# ICS

**中国建筑业协会团体标准 团体标准**

**P**  T/CCIAT xxxx— 20xx

建筑信息模型数字成果元数据标准

Metadata for digital products of fundamental

building information modeling

（征求意见稿）

**20xx— xx—xx 发布　　　20xx—xx —xx 实施**

**中国建筑业协会 发布**

中国建筑业协会团体标准

建筑信息模型数字成果元数据标准

Metadata for digital products of fundamental

building information modeling

T/CCIAT xxxx— 20xx

批准部门：中国建筑业协会

施行日期：20xx年xx月xx日

中国建筑工业出版社

20xx 北京

前  言

根据中国建筑业协会《关于开展第七批团体标准编制工作的通知》（建协函[2023]4号）的要求，标准编制组经过深入调查研究，认真总结工程实践经验，参考了国内外相关技术标准，进行了必要的理论研究和验证试验，并在广泛征求意见的基础上，制订本标准。

本标准共主要技术内容：1.总则、2.术语、3.基本规定、4.项目模型元数据、5.构件库元数据、附录A项目模型元数据定义、附录B构件模型元数据定义、附录C构件库元数据定义。

本标准（规范、规程）由中国建筑业协会负责管理，由中建工程产业技术研究院有限公司负责具体技术内容的解释。请各单位在执行过程中，总结实践经验，积累资料，随时将有关意见和建议反馈给中建工程产业技术研究院有限公司（地址：北京顺义林河大街15号；邮政编码：101300）。

本标准主编单位：中建工程产业技术研究院有限公司

本标准参编单位：×××、×××

本标准主要起草人员：×××、×××

本标准主要审查人员：×××、×××

目 次

[1 总则 1](#_Toc31261)

[2 术语与缩略语 2](#_Toc30786)

[3 基本规定 3](#_Toc30485)

[4 项目模型元数据 4](#_Toc9863)

[4.1 一般规定 4](#_Toc3429)

[4.2 外部依赖 5](#_Toc20063)

[4.3 分布式管理 5](#_Toc27424)

[5 构件库模型元数据 7](#_Toc29822)

[5.1 一般规定 7](#_Toc32114)

[5.2 构件定义 8](#_Toc26521)

[5.3 分布式管理 9](#_Toc32142)

[附录A：元数据定义开源项目 10](#_Toc21538)

[附录B：项目模型元数据定义 11](#_Toc1534)

[附录C：构件库元数据定义 32](#_Toc4819)

[本标准用词说明 53](#_Toc31619)

[引用标准名录 54](#_Toc28822)

[条文说明 55](#_Toc11751)

# 总则

* + 1. 为支持建筑信息模型创建相关成果有序管理，指导企业建筑信息模型资源库建设工作，制定本标准。
    2. 本标准适用于建筑信息模型和构件库的生产、分发及应用，其他数字成果可参照使用，不包含其他应用成果的
    3. 建筑信息模型数字成果的元数据，除应符合本标准及配套标准外，尚应符合国家、行业现行有关标准的规定。

# 术语

* + 1. 数字成果digital product

按一定的数据组织方式，以数字形式表达和存储的成果。

* + 1. 建筑信息模型数字成果BIM digital results

建筑信息模型应用产生的数字成果，在本标准里特指项目模型和构件库。

* + 1. 元数据 metadata

关于描述数据信息对象的数据，是对某个潜在信息性对象做出的陈述。

* + 1. 发布库repository

应用元数据采集、存储、管理和分享建筑信息模型数字成果的管理系统。

* + 1. 外部依赖dependency

特指项目模型对其他模型的依赖或引用。

# 基本规定

1. 宜将建筑信息模型数字成果元数据标准用于本地数据管理和远程存储库管理。
2. 本标准参照《元数据的XML Schema置标规则》（GB/T 24639-2009）定义元数据，建立了开源项目（详见附录A），从元数据定义模型（详见附录B和附录C）生成元数据XML模式定义，以及XML和JSON格式的元数据文件Java读写代码。应用其他技术可参考执行。
3. 元数据中宜包括问题跟踪系统的集成信息（详见附录B和附录C中的IssueManagement定义及引用）。
4. 元数据中宜包括版本管理系统的集成信息（详见附录B和附录C中的Scm定义及引用）。
5. 元数据中宜包括许可协议的信息（详见附录B和附录C中的License定义及引用）。
6. 元数据中宜包括创建人及审核人的信息（详见附录B和附录C中的Person定义及引用）。

# 项目模型元数据

## 一般规定

### 项目模型元数据应包含所用元数据标准的版本信息（详见附录B中Model的metaVersion属性定义），当前版本为1.0.0。

### 项目模型元数据宜添加分组信息（详见附录B中Model的groupId属性定义）。

### 项目模型元数据应包含标识信息（详见附录B中Model的modelId属性定义），标识信息不可以包含\、/、:、\*、?、"、<、>、|等特殊字符。

### 项目模型元数据应包含版本信息（详见附录B中Model的version属性定义）。

### 项目模型元数据应包含存储格式信息（详见附录B中 Model的packaging属性定义）。

### 项目模型元数据宜添加标签信息（详见附录B中Model的tags属性定义）。

### 项目模型元数据宜添加备注信息（详见附录B中Model的description属性定义）。

### 项目模型元数据宜添加创建时间信息（详见附录B中 Model的creationTime属性定义），时间表示法应符合国家标准GB/T 7408-2005《数据元和交换格式·信息交换·日期和时间表示法》中定义的扩展格式表示，YYYY-MM-DDThh:mm:ss。

