

ICS 27.100
F 21
备案号：57156-2017



中华人民共和国电力行业标准

DL/T 1230 — 2016
代替 DL/T 1230 — 2013

电力系统图形描述规范

Graphic description specification for electric power system

2016-12-05发布

2017-05-01实施

国家能源局 发布

目 次

前言	II
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 和其他标准的关系	2
5 用例	3
6 图形文件和定义文件的结构	4
6.1 文件结构	4
6.2 图形文件结构	4
6.3 定义文件结构	5
7 基本形状图形元素	6
8 电力系统图形元素和图形显示	7
8.1 概述	7
8.2 自定义的电力系统图元	7
8.3 自定义图元的引用	8
8.4 断路器图元示例	9
8.5 动态文本图元	10
8.6 连接关系图元	10
8.7 间隔图元	11
8.8 电压等级图	11
8.9 变电站或电厂图	12
8.10 电网接线图	12
9 实时展示机制	13
9.1 实时本地展示机制	13
9.2 运行时远程浏览机制	14
9.3 菜单定义	14
附录 A (资料性附录) 电力系统图元定义	17
附录 B (资料性附录) 基于电压等级的色彩配置定义	22
附录 C (资料性附录) 图形显示风格默认值	23
附录 D (资料性附录) 菜单定义范例	24
附录 E (资料性附录) 厂站单线图范例	25

前　　言

本标准是能量管理系统（EMS）应用程序接口（API）DL 890 系列标准的一部分。

DL 890.3xx 系列标准定义了公共信息模型（CIM）：CIM 是一个抽象模型，它采用统一建模语言（UML），它以面向对象的方式对电力企业各类设备进行可视化的描述。

本标准基于 XML 定义图形文件格式，并描述如何交换基于 CIM 模型的电力图形对象，以及图形对象在人机交互界面中如何绘制。该格式有助于图形数据的高效传输，并满足调控中心间，调控中心和变电站之间的电力图形在线实时远程浏览的需求。

本标准由中国电力企业联合会提出。

本标准由全国电网运行与控制标准化技术委员会（SAC/TC 446）归口。

本标准主要起草单位：国网四川省电力公司、国家电网公司国家电力调度控制中心、国电南瑞科技股份有限公司、国家电网华东电力调控分中心、国家电网华中电力调控分中心、中国南方电网电力调度控制中心、中国电力科学研究院。

本标准主要起草人：辛耀中、李伟、熊志杰、陈国平、许洪强、陶洪铸、石俊杰、程芸、严亚勤、黄昆、孟鑫、曹蓉蓉、张亮、刘涛、周华峰、米为民、翟明玉、尚学伟、曹阳。

本标准在执行过程中的意见或建议反馈至中国电力企业联合会标准化管理中心（北京市白广路二条一号，100761）。

电力系统图形描述规范

1 范围

本标准定义了一种基于 CIM 的图形交换格式 (CIM/G)，包括：图形文件结构和定义的结构，基本形状图形元素，电力系统图形元素和图形显示，实时展示机制。

本标准适用于调控中心之间及调控中心与变电站间由不同供应商提供的异构 SCADA/EMS 系统之间的图形文件离线交换和在线远程浏览。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是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 30149—2013 电网通用模型描述规范

GB/T 4728 电力工程制图标准

DL/T 890.1—2007 能量管理系统应用程序接口 (EMS-API) 第 1 部分：导则和一般需求 (IEC 6970-1)

DL/T 890.2—2010 能量管理系统应用程序接口 (EMS-API) 第 2 部分：术语 (IEC 61970-2)

DL/T 890.301—2016 能量管理系统应用程序接口 (EMS-API) 第 301 部分：公共信息模型基础 (CIM) 基础 (IEC 61970-301: 2013, IDT)

DL/T 890.452—2015 能量管理系统应用程序接口 (EMS-API) 第 452 部分：CIM 稳态输电网络模型子集 (IEC 61970-452: 2013)

DL/T 890.453—2012 能量管理系统应用程序接口 (EMS-API) 第 453 部分：基于 CIM 的图形交换 (IEC 61970-452:2008)

DL/T 890.402—2011 能量管理系统应用程序接口 (EMS-API) 第 402 部分：公共服务 (IEC 61970-402:2003)

ISO 10646 信息技术通用多八位编码字符集 (Universal multiple-octet coded character set)

W3C: Extensible Markup Language (XML) 1.0

W3C: Scalable Vector Graphics (SVG) 1.1

3 术语和定义

3.1

领域对象 domain object

类的实例化，该类是对具有唯一 ID 的真实世界对象的建模。

注：领域对象从 CIM IdentifiedObject 继承。领域对象通常不是图形对象。领域对象的定义参见 DL/T 890.453，该标准描述电力系统设备的图形模型。

3.2

图形显示 diagram

无缝平面图纸的电子化等价物。

注：图形显示是若干图形对象的标识容器。图形显示的例子包括厂站图、输配电网络正交示意图或准地理示意图，图形显示明确定义了坐标空间。一幅图形显示是电力系统图形元素、基本图形元素、动态和静态文本的

DL/T 1230—2016

集合，图形显示的定义参见 DL/T 890.453。

3.3**图形显示对象 diagram object**

领域对象的图形显示或静态背景图。

注：领域对象的例子如断路器，静态背景图的例子如湖泊，用户交互元素的例子如按钮，图形对象的定义参见 DL/T 890.453。图形对象在本标准定义为图形元素，图形元素包括基本图形形状元素和电力系统图形元素。

3.4**图形显示对象表现风格 diagram object style**

定义了如何根据领域对象的状态渲染图形对象。

注：通常系统都会采用特有的图形对象显示风格，该定义参见 DL/T 890.453。本标准中定义为基本图形元素和电力系统图形元素渲染的风格。

3.5**图层 layer**

根据图形元素所属应用主题的不同，图层用于确定当前图层所含图形元素是否显示。每一个图层根据配置的缩放参数分别确定当前图层所含图形元素是否显示，以满足电网模型在不同应用场景下不同的显示要求。

3.6**基本形状图形元素 basic shape graphic element**

基本形状图形元素用于绘制基本形状，比如长方形、圆形、椭圆形、线段和多边形等，基本形状图形元素的定义兼容 SVG 基本形状定义的语法规则。

3.7**电力系统图形元素 power system graphic element**

电力系统图形元素用于绘制电力设备和拓扑的图形对象。它是领域对象的图形显示和 DL/T 890.453 定义的显示对象的具体化。

4 和其他标准的关系

本标准作为 DL/T 890 CIS 标准 Level 2 的 5xx 系列标准，是 DL/T 890 标准系列的一部分。如图 1 所示，它基于 DL/T 890.3xx 和 DL/T 890.4xx 系列，并与 DL/T 890.453 标准具有紧密的关系，本标准继承 DL/T 890.301 的图表布局模型，同时实现并扩展了 DL/T 890.453 描述的图形显示和渲染机制。图

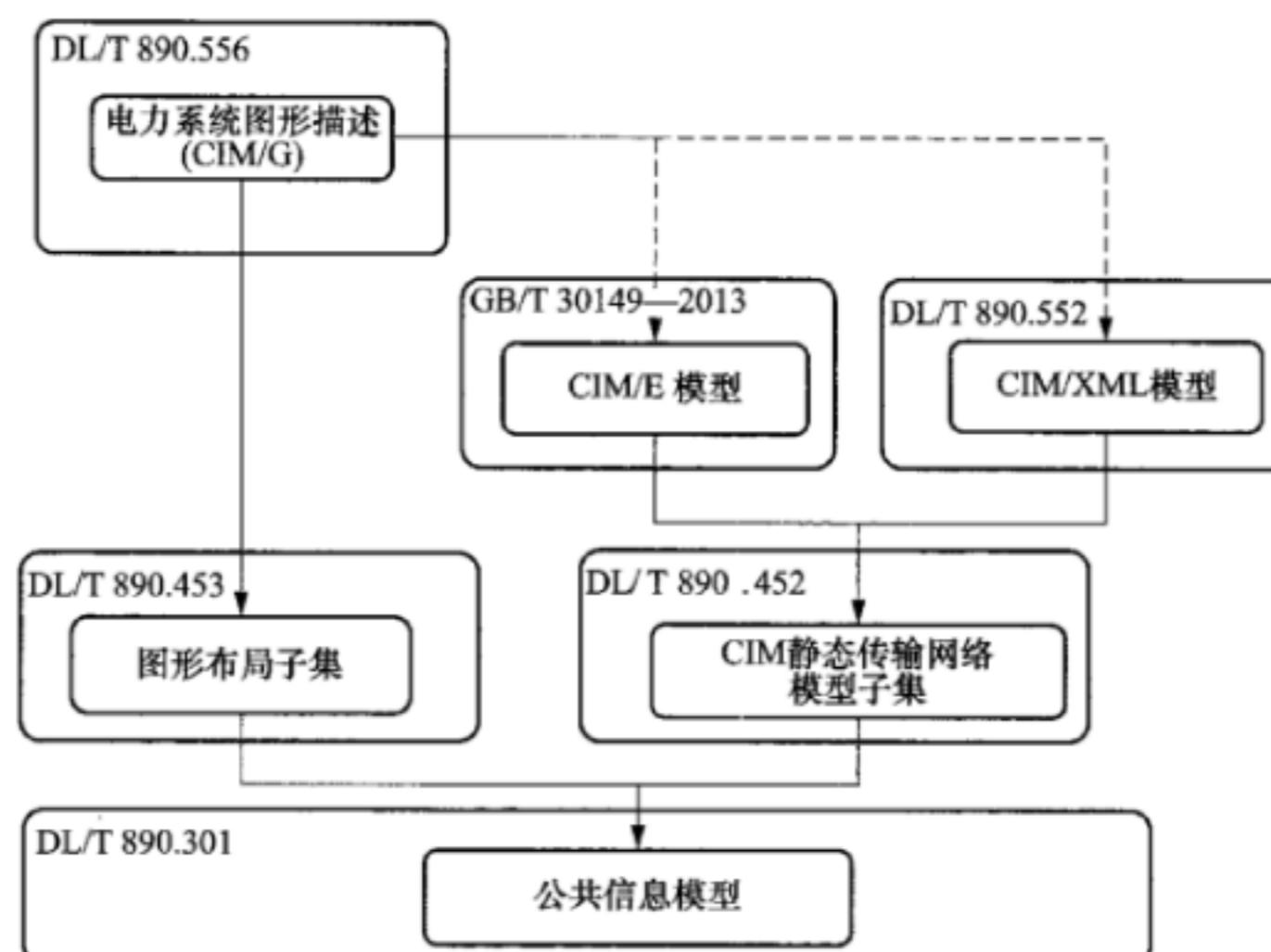


图 1 与其他标准的关系

形布局子集和电力设备模型间的关联关系在 DL/T 890.453 中定义；而图形在线显示和离线交换的详细格式则在本标准中定义。

5 用例

本标准应用于以下场景：

- 1) DL/T 890.453 定义图形布局子集和图形对象交换规则，本标准则定义交换实体，即 CIM 模型中电力设备的图形绘制方式和基于 XML 文件格式的存储方式的定义。本标准可适用于分布在不同地点、由不同厂商通过的异构 SCADA/EMS 间的图形离线交换。比如，控制中心与控制中心之间，控制中心与变电站或者电厂之间。基于本标准，图形和模型之间能够高效地完成相互转换，实现图形到模型、模型到图形的自动生成。图 2 所示为离线交换应用场景。

