

# 基于知识库的 IFC 模型存储技术研究

张越美<sup>1,2,3</sup> 高歌<sup>1,2,3</sup> 彭程<sup>1,2,3</sup> 刘寒<sup>1,2,3</sup> 顾明<sup>1,2,3</sup>

(1. 清华大学软件学院, 北京 100084; 2. 北京信息科学与技术国家研究中心, 北京 100084;  
3. 信息系统安全教育国家重点实验室, 北京 100084)

**【摘要】**随着 BIM 技术的广泛应用, 相关模型设计软件对于 IFC 标准的支持日渐完善, IFC 模型的规模和语义信息更加丰富, 如何高效存储 IFC 模型以促进模型的应用是学者们关心的热点问题。本文使用知识库的方法, 阐述了如何将 IFC 数据模型和 IFC 实例文件存储在图数据库中, 建立元数据知识库和实例知识库, 分析两个知识库模型的关系, 在实际的商业综合体模型中验证了知识存储的时间效率和空间效率。在知识库建立的基础上, 本文分析了 IFC 知识库上使用 Cypher 语言查询的案例, 有效地解决了 IFC 版本对比、实体关系梳理、空间关系发掘相关的问题, 对知识库与土木建筑行业的结合提供了重要参考。

**【关键词】**BIM; IFC; 知识库; 图数据库; 模型查询

**【中图分类号】**TU17 **【文献标识码】**A

**【版权声明】**本文被《土木工程信息技术》、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

建筑信息模型 (Building Information Model, BIM) 是建设领域的全周期信息模型。BIM 由数字技术作支撑, 在简化设计流程、计算工程算量、模型交付和检查、后期运维等方面发挥了极大的作用<sup>[1]</sup>。BIM 在建筑领域逐渐得到广泛的应用, buildingSMART 组织推行了工业基础类 (Industry Foundation Classes, IFC) 的标准数据模型, 并被纳入 ISO 标准 (ISO16738:2013), 以解决多种 BIM 设计软件之间的数据交换问题<sup>[2]</sup>。

随着建筑设计软件功能的完善, 建筑模型的设计细节丰富, 建模支持的专业增多, 使模型的语义信息增长迅速, 导出的 IFC 模型体积逐渐增大。在实际的工程项目中, 单专业的 IFC 建筑模型体积已达百兆数量级, 多专业模型体积接近千兆数量级。因此, 如何高效存储 IFC 模型, 如何选取 IFC 存储的数据库类型以贴合 IFC 模型的结构, 以及如何在

IFC 模型中建立数据字典与数据模型的关联是 IFC 模型在土木建筑行业应用中的挑战。

## 1 IFC 模型存储技术类型

### 1.1 文件系统

文件系统是目前 IFC 模型的主要存储方式。然而, IFC 模型中对象的种类、数量繁多, 对象中存在着大量的引用、连接、组合、包含、表示、分配等各类关系。IFC 文件中复杂的构件、关系信息被以非结构化信息的方式以构件为单位逐行存储在 IFC 文件中, 需要全部载入内存才能进行处理与利用。随着 IFC 数据规模的增大, 带来了数据处理性能与可用性上的严重挑战。

### 1.2 关系数据库

关系数据库是目前大规模数据表示的重要方式。采用关系型数据库存储 IFC, 通常是将 IFC Schema 转化实体—关系 (ER) 模型, 为 IFC 中的各种实体建立数据表, 以实体内部的属性作为列, 对

**【基金项目】** 国家自然科学基金项目 (编号:61527812); 核高基项目 (编号:2016ZX01038101); 科技部科技支撑计划项目 (编号:2015BAG14B01-02); 科技部重点研发计划项目 (编号:2016QY07X1402)