### 项目模型元数据宜添加组织信息（详见附录B中Model的organization属性定义）。

### 项目模型元数据宜通过属性信息扩展定义（详见附录B中Model的properties属性定义）。

## 外部依赖

### 项目模型宜包含外部依赖信息（详见附录B中Model的dependencies属性定义）。

### 外部依赖元数据应包括分布式管理、标识信息、存储格式、版本等信息，宜包括分组信息（详见附录B中Dependency的repository、groupId、modelId、packaging、version属性定义）。

### 外部依赖元数据可包括排除信息（详见附录B中 Dependency的exclusions属性定义）。

### 外部依赖元数据可包括类型信息（详见附录B中 Dependency的type属性定义），缺省是依赖dependency，也可为引用reference，对引用不计算进一步的依赖关系。

## 分布式管理

### 项目模型元数据宜包括发布库信息（详见附录B中Model的repository属性定义）。

### 发布库元数据应包含URL信息（详见附录B中Repository中的url属性定义）。

### 发布库元数据宜包含命名信息（详见附录B中Repository的name属性定义）。

# 构件库模型元数据

## 一般规定

### 构件库元数据应包含所用元数据标准的版本信息（详见附录C中Model的metaVersion属性定义），当前版本为1.0.0。

### 构件库元数据宜添加分组信息（详见附录C中Model的groupId属性定义）。

### 构件库元数据应包含标识信息（详见附录C中Model的libraryId属性定义），标识信息不可以包含\、/、:、\*、?、"、<、>、|等特殊字符。

### 构件库元数据应包含版本信息（详见附录C中Model的version属性定义）。

### 构件库元数据应包含存储格式信息（详见附录C中Model的packaging属性定义）。

### 构件库元数据宜添加标签信息（详见附录C中Model的tags属性定义）。

### 构件库元数据宜添加备注信息（详见附录C中Model的description属性定义）。

### 构件库元数据宜添加创建时间信息（详见附录C中Model的creationTime属性定义），时间表示法应符合国家标准GB/T 7408-2005《数据元和交换格式·信息交换·日期和时间表示法》中定义的扩展格式表示，YYYY-MM-DDThh:mm:ss。

### 构件库元数据宜添加组织信息（详见附录C中Model的organization属性定义）。

### 构件库元数据宜通过属性信息扩展定义（详见附录C中 Model的properties属性定义）。

## 构件定义

### 构件库应至少包含至少一项构件定义（详见附录C中 Model的components属性定义）。

### 构件元数据应包含构件名称（详见附录C中Component的name属性定义）。

### 构件元数据宜添加分类名称（详见附录C中Component的classificationName属性定义）。

### 构件元数据宜添加分类编码（详见附录C中Component的classificationCode属性定义）。

### 构件元数据宜包含所用分类系统名称（详见附录C中 Component的classificationSystem属性定义），缺省值是GB/T 51269-2017。

### 构件元数据宜添加注释（详见附录C中Component的classificationCode属性定义）。

### 构件元数据宜添加标签信息（详见附录C中Component的tags属性定义）。

### 构件元数据宜添加缩略图信息（详见附录C中Component的thumbnailURL属性定义）。

## 分布式管理

### 构件库元数据宜包含分布式管理信息（详见附录C中Model的distributionManagement属性定义）。

### 分布式管理元数据应包括发布库信息（详见附录C中DistributionManagement的repository属性定义）。

### 发布库元数据应包含URL信息（详见附录C中Repository的url属性定义）。

### 发布库元数据宜包含命名信息（详见附录C中Repository的name属性定义）。

### 分布式管理元数据宜包括网站信息（详见附录C中DistributionManagement的site属性定义及Site定义）。

### 分布式管理元数据可构件库的下载地址信息详见附录C中DistributionManagement的downloadUrl属性定义）。

附录A：元数据定义开源项目

标准编写组建立了与标准内容对应的开源项目，支持标准的落地实施。

开源项目网址：[https://github.com/zgjzy/zgjzy-bim-datameta，开源项目采用Apache 2.0](https://github.com/QiuKuining/zgjzy-bim-datameta，源项目采用Apache%202.0)开源协议。

附录B：项目模型元数据定义

项目模型元数据定义采用Modello模型定义，内容如下：

<?xml version="1.0" encoding="UTF-8"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one

or more contributor license agreements. See the NOTICE file

distributed with this work for additional information

regarding copyright ownership. The ASF licenses this file

to you under the Apache License, Version 2.0 (the

"License"); you may not use this file except in compliance

with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,

software distributed under the License is distributed on an

"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

KIND, either express or implied. See the License for the

specific language governing permissions and limitations

under the License.

-->

<model xmlns="http://codehaus-plexus.github.io/MODELLO/2.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://codehaus-plexus.github.io/MODELLO/2.0.0

https://codehaus-plexus.github.io/modello/xsd/modello-2.0.0.xsd"

xml.namespace="http://zgjzy.org.cn/model-meta/${version}"

xml.schemaLocation="https://zgjzy.org.cn/xsd/model-meta-${version}.xsd">

<id>model-meta</id>

<name>ModelMeta</name>

<description>

<![CDATA[

<p>这是BIM中使用的模型元描述符的参考实现。</p>

]]>

</description>

<defaults>

<default>

<key>package</key>

<value>org.zgjzy.meta.model</value>

</default>

</defaults>

<classes>

<class rootElement="true" xml.tagName="model" java.clone.hook="cloneHook" java.clone="deep">