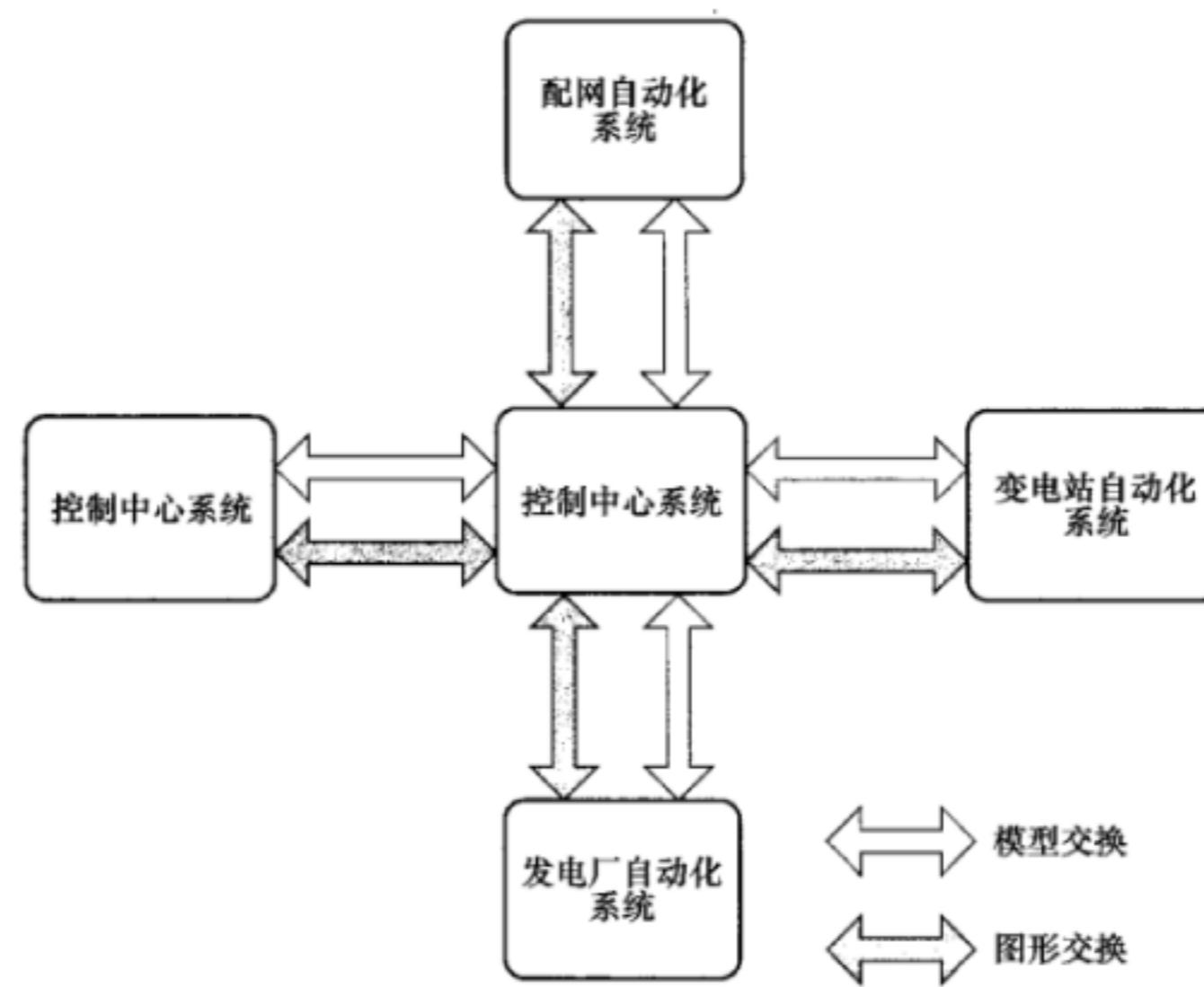


图 2 离线交换应用场景

- 2) 在不同区域控制中心、不同厂商间的异构 SCADA/EMS 间提供在线远程浏览，比如：控制中心与控制中心之间、控制中心与变电站或者发电厂之间。图 3 所示为统一人机交互界面和在线浏览应用场景。

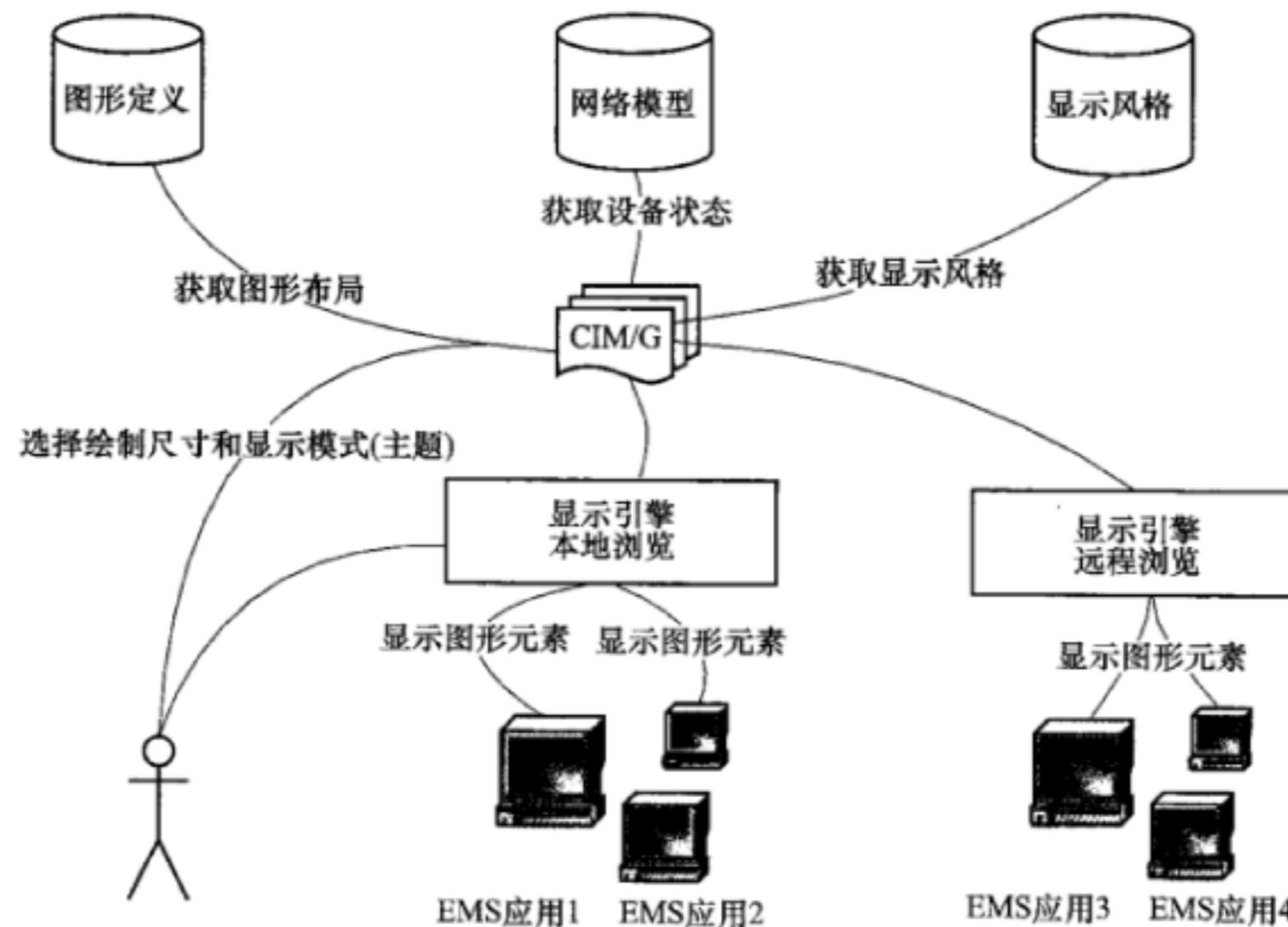


图 3 统一人机交互界面和在线浏览应用场景

DL/T 1230—2016

- 3) 基于 CIM/G 的在线图形浏览机制就像基于 HTML 的 web 浏览器，没有规范的 HTML 描述就不能够实现基于 Internet 的信息浏览。CIM/G 规范的作用类似于 HTML 规范的作用，它只是描述电力设备的在线图形浏览。在变电站或者发电厂中有部分图形很少被控制中心浏览，但当需要浏览时应能快速浏览，采用在线图形浏览机制能够满足该需求。

6 图形文件和定义文件的结构

6.1 文件结构

CIM/G 文件是基于 XML 格式的纯文本文件。CIM/G 文件分为图形文件和定义文件两类文件格式。图形文件格式用于厂站图、电网潮流图、电网 GIS 图的公共信息引用和图形元素的描述。定义文件格式用于图元、着色配置、动态显示风格和交互菜单的描述。

6.2 图形文件结构

图形文件包含声明和实体，实体以<G>开始，以</G>结束。实体包含 Include、Layer 和图形元素等标签。图形文件以.g 作为文件后缀名称，通用结构如图 4 所示。

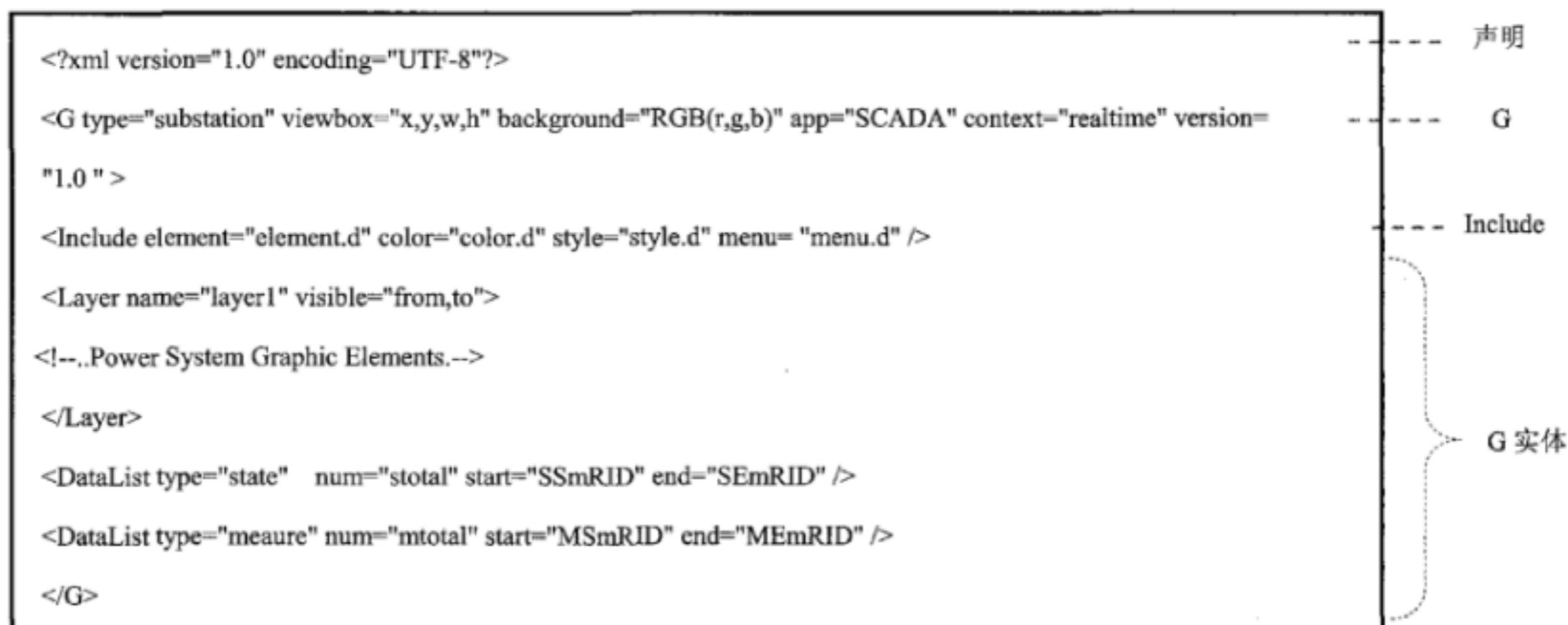


图 4 图形显示文件结构

声明部分：声明 XML 的版本和使用的字符编码。

G 标签：“G”是 CIM/G 文件的根标签，以<G>作为开始标签，并以</G>作为结束标签。G 标签内定义绘制图形的基本形状图形元素和电力系统图形元素。在 G 标签中所有图形元素都是用户交互的敏感点或敏感区域，类似 Web 页面中的“超链接”。它包含六个属性：type，viewbox，background，app，context，version。

type：表示指图形类型，例如电网接线图、厂站接线图和 GIS 图等；

viewbox：表示图形显示可见矩形区域 (x, y, w, h)，包括左上角的坐标和图形宽、高的定义；

background：表示背景颜色或背景图片，背景颜色使用颜色名称或 RGB 值定义，其他的属性可由用户定义；

app：表示图形所属应用，例如：“SCADA”“AGC”和“DTS”等；

context：表示图形所属的态，例如：实时态、测试态和未来态等；

version：表示当前 CIM/G 文件的版本号，版本号分为主版本号与次版本号两部分，例如：version=“1.2”，主版本号为 1，次版本号为 0.2，主版本用于文件格式具有较大修订时更新，而次版本号则用于日常少量的、频繁的修订时更新。