**【作者简介】** 张越美 (1994-), 女, 硕士研究生, 主要研究方向: BIM、领域知识库; 通讯作者: 高歌 (1985-), 男, 博士后, 主要研究方向: BIM 标准与支撑技术、领域知识库。

象之间的关系需要采用外键来表示。例如 Cruz 采用了 Oracle 进行 IFC 数据存储,开发了基于 Web 的工程管理平台 Active3D<sup>[3]</sup>。刘照球等基于关系数据库开发了工程设计模型数据库应用系统<sup>[4]</sup>。张洋等利用关系数据库 SQLServer 开发开发了基于 IFC 的信息集成系统<sup>[5]</sup>。然而,IFC Schema 通常较大,如 IFC 4x2 具有 816 个不同的实体类型,且实体之间具有非常复杂的关系。因此采用关系数据库存储 IFC,需要建立极为复杂的数据表,而且需要进行大量耗时的跨表连接操作才能进行各类复杂查询。另外,IFC 实体的属性较多,在关系数据库的实际存储中较为稀疏。此外,由于 BIM 发展得较快,IFC 标准还在不断迭代之中,IFC 标准版本的变化要求大幅度改动数据库结构,使得数据库维护成本非常高,因此关系数据库不适用于 IFC 模型的存储。

### 1.3 面向对象数据库

IFC 模型具备面向对象的特征,针对于这一特性,很多学者提出了面向对象数据库实现对 IFC 数据的存储。如 Tanyer 利用面向对象数据库 EXPRESS Data Manager 存储 IFC 数据<sup>[6]</sup>。Farag 利用面向对象数据库 ObjectStore 存储 IFC 数据并开发了基于 Web 的 IFC 共享环境<sup>[7]</sup>。

陆宁等基于 SQLServer 与 Versant 分别实现了关系数据库和对象关系数据库的 IFC 管理系统,并经过实验比较指出面向对象数据库在性能上具有显著优势<sup>[8]</sup>。然而,面向对象型的 BIM 数据库本身技术尚不成熟、标准化程度较低,相比于现代的 NoSQL 数据库,可扩展性较差。

### 1.4 NOSQL 数据库

NOSQL 数据库是无固定表结构、不采用 SQL 查询、避免使用连接操作的数据库,通常包括键值数据库、文档数据库、列数据库等,近年来发展十分迅速。Beetz 等基于键值型数据库 BerkeleyDB 开发了开源 BIM 服务器 BIMserver,提供了模型增量存储、提取、转换等功能<sup>[9]</sup>。林雅虹等人基于文档数据库 MongoDB 实现了 IFC 的存储并应用在室内路径规划中<sup>[10]</sup>。Ma 等人基于 MongoDB 开发了基于 Web 的 BIM 协同协同,可在线查询和编辑模型中的对象<sup>[11]</sup>。余芳强等人基于列数据库 Hbase 实现了半结构化的数据库,将 IFC 资源实体的数据直接存储在使用它的可独立交换实体的记录中,减少了连接操作,但也增加存储空间和数据冗余<sup>[12]</sup>。NOSQL

数据库通常在查询速度和扩展性等方面具有一定优势,但其数据查询访问方式上通常具有一定的局限性,占用的存储空间也较多。

知识库(或知识图谱)是采用网状的结构对现实世界的事物及其相互关系进行形式化地描述与表达。知识库通常采用图数据库进行存储,通常利于处理大量复杂、互相关联的数据。并且具有灵活的查询方式。可以为信息查询、专家系统、问答系统等多种应用提供支撑。IFC 模型数据规模较大,并且具有近千种对象类型,并且特定构件的表示通常不具有唯一性,而是多种对象互相关联表示,与知识库的特点较为契合。因此,本文采用了知识库建模的方式,以构件为节点,构件关系为边,将 IFC 模型组织成具有图结构的知识库,为开展 IFC 模型知识库上的查询、管理、检查等应用打下基础。

## 2 IFC 模型的知识库构建方法

### 2.1 知识库数据存储方法与 IFC

知识库(或知识图谱)是采用网状的结构对现实世界的事物及其相互关系进行形式化地描述与表达。知识库通常采用图数据库进行存储,通常利于处理大量复杂、互相关联的数据。并且具有灵活的查询方式。可以为信息查询、专家系统、问答系统等多种应用提供支撑。IFC 模型数据规模较大,并且具有近千种对象类型,并且特定构件的表示通常不具有唯一性,而是多种对象互相关联表示,与知识库的特点较为契合。因此,本文采用了知识库建模的方式,以构件为节点,构件关系为边,将 IFC 模型组织成具有图结构的知识库,为开展 IFC 模型知识库上的查询、管理、检查等应用打下基础。