<name>Model</name>

<description>

<![CDATA[

<code>&lt;model&gt;</code> 是元数据描述的根元素。以下是可能的子元素。

]]>

</description>

<version>1.0.0+</version>

<fields>

<!-- ================== -->

<!-- 元数据格式的版本 -->

<!-- ================== -->

<field>

<name>metaVersion</name>

<version>1.0.0+</version>

<required>true</required>

<description>

描述模型元数据格式的版本号，例如当前元数据格式的版本是1.0.0。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 模型分组结构 -->

<!-- =============== -->

<field xdoc.separator="blank">

<name>groupId</name>

<version>1.0.0+</version>

<required>false</required>

<description>

分组结构，用于根据个人、项目、企业的需求或规定，组织、管理模型数据，其中包含的\，用于分割目录，形成目录结构。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 模型标识 -->

<!-- =============== -->

<field>

<name>modelId</name>

<version>1.0.0+</version>

<required>true</required>

<description>

<![CDATA[

相当于模型的文件名，用于标识模型，命名不可以包含\/:\*?"<>|等特殊字符。

]]>

</description>

<type>String</type>

</field>

<!-- ============== -->

<!-- 模型版本 -->

<!-- ============== -->

<field>

<name>version</name>

<version>1.0.0+</version>

<required>true</required>

<description>

模型的当前版本。

</description>

<type>String</type>

</field>

<!-- ================ -->

<!-- 模型存储格式 -->

<!-- ================ -->

<field>

<name>packaging</name>

<version>1.0.0+</version>

<required>true</required>

<description>

相当于模型文件后缀，用于标识模型的格式类型，例如，ifc、dwg、dgn、rvt、skp等所有可能的文件格式，也可以是自定义。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 模型描述信息 -->

<!-- =============== -->

<field>

<name>description</name>

<version>1.0.0+</version>

<description>

详细描述模型的信息。

</description>

<type>String</type>

</field>

<!-- ================ -->

<!-- 模型创建 -->

<!-- ================ -->

<field>

<name>creationTime</name>

<version>1.0.0+</version>

<required>true</required>

<description>

时间表示法应符合国家标准GB/T 7408-2005《数据元和交换格式·信息交换·日期和时间表示法》中定义的扩展格式表示，YYYY-MM-DDThh:mm:ss。

</description>

<type>String</type>

</field>

<!-- ================== -->

<!-- 组织机构信息 -->

<!-- ================== -->

<field>

<name>organization</name>

<version>1.0.0+</version>

<description>

描述模型所属组织机构的属性，可用于声明模型的版权声明和链接。

</description>

<alias>organisation</alias>

<association>

<type>Organization</type>

</association>

</field>

<!-- =============== -->

<!-- 许可信息 -->

<!-- =============== -->

<field>

<name>licenses</name>

<version>1.0.0+</version>

<description>

描述模型的所有许可信息，若有多个许可，可以分开描述，并假定用户可以选择其中部分使用，而不是全部。

另外，模型的许可信息仅适用于本模型，不适用于依赖的模型。

</description>

<association>

<type>License</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- ==================== -->

<!-- 模型创建者信息 -->

<!-- ==================== -->

<field xdoc.separator="blank">

<name>developers</name>

<version>1.0.0+</version>

<description>

模型创建的贡献者，可以有多个。

</description>

<association>

<type>Person</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- ================== -->

<!-- 模型检查者信息 -->

<!-- ================== -->

<field>

<name>examiners</name>

<version>1.0.0+</version>

<description>

模型检查者，可以有多个。

</description>

<association>

<type>Person</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- =================== -->

<!-- 版本管理系统信息 -->

<!-- =================== -->

<field xdoc.separator="blank">

<name>scm</name>

<version>1.0.0+</version>

<description>

模型的版本管理系统信息，例如Git、Subversion、CVS等。

</description>

<association>

<type>Scm</type>

</association>

</field>

<!-- ====================== -->

<!-- 问题追踪系统 -->

<!-- ====================== -->

<field>

<name>issueManagement</name>

<version>1.0.0+</version>

<description>

模型的问题追踪系统信息。

</description>

<association>

<type>IssueManagement</type>

</association>

</field>

<field xdoc.separator="blank">

<name>repository</name>

<version>1.0.0+</version>

<description>

模型的远程存储库部署信息。

</description>

<association>

<type>Repository</type>

</association>

</field>

<field xdoc.separator="blank">

<name>properties</name>

<version>1.0.0+</version>

<description>

<![CDATA[

可以在整个元数据中作为替代使用的属性，并且在启用时用作资源中的过滤器。

格式：<code>&lt;name&gt;value&lt;/name&gt;</code>。

]]>

</description>

<type>Properties</type>

<association xml.mapStyle="inline">

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- =============== -->

<!-- 标签信息 -->

<!-- =============== -->

<field>

<name>tags</name>

<version>1.0.0+</version>

<description>

描述模型的标签信息，用于检索模型。

</description>

<association>

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<field>

<name>dependencies</name>

<version>1.0.0+</version>

<description>

模型所有的依赖项，可以自动下载到本地。

</description>

<association>

<type>Dependency</type>

<multiplicity>\*</multiplicity>

</association>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* 元数据文件。

\*/

private java.io.File metaFile;

private void cloneHook( Model copy )

{

copy.metaFile = metaFile;

}

/\*\*

\* 如果有，获取模型的元数据文件。

\*

\* @return 返回本地的模型元数据文件，如果是从远程库得到元数据信息，则返回空值 {@code null} 。

\*/

public java.io.File getMetaFile()