Include 标签: **Include** 标签用于引入预先定义的公用图形信息, 定义文件的格式参照 8.2 节。 **Include** 标签含有四个属性: **element**、**color**、**style** 和 **menu**。

element=“element.d” 项列出预先定义的电力系统图元, **element.d** 定义参见附录 A;
color=“color.d” 列出预先定义的颜色编码、名称、对应的 RGB 值和电压等级, 所有定义都表示同一个含义, **color.d** 的定义参见附录 B;

style=“style.d” 项列出图形样式的默认值, **style.d** 定义参见附录 C;

menu=“menu.d” 项列出预定义的功能菜单, **menu.d** 定义参见附录 D。

注: 以上附录定义的具体内容只是作为参考信息, 并不具有标准的强制约束性, 但推荐用户或者系统开发商按照以上格式进行信息集成或者信息交换。

Layer 标签: **Layer** 标签用于定义图形元素所属图层。每一个图层可对应特定的缩放级别。

name 属性指图层名;

visible 属性指可见的缩放级别, 通过 “from” 和 “to” 两个值来定义。其他属性可由用户自定义。

DataList 标签: **DataList** 标签用于定义电力设备图形元素在图形显示时所关联的动态数据集。在图形远程浏览时, 可用于加速动态数据刷新的效率。它包含四个属性: **type**, **num**, **start**, **end**。

type 属性指动态数据的类型, 包含遥信值、遥测值和统计值等动态数据;

num 属性指动态数据的数量, 动态数据按序排列;

start 属性指第一个动态数据的编号;

end 属性指最后一个动态数据的编号。

6.3 定义文件结构

定义文件采用 “**defs**” 标签定义图元库、着色配置库、动态显示风格库和交互菜单库。图元库存储预定义的公共电气图元; 着色配置库存储预定义的着色配置方案; 动态显示风格库存储预定义的图形动态显示风格, 比如: 闪烁方式、填充模式、线型和线色等; 交互菜单库存储预定义的用户配置的各类功能菜单。所有的定义并不用于直接显示, 而是通过图形文件的引用实现图形风格和样式在线渲染。定义文件的通用结构如图 5 所示。

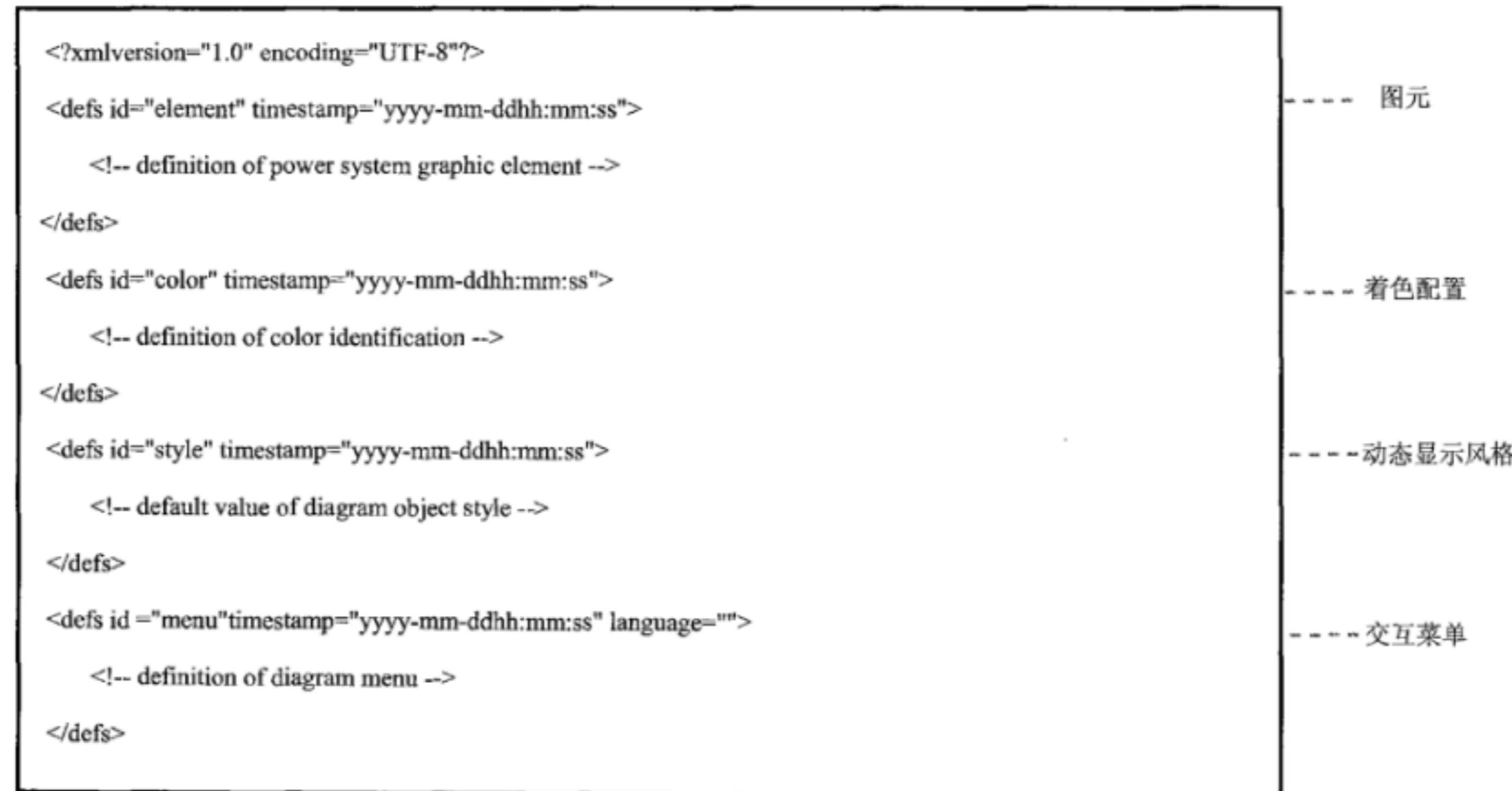


图 5 图形显示文件结构

定义文件以.d 作为文件后缀, 并针对图元、色彩配置、风格定义和菜单的定义分别保存为 **element.d**、**color.d**、**style.d** 和 **menu.d** 四个文件, 具体描述参见附录 A~附录 D。

DL/T 1230—2016

声明部分：定义文件第一行为 XML 的申明标签，用于说明 XML 的版本和使用的字符编码。

defs: **defs** 标签采用 SVG 标签，它用于定义图元、着色配置、动态显示风格和交互菜单，**defs** 标签包含两个基本属性 **id** 和 **timestamp**，用户或者系统开发商也根据需要扩展定义其他属性。

id: 指定的类型，例如图元、着色配置、动态显示风格和交互菜单。该属性可根据需要进行扩展。

timestamp: 编辑定义文件的时间戳。“yyyy-mm-ddhh:mm:ss”表示定义文件编辑时的年、月、日、小时、分钟、秒，它用于不同调度中心间定义文件包的版本比较。

language: 指菜单所使用的语言，例如英语、中文等。所有菜单都采用同一语言设置。

7 基本形状图形元素

基本形状图形元素定义直线、折线、矩形、圆形、椭圆、多边形、路径、文本等图形元素。基本图形元素用于定义电力设备图元和绘制静态图形如河流、道路等地理信息背景。基本形状图形元素的属性定义参考 SVG 的语法规则，为了更符合电力系统的图元特征，在 CIM/G 中对 SVG 的基本图形元素名称和属性进行了适当的简化，如表 1 所示。

表 1 SVG 的常用图元和属性

图元名称	主 要 属性
rect	x, y, width, height, rx, ry
circle	cx, cy, r
ellipse	cx, cy, rx, ry
line	x1, y1, x2, y2
polyline	Points
polygon	Points
path	D
text	x, y, font-family, font-size
image	x, y, width, height, xlink:href
animate	from, to

如表 1 所示，矩形、圆、椭圆、线、折线、多边形描述图形的基本形状，文字表示单字符串，路径（path）描述事物的几何轮廓，以下命令用于路径（path）行为的定义：M（移动至），L（直线至），C（曲线至），a（弧线），z（结束路径）；在路径的属性中大写字母表示绝对坐标，小写表示相对坐标；在图像的属性中，“xlink:href”定义引用外部文件；在动画的属性中，“from” 和 “to” 用于表示移动的路径。

CIM/G 支持 SVG 中所有的基本图形元素，在此基础上，CIM/G 还允许根据公共信息模型（CIM）自定义任意的电力系统图形元素。具体参见第 8 章《电力系统图形元素和图形显示》。为了提升效率，CIM/G 缩写了 SVG 的属性名称。基本图形元素绘制通用属性如表 2 所示。

表 2 基本绘图属性

描述	CIM/G 的属性名称	SVG 的属性名称
宽	w	width
高	h	height
线色	lc	stroke

表 2 (续)

描述	CIM/G 的属性名称	SVG 的属性名称
线宽	lw	stroke-width
线型	ls	stroke-dasharray
填充色	fc	fill
填充模式	fm	fill-rule
坐标变换	tf	transform
字体大小	fs	font-size
字体类型	ff	font-family

通用绘制属性和样式的默认值参见附录 C。

8 电力系统图形元素和图形显示

8.1 概述

电力系统图形元素除了形状定义外还映射 CIM 模型，并具有动态属性。电力系统图形元素分为图元定义和图元引用，形状采用 SVG 图形元素定义的图元，图形显示时通过引用图元定义来实现图形对象的实例化。通过 XML 标签直接描述 CIM 模型，实现 CIM 模型与电力图形对象的一一映射关系，在 CIM/G 定义文件中电力系统图形元素名称就是 CIM 模型中类名称，例如 Disconnector、Breaker、PowerTransformer、Bay 等。通过电力系统图形元素的全局设备资源 ID 与 CIM 模型关联，并在图形在线显示时获取动态属性。

电力系统图形元素还直接定义电压等级、变电站、发电厂和电网等设备容器，在图形显示时不需要引用预先定义的图元直接绘制。

图形显示中所有图元都可以是敏感点或敏感区域，类似网页中的“超链接”，用户可以通过使用鼠标或键盘设备与之交互。

8.2 自定义的电力系统图元

电力系统图元应该与公共信息模型中定义的设备模型对应，并用“defs”标签声明为图元定义。随着公共信息模型在配电、变电、电力市场等其他专业领域的扩展，用户或开发商可以根据以上应用场景扩展定义相对应的电力系统图元；同时，用户也可以根据自身需求，针对同一类型的电力设备，定义不同形状的图形。图 6 为电力系统图元定义模板。

```
<defs id="element">
  <UserDefinedElement id="" loc="" data="#mRID" show="status" box="X,Y,W,H"
    glue="x1,y1 x2,y2... A="" app="">
    <! –the element could be defined by basic graphic elements or other user defined elements-->
  </UserDefinedElement>
</defs>
```

图 6 电力系统图元定义

UserDefinedElement：表示电力系统图元名称，例如：Disconnector、Breaker、PowerTransformer、BusBarSection、ACLineSegment、DCLineSegment 等在 DL/T 890.301 中定义的设备名称。而配电、变

电、电力市场等专业领域的设备模型也可以通过以上方法进行定义。附录 A 列出常用的电力系统图元定义，用户或者供应商也可行根据需要重新定义或扩展。

所有自定义图元都包含八个公共属性：“id”“loc”“data”“show”“A”“box”“glue”和“app”。“id”“loc”“data”“show”和“A”属性是图元的形式属性（类似于程序调用中的形式参数），形式属性在图元定义文件可以为空甚至不显示。实际参数由引用的图元传递。“box”“glue”和“app”为实体属性。各个属性的具体含义如下：

id: 指当前图形显示中图元唯一的编号。id 在定义文件中可以为空，但引用文件中必须传递。

loc: 指图元在画布中的位置，x 和 y 是图元左上角的坐标，w 和 h 是指画布中图元的宽度和高度。

loc 属性在定义文件中可以为空，但引用文件中必须传递。这里的 x、y、w、h 都必须为小写字母。

data: 指图元所关联的 CIM 模型中的数据或对象 ID。data 属性有两种类型，一种是实际的显示值，另一种是以“#”开头的全局设备资源 ID。图元属性的多个数据以逗号分隔。data 属性在定义文件中可以为空，但引用文件中必须传递。

show: 指图元的状态或样式。格式为 show=“Q, T, F, S”。其中，Q、T、F 和 S 都是小于 255 的整型值（最大为 1 字节）。show 属性在定义文件中可以为空，但引用文件中必须传递。详细释义请参照 8.3 节。

box: 指图元的最小包围矩形，X, Y 是矩形左上角的坐标，W、H 是矩形的宽和高。X、Y、W、H 均为大写字母。在定义文件中这些值必须被定义。

glue: 指图元的连接点。x0、y0 至 xn、yn 表示端子坐标。端子数量取决于所关联的电气设备。例如：断路器有两个端子，三绕组变压器有三个端子，而母线的端子数则取决于连接到母线的线数，这个数值有可能是 n。glue 属性按照坐标顺序排列，例如：point (x0, y0) 对应于 glue0, point (x1, y1) 对应于 glue1, point (xn, yn) 对应于 gluen。该属性在定义文件中必须被定义。

