

信息技术 BIM 软件 第 3 部分：模型视图定义

1 范围

本文件规定了模型视图定义（MVD）的开发、执行和管理机制等要求，提供了一个轻量级的模型视图定义语言。

本文件适用于建筑信息模型的数据交换中所需的模型视图定义的开发和执行。

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 13000 信息技术 通用多八位编码字符集（UCS）

GB/T 36456.3 面向工程领域的共享信息模型 第3部分：测试方法

GB/T XXXXX.1 信息技术 BIM软件 第1部分：应用参考架构

3 术语和定义

下列术语和定义适用于本文件。

3.1

信息模型 information model

反应现实世界特定领域对象的形式、特性、关系的模式化数据。

[来源：GB/T 36456.1-2018, 3.1]

3.2

数据交换 data exchange

数据的存储、访问、传输和归档。

[来源：GB/T 16656.1-2008, 3.2.17]

3.3

模型视图 model view

信息模型中的由节点与边组成的满足特定交换需求的子集。

3.4

模型视图定义 model view definition

描述模型视图应符合的交换需求的计算机可执行规则集。

3.5

信息交付指南 information delivery manual

描述数据交换的业务需求与交换需求的文档，以文本、表格、图表等自然语言形式表示。

3.6

业务需求 business requirement

对数据交换的一个业务场景的自然语言描述。业务需求通常包括对数据交换的目的、参与方、流程等信息的描述，以及对涉及的数据内容的概述。

3.7

交换需求 exchange requirement

针对特定业务，对所需模型视图的信息单元与相应信息约束的描述。

3.8

信息单元 information unit

交换需求中的用自然语言描述的信息项。

注：一个信息单元可能是可以为一个单独的对象，也可能是可以为一个对象与特性、关系、表现之一的组合。单独的对象，如，墙、门、窗等；对象与特性、关系、表现之一的组合，如，墙的厚度、墙与门之间的连接关系等。

3.9

信息约束 information constraint

对信息单元的内容限制与使用限制的自然语言描述。

注：内容限制，如，节点类型、值域、单位、精度等；使用限制，如，节点存在与否、是否必填、数量、唯一性等。

3.10

对象 object

组成信息模型的、在工程领域具有共性意义的事物。

注：对象包括具有物理实体的事物与抽象的事物。具有物理实体的事物，如，构件、空间、人员、材料等；抽象的事物，如，流程、指令、集合等。

3.11

特性 attribute

对象的某一方面特征的定性或定量的信息。

3.12

关系 relationship

事物之间的互相作用的信息。

3.13

表现 representation

对象的外观与形态的描述性信息。

3.14

信息模式 information schema

描述信息模型的数据分类与数据结构约束的计算机可执行规则集。

3.15

数据分类 data classification

描述信息模型允许具有的节点类型与边类型，以及规定节点类型之间、边类型之间的父子类关系的规则。

3. 16

数据结构约束 data structure constraint

描述信息模型中节点与边之间允许的连接方式与连接数量的规则。

3. 17

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. 18

边 edge

信息模型中节点之间的有向连接，表示从一个节点出发，访问表示某一方面信息的其它节点的方法。

3. 19

子图模板 subgraph template

组成MVD的一类规则，描述信息模式中的多种节点类型与边类型组成的图结构。

3. 20

子图表达式 subgraph expression

组成MVD的一类规则，针对一个子图模板制定，表示对子图实例执行的逻辑判定。

3.21

定义规则 definition rule

一类子图表达式规则，用于定义扩展节点类型。

3. 22

约束规则 constraint rule

一类子图表达式规则，用于描述一项信息约束。

3. 23

固有节点类型 intrinsic node type

在MVD引用的信息模式中规定的节点类型。

3.24

扩展节点类型 extended node type

在MVD中扩展规定的节点类型，继承自固有节点类型。扩展节点类型在信息模型中对应的节点集是其继承的固有节点类型对应的节点集的一个子集。

3. 25

子图路径 subgraph path

子图模板中两个节点位置之间的一条有向通路，由多个节点类型与边类型交替组成。

3.26

子图实例 subgraph instance

在信息模型中与子图模板匹配的图结构，由若干节点实例与边实例组成。子图实例中节点实例与边实例的组合方式与子图模板中节点类型与边类型的组合方式一致。

3. 27

根概念 root concept

当子图表达式对应的子图模板以一个节点类型作为图结构的根，称这个节点类型为这个子图表达式的根概念。根概念通常与信息单元中的一类对象对应。

3. 28

根节点集 root node collection

当子图表达式的根概念在信息模型中对应一个节点集，称这个节点集为这个子图表达式的根节点集。

3. 29

MVD 开发 MVD development

编制IDM文档并根据IDM中的交换需求编制MVD的过程。

3. 30

MVD 执行 MVD execution

计算机程序针对输入的信息模型执行MVD并输出结果的过程。

3. 31

MVD 语言 MVD language

具有描述子图模板与子图表达式的能力的计算机语言。

3. 32

完备性检查 completeness checking

关于信息模型中的组件、属性和关系与全部规定要求相一致的检查。

[来源：GB/T 36456.3-2018, 3.2]

3. 33

模型视图抽取 model view extraction

计算机程序根据MVD从信息模型中抽取一个模型视图，并输出为新的信息模型的过程。新的信息模型符合原信息模型的信息模式。

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)

IDM: 信息交付指南 (Information Delivery Manual)

IFC: 工业基础类 (Industry Foundation Classes)

LOD: 多细节层次 (Level of Details)

MVD: 模型视图定义 (Model View Definition)

MVDLite: 轻量型模型视图定义语言 (Model View Definition Lite Language)