IFC 标准的表达主要分为两个部分:标准 SCHEMA 的表达和实例模型文件的表达。SCHEMA 文件在 buildingSMART 上以 EXPRESS 文件的形式定义。EXPRESS 是 ISO 10303 - 11 中规定的数据库模型表示方式。实例模型文件以 STEP 文件的形式存储,STEP 是 ISO 10303 - 21 中规定的实例数据表示方法。本研究中知识库的生成,将紧密结合 IFC 的 SCHEMA 文件与 STEP 文件标准,解析关键语义信息存储在知识库中。

在构建知识库中,本研究选取了图数据库 Neo4j 作为 IFC 模型的存储介质。图数据库底层最大限度地贴合图结构,与图相关的查询具有语言简

单、高效的特点,适合表达 IFC 模型中的复杂关系图谱。

Neo4j 作为图数据库的代表之一,是第一个使用原生图数据存储的数据库<sup>[13]</sup>。在容量方面,社区版最多支持 320 亿个节点,640 亿个属性和 320 亿个关系,百兆 IFC 模型经过优化后的存储需要的节点数量为十万级别,因此对于 IFC 相关模型的存储完全够用。在速度方面,由于 Neo4j 是原生的图数据库存储,每个节点中存储了每个边的指针,因此遍历效率非常高,属性信息中也可以建立索引,加快属性查找速度。除了 Cypher 查询语言之外,Java 平台有原生的 Embedded API 可以更高效地完成图数据库的增删改查操作,本研究在模型入库阶段即使用了其 Embedded API<sup>[14]</sup>。因此,本文采用 Neo4j 数据库作为知识库的载体。

2.2 元数据知识的建模存储

知识库建立的第一步是对 IFC SCHEMA 在知识库中建模、解析存储。存储 IFC SCHEMA 文件的作用有以下几点:第一,IFC 版本众多,对模型的版本管理尤为重要,知识库可以验证模型文件的版本匹配;第二,对 IFC 模型使用知识库建模时,需要依赖 SCHEMA 知识进行属性推理。因此,本节将介绍 IFC SCHEMA 在知识库中的建模存储方法。

如图 1 所示,将 IFC SCHEMA 文件建模存储为知识库中的元数据知识的步骤分为使用语法分析器 ANTRL 解析 SCHEMA 文件、抽取 IFC 实体信息、补充推理实体属性信息、属性类型关联、Neo4j 数据库存储。

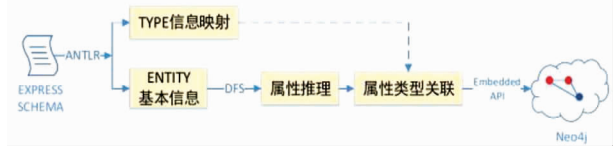


图 1 元数据知识库建模流程

根据 ANTLR<sup>[15]</sup>对于 Express Schema 的解析结果,即可在 Neo4j 图数据库中建立元数据知识库。首先,在 Neo4j 中定义两个标签:Node 和 Attribute,分别代表实例节点和属性节点标签。实例节点包含两个属性:实体名称(name)和实体所属版本(version),属性节点也包含两个属性:属性名称(name)和属性下标(index)。其次,在 Neo4j 中定义了三种关系:HAS\_ATTR 表示实例节点具有某个属性,

LINK\_TO 代表属性节点的限定关系,即这个属性的取值必须是哪一个实体类型,SUBTYPE\_OF 代表节点之间的继承关系。

抽取实体信息后,基于 IFC 实体之间的继承属性,实体的全部属性应为其父类的全部属性和自身属性的并集,因此使用深度优先搜索的方法推理出每个实体的完整属性。最终根据堆属性的引用实体类型定义,将每个属性节点与相应的引用实体连接,形成完整的元数据知识库模型。