A: 指当鼠标点击该设备时需要调用的交互函数或图形，该属性具有热点或者敏感替代图元功能。

app: 电力设备图元所属应用。

8.3 自定义图元的引用

自定义图元采用隐式和显式两种引用方式，隐式引用直接调用 element.d 文件中默认定义的设备图元，显式引用则需要采用“devref”属性指定引用的外部定义图元。图 7 为自定义图元的隐式引用，图 8 为自定义图元的显式引用。

```
<G>
  <UserDefinedElement id="id" loc="x, yw, h" data="#mRID" show="status" A="func|exec|script|pic"/>
</G>
```

图 7 自定义图元的隐式引用

```
<G>
  <UserDefinedElement id="id" loc="x, yw, h" data="#mRID" show="status" A="func|exec|script|pic" devref=
  "userdefinelement_name"/>
</G>
```

图 8 自定义图元的显式引用

注：在实际应用中，从显示效率和画面显示一致性考虑，默认采用自定义图元的隐式引用。

“id”“loc”“data”“show”和“A”通过直接引用自定义图元的方式赋值。

id: 指当前图形中该图元唯一的编号。

loc: 指图元在画布中的位置, x 和 y 是图元左上角的坐标, w 和 h 是指画布中图元的宽度和高度。loc 属性在定义文件中可以为空, 但引用文件中必须传递。这里的 x、y、w、h 都必须为小写字母。当图形中一个图元被绘制时, 其实际的坐标按照以下方法计算:

```
x_real = XX_definition+x_reference;
```

```
y_real = YY_definition+y_reference;
```

当“w”和“h”与图元定义中的“W”和“H”不一致时, 图元中的“W”和“H”被赋值为“w”和“h”。

data: 指图元所关联的 CIM 模型中的数据或对象 ID。Data 属性有两种类型, 一种是实际的值, 另一种是间接的数据 ID 或者以“#”开头的索引。图元属性的多个数据以逗号分隔。

show: 指图元的状态或样式。格式为 show=“Q, T, F, S”。其中, Q, T, F 和 S 都是小于 255 的整型值。show 属性在定义文件中可以为空, 但引用文件中必须传递。

“Q”指数据质量, 其定义参照 DL/T 890.301。

“T”指拓扑状态(带电、停电和接地), 其对应的颜色参照附录 B。

“F”指刷新状态, F=0 表示不刷新, F=1 表示每秒刷新一次, 其他值可由用户定义。

“S”指形状改变标记, S=0 表示形状没有改变, 其他值可由用户定义。

举例来说, show=“0, 3, 0, 0”表示数据质量有效, 拓扑色为 500kV 电压等级色, 数据不刷新, 形状无改变。很多情况下 Q, T, F, S 可以简化为“status”。

A: 指当鼠标点击该设备时需要调用的交互函数或图形, 该属性具有热点或者敏感替代图元功能。

交换函数可以采用进程内函数, 如: A=“zoomin()”, 或者进程外可执行函数, 如: A=“exec:openfile(name)”, 甚至可以采用遵循 EMCA Script 标准定义的脚本文件, 如: A=“setbreaker12.js”。调用图形则定义为: A=“substation2.g”。

app: 电力设备图元所属应用。图形显示时采用该图元定义的 app 属性值, 但电力设备图元 app 属性值可以为空, 当电力设备图元 app 属性值为空时, 该电力设备图元默认采用当前图形文件定义的所属应用。

六个属性(id, loc, data, show, A, app)需要由引用图形传递, 分别表示实际的图元编号、位置、动态数据、显示样式和交互函数。基于这些属性, 自定义图元可以在图形画布中被绘制。

8.4 断路器图元示例

断路器图元是电力系统中断路器的图形表示。设备的颜色可根据电压等级色(参照附录 A), 或具体颜色名称, 或 RGB 值设置。通常用是否填充来表示断路器的开合状态。断路器图元也包含其他基础图元所具备的通用属性。

1) element.d 中定义断路器。

断路器图元定义如表 3 所示。所有图元的定义请参照附录 A。

表 3 断路器图元定义

图元定义	形状
<pre><defs> <Breaker id="breaker0" loc="" data="" show="" box="0, 042, 18" glue="18, 3 18, 15" A=""> <rect x="4" y="3" w="12" h="28" fm="0"/> </Breaker> </defs></pre>	

2) 图形中断路器图元的引用。

断路器图元通过以下两种方式被引用：

```
<Breaker id="CB5013" loc="50, 56 16, 32" data="#765013" show="0, C, 0, 0" A="fun1"/>
```

或者

```
<Breaker id="CB5013" loc="50, 56 16, 32" data="#765013" show="0, C, 0, 0" A="fun1" devref="breaker0" />
```

show=“0, C, 0, 0”表示数据质量有效，拓扑色为标准电压等级色，数据不刷新，形状无改变。

8.5 动态文本图元

动态文本图元用来描述电力量测数据，如 P 、 Q 、 I 等。它不是电力系统设备，但是可以作为电力系统图元使用。它的定义结构如图 9 所示。

```
<defs>
  <DText id="dtext1" loc="" data="" show="status" box="X,Y W,H" A="F1" />
</defs>
```

图 9 动态文本图元的定义

一个动态文本图元的引用例子如图 10 所示。

```
<G>
  <DText id="id1" loc="x,y,w,h" data="#mRID" show="Q,T,F,S" A="F1" precision="0"/>
</G>
```

图 10 动态文本图元的引用

“id”“loc”“data”和“show”的属性使用可以通过直接引用标签。这些属性的含义参见 8.2 和 8.3 节。precision 属性描述数据显示精度，根据属性值决定显示小数点位数，比如：precision=“0”显示整型，precision=“2”显示两位小数。

8.6 连接关系图元

连接关系图元是一种特殊的图元，它不是通常的设备，仅仅是连接两个或多个设备之间的一条短线，它没有任何的量测和阻抗，它展示的是 DL/T 890.301 中定义的连接点的概念。电力系统图元可以通过他们的连接点直接关联，大多数情况下，它是不需要用连接关系图元的。连接关系图元有 3 个属性：连接点集、关联和显示。

一个连接关系图元的例子如图 11 所示。

```
<G>
  <Link points="x1, y1 x2, y2" connect="glue0, equip0;glue1, equip1" show="Q, T, F, S" />
</G>
```

图 11 连接关系图元的引用

连接点集：定义连接点的坐标集。

关联：表示设备图元之间的连接关系。定义的格式为 connect=“glue0, equip-A;glue1, equip-B”，表示设备 A 的连接点 0 与设备 B 的连接点 1 通过连接关系图元相关联。

8.7 间隔图元

间隔图元是 DL/T 890.301 中定义的间隔模型的图形化展示，用来描述断路器、隔离开关和接地开关在间隔区域的配置。因为大部分的间隔可以用间隔图元概括，所以厂站图可以简单地通过几个间隔图元绘制。

间隔由于在同一种接线方式下设备之间的连接关系是相对固定的，所以通过 Bay 可非常简洁的描述厂站内部结构。间隔图元采用类似于电力设备图元先定义、后引用的图形描述方式；另外，通过间隔图元定义的 conf 参数，可以设置间隔模板电力设备图元的消隐，从而实现一个间隔图元表达多个间隔实例的图形展示方式。

volt：表示当前间隔所属电压等级；

cfg：设置间隔图元引用时设备的消隐状态，对应位为 1 则显示设备图元，0 则隐藏设备图元；

index：表示间隔图元的 conf 顺序号，从 0 开始编号，适用于间隔模板内所有动态图形元素对象。

间隔图元定义示例如图 12 所示。

图元定义	形状
<pre> <defs> <Bay id="bay1-3/2" volt="500kV" loc="" data="" show="" box="X, Y, W, H" glue="x1, y1" A=""> <Disconnector id="d1" loc="54, 9620, 20" data="" show="" /> <Breaker id="b1" loc="50, 5620, 20" data="" show="" /> <Disconnector id="d2" loc="54, 18120, 20" data="" show="" /> <Disconnector id="d3" loc="54, 27220, 20" data="" show="" /> <Breaker id="b2" loc="50, 23220, 20" data="" show="" /> <Disconnector id="d4" loc="54, 35720, 20" data="" show="" /> <Disconnector id="d5" loc="54, 44820, 20" data="" show="" /> <Breaker id="b3" loc="50, 40820, 20" data="" show="" /> <Disconnector id="d6" loc="50, 520, 20" data="" show="" /> <Disconnector id="d7" loc="5, 15120, 20" data="" show="" /> <Disconnector id="d8" loc="65, 32720, 20" data="" show="" /> <Link points="59, 44059, 452" connect="1, d7" show="" /> <Linkpoints="59, 26459, 276" connect="1, d8" show="" /> </Bay> </defs> </pre>	

图 12 间隔图元定义示例

间隔图元的引用如图 13 所示。

```

<G>
  <Bay id="bay1-3/2" Volt="500kV" loc="x, y, w, h" data="#mRID" show="Q, T, F, S" A="F1" />
</G>

```

图 13 间隔图元的引用

conf 属性值中的各数值位分别对应间隔模板内 index 编号的各电力设备图元，1 表示对应的设备图元显示，0 表示对应的设备图元隐藏。对于隐藏的设备图元，需要处理隐藏后的连接线及连接关系。通过 conf 参数的配置，能够实现一个间隔模板生成若干间隔实例，如图 14 所示。

8.8 电压等级图

电压等级图是一个组件，可以无需预先定义直接使用。一幅电压等级图中可以包含多条母线和多个间隔。属性 station 表示当前电压等级所属厂站。图 15 中的例子包括两条母线和两个间隔。