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

5 综述

5.1 MVD 的组成

MVD是GB/T XXXXX.1中交换需求标准的内容。MVD包括子图模板与子图表达式两类规则。

- a) 子图模板规则用于描述信息模型中可能具有的一种图结构，该结构以一个节点类型作为图结构的根，若干子图路径指向与此节点类型相关的各项信息在信息模型中的位置。
- b) 子图表达式规则用于描述信息模型中与子图模板匹配的子图实例执行的规则，分为定义规则与约束规则：
 - 1) 定义规则用于定义扩展节点类型，以信息模型中符合定义规则的子图实例的根节点组成的集合作为扩展节点类型对应的节点集；
 - 2) 约束规则用于描述针对子图实例的一项信息约束。

5.2 MVD 与信息模式之间的依赖关系

MVD引用一个信息模式，基于此信息模式编制MVD中的所有规则。MVD与信息模式之间的依赖关系如图1所示。

- a) 子图模板规则依赖信息模式中的数据分类规则与数据结构约束规则。
- b) 子图表达式规则依赖子图模板规则。
- c) 约束规则依赖定义规则。

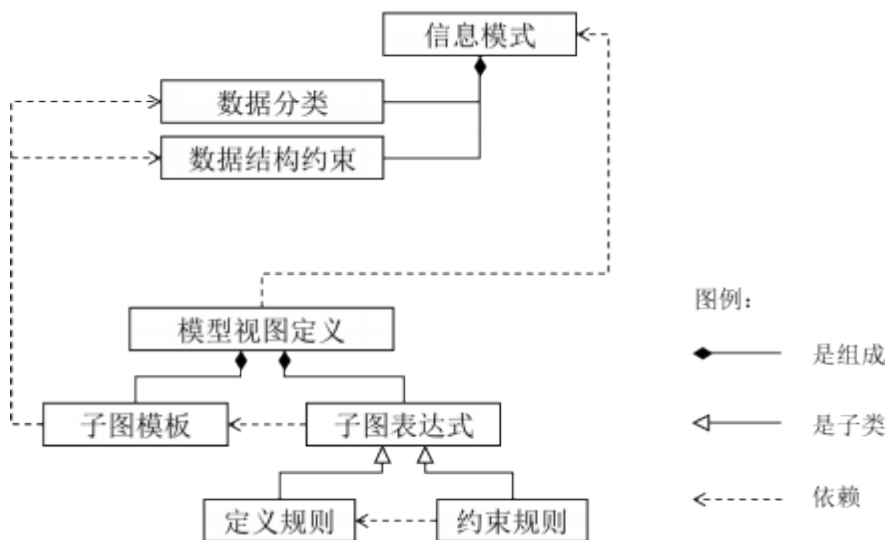


图1 MVD与信息模式之间的依赖关系

5.3 MVD 语言

用于描述MVD的文本语言应具备以下能力：

- a) 描述子图模板与子图表达式的能力；
- b) 为规则附加自然语言标记的能力。

第9章给出的MVD Lite语言是一种满足上述要求的、专门用于描述MVD的语言。

5.4 MVD 开发和执行过程

MVD开发和执行过程见图2。

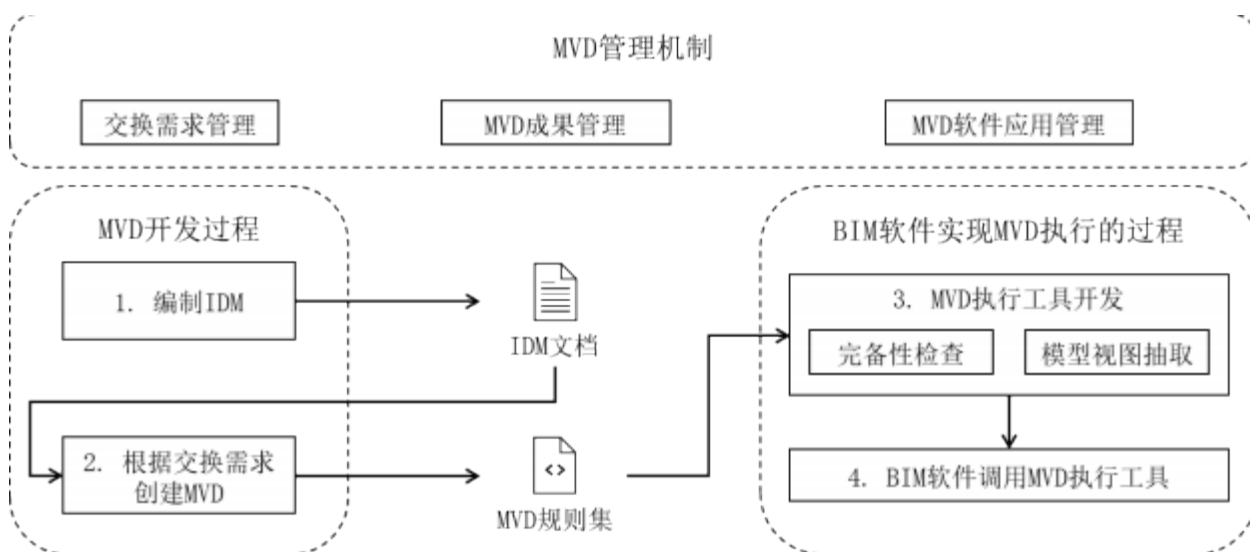


图2 MVD 开发和执行过程

6 MVD 开发

6.1 MVD 开发步骤

MVD的开发包括以下步骤:

- a) 编制 IDM: 描述工程领域中的业务场景与业务需求, 并明确列举数据的交换需求, 形成 IDM 文档;
- b) 根据交换需求创建 MVD: 针对特定的信息模型, 将 IDM 文档中的交换需求转换为计算机可执行的 MVD。