{

return metaFile;

}

public void setMetaFile( java.io.File metaFile )

{

this.metaFile = ( metaFile != null ) ? metaFile.getAbsoluteFile() : null;

}

/\*\*

\* 如果有，获取相应模型的根目录。

\*

\* @return 返回根目录，如果是从远程库得到元数据信息，则返回空值 {@code null} 。

\*/

public java.io.File getProjectDirectory()

{

return ( metaFile != null ) ? metaFile.getParentFile() : null;

}

/\*\*

\* @return 模型标识 <code>groupId:modelId:packaging:version</code>

\*/

public String getId()

{

StringBuilder id = new StringBuilder( 64 );

id.append( getGroupId() );

id.append( ":" );

id.append( getModelId() );

id.append( ":" );

id.append( getPackaging() );

id.append( ":" );

id.append( getVersion() );

return id.toString();

}

@Override

public String toString()

{

return getId();

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Person</name>

<description>

描述人员信息.

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>id</name>

<version>1.0.0+</version>

<description>

人员的唯一标识。

</description>

<type>String</type>

</field>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>姓名全名。</description>

<type>String</type>

</field>

<field>

<name>email</name>

<version>1.0.0+</version>

<description>

人员邮箱信息。

</description>

<type>String</type>

</field>

<field>

<name>organization</name>

<alias>organisation</alias>

<version>1.0.0+</version>

<description>

人员所属组织。

</description>

<type>String</type>

</field>

<field>

<name>roles</name>

<version>1.0.0+</version>

<description>

人员角色。

</description>

<association>

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<field>

<name>properties</name>

<version>1.0.0+</version>

<description>

人员属性。

</description>

<type>Properties</type>

<association xml.mapStyle="inline">

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "Person {name=" + name + ", email=" + email + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Dependency</name>

<version>1.0.0+</version>

<description>

<![CDATA[

The <code>&lt;dependency&gt;</code> element contains information about a dependency

of the project.

]]>

</description>

<fields>

<field>

<name>repository</name>

<version>1.0.0+</version>

<description>

模型所在库。

</description>

<association>

<type>Repository</type>

</association>

</field>

<field>

<name>groupId</name>

<version>1.0.0+</version>

<required>true</required>

<description>

<![CDATA[

The project group that produced the dependency, e.g.

]]>

</description>

<type>String</type>

</field>

<field>

<name>modelId</name>

<version>1.0.0+</version>

<required>true</required>

<description>

模型标识

</description>

<type>String</type>

</field>

<field>

<name>packaging</name>

<version>1.0.0+</version>

<required>true</required>

<description>

模型存储类型

</description>

<type>String</type>

</field>

<field>

<name>version</name>

<version>1.0.0+</version>

<description>

<![CDATA[

模型的版本，例如 <code>3.2.1</code>，也可以是版本的范围。

]]>

</description>

<type>String</type>

</field>

<field>

<name>exclusions</name>

<version>1.0.0+</version>

<description>

当计算传递依赖时，应从此依赖项中排除的模型。

</description>

<association>

<type>Exclusion</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<field>

<name>type</name>

<version>1.0.0+</version>

<description>

<![CDATA[

依赖类型，分为：Dependency和Reference两类。在计算依赖传递时，仅对类型为Dependency的依赖项计算，不对类型为Reference的计算。

]]>

</description>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "Dependency {groupId=" + groupId + ", modelId=" + modelId + ", version=" + version + ", type=" + type + "}";

}

]]>

</code>

</codeSegment>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

private String managementKey;

/\*\*

\* @return the management key as <code>groupId:artifactId:type</code>

\*/

public String getManagementKey()

{

if ( managementKey == null )

{

managementKey = groupId + ":" + modelId + ":" + type;

}

return managementKey;

}

/\*\*

\* Clears the management key in case one field has been modified.

\*/

public void clearManagementKey()

{

managementKey = null;

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Exclusion</name>

<version>1.0.0+</version>

<description>

<![CDATA[

<code>&lt;exclusion&gt;</code> 包含计算模型依赖关系时，排除的模型。

]]>

</description>

<fields>

<field>

<name>groupId</name>

<version>1.0.0+</version>

<description>

分组结构。

</description>

<type>String</type>

<required>true</required>

</field>

<field>

<name>modelId</name>

<version>1.0.0+</version>

<description>

模型标识。

</description>

<type>String</type>

<required>true</required>

</field>

</fields>

</class>

<class java.clone="deep">

<name>IssueManagement</name>

<description>

问题跟踪系统描述。

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>system</name>

<version>1.0.0+</version>

<description>

问题跟踪系统名称。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<description>

问题跟踪系统URL。

</description>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "IssueManagement {system=" + system + ", url=" + url + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>License</name>

<description>

描述模型的使用许可证，仅适用于限于本模型，而不适用于依赖的模型。

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

许可证的法定全称。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<description>

许可证文本的官方URL。

</description>

<type>String</type>

</field>

<field>

<name>comments</name>

<description>

与许可证有关的注释信息。

</description>

<version>1.0.0+</version>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "License {name=" + name + ", url=" + url + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Organization</name>

<description>

组织信息。

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

组织的全称。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<description>

组织主页。

</description>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "Organization {name=" + name + ", url=" + url + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Scm</name>