得到了元数据知识模型,可以通过可视化工具更为具体地了解不同的 IFC 版本中的区别与联系,在 IFC 模型文件的建模过程中进行版本验证。也可通过简单的语句对于 IFC 版本信息进行挖掘和推理。

2.3 实例知识的建模与存储

知识库建立的第二步是对 IFC 实例知识建模存储。对 IFC 实例文件的存储是知识库的关键部分,是在知识库上查询、检查等应用的基础。在元数据知识的基础下,实例知识的建立分为图 2 几个步骤:IFC 文件解析、实体节点验证存储、引用关系建立。

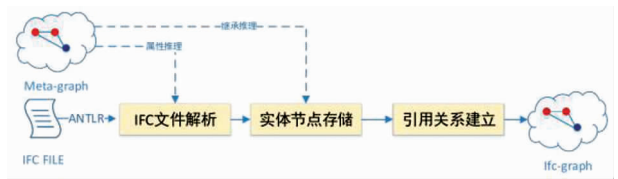


图 2 实例知识库建模流程

IFC 模型文件解析的关键点在 ANTLR 对 IFC 文件的解析方式,如何建立语法树来兼顾存储速度和存储所占内存大小;实例节点验证存储的关键点在如何通过元数据知识库获取属性顺序和对应名称,并在图数据库中建立索引的方式;节点引用关系建立的关键点在于引用关系的发现和建立关系后的关系化简过程。

实例知识库建立的第一个关键点是:提高 IFC 模型文件解析速度、减少解析模型占用的内存空间。基于 IFC 文件解析开源库实现模型文件的解析有两点不足,第一,解析方法包含知识库所不需要的几何造型重构信息导致解析速度慢;第二,基于将模型全部解析入计算机内存对存储空间开销巨大。因此,本研究采用 ANTLR 的 Visitor 模式,独立构建实例文件的语法树,并使用如图 3 语法树拆分算法解决语法树过大问题。高效对接实例知识

库,极大地提高了实例知识库建立的时间、空间效率。

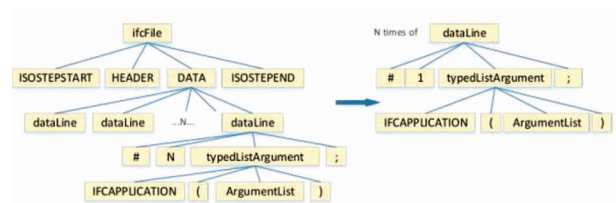


图3 语法树拆分示意图

实例知识库的第二个关键点是实例节点的存储。在节点存储的过程包含三个步骤:实例节点过滤、实例节点属性推理、引用关系建立。IFC 实例文件中有大量与语义信息无关的底层几何信息,底层几何信息节点的数量甚至占整个模型节点数量的一半以上。实例节点过滤的目的即为过滤无语义信息的节点,节省实例知识库空间,提高实例知识库生成速度。

实例节点的属性推理原因是 STEP 文件中每个实体的属性列表中是每个属性的值,没有属性的名称。而在建模过程中,属性名称是两个节点关系建立时的关系名称,因此必须通过推理元数据知识库得到属性对应的名称。其次,在推理属性名称的同时也可以检查这个模型文件是否符合 SCHEMA 的基本规定,例如属性数量是否与 SCHEMA 不对应。属性对应关系如图 4 所示。因此,通过查询元数据只是库中的属性名称,与实例知识库实体进行匹配,即可推理得到实例节点的具体属性键值对。最终,单个实例节点满足以下原则:第一,节点的类型标签为其自身 IFC 类型及其所有父类类型;第二,节点属性为键值对分别对应了 IFC SCHEMA 中的属性名和模型文件中的属性值;第三,每个节点加入 lineId 属性代表节点编号,model 属性代表模型名称。

