息定义",
  "type": "object",
  "description": "关系信息数据结构定义",
  "properties": {
    "relating": {
      "type": "integer"
    },
    "related": {
      "type": "array",
      "items": [
        {
          "type": "integer"
        }
      ]
    },
    "connection": {
      "type": "integer"
    },
    "extensions": {},
    "extras": {}
  },
  "additionalProperties": true,
  "required": [
    "id",
    "relationName",
    "relating",
    "related"
  ]
}

```

A.6 属性信息和属性集信息数据结构

以下是使用 JSON 描述的模型属性信息和属性集信息的数据结构。

```

name: BIM_semantic.property.schema.json

{
  "$ schema": "http://json-schema.org/draft-04/schema",
  "title": "属性信息定义",
  "type": "object",
  "description": "属性信息数据结构定义",
  "properties": {
    "name": {
      "type": "string"
    },
  },
}

```

```

    "valCategory": {
      "type": "string",
      "enum": [
        "单值",
        "枚举值",
        "有界值",
        "序列值",
        "表格值",
        "引用值",
        "复杂值"
      ]
    },
    "valType": {
      "type": "string"
    },
    "value": {
      "type": "array",
      "items": [
        {
          "type": "string"
        }
      ]
    },
    "extensions": {},
    "extras": {}
  },
  "additionalProperties": true,
  "required": [
    "name",
    "valType"
  ]
}

```

name: BIM_semantic.propertySet.schema.json

```

{
  "$schema": "http://json-schema.org/draft-04/schema",
  "title": "属性集定义",
  "type": "object",
  "description": "属性集数据结构定义",
  "properties": {
    "name": {
      "type": "string"
    },
    "description": {
      "type": "string"
    }
  }
}

```

```
    },  
    "valIndex": {  
      "type": "array",  
      "items": [  
        {  
          "type": "integer"  
        }  
      ]  
    },  
    "extensions": {},  
    "extras": {}  
  },  
  "additionalProperties": true,  
  "required": [  
    "name",  
    "description",  
    "valIndex"  
  ]  
}
```



附 录 B
(资料性)
网格模型数据结构 JSON 描述

B.1 模型根节点数据结构

以下是使用 JSON 描述的模型根节点数据结构。

| |
|---|
| name: root.BIM_semantic.schema.json |
| <pre> { "\$schema": "http://json-schema.org/draft-04/schema", "title": "BIM_semantic root extension", "type": "object", "description": "模型根节点数据结构定义", "properties": { "metadata": { "oneOf": [{ "type": "object", "allOf": [{ "\$ref": "BIM_semantic.schema.json" }] }, { "type": "string", "format": "uri" }] }, "extensions": {}, "extras": {} }, "additionalProperties": true } </pre> |

B.2 项目节点数据结构

以下是使用 JSON 描述的模型项目节点数据结构。

| |
|---|
| name: scene.BIM_semantic.schema.json |
| <pre> { "\$schema": "http://json-schema.org/draft-04/schema", "title": "BIM_semantic scene extension", "type": "object", </pre> |

```

"description": "项目节点数据结构定义",
"properties": {
  "id": {
    "type": "string"
  },
  "extensions": {},
  "extras": {}
},
"additionalProperties": true,
"required": [
  "id"
]
}

```

B.3 空间节点和产品节点统一的数据结构

以下是使用 JSON 描述的模型空间节点和产品节点统一的数据结构。

```

name: node.BIM_semantic.schema.json

{
  "$schema": "http://json-schema.org/draft-04/schema",
  "title": "BIM_semantic node extension",
  "type": "object",
  "description": "空间节点数据结构定义 & 产品节点数据结构定义",
  "properties": {
    "id": {
      "type": "string"
    },
    "isAssembly": {
      "type": "boolean"
    },
    "isProduct": {
      "type": "boolean"
    },
    "extensions": {},
    "extras": {}
  },
  "additionalProperties": true,
  "required": [
    "index",
    "isAssembly",
    "isProduct"
  ]
}

```

附录 C

(资料性)

层次细节模型数据结构 JSON 描述

C.1 瓦片集数据模式和元数据示例

以下是使用 JSON 描述的层次细节模型瓦片集数据模式和元数据示例。

| |
|---|
| name: tileset.json |
| <pre> { ... "schema": { "classes": { "BIM_semantic_data": { "properties": { "url": { "type": "STRING" } } } } }, "metadata": { "class": "BIM_semantic_data", "properties": { "url": "./BIM_semantic.json" } }, ... } </pre> |

C.2 瓦片集节点索引信息数据结构

以下是使用 JSON 描述的层次细节模型瓦片集节点索引信息数据结构。

| |
|--|
| name: tileset.BIM_semantic.schema.json |
| <pre> { "\$schema": "http://json-schema.org/draft-04/schema", "title": "BIM_semantic tileset extension", "type": "object", "description": "瓦片集节点索引信息数据结构定义", "properties": { "tilesetType": { "type": "string", "enum": [</pre> |

```

    "根节点",
    "项目节点",
    "空间节点",
    "产品节点"
  ]
},
"tilesetId": {
  "type": "string"
},
"extensions": {},
"extras": {}
},
"additionalProperties": true
}

```

C.3 瓦片节点索引信息数据结构

以下是使用 JSON 描述的层次细节模型瓦片节点索引信息数据结构。

```

name: EXT_structural_metadata.BIM_semantic.schema.json

{
  "schema": {
    "classes": {
      "BIM_semantic": {
        "properties": {
          "id": {
            "type": "string"
          }
        }
      }
    }
  }
}

```

参 考 文 献

- [1] ISO/IEC 12113 Information technology—Runtime 3D asset delivery format—Khronos glTF™2.0
-