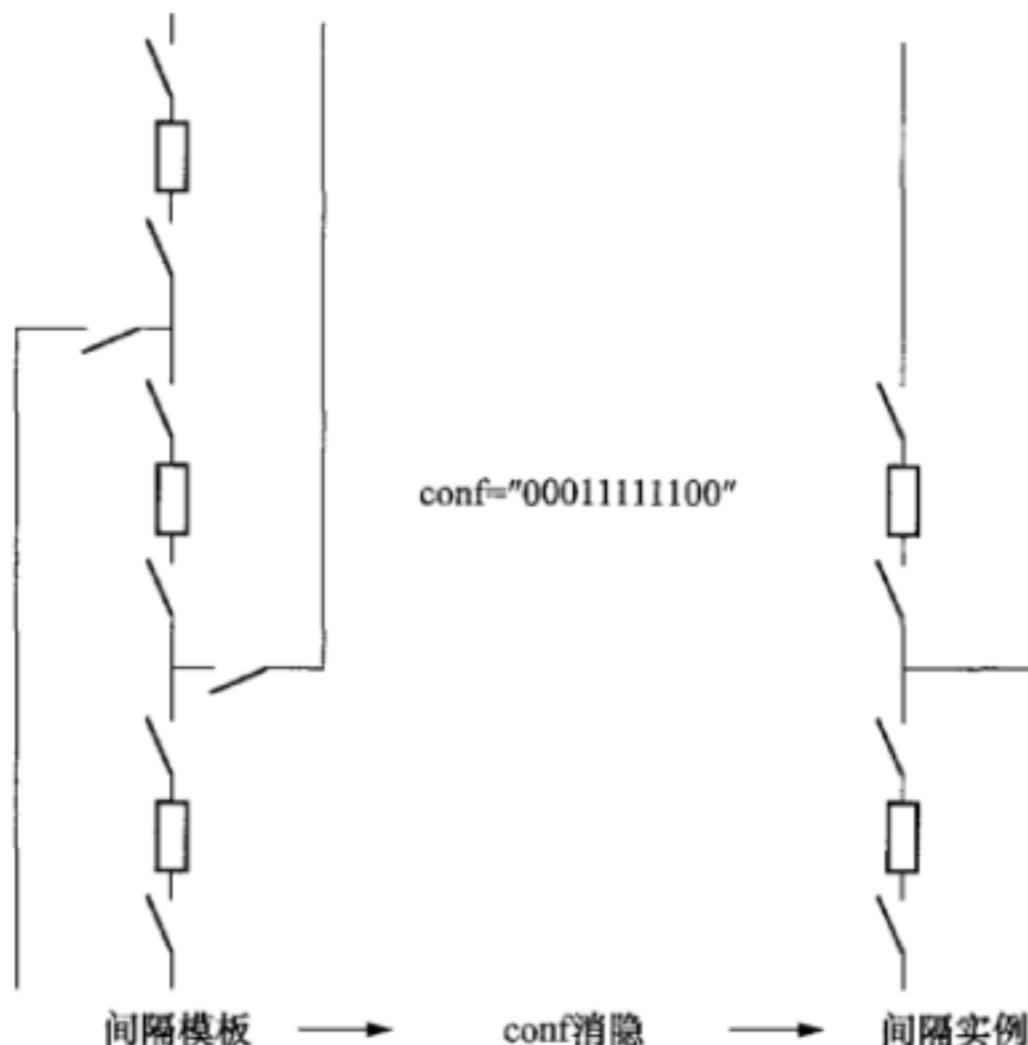


图 14 间隔消隐效果示例

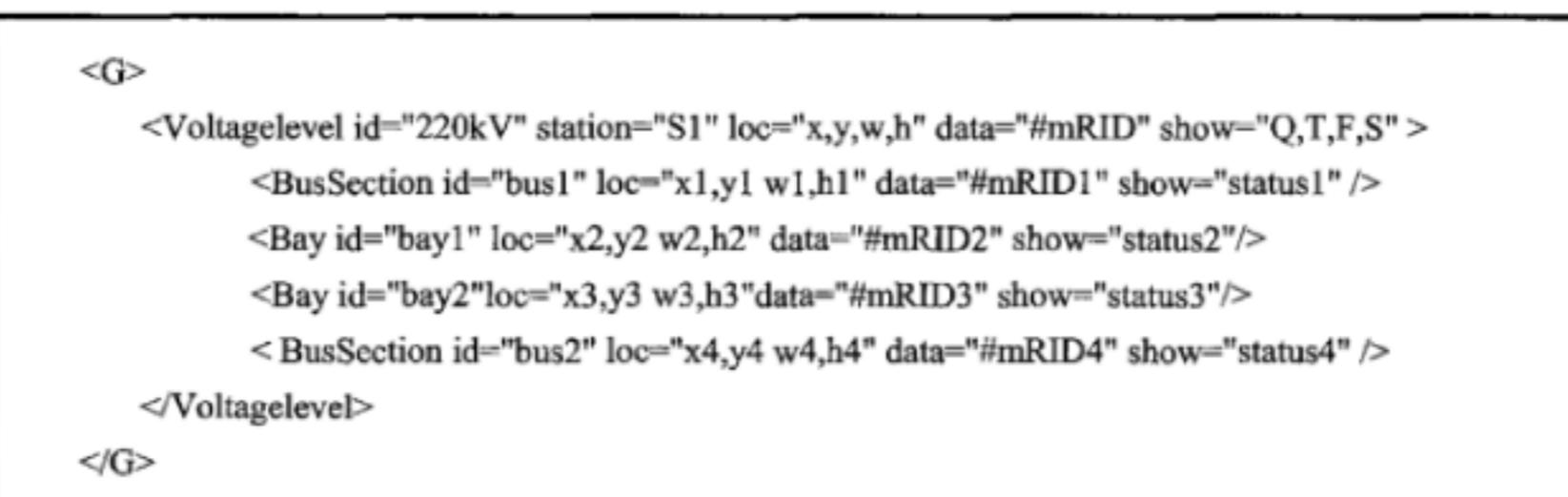


图 15 电压等级图示例

8.9 变电站或电厂图

变电站或电厂图是一个组件，可以无需预先定义直接使用。变电站或电厂图中包含多个设备和电压等级。属性 grid 表示当前厂站所属电网。图 16 是一个包含三个电压等级的变电站图示例。

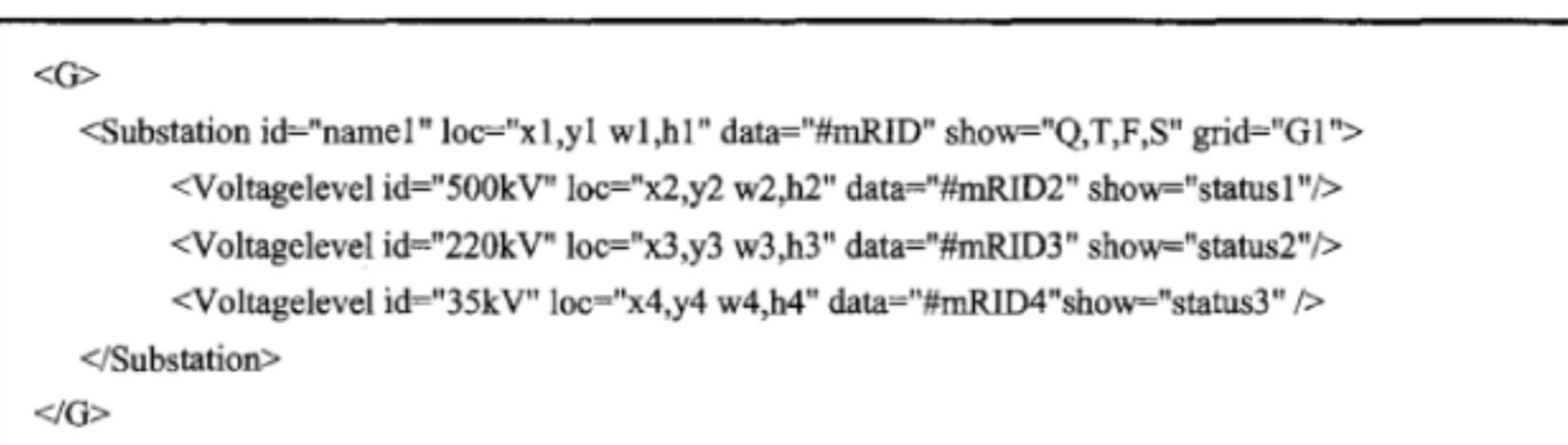


图 16 带有电压等级的变电站图示例

在某些情况下，间隔或者设备可以采用不带有电压等级的图元描述，示例如图 17 所示。

在本例中，变电站标签中包含电压属性，所以图中没有电压标签。如果每个间隔的电压等级都不同，则每个间隔图元都需要加入电压属性标签。

8.10 电网接线图

电网接线图是由输电线路和变电站或者电厂的拓扑连接组成的。通常变电站或电厂在图中简化为一个点或者特殊的图标。电网接线图分为两种类型，一种是普通的电网接线图，另一种是基于 GIS 的地理电网接线图。

```

<G>
  <Substation id="name1" loc="x1,y1 w1,h1" data="#mRID" show="Q,T,F,S" volt="500kV"
    grid="G1">
    <Bus id="bus1" loc="x1,y1 w1,h1" data="#mRID1" show="status1" />
    <Bay id="bay1" loc="x2,y2 w2,h2" data="#mRID2" show="status2" />
    <Bay id="bay2" loc="x3,y3 w3,h3" data="#mRID3" show="status3" />
    <Bus id="bus2" loc="x4,y4 w4,h4" data="#mRID4" show="status4" />
    <DataList type="state" num="stotal" start="SSmRID" end="SEmRID" />
    <DataList type="meaure" num="mtotal" start="MSmRID" end="MEmRID" />
  </Substation>
</G>

```

图 17 没有电压等级的变电站图示例

在普通电网接线图中，当放大到一定比例时显示为单线图，当缩小到一定比例时单线图显示为变电站或者电厂图元。

在地理电网接线图中，变电站或者电厂是根据实际地理坐标绘制的，在图中标记为变电站或者电厂图元。当图形缩放时，根据不同的缩放比例显示不同的地理图层。

电网接线图是最高层次的容器图元，无需预先定义就可直接使用。图 18 是一幅由四条线路和四个变电站组成的电网接线图示例。

```

<G>
  <PowerGrid id="grid1" box="x,y,w,h" type="OneLine" volt="500kV" >
    <ACLineSegment id="L1" loc="x1,y1,w1,h1" data="#mRID1" show="status1" A="F1" />
    <ACLineSegment id="L2" loc="x2,y2,w2,h2" data="#mRID2" show="status2" A="F2" />
    <ACLineSegment id="L3" loc="x3,y3,w3,h3" data="#mRID3" show="status3" A="F3" />
    <ACLineSegment id="L4" loc="x4,y4,w4,h4" data="#mRID4" show="status4" A="F4" />

    <Substation id="S1" loc="x5,y5,w5,h5" data="#mRID5" show="status5" A="F5" />
    <Substation id="S2" loc="x6,y6,w6,h6" data="#mRID6" show="status6" A="F6" />
    <Substation id="S3" loc="x7,y7,w7,h7" data="#mRID7" show="status7" A="F7" />
    <Substation id="S4" loc="x8,y8,w8,h8" data="#mRID8" show="status8" A="F8" />

    <DataList type="state" num="stotal" start="SSmRID" end="SEmRID" />
    <DataList type="meaure" num="mtotal" start="MSmRID" end="MEmRID" />
  </PowerGrid>
</G>

```

图 18 电网接线图示例

电网接线图通常包括四种属性：

id: 表示电网接线图的名称。

box: 表示电网接线图的矩形区域。

type: 表示电网接线图的类型，type 的值可以是“**OneLine**”或者“**GIS**”，“**OneLine**”表示普通电网接线图，“**GIS**”表示地理电网接线图。

volt: 表示电网接线图的电压等级。

9 实时展示机制

9.1 实时本地展示机制

在不同的控制中心之间的在线远程图形浏览机制如图 19 所示，步骤如下：

a) 图形浏览器向图形服务端发送画面 A 的请求；

- b) 图形服务端响应并且发送文件 ‘A.g’ 到图形浏览器;
- c) 图形浏览器基于收到的 ‘A.g’ 展示静态图形;
- d) 图形服务端不间断地向浏览器发送实时数据;
- e) 图形浏览器刷新动态数据展示;
- f) 当图形浏览器发送一个终止或者退出的请求, 图形服务端停止发送数据。