<version>1.0.0+</version>

<description>

<![CDATA[

源代码控制管理（Source Control Management，<code>&lt;scm&gt;</code> ）信息描述。

]]>

</description>

<fields>

<field>

<name>connection</name>

<version>1.0.0+</version>

<description>

源代码控制管理系统的URL。

</description>

<type>String</type>

</field>

<field>

<name>tag</name>

<version>1.0.0+</version>

<description>

当前版本的标签。

</description>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "Scm {connection=" + connection + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Repository</name>

<version>1.0.0+</version>

<description>

包含建立远程存储库连接所需的信息。

</description>

<fields>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

便于阅读的命名。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<required>true</required>

<description>

<![CDATA[

存储库的URL，一般表示为<code>protocol://hostname/path</code>。

]]>

</description>

<type>String</type>

</field>

</fields>

</class>

<class locationTracker="locations" java.clone="shallow">

<name>InputLocation</name>

<version>1.0.0+</version>

<fields>

<!-- line, column and source fields are auto-generated by Modello -->

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

@Override

public String toString()

{

return getLineNumber() + " : " + getColumnNumber() + ", " + getSource();

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class sourceTracker="source" java.clone="shallow">

<name>InputSource</name>

<version>1.0.0+</version>

<fields>

<field>

<name>modelId</name>

<version>1.0.0+</version>

<type>String</type>

<description>

<![CDATA[

元数据格式中标识 {@code <groupId>:<modelId>:<version>}。

]]>

</description>

</field>

<field>

<name>location</name>

<version>1.0.0+</version>

<type>String</type>

<description>

<![CDATA[

元数据路径或URL，如果未知则为空 {@code null} 。

]]>

</description>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

@Override

public String toString()

{

return getModelId() + " " + getLocation();

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

</classes>

</model>

附录C：构件库元数据定义

构件库元数据定义采用Modello模型定义，内容如下：

<?xml version="1.0" encoding="UTF-8"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one

or more contributor license agreements. See the NOTICE file

distributed with this work for additional information

regarding copyright ownership. The ASF licenses this file

to you under the Apache License, Version 2.0 (the

"License"); you may not use this file except in compliance

with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing,

software distributed under the License is distributed on an

"AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

KIND, either express or implied. See the License for the

specific language governing permissions and limitations

under the License.

-->

<model xmlns="http://codehaus-plexus.github.io/MODELLO/2.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://codehaus-plexus.github.io/MODELLO/2.0.0 https://codehaus-plexus.github.io/modello/xsd/modello-2.0.0.xsd"

xml.namespace="http://http://www.zgjzy.org.cn/component-meta/${version}"

xml.schemaLocation="https://www.zgjzy.org.cn/xsd/component-meta-${version}.xsd">

<id>component</id>

<name>Component</name>

<description>

<![CDATA[

<p>这是BIM中使用的构件库元描述符的参考实现。</p>

<p>An XSD is available at:</p>

]]>

</description>

<defaults>

<default>

<key>package</key>

<value>org.zgjzy.meta.component</value>

</default>

</defaults>

<classes>

<class rootElement="true" xml.tagName="library" java.clone.hook="cloneHook" java.clone="deep">

<name>Model</name>

<description>

<![CDATA[

<code>&lt;library&gt;</code> 是元数据描述的根元素。以下是可能的子元素。

]]>

</description>

<version>1.0.0+</version>

<fields>

<!-- =============== -->

<!-- 元数据格式的版本 -->

<!-- =============== -->

<field>

<name>metaVersion</name>

<version>1.0.0+</version>

<required>true</required>

<description>

描述构件库元数据格式的版本号，例如当前元数据格式的版本是1.0.0。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!--构件库分组结构 -->

<!-- =============== -->

<field xdoc.separator="blank">

<name>groupId</name>

<version>1.0.0+</version>

<required>false</required>

<description>

分组结构，用于根据需求或规定，组织、管理构件库数据，其中包含的\，用于分割目录，形成目录结构。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件库标识 -->

<!-- =============== -->

<field>

<name>libraryId</name>

<version>1.0.0+</version>

<required>true</required>

<description>

<![CDATA[

相当于构件库的文件名，用于标识构件库，命名不可以包含\/:\*?"<>|等特殊字符。

]]>

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件库版本 -->

<!-- =============== -->

<field>

<name>version</name>

<version>1.0.0+</version>

<required>true</required>

<description>

构件库的当前版本

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件库存储格式 -->

<!-- =============== -->

<field>

<name>packaging</name>

<version>1.0.0+</version>

<required>true</required>

<description>

相当于构件库文件后缀，用于标识构件库的格式类型，也可以是自定义。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件库描述信息 -->

<!-- =============== -->

<field>

<name>description</name>

<version>1.0.0+</version>

<description>

详细描述构件库的信息。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件库创建时间 -->

<!-- =============== -->

<field>

<name>creationTime</name>

<version>1.0.0+</version>

<required>true</required>

<description>

时间表示法应符合国家标准GB/T 7408-2005《数据元和交换格式·信息交换·日期和时间表示法》中定义的扩展格式表示，YYYY-MM-DDThh:mm:ss。

</description>

<type>String</type>

</field>

<!-- =============== -->

<!-- 组织机构信息 -->

<!-- =============== -->

<field>

<name>organization</name>

<version>1.0.0+</version>

<description>

描述构件库所属组织机构的属性，可用于声明构件库的版权声明和链接。

</description>

<alias>organisation</alias>

<association>

<type>Organization</type>

</association>

</field>

<!-- =============== -->

<!-- 许可信息 -->

<!-- =============== -->

<field>

<name>licenses</name>

<version>1.0.0+</version>

<description>

描述构件库的所有许可信息，若有多个许可，可以分开描述，并假定用户可以选择其中部分使用，而不是全部。

</description>

<association>

<type>License</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- =============== -->