注: MVD的开发通常需要工程领域从业者与信息领域从业者的协作, 对工程领域中的业务场景与业务需求进行分析, 明确描述交换需求中的信息单元与信息约束, 并将其转换为计算机可执行的规则集。

6.2 编制 IDM

6.2.1 IDM 的内容

6.2.1.1 IDM 的组成

IDM应对业务场景与交换需求进行详细描述, 使目标读者能明确地识别出各项业务场景对应的交换需求, 以及各项交换需求中的信息单元与信息约束。IDM的主要组成见图3。

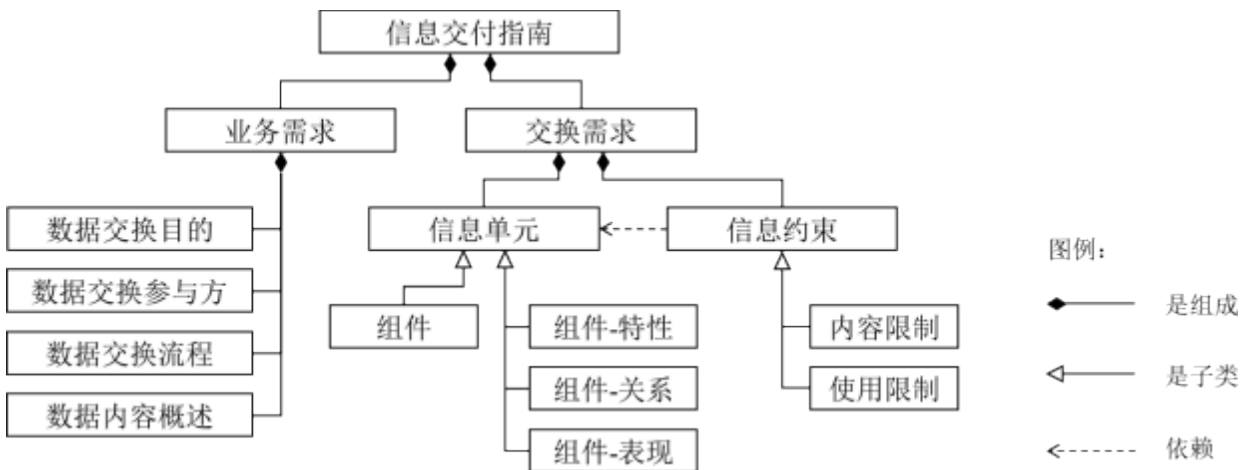


图3 IDM 的主要组成

6.2.1.2 IDM 业务需求

IDM中的业务需求描述应包括以下内容:

- a) 数据交换目的: 数据交换的目的与应实现的目标;
- b) 数据交换参与方: 数据发送方与接收方的角色, 以及双方的交互;
- c) 数据交换流程: 工程项目中数据交换的流程图, 以及流程图中每个数据交换场景的概况;
- d) 数据内容概述: 数据交换涉及的数据格式与内容范围的概况。

6.2.1.3 IDM 交换需求

IDM中的交换需求描述应包括对信息单元与信息约束的列举。

- a) 列举单一对象信息单元：将业务需求中涉及到的对象按照其功能、用途、形态等特征进行分类，说明将某一类对象与其它类型对象区分开的特征。
- b) 列举对象与特性、关系、表现组合的信息单元：列出每种对象涉及的特性、关系、表现信息，以及上述信息与业务需求中的项目的对应关系。

- c) 列举信息约束：针对每个信息单元列出内容限制与使用限制描述，以及上述约束与业务需求中的项目的对应关系。其中：
- 1) 内容限制可包括对象命名要求、对象分类标识要求、特性类型要求、特性取值要求、几何表现要求、颜色表现要求、材质表现要求等；
 - 2) 使用限制包括对象、特性、关系、表现是否存在的要求，以及特性、关系、表现应当满足的数量限制、唯一性限制等。

6.2.2 IDM 引用语义字典

在实施对象分类以及描述对象类型的识别特征的过程中，应引用针对工程领域对象的语义字典标准或分类标准。当IDM中的对象名称与所引用的分类标准不一致时，应按照语义字典的规定进行领域规范术语映射。

6.2.3 IDM 文档成果

IDM文档是编制IDM的成果。IDM文档成果中除了涵盖各种局部对象的业务需求与交换需求外，还应涵盖模型文件要求与模型全局要求。

- a) 模型文件要求：信息模型的文件格式、文件命名规则等要求。
- b) 模型全局要求：信息模型中全局唯一的或具有全局影响的信息单元与信息约束，如模型的基点坐标、计量单位、空间组织结构等要求。

6.3 根据交换需求创建 MVD

6.3.1 MVD 的创建步骤

根据交换需求创建MVD的过程包括以下步骤：

- a) 选择 MVD 引用的信息模式：MVD 必须针对特定的信息模式制定，应根据业务需求，确定一个或多个引用的信息模式；
- b) 转译信息单元与信息约束：针对信息模式，使用 MVD 语言将信息单元与信息约束转译为 MVD 规则；
- c) MVD 校验模型准备：创建一个或多个信息模型，并声明完备性检查或模型视图抽取的预期结果，用于校验 MVD 的正确性；
- d) MVD 规则正确性校验：在 MVD 校验模型上执行完备性检查或模型视图抽取，将实际结果与预期结果进行比较，并修正 MVD 规则。

6.3.2 选择 MVD 引用的信息模式

MVD引用的信息模式的选择应考虑以下因素：

- a) 信息模式对信息单元中的对象、特性、关系、表现的支持情况，数据交换上下游软硬件系统支持输入、输出、转换的信息模式的情况，以及此过程中的信息损耗情况；
- b) 信息模式的可扩展性、可编辑性、可维护性；
- c) 用于编辑符合此信息模式的模型的工具的授权、使用期限等情况；
- d) 数据交换双方角色的其它约定。