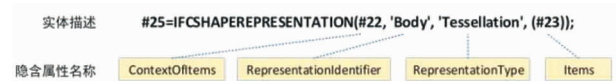


图4 属性推理示意图

实例节点存储后,知识库中的实例信息只有孤立的顶点,将有关关系的节点连接起来,才能形成真正可推理的知识库。引用关系的建立分为两个主要步骤:第一步是逐个遍历数据库中的每个实例节点以及其全部属性,发掘属性引用关系后建立有向

连接,由引用节点指向被引用节点,关系名称即为属性名称;第二步是对部分引用路径进行合并精简,减少不重要的节点。由于 IFC 实例文件中的实体的行号为唯一标识,引用关系的设置便以行号为基础。在使用行号查询引用节点时,由于 IFC 实例文件规模巨大,往往一个模型包含数十万节点,搜索开销较大。本文利用图数据库的索引机制,在节点存储时对行号建立索引,极大地降低了查询的时间开销。最终,IFC 实例文件在知识库中形成一张巨大的关系网。如图 5 可见一个模型中的门(IFC-DOOR)节点在知识库中的展示。

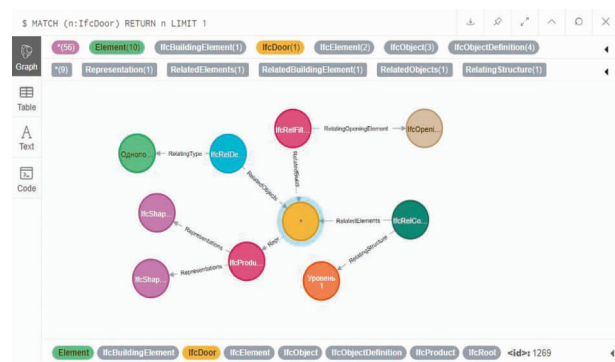


图5 IFCDOOR在实例知识库中的表示

### 3 IFC 知识库的效率分析

本文所选的测试模型均来自实际的商业综合体模型,测试模型有以下三个特点:其一,模型覆盖了两种主流 IFC 版本:IFC2X3 和 IFC4;其二,模型涵盖了建筑、暖通、排水三个专业,具有多专业的普适性;第三,大体量模型的体积超过百兆,使用传统的内存模型将严重受限于用户计算机的硬件配置,可以更加明显地体现出知识库存储的优势。

表 1 分析了测试模型载入知识库的时间效率,效率分析根据模型载入的几个重要步骤进行统计,包括:模型文件递归下降解析、实体节点插入知识库、节点之间关系建立三个步骤。通过分析可得出以下几个结论:第一,知识库建立的总体时间在 2 分钟以下,符合工程对于模型存储的要求;第二,三个步骤所需时间较为平均,其中模型的解析时间和模型大小呈现正相关,节点插入时间和关系建立时间和模型内容紧密相关。

表 2 分析了测试模型在知识库上查询的时间效率。我们以构件的存在性简单查询语句为例,以上模型需要的查询初始化用时、单次查询时间分别如

表 2 所示。

表 1 知识库建立时间效率分析

| 文件名    | 模型大小<br>(MB) | 模型解析<br>(ms) | 节点插入<br>(ms) | 关系建立<br>(ms) | 总时间<br>(ms) |
|--------|--------------|--------------|--------------|--------------|-------------|
| 综合体建筑  | 327          | 23 337       | 37 541       | 22 745       | 88 839      |
| 综合体暖通  | 174          | 17 957       | 15 755       | 17 252       | 51 622      |
| 综合体给排水 | 110          | 16 662       | 18 149       | 4 483        | 43 078      |

表 2 知识库查询时间效率分析

| 文件名    | 模型大小<br>(MB) | 节点数<br>(ms) | 初始化<br>时间(ms) | 查询时间<br>(ms) |
|--------|--------------|-------------|---------------|--------------|
| 综合体建筑  | 327          | 180 3931    | 1 758         | 350          |
| 综合体暖通  | 174          | 897 965     | 2 255         | 126          |
| 综合体给排水 | 110          | 503 401     | 2 333         | 64           |