<!-- 构件信息 -->

<!-- =============== -->

<field>

<name>components</name>

<version>1.0.0+</version>

<description>

构件库包含的构件。

</description>

<association>

<type>Component</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- =============== -->

<!-- 构件库创建者信息 -->

<!-- =============== -->

<field xdoc.separator="blank">

<name>developers</name>

<version>1.0.0+</version>

<description>

构件库创建的贡献者，可以有多个。

</description>

<association>

<type>Person</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- =============== -->

<!-- 构件库检查者信息 -->

<!-- =============== -->

<field>

<name>examiners</name>

<version>1.0.0+</version>

<description>

构件库检查者，可以有多个。

</description>

<association>

<type>Person</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- =============== -->

<!-- 版本管理系统信息 -->

<!-- =============== -->

<field xdoc.separator="blank">

<name>scm</name>

<version>1.0.0+</version>

<description>

构件库的版本管理系统信息，例如Git、Subversion、CVS等。

</description>

<association>

<type>Scm</type>

</association>

</field>

<!-- =============== -->

<!-- 问题追踪系统 -->

<!-- =============== -->

<field>

<name>issueManagement</name>

<version>1.0.0+</version>

<description>

构件库的问题追踪系统信息。

</description>

<association>

<type>IssueManagement</type>

</association>

</field>

<field xdoc.separator="blank">

<name>distributionManagement</name>

<version>1.0.0+</version>

<description>

构件库的远程存储库部署信息。

</description>

<association>

<type>DistributionManagement</type>

</association>

</field>

<field xdoc.separator="blank">

<name>properties</name>

<version>1.0.0+</version>

<description>

<![CDATA[

可以在整个元数据中作为替代使用的属性，并且在启用时用作资源中的过滤器。

格式：<code>&lt;name&gt;value&lt;/name&gt;</code>。

]]>

</description>

<type>Properties</type>

<association xml.mapStyle="inline">

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- ===================== -->

<!-- 标签信息 -->

<!-- ===================== -->

<field>

<name>tags</name>

<version>1.0.0+</version>

<description>

描述构件库的标签信息，用于检索构件库。

</description>

<association>

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* 元数据文件。

\*/

private java.io.File metaFile;

private void cloneHook( Model copy )

{

copy.metaFile = metaFile;

}

/\*\*

\* 如果有，获取构件库的元数据文件。

\*

\* @return 返回本地的构件库元数据文件，如果是从远程库得到元数据信息，则返回空值 {@code null} 。

\*/

public java.io.File getMetaFile()

{

return metaFile;

}

public void setPomFile( java.io.File metaFile )

{

this.metaFile = ( metaFile != null ) ? metaFile.getAbsoluteFile() : null;

}

/\*\*

\* 如果有，获取相应构件库的根目录。

\*

\* @return 返回根目录，如果是从远程库得到元数据信息，则返回空值 {@code null} 。

\*/

public java.io.File getProjectDirectory()

{

return ( metaFile != null ) ? metaFile.getParentFile() : null;

}

/\*\*

\* @return 构件库标识 <code>groupId:libraryId:packaging:version</code>

\*/

public String getId()

{

StringBuilder id = new StringBuilder( 64 );

id.append( getGroupId() );

id.append( ":" );

id.append( getLibraryId() );

id.append( ":" );

id.append( getPackaging() );

id.append( ":" );

id.append( getVersion() );

return id.toString();

}

@Override

public String toString()

{

return getId();

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Component</name>

<version>1.0.0+</version>

<description>

构件信息。

</description>

<fields>

<!-- =================== -->

<!-- 构件名称 -->

<!-- =================== -->

<field>

<name>name</name>

<version>1.0.0+</version>

<required>true</required>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件分类名称 -->

<!-- =============== -->

<field>

<name>classificationName</name>

<version>1.0.0+</version>

<required>false</required>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件分类码名称 -->

<!-- =============== -->

<field>

<name>classificationCode</name>

<version>1.0.0+</version>

<required>false</required>

<type>String</type>

</field>

<!-- =============== -->

<!-- 构件分类系统名称 -->

<!-- =============== -->

<field>

<name>classificationSystem</name>

<version>1.0.0+</version>

<required>false</required>

<type>String</type>

<defaultValue>GB/T 51269-2017</defaultValue>

</field>

<!-- ===================== -->

<!-- 构件注释 -->

<!-- ===================== -->

<field>

<name>description</name>

<version>1.0.0+</version>

<required>false</required>

<type>String</type>

</field>

<!-- ===================== -->

<!-- 构件标签统 -->

<!-- ===================== -->

<field>

<name>tags</name>

<version>1.0.0+</version>

<required>false</required>

<association>

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<!-- ===================== -->

<!-- 构件缩略图地址 -->

<!-- ===================== -->

<field>

<name>thumbnailURL</name>

<version>1.0.0+</version>

<required>false</required>

<type>String</type>

</field>

</fields>

</class>

<class java.clone="deep">

<name>Person</name>

<description>

人员信息描述。

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>id</name>

<version>1.0.0+</version>

<description>

人员的唯一标识。

</description>

<type>String</type>

</field>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

姓名全名。

</description>

<type>String</type>

</field>

<field>

<name>email</name>

<version>1.0.0+</version>

<description>

人员邮箱信息。

</description>

<type>String</type>

</field>

<field>

<name>organization</name>

<alias>organisation</alias>

<version>1.0.0+</version>

<description>

人员所属组织。

</description>

<type>String</type>

</field>

<field>

<name>roles</name>

<version>1.0.0+</version>

<description>

人员角色。

</description>

<association>

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

<field>

<name>properties</name>

<version>1.0.0+</version>

<description>

人员属性。

</description>

<type>Properties</type>

<association xml.mapStyle="inline">

<type>String</type>

<multiplicity>\*</multiplicity>

</association>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "Person {name=" + name + ", email=" + email + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>IssueManagement</name>