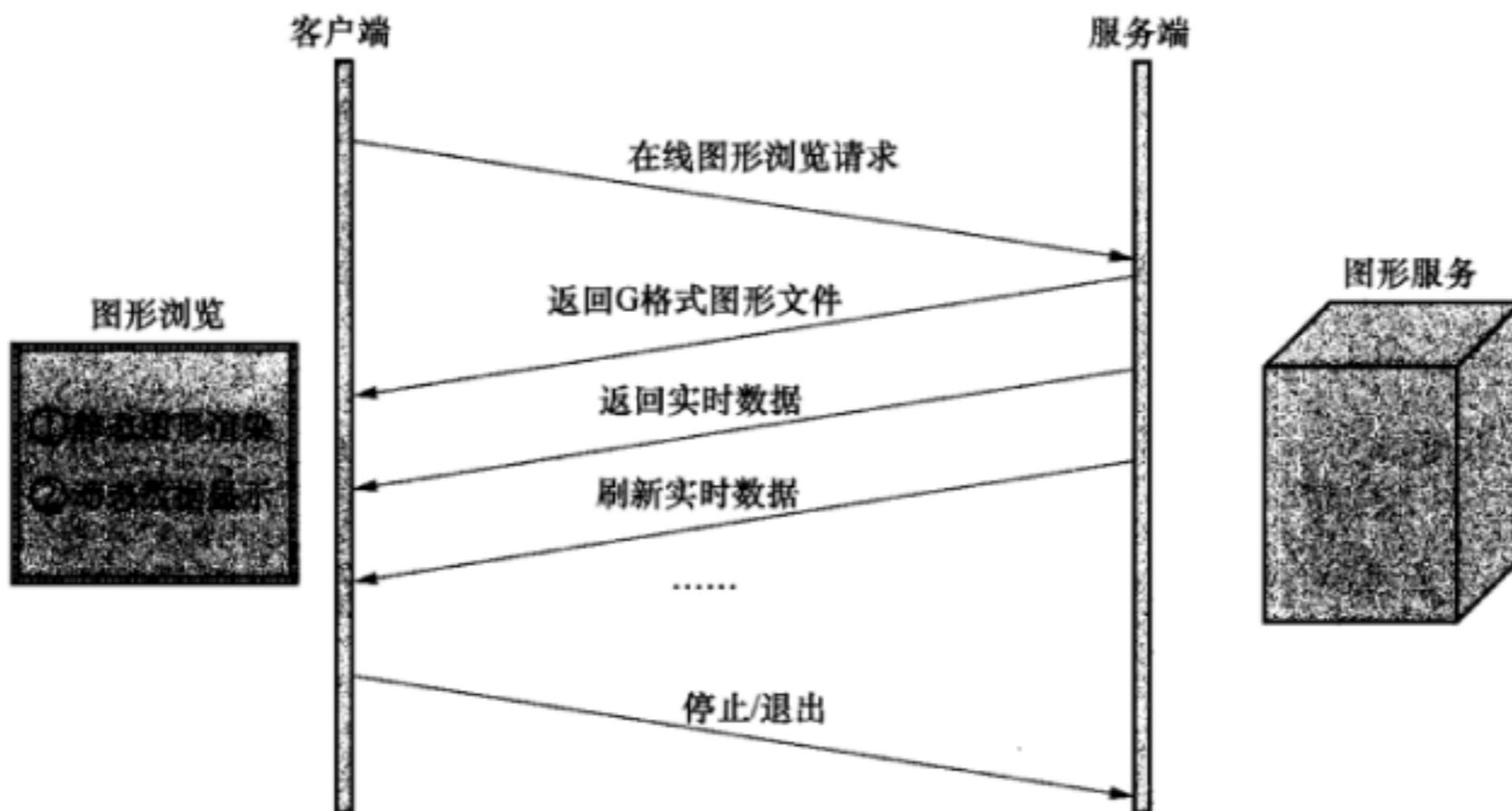


图 19 实时本地展示机制

9.2 运行时远程浏览机制

通用图形样式包在服务端定义; 个性化的图形样式包由用户在客户端定义。图形样式包括“element.d”“color.d”“style.d”和“menu.d”。当远程图形在本地显示时, 通用图形样式包或个性化图形样式包均可能被采用。当图形以通用图形样式显示时, 通用图形样式包和个性化的图形样式包需要进行比较, 如果两者是一致的, 本地个性化图形样式包用于控制图形的实际显示效果, 相反, 如果两者不一致, 需要从服务端下载通用图形样式包, 进而用于本地图形通用样式的显示。但是运行时远程浏览所采用的具体通信方式不属于本标准的范畴, 本标准仅覆盖了显示格式方面的内容。运行时远程浏览机制如图 20 所示。

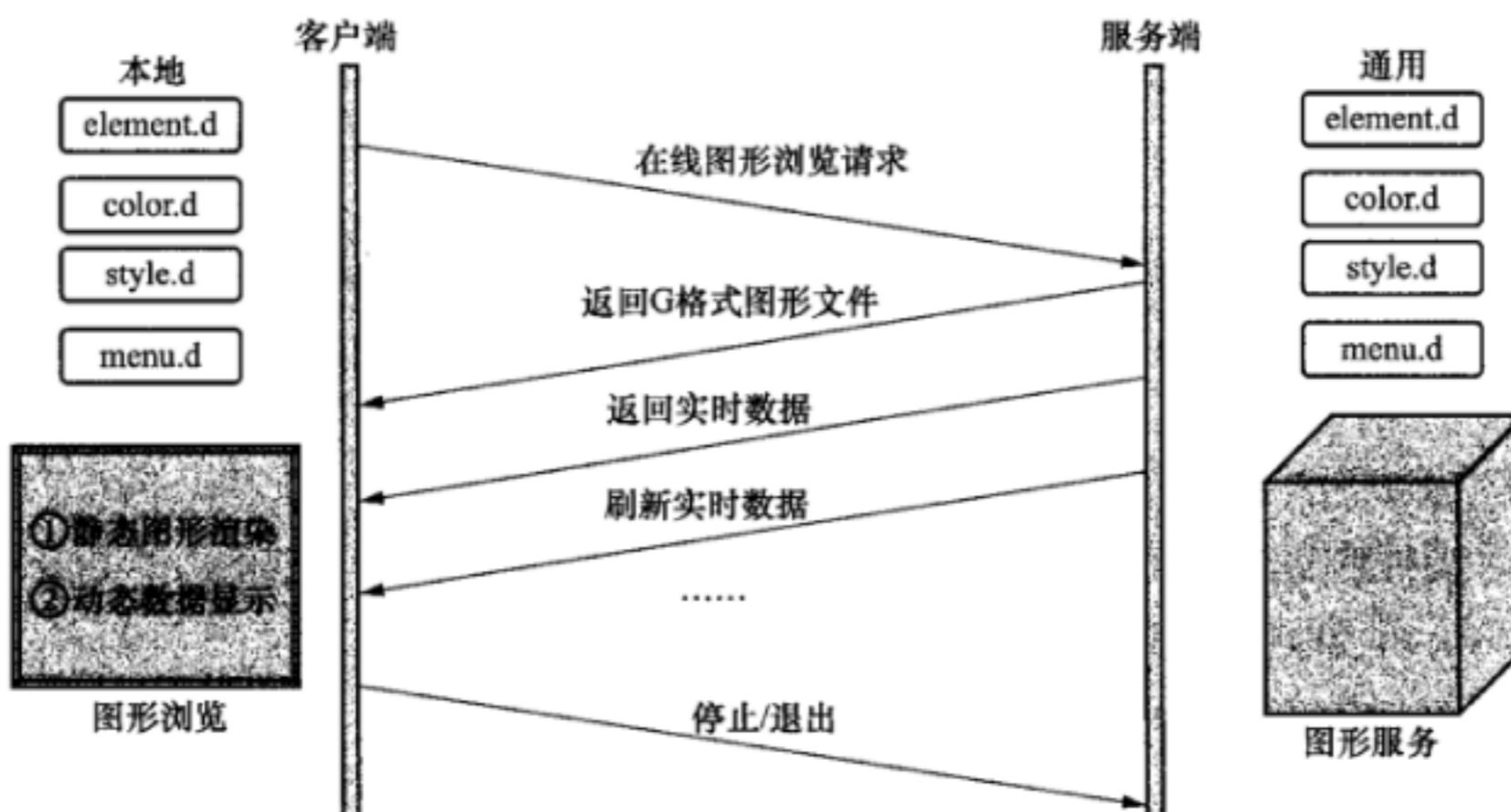


图 20 运行时远程浏览机制

9.3 菜单定义

定义一个菜单, 需要用到“application”“menu”和“button”标签。“application”标签是用来描述菜

单所属的应用环境，该标签可以包含多个“menu”标签，每个“menu”标签包含几个“button”标签。

菜单可以包含多个层次的嵌套菜单项，形成一个由根菜单和子菜单组成的分层结构。

每个“button”或下一级“button”有一个“act”属性（act 的 menu@前缀，用来区分直接调用还是间接调用），该属性可以是一个指定的动作（例如，调用一个函数、一段程序或打开另一幅图）。

所有的“button”标签中的“act”动作属性都关联一个具体动作。通过“separator”标签可以将“button”标签定义成不同的菜单组。菜单定义示例如图 21 所示。

```
<defs id="menu" timestamp="yyyy-MM-ddHH:mm:ss" language="English">
<application id="scada" context="realtime" devcice="Breaker" active="right_click">
    <menu id="m1_id" name="m1_name" type="m1_T, m1_L" buttons="3" enable="y/n">
        <button id="m1b1" name="m1b1_name" icon="m1b1_icon" act="m1b1_function" enable="" />
        <button id="m1b2" name="m1b2_name" icon="m1b2_icon" act="m1b2_function" enable="" />
        <button id="m1b3" name="m1b3_name" icon="m1b3_icon" act="menu@m2_id" enable="" />
    </menu>
    <separator/>
    <menu id="m2_id" name="m2_name" type="m2_T, m2_L" buttons="2" enable="y/n">
        <button id="m2b1" name="m2b1_name" icon="m2b1_icon" act="m2b1_function" enable="" />
    </menu>
</application>
</defs>
```

图 21 菜单定义示例

a) `defs` 标签用于定义系统的所有菜单项，至少包括三个基本属性：

“`id`”属性，该属性值必须是 `menu`，表示这个标签定义的是一个菜单；

“`timestamp`”属性表示菜单的最后修改时间，时间精确到秒；

“`language`”属性表示“menu”和“button”显示的语言，如英语、汉语等。该属性作用的范围包括 `def` 标签内的所有“menu”和“button”。该属性当系统初始化时设置。

b) `application` 标签用于定义“menu”所属的应用环境，其中包括至少四个基本属性：

“`id`”属性，`application` 标签的标识符。如果属性值是“`window`”，代表窗口菜单。如果 `id` 的值是 `application` 的名称，代表图中的设备的菜单。`id` 的值可以是字符串或整数。

“`context`”属性，描述 `application` 标签所属的态。态的值可以是字符串或整数，例如：实时态（值为 0），学习态（值为 1），测试态（值为 2）。

“`device`”属性，用于描述设备类型。不同的菜单可以被定义为不同的设备。该属性值通常是设备类型的名称。

“`active`”属性，表示激活菜单的方法，例如，属性值 `right-click` 表示通过鼠标右键激活菜单。

c) “`menu`”标签用来定义一个单独的菜单，至少包括五个基本属性：

“`id`”属性，`menu` 标签的标识符，属性值可以是字符串或整数。

“`name`”属性，`menu` 标签的名字，属性值是“`language`”属性中指定语言的字符串。

“`buttons`”属性，`menu` 标签中包括按钮数量。属性值是一个整数。

“`enable`”属性，表示是否启用或禁用 `menu`，属性值是“`y`”或“`n`”，“`y`”表示菜单可用，“`n`”表示菜单禁用。

“`type`”属性，表示 `menu` 标签的类型，由一对 (T, L) 属性组成，T 代表菜单类型，L 代表菜单在窗口的位置。例如，一个固定在窗口上方的菜单，`type` 值应该是 (`m_fixed, l_top`) 或 (0, 0)。

常用的菜单类型包括：①`m_fixed`：固定菜单，值为 0；②`m_pulldown`：下拉式菜单，值为 1；③`m_popup`：弹出式菜单，值为 2；④`m_bookmark`：书签菜单，值为 3；⑤`m_list`：列表菜单，值为 4；⑥`m_floating`：浮动菜单，值为 5。

常用的菜单位置包括：①L_TOP：窗口顶部，值是 0；②L_BOTTOM：窗口底部，值是 1；③L_LEFT：窗口左边，值是 2；④L_RIGHT：窗口右边，值是 3。

d) button 标签用来定义一个按钮，至少包括五个基本属性：

“id” 属性，button 标签的标识符，属性值是字符串或整数。

“name” 属性，button 标签的名称，属性值是“language” 属性中指定的语言。

“icon” 属性，button 标签的图标。

“act” 属性，button 标签关联的动作。当关联的动作是直接调用时，act 属性的值为调用的函数或程序的名称（包括路径）。当表示下一级菜单时，在标识符加上“menu@” 前缀，以区别于直接调用。

“enable” 属性，button 标签启用或禁用，表示方法和 menu 标签的“enable” 属性相同。当属性的值是“y” 时，表示属性可用。

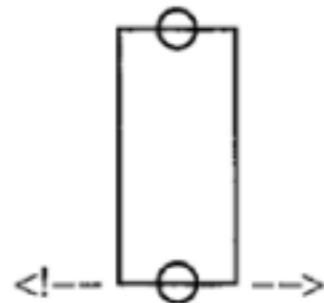
请参阅附录 D 菜单定义示例。

附录 A
(资料性附录)
电力系统图元定义

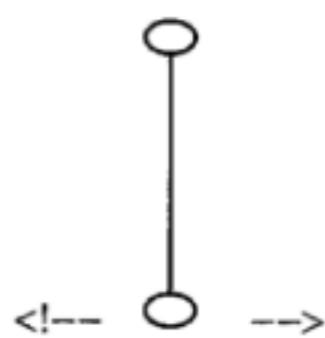
附录 A 描述一系列电力系统图元的标准定义。作为默认设置，以下定义图元在图形显示时首先引用，并作为电力系统图形元素的实体显示。以下定义根据应用需要可扩展。图元定义文件名称叫“element.d”。

```
<? xml version="1.0" encoding="UTF-8"?>
<defs id="element" timestamp="yyyy-MM-ddHH:mm:ss">

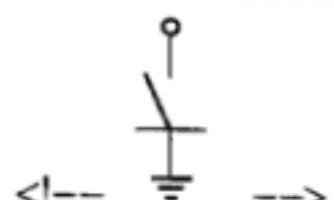
<Breaker id="" loc="" data="" show="" box="0,0 36,18" glue="18,3 18,15" >
    <rect x="4" y="3" w="28" h="12" fm="0" />
    <text x="40" y="8" w="10" h="15" value="*id" A="" />
</Breaker>
```



```
<Disconnectord id="" loc="" data="" show="" box="0, 048, 16" glue="11, 4 11, 42" >
    <circle cx="11" cy="7" r="3" />
    <line x1="11" y1="10" x2="11" y2="36"/>
    <circle cx="11" cy="39" r="3" />
    <text x="20" y="16" w="10" h="15" value="*id" A="" />
</Disconnectord>
```