6.3.3 转译信息单元与信息约束

使用MVD语言将信息单元与信息约束转译为MVD规则应包括：

- a) 从信息单元中的对象类型到信息模式中的根节点类型的映射；

- b) 对象类型的识别特征在 MVD 定义规则中的表达;
- c) 从信息单元中的特性、关系、表现到子图模板中的子图路径的映射;
- d) 针对某信息单元的信息约束在 MVD 约束规则中的表达。

6.3.4 MVD 校验模型准备

6.3.4.1 完备性检查的预期结果声明

MVD校验模型准备中，完备性检查的预期结果声明应包括：

- a) 信息单元中的对象类型与信息模型中的根节点集的对应关系;
- b) 根节点集中每个元素的特性、关系、表现的存在性与数量的声明;
- c) 根节点集中每个元素对某条信息约束的符合性的声明。

6.3.4.2 模型视图抽取的预期结果声明

MVD校验模型准备中，模型视图抽取的预期结果声明应包括：

- a) 信息单元中的对象类型与信息模型中的根节点集的对应关系;
- b) 根节点集中每个元素是否应包含在被抽取的模型视图中的声明;
- c) 根节点集中每个被包含在被抽取的模型视图中的元素是否应具有某项特性、关系、表现的声明。

6.3.5 MVD 规则正确性校验

6.3.5.1 MVD 规则正确性校验的目标

MVD规则正确性校验应在IDM编制者、MVD创建者、MVD执行应用开发者三者之间进行，以保证以下方面的正确性：

- a) IDM 中的描述以及 IDM 读者对其的理解与 IDM 编制者的真实意图符合;
- b) MVD 中的计算机可执行规则与 IDM 中的描述符合，且与其引用的信息模式兼容;
- c) MVD 校验模型已按照 IDM 的描述声明了预期的 MVD 执行结果;
- d) 执行 MVD 的结果与声明的预期结果符合。

6.3.5.2 MVD 规则正确性校验的实施方法

应按照第7章的要求实施MVD规则正确性校验，其中用于完备性检查的MVD规则的正确性校验按照7.1所述方法执行，用于模型视图抽取的MVD规则的正确性校验按照7.2所述方法执行。

7 MVD 执行

7.1 完备性检查

7.1.1 完备性检查的目的

基于MVD对信息模型执行完备性检查，以确认信息模型是否满足特定的交换需求。

7.1.2 完备性检查的流程

MVD完备性检查的流程应按照GB/T 36456.3所述方法执行。

7.1.3 子图表达式规则与测试项的对应

在基于MVD的完备性检查中，MVD中的一条定义规则对应一个根概念，MVD中的一条约束规则对应一个测试项。在完备性检查执行时，先执行定义规则，后执行约束规则。

7.1.4 子图表达式规则的执行步骤

一条MVD子图表达式规则的执行步骤包括根节点集获取、子图匹配、子图过滤、结果输出四个步骤，如图4所示。

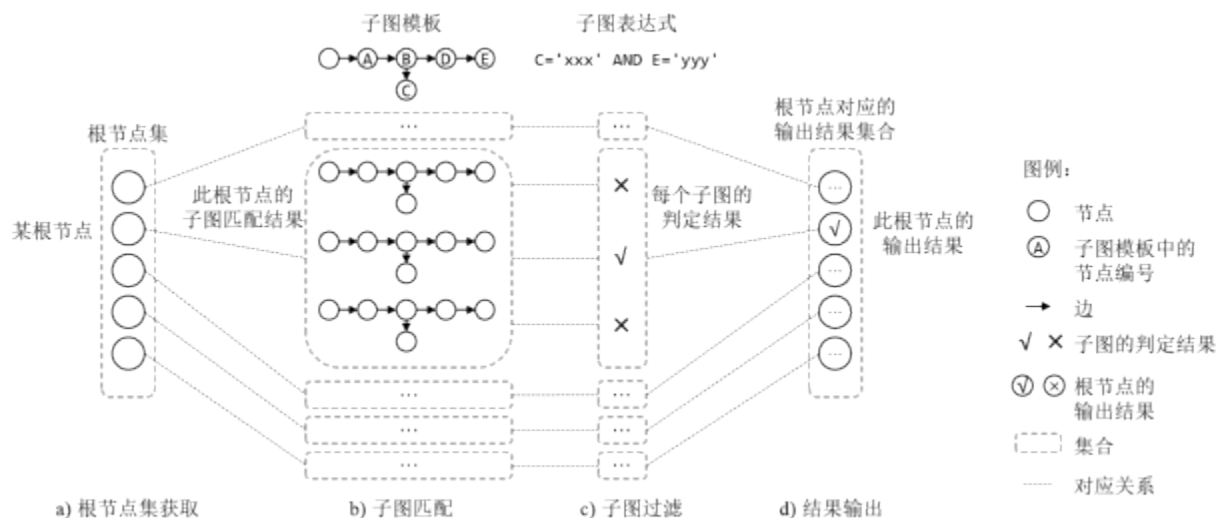


图4 子图表达式规则执行步骤

a) 根节点集获取

根节点集获取步骤中，从信息模型中获取一个节点集作为约束规则的根节点集。

- 1) 如果约束规则不引用定义规则而直接引用信息模式中的根节点类型，则使用信息模型中根节点类型对应的节点集作为根节点集。
- 2) 如果约束规则引用定义规则，则根据定义规则获取根概念对应的节点集作为根节点集。

b) 子图匹配

子图匹配步骤中，根据约束规则引用的子图模板，在信息模型中查找以根节点集中的每个元素为起始节点的所有匹配的子图实例。匹配子图实例包括与子图模板中的所有节点类型与边类型均匹配的完全匹配子图实例，以及无法找到与子图模板中某个位置的边类型与节点类型匹配的边与节点，从而自此位置之后的节点位置留空的部分匹配子图实例。

c) 子图过滤

子图过滤步骤中，针对子图实例集合中的每个子图实例使用子图表达式进行测试，并根据结果将子图实例标记为“符合”或“不符合”。

d) 结果输出

结果输出步骤中，输出的结果为根节点集中每个元素的“符合”与“不符合”的结果。如果存在以某根节点为起始节点的且标记为“符合”的子图实例，则此根节点的结果为“符合”，否则结果为“不符合”。