<description>

问题跟踪系统描述。

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>system</name>

<version>1.0.0+</version>

<description>

问题跟踪系统名称。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<description>

问题跟踪系统URL。

</description>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "IssueManagement {system=" + system + ", url=" + url + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>DistributionManagement</name>

<version>1.0.0+</version>

<description>

与构件库发布相关的信息描述。

</description>

<fields>

<field>

<name>repository</name>

<version>1.0.0+</version>

<description>

发布的库信息。

</description>

<association>

<type>Repository</type>

</association>

</field>

<field>

<name>site</name>

<description>

构件库的网站信息。

</description>

<version>1.0.0+</version>

<association>

<type>Site</type>

</association>

</field>

<field>

<name>downloadUrl</name>

<version>1.0.0+</version>

<description>

构件库的下载URL。

</description>

<type>String</type>

</field>

</fields>

</class>

<class java.clone="deep">

<name>License</name>

<description>

许可证信息，适用于构件库包含的所有构件。

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

许可证的法定全称。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<description>

许可证文本的官方URL。

</description>

<type>String</type>

</field>

<field>

<name>comments</name>

<description>

与许可证有关的注释信息。

</description>

<version>1.0.0+</version>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "License {name=" + name + ", url=" + url + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Organization</name>

<description>

组织信息。

</description>

<version>1.0.0+</version>

<fields>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

组织的全称。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<description>

组织主页。

</description>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "Organization {name=" + name + ", url=" + url + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Scm</name>

<version>1.0.0+</version>

<description>

<![CDATA[

源代码控制管理（Source Control Management，<code>&lt;scm&gt;</code> ）信息描述。

]]>

</description>

<fields>

<field>

<name>connection</name>

<version>1.0.0+</version>

<description>

源代码控制管理系统的URL。

</description>

<type>String</type>

</field>

<field>

<name>tag</name>

<version>1.0.0+</version>

<description>

当前版本的标签。

</description>

<type>String</type>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

/\*\*

\* @see java.lang.Object#toString()

\*/

public String toString()

{

return "Scm {connection=" + connection + "}";

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class java.clone="deep">

<name>Repository</name>

<version>1.0.0+</version>

<description>

包含建立远程存储库连接所需的信息。

</description>

<fields>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

便于阅读的命名。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<required>true</required>

<description>

<![CDATA[

存储库的URL，一般表示为<code>protocol://hostname/path</code>。

]]>

</description>

<type>String</type>

</field>

</fields>

</class>

<class java.clone="deep">

<name>Site</name>

<version>1.0.0+</version>

<description>

包含部署网站所需的信息。

</description>

<fields>

<field>

<name>name</name>

<version>1.0.0+</version>

<description>

便于人类阅读的名称。

</description>

<type>String</type>

</field>

<field>

<name>url</name>

<version>1.0.0+</version>

<description>

<![CDATA[

部署站点位置的URL，一般的形式是<code>protocol://hostname/path</code>。

]]>

</description>

<type>String</type>

</field>

</fields>

</class>

<class locationTracker="locations" java.clone="shallow">

<name>InputLocation</name>

<version>1.0.0+</version>

<fields>

<!-- 由Modello自动生成的行、列和源字段 -->

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

@Override

public String toString()

{

return getLineNumber() + " : " + getColumnNumber() + ", " + getSource();

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

<class sourceTracker="source" java.clone="shallow">

<name>InputSource</name>

<version>1.0.0+</version>

<fields>

<field>

<name>modelId</name>

<version>1.0.0+</version>

<type>String</type>

<description>

<![CDATA[

元数据标识 {@code <groupId>:<libraryId>:<version>}.

]]>

</description>

</field>

<field>

<name>location</name>

<version>1.0.0+</version>

<type>String</type>

<description>

<![CDATA[

元数据路径或URL，如果不知道则为{@code null} 。

]]>

</description>

</field>

</fields>

<codeSegments>

<codeSegment>

<version>1.0.0+</version>

<code>

<![CDATA[

@Override

public String toString()

{

return getModelId() + " " + getLocation();

}

]]>

</code>

</codeSegment>

</codeSegments>

</class>

</classes>

</model>

本标准用词说明

1 为便于在执行本标准（规范、规程）条文时区别对待，对于要求严格程度不同的用词说明如下：

1）表示很严格，非这样做不可的：

正面词采用“必须”；反面词采用“严禁”。

2）表示严格，在正常情况下均应这样做的：

正面词采用“应”；反面词采用“不应”或“不得”。

3）表示允许稍有选择，在条件许可时首先应这样做的：

正面词采用“宜”；反面词采用“不宜”。

4）表示有选择，在一定条件下可以这样做的，采用“可”。

2 条文中指明必须按其他标准、规范执行的写法为“按……执行”或“应符合……的规定”

引用标准名录

1. 《建筑信息模型分类和编码标准》GB/T51269-2017
2. 《元数据的XML Schema置标规则》（GB/T 24639-2009）
3. 《数据元和交换格式·信息交换·日期和时间表示法》GB/T 7408-2005