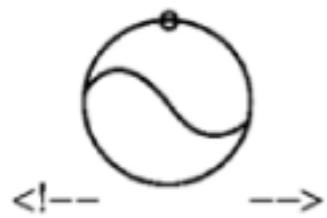


```
<GroundDisconnectord id="" loc="" data="" show="" box="0, 042, 18" glue="4, 12" >
    <circle cx="6" cy="12" r="2" />
    <line x1="6" y1="12" x2="14" y2="12" />
    <line x1="14" y1="9" x2="14" y2="15" />
    <line x1="14" y1="12" x2="22" y2="5" />
    <line x1="22" y1="12" x2="30" y2="12" />
    <line x1="30" y1="7" x2="30" y2="17" />
    <line x1="32" y1="8" x2="32" y2="16" />
    <line x1="34" y1="9" x2="34" y2="15" />
    <text x="40" y="10" w="10" h="15" value="*id" A="" />
</GroundDisconnectord>
```

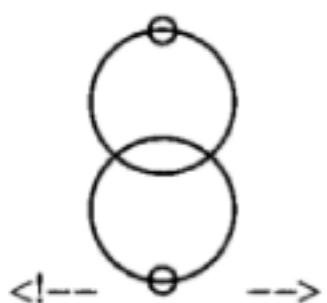


```
<Generator id="" loc="" data="" show="" box="0, 038, 40" glue="4, 12">
    <circle cx="19" cy="20" r="14" />
    <path d="M5, 20 a7, 7 1 0 14, 0"/>
    <path d="M33, 20 a7, 7 1 0 -14, 0"/>
    <text x="40" y="20" w="10" h="15" value="*id" A="" />
    <text x="40" y="40" w="20" h="15" value="P:" />
    <DText id="*id" loc="x, y, w, h" data="#Mrid1" />
    <text x="40" y="60" w="20" h="15" value="Q:" />
    <DText id="*id" loc="x, y, w, h" data="#Mrid2" />
```

</Generator>



```
<PowerTransformer2 id="" loc="" data="" show="" box="0, 062, 40" glue="20, 4 20, 56">
    <circle cx="20" cy="20" r="16" />
    <circle cx="20" cy="40" r="16" />
    <text x="40" y="40" w="10" h="15" value="*id" A="" />
    <text x="40" y="0" w="20" h="15" value="P:" />
    <DText id="*id" loc="x, y, w, h" data="#mRID1" show="status" />
    <text x="40" y="20" w="20" h="15" value="Q:" />
    <DText id="*id" loc="x, y, w, h" data="#mRID2" show="status" />
    <text x="40" y="60" w="20" h="15" value="P:" />
    <DText id="*id" loc="x, y, w, h" data="#mRID3" show="status" />
    <text x="40" y="80" w="20" h="15" value="Q:" />
    <DText id="*id" loc="x, y, w, h" data="#mRID4" show="status" />
</PowerTransformer2>
```



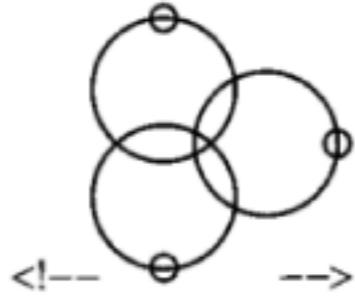
```
<PowerTransformer3 id="" loc="" data="" show="" box="0, 062, 40" glue="40, 4 14, 29 66, 29">
    <circle cx="40" cy="20" r="16" />
    <circle cx="25" cy="40" r="16" />
    <circle cx="55" cy="40" r="16" />
    <text x="0" y="60" w="10" h="15" value="*id" A="" />
    <text x="40" y="0" w="20" h="15" value="P:" />
    <DText id="*id" loc="x, y, w, h" data="#mRID1" show="status" />
    <text x="40" y="20" w="20" h="15" value="Q:" />
    <DText id="*id" loc="x, y, w, h" data="#mRID2" show="status" />
    <text x="40" y="60" w="20" h="15" value="P:" />
    <DText id="*id" loc="x, y, w, h" data="#mRID3" show="status" />
```

```

<text x="40" y="80" w="20" h="15" value="Q:" />
<DText id="*id" loc="x, y, w, h" data="#mRID4" show="status" />
<text x="40" y="100" w="20" h="15" value="P:" />
<DText id="*id" loc="x, y, w, h" data="#mRID5" show="status" />
<text x="40" y="120" w="20" h="15" value="Q:" />
<DText id="*id" loc="x, y, w, h" data="#mRID6" show="status" />

```

</PowerTransformer3>



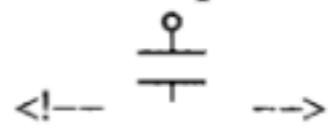
<ShuntCapacitor id="" loc="" data="" show="" box="0, 026, 40" glue="13, 8" >

```

<line x1="13" y1="8" x2="13" y2="16" />
<line x1="4" y1="16" x2="22" y2="16" />
<line x1="4" y1="22" x2="22" y2="22" />
<line x1="13" y1="30" x2="13" y2="38"/>
<text x="30" y="12" w="10" h="15" value="*id" A="" />

```

</ShuntCapacitor>



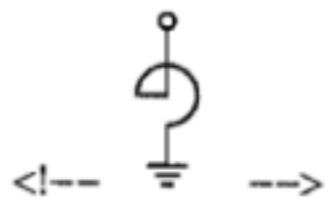
<ShuntReactor id="" loc="" data="" show="" box="0, 026, 40" glue="13, 3" >

```

<path d="M13, 10 a8, 8 0 1 0 -5 8"/>
<line x1="13" y1="3" x2="13" y2="10" />
<line x1="5" y1="18" x2="13" y2="18" />
<line x1="13" y1="18" x2="13" y2="37" />
<line x1="4" y1="37" x2="22" y2="37" />
<line x1="6" y1="39" x2="20" y2="39"/>
<line x1="8" y1="41" x2="18" y2="41" />
<text x="30" y="16" w="10" h="15" value="*id" A="" />

```

</ShuntReactor>



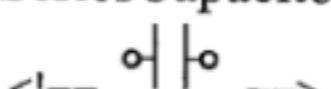
<SeriesCapacitor id="" loc="" data="" show="" box="0, 040, 26" glue="6, 11" >

```

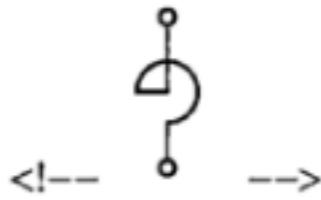
<line x1="7" y1="11" x2="15" y2="11" />
<line x1="16" y1="4" x2="16" y2="19"/>
<line x1="23" y1="4" x2="23" y2="19"/>
<linex1="23" y1="11" x2="32" y2="11"/>
<text x="38" y="10" w="10" h="15" value="*id" A="" />

```

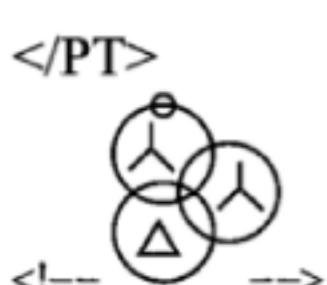
</SeriesCapacitor>



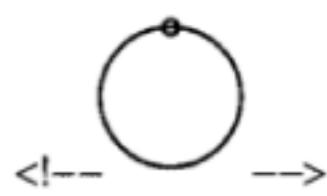
```
<SeriesReactor id="" loc="" data="" show="" box="0, 026, 40" glue="13, 5 13, 30">
    <line x1="13" y1="5" x2="13" y2="10" />
    <line x1="8" y1="18" x2="13" y2="18" />
    <line x1="13" y1="18" x2="13" y2="30" />
    <path d="M13, 10 a8, 8 0 1 0 -5 8" />
    <text x="38" y="10" w="10" h="15" value="*id" A="" />
</SeriesReactor>
```



```
<PT id="" loc="" data="" show="" box="0, 040, 44" glue="21, 5">
    <circle cx="20" cy="15" r="10"/>
    <circle cx="28" cy="29" r="10"/>
    <circle cx="13" cy="29" r="10" />
    <line x1="20" x2="20" y1="14" y2="19" />
    <line x1="24" x2="20" y1="10" y2="13" />
    <line x1="20" x2="20" y1="15" y2="15" />
    <line x1="15" x2="19" y1="10" y2="13" />
    <line x1="29" x2="29" y1="31" y2="31" />
    <line x1="9" x2="7" y1="27" y2="31" />
    <line x1="8" x2="14" y1="32" y2="32" />
    <line x1="14" x2="14" y1="34" y2="34" />
    <line x1="14" x2="12" y1="31" y2="27" />
    <text x="38" y="10" w="10" h="15" value="*id" A="" />
</PT>
```



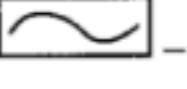
```
<CT id="" loc="" data="" show="" box="0, 050, 50" glue="23, 7">
    <circle cx="23" cy="21" r="14" />
    <text x="40" y="10" w="10" h="15" value="*id" A="" />
</CT>
```



```
<SubstationMark id="" loc="" data="" show="" box="0, 025, 25" glue="" >
    <circle cx="12" cy="12" r="10" />
    <text x="30" y="4" w="10" h="15" value="*id" />
    <DTText id="*id" loc="x, y, w, h" data="#mRID" show="status" A="" />
</SubStationMark>
```



```

<PowerPlantMark id="" loc="" data="" show="" box="0, 060, 40" glue="">
  <rect x="2" y="2" w="56" h="36" />
  <path d="M5, 20 a7, 7 1 0 14, 0"/>
  <path d="M33, 20 a7, 7 1 0 -14, 0"/>
  <text x="60" y="4" w="10" h="15" value="*id" />
  <DText id="*id" loc="x, y, w, h" data="#mRID" show="status" A="" />
</ PowerPlantMark>
<!--  -->

<Turbine id="" loc="" data="" show="" box="0, 080, 70" glue="">
  <polygon points="0, 0 6, 3 12, 0 12, 12 6, 9 0, 12" />
  <text x="85" y="4" w="10" h="15" value="*id" A="" />
</Turbine>
<!--  -->

<BusBarSection id="" loc="" data="" show="" box="0, 0, w, h" glue="">
  <path d="M0, 0 w, 0"/>
  <text x="60" y="4" w="10" h="15" value="*id" A="" />
  <text x="60" y="20" w="10" h="15" value="V;" />
  <DText id="*id" loc="x, y, w, h" data="#mRID" show="status" />
</ BusBarSection>

<ACLineSegment id="" loc="" data="" show="" box="0, 0w, h" glue="">
  <path d="M0, 0 w, h "/>
  <text x="60" y="4" w="10" h="15" value="*id" A="" />
  <text x="60" y="4" w="10" h="15" value="P;" />
    <DText id="*id" loc="x, y, w, h" data="#mRID1" show="status" />
  <text x="60" y="20" w="10" h="15" value="Q;" />
  <DText id="*id" loc="x, y, w, h" data="#mRID2" show="status" />
</ ACLineSegment>

<DCLineSegment id="" loc="" data="" show="" box="0, 0w, h" glue="">
  <path d="M0, 0 w, h "/>
  <text x="60" y="4" w="10" h="15" value="*id" A="" />
  <text x="60" y="20" w="10" h="15" value="P;" />
  <DText id="*id" loc="x, y, w, h" data="#mRID1" show="status" />
  <text x="60" y="30" w="10" h="15" value="I;" />
  <DText id="*id" loc="x, y, w, h" data="#mRID2" show="status" />
</ DCLineSegment>

</defs>

```

附录 B
(资料性附录)
基于电压等级的色彩配置定义

色彩配置主要用于电力系统图形的线色和填充色彩的绘制。色彩的定义通常基于带电设备的电压等级，可以用编号、颜色名称或 RGB 值定义同一电压等级的色彩配置，并能够用于所有图形显示。表 B.1 的定义是一个范例，用户可以根据实际需求重新定义或者扩展，本标准不作强制约束。