此外,我们将本文方法与其它方法存储过程的内存空间效率进行了对比。其中一种是将 IFC 解析为语义网三元组,存储到 Jena TDB 数据库;另一种是将 IFC 实体解析为链表,存储到 MongoDB 中。如表 3 所示,在 16GB 内存计算机的实验环境,300M 以上模型的三元组存储创建过程失败,而 MongoDB 的内存占用与模型大小成正比。基于知识库的模型存储架构采用逐行解析存储,内存占用与模型规模相关性较小,百兆模型的存储过程占用内存不超过 500MB,显著地提高了内存空间的使用效率。

表 3 内存空间效率分析对比

| 文件名   | 大小<br>(MB) | TDB<br>(MB) | MongoDB<br>(MB) | 本文方法<br>(MB) |
|-------|------------|-------------|-----------------|--------------|
| 综合体建筑 | 327        | 失败          | 13 190          | 434.5        |
| 综合体暖通 | 174        | 6 750       | 5 980           | 216.2        |

4 IFC 知识库的应用

在 Neo4j 上建立了元数据知识库和实例模型知识库,即可使用 Cypher 查询语言,借助 Neo4j 可视化平台,方便快捷地获取 IFC Schema 信息和 IFC 模型信息。

4.1 IFC SCHEMA 版本关系查询

在知识库的建立章节,本文介绍了将 IFC SCHEMA 文件建模存储入知识库形成元数据知识的方法。随着 BIM 的发展和应用,IFC 版本的也做出配合的更新,将版本信息化为结构化的知识,可以方便地查询 IFC 版本之间的关系和差异。如下 Cypher 脚本即可将知识库中的 IFC4X1 版本信息与

IFC4 版本信息做比对,查询出 IFC4X1 中的增加元素和相关属性。

```
MATCH(m;Entity | ifctype:'IFC4')
WITH COLLECT(distinct m.name) as ifcnames
MATCH(n;Entity | ifctype:'IFC4X1'),
WHERE n.name NOT in ifcnames
RETURN n
```

4.2 类型继承关系查询

Cypher 查询语句在关系查询上的模糊查询功能,对于图数据库的查询产生极大便利。在 IFC SCHEMA 定义的数据模型中,继承关系网较为复杂。获取一个实体的所有子类在官方文档上查询费时费力,在文档类型的数据库上的查询语言复杂,学习成本较高。而在图数据库上,通过 Cypher 语句对于不定长关系链的语言设置和图数据库的底层存储机制,使相关查询语句易于理解和书写,查询速度快。如下简单的 Cypher 脚本即可查询 IfcBuildingElement 实体的子类。

```
MATCH(:Node | name:'IfcBuildingElement', version:'IFC4') <- [:SUBTYPE_OF * 1..5] - (m;Node)
RETURN m.name
```

如图 6 在 Neo4j 平台的可视化展示,可以简洁明了地分析出 IfcBuildingElement 实体的继承关系网络。

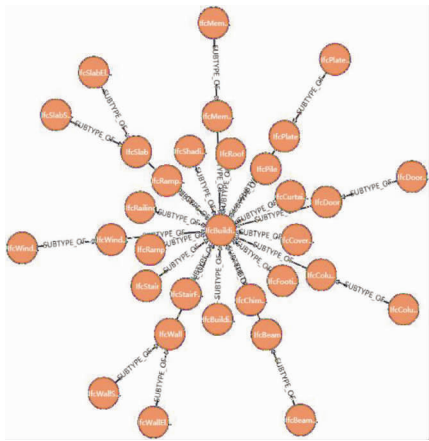


图 6 IFC 继承关系可视化

4.3 建筑空间结构查询

IFC 模型中使用 IfcSpace 实体表示建筑中的空间概念,一个楼层的房间联通关系在 IFC 文件中使

用 BoundedBy 关系表示,在传统的存储方式中,空间联通关系的查询难以直观显示,而图数据库对于拓扑关系的表达高效清晰,如下 Cypher 语句通过获取门和空间的关系形成图 7 所示的 building 的空间关系网络,快捷地获取楼层空间结构。

```
MATCH p = (n:IfcDoor) - [r:isBoundedBy] -> (m:IfcSpace)
RETURN p
```