中国建筑业协会团体标准

建筑信息模型数字成果元数据标准

Metadata for digital products of fundamental

building information modeling

条文说明

**制定说明**

《建筑信息模型数字成果元数据标准》（T/CCIAT xxxx— 20xx），经中国建筑业协会××××年××月××日以第××号公告批准发布。

本标准的主编单位是×××，参编单位是×××、×××，（参加单位是×××、×××，）主要起草人员是×××、×××。

本标准制订过程中，编制组进行了建筑信息模型数字成果元数据调查研究，总结了我国建筑信息模型数字成果元数据标准应用的实践经验，同时参考了国外先进技术法规、技术标准，通过项目实践取得了本标准的重要技术参考内容。

为便于广大设计、施工、科研、学校等单位有关人员在使用本标准时能正确理解和执行条文规定，编制组按章、节、条顺序编制了本标准的条文说明，对条文规定的目的、依据以及执行中需注意的有关事项进行了说明。但是，本条文说明不具备与标准正文同等的法律效力，仅供使用者作为理解和把握标准规定的参考。

目  次

[1总则 58](#_Toc14469)

[2术语 59](#_Toc2375)

[3 基本规定 60](#_Toc12302)

[4 项目模型元数据 61](#_Toc17729)

[5 构件库元数据 62](#_Toc11009)

# 1总则

1.0.3 本标准规定了工程信息数字成果元数据的基本要求和元数据内容。当本标准（或规范、规程）与国家法律、行政法规的规定相抵触时，按国家法律、行政法规的规定执行。

# 2术语

2.0.3 元数据一般包括数据的标识、覆盖范围、质量、空间和时间模式、空间参照系和分发等信息。

# 3 基本规定

3.0.2 标准采用Modello模型定义元数据，基于此模型及Modello技术可自动生成XML和Json格式元数据的Java读写程序，以及XSD定义。采用其他语言工具可参考执行。

3.0.3 问题跟踪系统，既可以是BugZilla、Trac、JIRA、Basecamp等软件开发工具，也可以是面向协同设计、协同管理的系统。

3.0.4 版本管理系统如Git、Subversion、CVS等。

3.0.5 许可协议注明数字成果的知识产权所属，若有多个许可，需要分开描述，并假定用户能选择其中部分使用，而不是全部。

3.0.6 创建人及审核人通常有多人。

# 4 项目模型元数据

**4.1一般规定**

4.1.2 建议用分组信息组织项目模型的布局，分组信息中的“\”，用于分割目录，形成目录结构。

4.1.3 项目模型的标识信息相当于模型的文件名，命名不包含\、/、:、\*、?、"、<、>、|等特殊字符，因文件系统命名不能包含这些特殊字符。

4.1.4 建议用递进的版本规则标识版本信息，例如1.0.0、2.1.2等，以便识别出最新的版本。

4.1.5 存储格式信息相当于文件后缀，用于标识模型的格式类型，例如，ifc、dwg、dgn、rvt、skp等所有可能的文件格式，也允许自定义。格式信息不分大小写，及IFC、Ifc、ifc被认为是相同的。

4.1.6 标签信息一般被用于查询或分类模型。

4.1.7 项目模型元数据最好添加备注信息（详见Model的description属性定义）。

4.1.8 在版本信息相同的情况下，创建时间一般用于查询最新的模型。

4.1.9 组织信息不代表知识产权的所属，对于知识产权声明还要在许可协议中规定。

4.1.10属性一般以名值对的形式出现，如在xml中<name>value</name>，在json中” name”：value；。

**4.2外部依赖**

4.2.2 模型对其他模型的依赖，一般不允许形成循环，即A依赖B，B依赖C、C依赖A。

4.2.3 B依赖C，在A中一般能定义对B的依赖，但排除C。

**4.3分布式管理**

4.3.2 一般表示为<code>protocol://hostname/path</code>，注意：不要将系统认证信息包含在url里。

# 5 构件库元数据

**5.1一般规定**

5.1.2 建议用分组信息组织构件库的布局，分组信息中的“\”，用于分割目录，形成目录结构。

5.1.3 构件库的标识信息相当于构件库的文件名，命名不包含\、/、:、\*、?、"、<、>、|等特殊字符，因文件系统命名不能包含这些特殊字符。

5.1.4 建议用递进的版本规则标识构件库的版本信息，例如1.0.0、2.1.2等，以便识别出最新的版本。

5.1.5 存储格式信息相当于文件后缀，用于标识构件库的格式类型，也允许自定义。格式信息不分大小写。

5.1.6 标签信息一般被用于查询或分类构件库。

5.1.7 构件库元数据最好添加备注信息（详见Model的description属性定义）。

5.1.8 在版本信息相同的情况下，创建时间一般用于查询最新的构件库。

5.1.9 组织信息不代表知识产权的所属，对于知识产权声明还要在许可协议中规定。

5.1.10属性一般以名值对的形式出现，如在xml中<name>value</name>，在json中” name”：value；。

**5.2构件定义**

5.2.7 标签信息通常被用于查询或分类构件。

5.2.8 标签信息通常被用于查询或分类构件。

**5.3分布式管理**

5.3.3 一般表示为<code>protocol://hostname/path</code>，注意：不要将系统认证信息包含在url里。

5.3.5 通常利用网站，详细介绍构件库及构件的信息。

5.3.6 如果不在发布库里包含构件库的源文件，通常会单独提供下载链接。