表 B.1 色彩配置值和图例示例

编号	电压等级	颜色名称	RGB 值	颜色图例
1	1000kV	blue	RGB (0, 0, 255)	
2	800kV	blue	RGB (0, 0, 255)	
3	750kV	orange	RGB (250, 128, 10)	
4	660kV	orange	RGB (250, 128, 10)	
5	500kV	red	RGB (255, 0, 0)	
6	330kV	bright blue	RGB (30, 144, 255)	
7	220kV	purple	RGB (128, 0, 128)	
8	110kV	vermeil	RGB (240, 65, 85)	
9	66kV	gold	RGB (255, 204, 0)	
10	35kV	yellow	RGB (255, 255, 0)	
11	20kV	brown	RGB (226, 172, 6)	
12	15kV	darkgreen	RGB (0, 128, 0)	
13	13kV	lightgreen	RGB (0, 210, 0)	
14	10kV	crimson	RGB (185, 72, 66)	
15	6kV	darkblue	RGB (0, 0, 139)	
16	0kV	grey	RGB (128, 128, 128)	

以下定义为电压等级色彩配置，定义文件名称为“color.d”，如图 B.1 所示。

```
<? xml version="1.0" encoding="UTF-8"?>
<defs id="color" timestamp="yyyy-MM-ddHH:mm:ss">
<color code= 1 name=" blue" voltage ="1000" rgb="0,0,255" />
<color code= 2 name=" blue" voltage ="800" rgb="0,0,255" />
<color code= 3 name=" orange" voltage ="750" rgb="250,128,10" />
<color code= 4 name=" orange" voltage ="660" rgb="250,128,10" />
<color code= 5 name=" red" voltage ="500" rgb="255,0,0" />
<color code= 6 name=" bright blue" voltage ="330" rgb="30,144,255" />
<color code= 7 name=" purple" voltage ="220" rgb="128,0,128" />
<color code= 8 name=" vermeil" voltage ="110" rgb="240,65,85" />
<color code= 9 name=" gold" voltage ="66" rgb="255,204,0" />
<color code= 10 name=" yellow" voltage ="35" rgb="255,255,0" />
<color code= 11 name=" brown" voltage ="20" rgb="226,172,6" />
<color code= 12 name=" darkgreen" voltage ="15" rgb="0,128,0" />
<color code= 13 name=" lightgreen" voltage ="13" rgb="0,210,0" />
<color code= 14 name=" crimson" voltage ="10" rgb="185,72,66" />
<color code= 15 name=" darkblue" voltage ="6" rgb="0,0,139" />
<color code= 16 name=" grey" voltage ="0" rgb="128,128,128" />
</defs>
```

图 B.1 电压等级色彩配置定义

附录 C
(资料性附录)
图形显示风格默认值

图形显示风格的默认值包括以下属性：线型、线宽、线色、填充模式、填充色、坐标变换、缩放、字体和字体尺寸等。如果用户确定图形显示的风格，在图形显示中的图形元素（图元）将首先采用“style.d”文件中定义的风格缩写名称进行图形渲染，如图C.1所示。该定义只是一个参考范例，它并不被本标准强制约束，并且能够根据用户需求重新定义或者扩展。

```

<? xml version="1.0" encoding="UTF-8"?>
<defs id="style" timestamp="yyyy-MM-ddHH:mm:ss">
  <style name="default_line_style" ls="1" />
  <style name="default_line_width" lw="1" />
  <style name="default_line_color" lc="0,0,255" />
  <style name="default_fill_mode" fm="0" />
  <style name="default_fill_color" fc="0,0,255" />
  <style name="default_transform" tf="rotate(0)scale(1,1)" />
  <style name="default_font_family" ff="Arial" />
  <style name="default_font_size" tfs="14" />
  <style name="station_font_family" s_ff="Arial" />
  <style name="station_font_size" s_fs="24" />
  <style name="station_font_color" s_lc="white" />
  <style name="name_font_family" n_ff="Arial" />
  <style name="name_font_size" n_fs="12" />
  <style name="name_font_color" n_lc="yellow" />
  <style name="data_font_family" d_ff="Arial" />
  <style name="data_font_size" d_fs="14" />
  <style name="data_font_color" d_lc="green" />
  <style name="menu_font_family" m_ff="Arial" />
  <style name="menu_font_size" m_fs="16" />
  <style name="menu_font_color" m_lc="black" />
</defs>

```

图C.1 图形显示风格默认值

附录 D
(资料性附录)
菜单定义范例

本附录只是菜单定义的示例，该定义包括以下属性：菜单名称、菜单类型、图标和动作。所有机界面的菜单根据“menu.d”文件定义进行绘制，如图 D.1 所示。

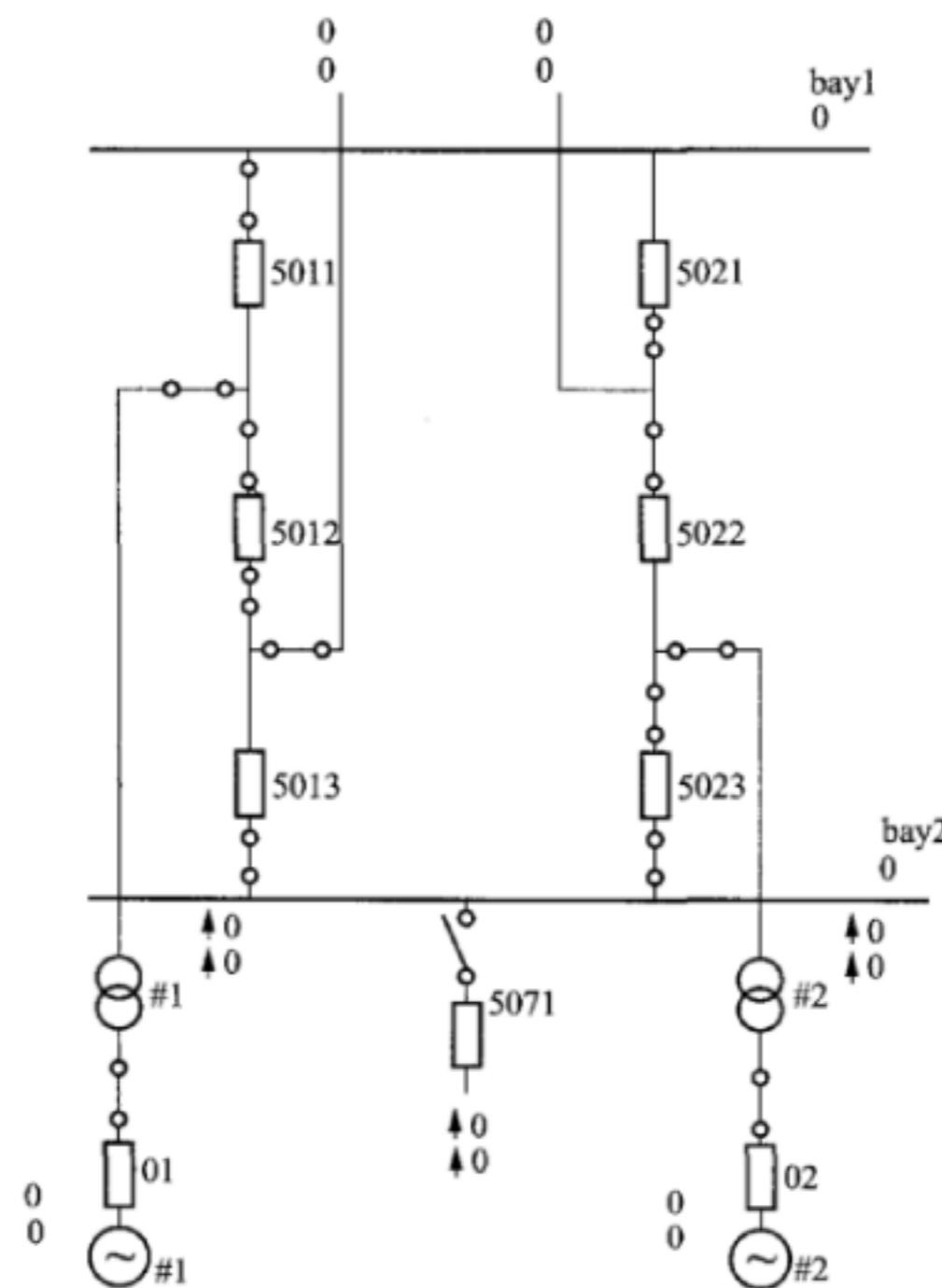
```

<? xml version="1.0" encoding="UTF-8"?>
<defs id="menu" timestamp="yyyy-MM-ddHH:mm:ss" language="English">
<application id="window" context="" device="" active="">
<menu id="m1_id" name="File" type="1,1" buttons="5" enable="y">
<button id="m1b1" name="Open" icon="open.png" act="FileOpen()" enable="y"/>
<button id="m1b2" name="Save" icon="save.png" act="FileSave(file_name)" enable="y"/>
<separator/>
<button id="m1b3" name="Forward" icon="arrowf.png" act="Forward()" enable="y"/>
<button id="m1b4" name="Backward" icon="arrowb.png" act="Backward()" enable="y"/>
<button id="m1b5" name="Home" icon="home.png" act="HomePage()" enable="y"/>
</menu>
<menu id="m2_id" name="Operation" type="1,1" buttons="3" enable="y">
<button id="m2b1" name="ZoomIn" icon="in.png" act="ZoomIn(step)" enable="y"/>
<button id="m2b2" name="ZoomOut" icon="out.png" act="ZoomOut(step)" enable="y"/>
<button id="m2b3" name="Drag" icon="drag.png" act="Drag(from,to)" enable="y"/>
</menu>
</application>
<application id="scada" context="realtime" device="Breaker" active="right_click">
<menu id="m3_id" name="CBreaker" type="1,1" buttons="3" enable="y">
<button id="m3b1" name="Para" icon="" act="GetPara()" enable="y"/>
<button id="m3b2" name="Token" icon="" act="SetToken()" enable="y"/>
<button id="m3b3" name="Remote Ctrl" icon="" act="RemoteCtrl()" enable="y"/>
</menu>
</application>
<application id="scada" context="realtime" device="DText" active="right_click">
<menu id="m4_id" name="Dynamic_data" type="1,1" buttons="2" enable="y">
<button id="m4b1" name="Para" icon="" act="GetPara()" enable="y"/>
<button id="m4b2" name="Curve" icon="" act="GetCurve(date_time)" enable="y"/>
</menu>
</application>
</defs>
```

图 D.1 菜单定义范例

附录 E
(资料性附录)
厂站单线图范例

以下范例描述如何采用间隔模板绘制一幅厂站单线图。该变电站有两个方式的间隔，名称分别为“bay1”和“bay2”的间隔是从3/2接线的间隔模板实例化的，在该变电站内还有两条母线、两条交流线段和两个发电机，图形显示如图E.1所示。



图E.1 厂站单线图范例

图形显示对应的CIM/G存储文本如图E.2所示。

```
<? xml version="1.0" encoding="UTF-8"?>
<G viewbox="0,0,645,1050" background="#236,236,236">
<Include element="Element.d" color="Color.d" style="Style.d" />
<PowerPlant id="plant-A">
<BusbarSection id="bus1" loc="64,246,469,0" data="#105081" show="status1"/>
<BusbarSection id="bus2" loc="314,553,10,490" data="#105082" show="status2" />
<Bay id="bay501" loc="104,246,20,20" data="#905083" show="status3" />
<Bay id="bay502" loc="358,246,20,20" data="#905084" show="status4" />
<PowerTransformer id="T1" loc="62,788,20,20" data="#405085" show="status5" />
<PowerTransformer id="T2" loc="465,788,20,20" data="#405086" show="status6" />
<Disconnecter id="dis01" loc="77,859" data="#605087" show="status7" />
<Disconnecter id="dis02" loc="480,864" data="#605088" show="status8" />
<Breaker id="CB01" loc="73,912,20,20" data="#305089" show="status9" />
<Breaker id="CB02" loc="476,913,20,20" data="#305090" show="status10" />
<Generator id="G1" loc="93,981,20,20" data="#705091" show="status11" />
<Generator id="G2" loc="496,981,20,20" data="#75092" show="status12" />
<ACLineSegment id="L1" loc="62,981 20,20" data="#805093" show="status13" />
<ACLineSegment id="L2" loc="462,981 20,20" data="#805094" show="status14" />
</PowerPlant>
<DataList type="state" num="8" start="1" end="8" />
<DataList type="meaure" num="12" start="9" end="20" />
</G>
```

图 E.2 厂站单线图 CIM/G 文本