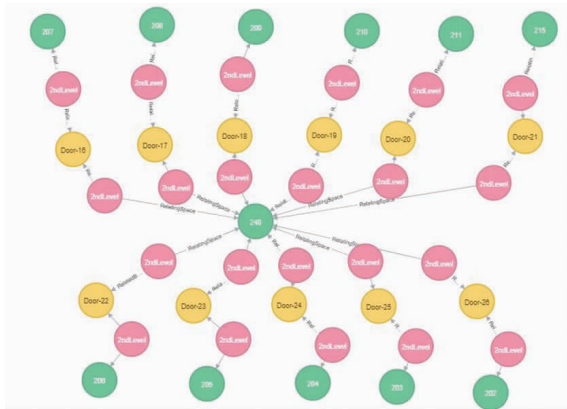


图 7 实例知识空间关系可视化

5 讨论与总结

随着 BIM 技术的发展,IFC 作为 BIM 国际交换标准的普及率逐渐增加,IFC 模型的规模增大、语义信息复杂,IFC 模型的存储是 IFC 模型查询的基础,IFC 模型的存储方案研究关系着 IFC 模型相关应用的便捷性和高效性。本文介绍了基于知识库的 IFC 模型存储方案和基于知识库的 IFC 模型查询应用。在 IFC 知识库的建立中,本文分别实现了以 IFC Schema 建模为基础的元知识库建模、以 IFC 模型文件为基础的实例知识库建模。元数据库的建模为实例知识库的建立打下基础,实例知识库为 IFC 查询提供直接依据。

本文提出的基于知识库的 IFC 存储技术,与文件系统相比,模型不用全部加载到内存,提高了模型信息访问的效率以及对大模型的支持。与基于关系型数据库存储技术相比,关系型数据库需要定义明确的表结构以及进行很多跨表操作。而 IFC 的复杂数据模型、稀疏的属性以及丰富的关系导致建立与维护表结构是困难的,跨表操作是低效的。与面向对象数据库相比,知识库方案所采用的图数据库技术更加成熟,成本更低。以 MongoDB 为代表的

文档类数据库是以 JSON 格式为基础存储数据,不受表结构的限制,但存在难以完全表示 IFC 模型的图结构信息的缺点。而本文提出的存储方法可以完善地表示 IFC 模型中的图结构信息,并提供了更加灵活的查询方法。相较于其他图数据库的存储方案(如 ifcOWL)<sup>[16]</sup>,本文采用属性标签图对 IFC 在图数据库中直接存储,不需将 IFC 模型转换成更复杂图结构,提高了存储效率。并通过元数据知识库实现了 Schema 的动态绑定,使模型存储不受制于 IFC 版本实现。在基于知识库的应用方面,本文主要基于 Cypher 语言对模型进行查询与可视化。相较于 SPARQL<sup>[17]</sup>、SHACL<sup>[18]</sup> 等固定路径的查询语言,基于知识库的方法具有自适应、不需指定匹配路由或路径长度的特性。该方法一方面良好地适配了 IFC 数据关系复杂、对象的表示方式多样化的特点,另一方面显著降低了用户的查询难度,使用户只需关注查询的目标,而不用关心查询的实现方式。

本文以 IFC 版本查询、类型继承关系查询、建筑空间结构查询等功能为例说明了基于知识库的 IFC 查询的灵活、高效性。在本文构建的 IFC 知识库的基础上,可以利用简洁灵活的语句实现更复杂的信息查询,为进一步基于知识库进行模型检查、模型提取等应用打下良好的基础。

参考文献

[ 1 ] Eastman C, Teicholz P, Sacks R, et al. BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors [ M ]. John Wiley & Sons, 2011.

[ 2 ] Industry foundation classes ( ifc ) for data sharing in the construction and facility management industries [ EB/OL ]. 2019. <https://www.iso.org/standard/38047.html>

[ 3 ] Cruz C, Nicolle C, Neveu M. The Active3D – Build; a Web-based civil engineering platform[ J ]. IEEE Multimedia, 2002, 9(4): 87-90.