其中，对于定义规则，结果为“符合”的根节点组成定义规则对应根概念的节点集；对于约束规则，结果为“符合”的根节点组成通过测试项的节点集，结果为“不符合”的根节点组成不通过测试项的节点集。

注：例如执行如下一条定义规则：

定义

养老院 继承 IfcBuilding

养老院->类型标签 = '养老院'

约束

养老院->实例属性集->实例属性(->属性名称='建筑高度')->属性值!=空

执行步骤如下：

- 根据定义规则，获取所有“类型标签 = '养老院'”的IfcBuilding作为根节点集；
- 根据约束规则“->实例属性集->实例属性->属性值”搜索每个根节点对应的匹配或部分匹配子图；
- 在上一步的搜索结果中，根据“属性名称='建筑高度'”及“属性值!=空”过滤子图，将符合该条件的子图标记为“符合”，反之标记为“不符合”；
- 对根节点集中的每个根节点，如果存在标记为“符合”的子图，则此根节点输出结果为真；否则此根节点输出结果为假。

7.2 模型视图抽取

7.2.1 目的

基于MVD，可对信息模型执行模型视图抽取，从信息模型中抽取出一个轻量化子集，并形成新的信息模型。

7.2.2 模型视图抽取的输出

模型视图抽取生成的新的信息模型应符合下列要求：

- a) 符合原信息模型的信息模式的数据分类与数据结构约束要求；
- b) 交换需求中声明需要的对象、特性、关系、表现的信息包含在此信息模型中；
- c) 交换需求中要求排除的对象、特性、关系、表现的信息不包含在此信息模型中。

7.2.3 模型视图抽取的规则集

模型视图抽取使用的规则集由MVD中的定义规则、约束规则以及信息模式中的关键边保留规则组成。其中：

- a) MVD 中的一条定义规则对应一个根概念，其用途与 6.2 中所述相同；
- b) MVD 中的一条约束规则表示一项信息单元存在性声明，分为根节点保留规则、根节点排除规则、路径保留规则、路径排除规则四种类型；
- c) 关键边保留规则是由信息模式中规定为必须存在的边类型组成的规则，其目的是使模型视图抽取生成的新的信息模型符合原信息模型的信息模式的数据分类与数据结构约束要求。

7.2.4 执行模型视图抽取的软件功能

执行模型视图抽取的软件模块应包括以下功能：

- a) 根概念定义规则执行与根节点查找：根据定义规则，查找每个需要抽取的根概念在信息模型中对应的根节点集；

- b) 信息单元存在性标记：根据信息单元存在性声明与关键边保留规则，对信息模型中的节点与边进行存在性标记，将其标记为“保留”或“排除”；
- c) 数据抽取与输出：抽取标记为“保留”的节点与边，并输出为新的信息模型。

7.2.5 规则执行结果

信息单元存在性标记与数据抽取与输出的执行结果应符合以下要求：

- a) 根节点保留规则：符合保留规则的根节点集中的节点标记为“保留”，并包含在输出的信息模型中；
- b) 根节点排除规则：符合排除规则的根节点集中的节点标记为“排除”，并在输出的信息模型中不出现；
- c) 路径保留规则：从标记为“保留”的根节点集出发的、符合路径保留规则的子图实例，路径中沿途所有节点与边标记为“保留”，并包含在输出的信息模型中；
- d) 路径排除规则：从标记为“保留”的根节点集出发的、符合路径排除规则的子图实例，路径的最后一段边标记为“排除”，并在输出的信息模型中不出现；
- e) 关键边保留规则：与标记为“保留”的节点与边相关的、符合关键边保留规则的节点与边标记为“保留”，并包含在输出的信息模型中。

7.2.6 规则优先级

在信息单元存在性标记中，当出现规则之间的冲突时，排除规则的优先级高于保留规则，根节点规则的优先级高于路径规则。

8 MVD 管理机制

8.1 交换需求管理

交换需求管理符合如下要求：

- a) 数据交换的需求方应整理描述业务需求与交换需求所需的术语概念，在 IDM 中统一术语定义；
- b) 数据交换的需求方应编制适于描述 IDM 的文档模板；
- c) MVD 所涵盖的工程信息模型交换场景，应符合工程建设的实际流程需要，涵盖设计协同、设计算量、设计施工深化、竣工验收等用例流程。各场景的应用要求标准，应符合相关国标或行业标准要求。

8.2 MVD 成果管理

MVD成果管理符合如下要求：

- a) MVD 成果管理应覆盖 IDM 文档与 MVD 规则文件；
- b) 针对同一需求、由不同方编制的 MVD 规则文件，应在统一的评审机制下，完成去重、冲突解决和合并等工作，最终形成统一的 MVD 规则文件存档；
- c) 应对 MVD 成果进行分类和索引，备注来源、历史、继承或变更信息；
- d) MVD 规则文件中，同一交换需求可针对不同交换场景对应多个不同的约束规则；此时应使用标记项明确规定其对应的场景，并在 MVD 执行中据此筛选适用于当前场景的规则。

8.3 MVD 软件应用管理

MVD软件应用管理符合如下要求：

- a) BIM 软件调用MVD 执行时，应确保当前使用的 BIM 信息模式与交换场景与 MVD 规则文件声明的信息模式与适用场景一致；
- b) 针对 MVD 的内容，宜配套形成面向主要 BIM 软件的配置文件，以提高用户建模效率；
- c) BIM 软件宜支持客户端与云平台之间的 MVD 规则同步与更新。

9 MVDLite 语言

9.1 基本规定

9.1.1 MVDLite 是一种 MVD 语言，其语法由词法、句型结构、标记三部分规则组成。其中：

- a) 词法规则规定了 MVDLite 语言中的标识符、数值、字符串、符号、关键字等成分表示方法，应符合附录 A.1 的规定；
- b) 句型结构规则规定了 MVDLite 语言表示子图模板规则、定义规则、约束规则的方法，应符合附录 A.2 的规定；
- c) 标记规则规定了在 MVDLite 句型结构基础上附加标记信息的方法，应符合附录 A.2 的规定。

9.1.2 MVDLite 语言使用的字符集为 GB 13000 字符集。

9.1.3 MVDLite 规则文件应使用“.mvd lite”或“.mdl”作为扩展名。

9.1.4 适用于 IFC 信息模式的常用 MVDLite 模板参见附录 B。

9.2 词法规则

9.2.1 标识符

标识符是 MVDLite 语言中用于表示节点、边、子图路径、标记的名称，类似于编程语言中的变量名。

标识符由大小写字母 (A-Z、a-z)、数字 (0-9)、汉字字符、下划线 () 组成，且首字符不能为数字。其中汉字字符为 GB 13000 字符集中从 \u4e00 到 \u9fa5 之间的字符。

当工程领域术语中含有超出上述字符范围的特殊符号，然而必须使用此带有特殊符号的术语作为标识符时，应使用标识符字符串表示，以代替通常的标识符。

9.2.2 整数

整数由 1 至多个数字，以及可选的符号前缀组成。

符号前缀包括正号 (+) 与负号 (-)，其中正号可以省略。

9.2.3 浮点数

浮点数由以小数点分隔的两组 0 至多个数字，以及可选的符号前缀与指数后缀组成。其中至少有一组数字的长度不为 0。

普通浮点数由以小数点分隔的两组 0 至多个数字，以及可选的符号前缀组成，其中至少有一组数字的长度不为 0。

科学计数法浮点数表示 10 进制科学计数法，由一个整数或普通浮点数作为前缀，以字母 e 或 E 后接表示指数的另一个整数作为后缀。

9.2.4 值域

值域表示实数轴上的一个连续闭区间。值域表示分为上下界型值域与中心误差型值域两种，具体内容如下：

- a) 上下界型值域由波浪线 () 连接的左右两个浮点数或整数组成，分别表示闭区间的下界与上界，其中左侧的数值应更小。如：0.1~0.9；

- b) 中心误差型值域由正负号（+-）连接的左右两个浮点数或整数组成，左侧为中心值，右侧为误差值，其中右侧值应为正数。如：3.5+-0.01。

9.2.5 字符串与转义字符

字符串由一对半角双引号（"）或一对半角单引号（'）包围的0至多个字符组成。字符串中的内容不能超过一个物理行，其中的换行符（\n）与回车符（\r）必须使用转义表示。

字符串中的转义字符由反斜杠（\）后接特定字符组成，用于表示特殊字符。需要转义的字符如表1所示。

表1 MVDLite 转义字符

符号名称	符号表示
双引号	\"
单引号	\'
反斜杠	\\
换行符	\n
回车符	\r
制表符	\t

9.2.6 部分匹配字符串

部分匹配字符串是在字符串前增加表示部分匹配的符号，用于表示MVD规则中对字符串值进行部分匹配判定的运算。

部分匹配的符号类型如表2所示。

表2 MVDLite 部分匹配字符串

部分匹配类型	部分匹配符号	示例	满足部分匹配条件的字符串样例
前缀匹配	s	s"app"	"apple", "apply", "append"
后缀匹配	e	e"ing"	"doing", "interesting", " boring"
包含匹配	c	c"ly"	"lyrics", "happily", "flying"

9.2.7 正则匹配字符串

正则匹配字符串是MVD规则中对字符串值进行正则表达式匹配判定的运算。

正则匹配字符串以表示正则匹配的符号（r）开头，后接由一对半角双引号（"）或一对半角单引号（'）包围的正则表达式。

9.2.8 标识符字符串

标识符字符串是在字符串前增加表示标识符的符号（\$），在MVDLite句型结构中的作用等价于标识符。

由于标识符中不允许出现空白字符（空格、换行符等）与除下划线以外的特殊符号，因此可以使用标识符字符串来表示包含空白字符与特殊符号的标识符。

9.2.9 逻辑值及其它特殊值

MVDLite中的逻辑值及其它特殊值如表3所示。

表3 MVDLite 逻辑值及其它特殊值

英文表示	中文表示
TRUE	真
FALSE	假
UNKNOWN	未知

英文表示	中文表示
NULL	空
SELF	自身

9.2.10 度量符号

MVDLite中的度量符号如表4所示。

表4 MVDLite 度量符号

英文度量符号	中文度量符号
[Value]	[值]
[Type]	[类型]
[Size]	[数量]
[Exists]	[存在]
[Unique]	[唯一]

9.2.11 比较符号

MVDLite中的比较符号如表5所示。

表5 MVDLite 比较符号

符号表示	描述
=	等于
!=	不等于
>	大于
>=	大于或等于
<	小于
<=	小于或等于

对于各种数据类型的[Value]度量，比较符号的含义如表6所示。

表6 MVDLite[Value]度量比较符号的含义

左侧数据类型	右侧数据类型	比较符号含义
--------	--------	--------

字符串	字符串	按字典序排序的字符串顺序比较
整数、浮点数	整数、浮点数	按实数轴排序的数值大小比较
	值域	左侧值与右侧值域在实数轴上的位置关系
逻辑值	逻辑值	只支持“=”与“!=”符号