[ 4 ] 刘照球, 李云贵, 张汉义. 工程设计模型数据库应用系统开发[ J ]. 土木建筑工程信息技术, 2011, 3(3): 16-21.

[ 5 ] 张洋. 基于 BIM 的建筑工程信息集成与管理研究 [ D ]. 清华大学, 2009.

[ 6 ] Tanyer A M, Aouad G. Moving beyond the fourth dimension with an IFC-based single project database[ J ]. Automation in Construction, 2005, 14(1): 15-32.

- [ 7 ] Faraj I, Alshawhi M, Aouad G, et al. An industry foundation classes Web-based collaborative construction computer environment: WISPER [ J ]. Automation in Construction, 2000, 10(1): 79-99.
- [ 8 ] 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, 52(6): 836-842.
- [ 9 ] Beetz J, Berlo L, Laat R, et al. Bimserver.org – an Open Source IFC model server [ C ] // in Proceedings of the CIB W78 27th International Conference on Applications of It in the AEC Industry CIB – W78, Cairo. 2010;1-8.
- [10] Lin Y H, Liu Y S, Gao G, et al. The IFC-based path planning for 3D indoor spaces [ J ]. Advanced Engineering Informatics, 2013, 27(2): 189-205.
- [11] Ma L, Sacks R. A cloud-based BIM platform for information collaboration [ M ]. ISARC, 2016.
- [12] 余芳强, 张建平, 刘强, 等. 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): 1-6.
- [13] Miller J J. Graph database applications and concepts with Neo4j [ C ] // Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA. 2013;141-147.
- [14] Holzschuher F, Peinl R. Performance of graph query languages: comparison of cypher, gremlin and native access in Neo4j [ C ] // Proceedings of the Joint EDBT/ICDT 2013 Workshops. ACM, 2013;195-204.
- [15] Parr T J, Quong R W. ANTLR: A predicated - LL(k) parser generator [ J ]. Software: Practice and Experience, 1995, 25(7): 789-810.
- [16] Terkaj W, Šojic A. Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology [ J ]. Automation in Construction, 2015, 57:188-201.
- [17] SPARQL Query Language for RDF [ EB/OL ]. [ 2019-06-11 ]. <https://www.w3.org/TR/rdf-sparql-query/>.
- [18] Shapes Constraint Language ( SHACL ) [ EB/OL ]. [ 2019-06-11 ]. <https://www.w3.org/TR/shacl/>.

## Research on IFC Model Storage Mechanism Based on Knowledge Base

Zhang Yuemei<sup>1,2,3</sup>, Gao Ge<sup>1,2,3</sup>, Peng Cheng<sup>1,2,3</sup>,  
Liu Han<sup>1,2,3</sup>, Gu Ming<sup>1,2,3</sup>

- (1. School of Software, Tsinghua University, Beijing 100084, China;
- 2. Beijing National Research Center for Information Science and Technology, Beijing 100084, China;
- 3. Key Laboratory for Information System Security, Ministry of Education, Beijing 100084, China)

**Abstract:** With the popularity of application of BIM technology, the related BIM authoring tools are becoming more powerful to support the IFC standard, and scale and semantic details of IFC models are becoming more complicated. Then, how to store IFC model efficiently to promote the model application is a challenge in the research area. This paper applies the method of knowledge base to describe how to store IFC data model and IFC case file in the graph database. A metadata knowledge base and a case knowledge base are both established, and the dependency of the two above knowledge base models is analyzed. In an actual commercial complex model, the time efficiency and space efficiency of knowledge base storage are verified. Based on the establishment of knowledge base, this paper further analyzes the case of using the Cypher language to query the IFC knowledge base, which effectively solves the problems related to IFC version comparison, entity relationship sorting and spatial relationship mining. The study provides an important reference for the combination of knowledge base and the AEC industry.

**Key Words:** BIM; IFC; Knowledge Base; Graph Database; Model Query