对于[Type]度量，比较符号的含义如表7所示。

表7 MVDLite[Type]度量比较符号的含义

符号表示	含义
A = B	A与B是同一个类型
A != B	A与B是不同类型

符号表示	含义
$A > B$	A是B的子类
$A < B$	A是B的父类
$A \geq B$	A是B的子类或者A与B相同
$A \leq B$	A是B的父类或者A与B相同

注：此处将概念的分类系统抽象为横向的树结构，“ $A < B$ ”表示“A作为父类有若干子类分支，其中一个分支是B”；“ $A > B$ ”表示“B作为父类有若干子类分支，其中一个分支是A”。

9.2.12 逻辑运算符

MVDLite中的逻辑运算符如表8所示。

表8 MVDLite 逻辑运算符

英文表示	中文表示	符号表示
AND	且	&
OR	或	
XOR	异或	^
NOT	不、非	!

9.2.13 其它符号

MVDLite中的其它符号如表9所示，其中字符均为半角符号字符。

表9 MVDLite 其它符号

符号名称	符号表示
指向	->
左括号	(
右括号)

9.2.14 关键字

MVDLite中的关键字如表10所示。

表10 MVDLite 关键字

英文关键字	中文关键字

definition	定义
constraint	约束
as	作为
extends	继承

9.2.15 空格

在MVDLite语言中，词法规则中所述各成分之间可以使用1至多个空格分隔。当两个词法成分连写不会造成歧义时，空格可以省略。

空格包括空格符、制表符、换行符、回车符四种。

空格不参与语法解析，且不能出现在除字符串、部分匹配字符串、正则匹配字符串、标识符字符串以外的其它词法成分中。

在字符串、部分匹配字符串、正则匹配字符串、标识符字符串中，空格符正常书写，制表符、换行符、回车符应使用转义字符表示。

9.2.16 注释

在MVDLite语言中，注释内容不参与语法解析，仅在人类直接阅读MVDLite时提供辅助说明。注释分为单行注释与多行注释两种。

- a) 单行注释以双斜杠 (//) 开始，直到行尾结束。
- b) 多行注释以斜杠星号 (/*) 开始，以星号斜杠 (*/) 结束。

9.3 句型结构规则

9.3.1 规则文件

规则文件由头部的预定义缩写与若干正文块组成，其中预定义缩写可省略。

9.3.2 预定义缩写

预定义缩写位于规则文件头部，由若干缩写表达式构成。

9.3.3 正文块

正文块是规则文件中用于组织MVD规则的主要结构。一个规则文件可以具有0至多个正文块。

正文块分为定义块与约束块两种，每个正文块由关键字“定义”或“约束”开头，直到下一个“定义”或“约束”关键字之前，或文件结尾之前结束。

9.3.4 定义块

定义块是表示MVD定义规则的正文块，用于规定在信息模型中查找某根概念对应的节点集的规则。定义块由关键字“定义”开头，其内部可含有若干概念表达式与规则表达式。

9.3.5 约束块

约束块是表示MVD约束规则的正文块，用于规定信息模型中与子图模板匹配的子图实例应当满足的约束规则。

约束块由关键字“约束”开头，其内部可含有若干规则表达式。

9.3.6 缩写表达式

缩写表达式用于将MVDLite的规则链中的一些常用的规则段序列缩写为一个标识符，以便于其在规则表达式中引用。

- a) 缩写表达式由以关键字“as”连接的前后两部分组成。前部分为一个标识符，是缩写表达式的名称；后部分以括号包围的根节点类型开头，后接一个由若干中间规则段组成的序列。其中根节点类型可为空，此时表示不限制根节点类型。

- b) 在正文的规则表达式中，可以使用缩写表达式的标识符代替其后部分的规则段序列。
- c) 在缩写表达式的后部分的规则段序列中，也可以嵌套使用其它缩写标识符，但是不允许形成循环引用。

9.3.7 概念表达式

概念表达式用于声明一个新的扩展节点类型，作为已有的节点类型的一个子类。

- a) 概念表达式由以关键字“extends”连接的前后两部分组成。前部分为一个标识符，是新的扩展节点类型的名称；后部分为一个已有的节点类型，或者多个已有的节点类型的并集，用“或”分隔（可以为汉字“或”，大写英文“OR”，或者竖线“|”）。
- b) 概念表达式只能出现在定义块中，并与定义块中的其它以此节点类型为根概念的规则表达式共同组成此节点类型的定义规则。

注：以一个节点类型为根概念的规则表达式允许出现在与概念表达式所处的定义块不同的其它定义块中。

9.3.8 规则表达式

规则表达式是正文块中表示定义规则或约束规则的主要结构。

- a) 规则表达式可以为一条单一的规则语句，也可以为若干条规则语句、逻辑连接符号、括号的组合。
- b) 当规则表达式中含有多条规则语句时，每条规则语句的根概念应相同。
- c) 当使用逻辑连接符号为“XOR”时，当前语句（或当前括号内的子句）只能有两项。
- d) 当逻辑连接符号“NOT”只能位于当前语句（或当前括号内的子句）的开头，且此句只能有一项。

9.3.9 规则语句

规则语句是规则表达式中的基本单元。

- a) 一个规则语句由一个根概念后接一个度量规则链组成，表示根概念对应的节点集上的一个过滤条件。
- b) 规则语句执行的结果是根概念对应的节点集中满足度量规则链的节点子集。

9.3.10 规则链

规则链是由若干规则段串联组成的链条，是MVDLite句型结构中用于表示一段子图路径以及这段子图路径上的规则的结构。

规则链按照功能分为度量规则链与路径规则链两种，以其结尾规则段的类型作为区分。

9.3.11 度量规则链

度量规则链表示对此规则链之前的节点集的过滤条件，应以度量段、对偶度量段或复合度量段结尾。

9.3.12 路径规则链

路径规则链表示以此规则链之前的节点集为起点的子图路径，应以属性段或复合属性段结尾。

9.3.13 规则段

规则段分为属性段、度量段、对偶度量段、复合属性段、复合度量段五种。

其中，度量段只能出现在规则链的结尾，不能出现在规则链的中间；其它四种规则段可以出现在规则链的任意位置。

注1：如需以度量段作为规则链结尾，应使用括号将度量段封装为复合度量段。

注2：如需将一个以复合度量段结尾的规则链作为路径规则链使用，应在其后添加一个“->SELF”属性段。

9.3.14 属性段

属性段是表示子图路径中的一个边类型的规则段，形如“->边类型标识符:目标节点约束标识符”。其中“:目标节点约束标识符”可省略，表示不限制此边类型的目标节点类型。

属性段的目标节点集为目标节点的无序不重复集合。具体分为以下情况：

- a) 当边指向单一目标节点时，将目标节点加入目标节点集；
- b) 当边指向包含多个节点的节点集时，将其中的每个节点加入节点目标节点集；
- c) 当边指向包含两层以上嵌套的节点集时，仅展开最外层集合，将其内部的每个嵌套集合看作一个目标，加入目标节点集。

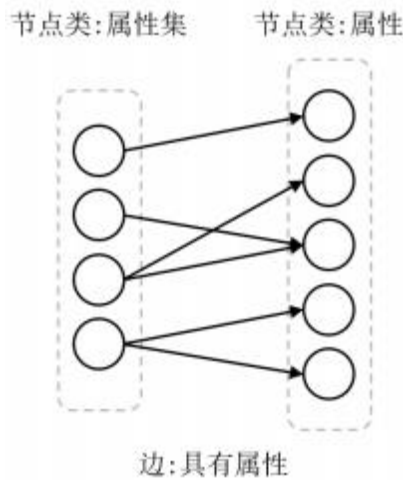


图5 属性段示例

注：边类型标识符允许使用一个缩写表达式名称，此时使用此缩写对应的规则段序列替换此属性段；如果存在“:”表示的目标节点类型约束，则将此约束应用于规则段序列的末尾。

9.3.15 度量段

度量段是表示对当前节点集的一个过滤条件的规则段，由一个比较符号与左侧的度量符号、右侧的值组成，如“[Size] > 3”。其中，当度量符号为“[Value]”时，度量符号可以省略。

根据度量段的行为的不同，度量段分为单度量段与集合度量段两类。

9.3.16 单度量段

单度量是可以由单一节点确定的度量，其输出的节点集是对当前输入节点集的过滤。

单度量段的符号为“[Value]”与“[Type]”两种。

- a) [Value]度量段根据当前输入节点集的每个节点的值与比较符号右侧的值的比较结果，对当前输入节点集进行过滤。当度量符号为“[Value]”，且值为部分匹配字符串或正则匹配字符串时，比较符号只能为“=”或“!=”，示例参见图 6。
- b) [Type]度量段根据当前输入节点集的每个节点的类型与比较符号右侧的类型在节点的分类体系中的关系进行过滤。当度量符号为“[Type]”时，比较符号右侧只能为字符串值，表示节点类型名称；此时比较符号的示例参见图 7。

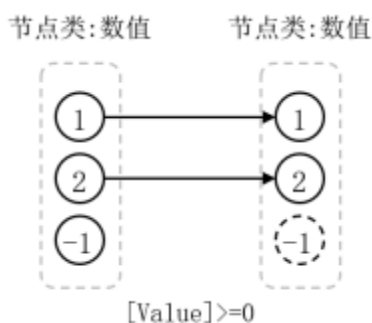


图6 [Value]度量段示例

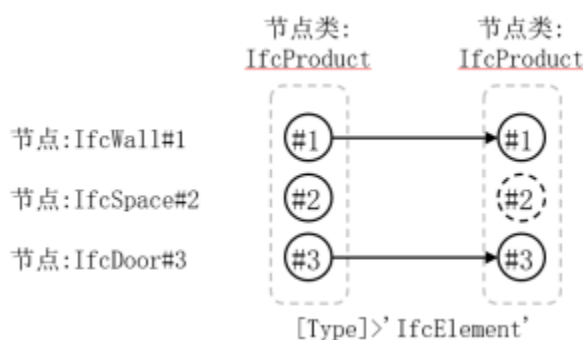


图7 [Type]度量段示例

9.3.17 集合度量段

集合度量段是由当前节点集与其以属性段或复合属性段关联的上一级节点集的引用关系进行判定的度量，其输出的节点集是对上一级节点集的过滤。

集合度量段符号为 “[Size]” “[Exists]” “[Unique]” 三种。

- [Size]度量段根据上一级节点集中的每个节点指向当前节点集中节点的数量，对上一级节点集进行过滤。当度量符号为 “[Size]” 时，值只能为整数。
- [Exists]度量段根据上一级节点集中的每个节点是否存在指向当前节点集中节点，对上一级节点集进行过滤。当度量符号为 “[Exists]” 时，比较符号只能为 “=” 或 “!=”，值只能为逻辑值。
- [Unique]度量段根据上一级节点集中的每个节点指向当前节点集中的节点是否与其它上一级节点互斥，对上一级节点集进行过滤。当指向存在且互斥的情况下[Unique]度量为 TRUE，否则为 FALSE。当度量符号为 “[Unique]” 时，比较符号只能为 “=” 或 “!=”，值只能为逻辑值。

注1：由于度量段只能出现在规则链的结尾，因此上一级节点集是当前规则链中的最后一个属性段或复合属性段的输入节点集。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：

<https://d.book118.com/686224030032010215>