

统一时空电网数据模型与访问技术^①

陈 升

(国网信息通信产业集团厦门亿力吉奥信息科技有限公司, 厦门 361008)

(国网信通亿力科技有限责任公司, 福州 350003)

摘 要: 在电网的地理信息系统发展过程中, 存在着多种不同的文件或空间数据库格式, 也产生了大量的电网时空数据. 为解决数据集中或各个系统间数据交互时, 面临的异构数据问题以及时空数据应用问题, 特提出统一时空电网数据模型与访问技术. 本文从实际应用出发, 利用元数据思想及软件插件模式, 从数据及访问层面解决了异构数据和时态数据的使用. 通过应用证明, 采用统一空间数据模型与访问技术方式, 能够有效地减少数据准备、系统开发的工作量, 并且可以整合利用不同数据源各自的优点.

关键词: 时空数据; 时态 GIS; 电网 GIS; 统一数据模型; 统一数据访问

Uniform Spatio-Temporal Power-Grid Data Model and Access Technology

CHEN Sheng

(Xiamen Great Power Geo Information Technology Co., Ltd., State Grid Information & Telecommunication Group, Xiamen 361008, China)

(State Grid Info-Telecom Great Power Science and Technology Co., Ltd., Fuzhou 350003, China)

Abstract: In the development of power-grid GIS, there are varieties of files and many types of spatial database formats, and also a large number of grid spatio-temporal data are produced. In order to solve the problem of heterogeneous data and the application of spatio-temporal data in data exchange between different systems, data model and access technology of unified spatio-temporal power grid are proposed. Based on practical applications, this paper solves the use of heterogeneous data and temporal data from the data and access level by using the metadata concept and software plug-in model. It is proved by the application that the unified spatial data model and access technology can effectively reduce the workload of data preparation and system development, and can integrate the advantages of different data sources.

Key words: spatio-temporal data; temporal GIS; power-grid GIS; uniform data model; uniform data access

在电网的信息化过程中, 地理信息系统(GIS)获得了广泛的应用^[1-4]. 这一过程中, 由许多不同厂商分别构建的信息系统, 形成多种不同的文件或数据库格式, 即异构数据^[5]. 当用户使用了某个公司的数据存储格式之后, 常常就被迫使用此公司的一系列相关软件, 用以访问他们自己的数据. 如果遇到必须使用其它公司的软件的情况, 往往需要耗费大量的人力、物力把数据从一种格式转换成另一种格式, 或者想方设法的去读写这些专有格式的数据. 若数据来源于不同的数据库或文件, 就需要将这些数据或文件迁移到指定的

数据库中. 在电力信息系统中, 由于平台发展历程及技术选型的不同, 也导致存在大量异构数据的问题, 使得数据更新升级、业务功能开发等存在一系列的障碍.

另外一方面, 人们对于电网的理解都是基于生产环境中的现有运行电网. 电网资源的存储也仅仅是现有运行态的电网资源, 不具有处理数据的时间动态性, 只是描述数据的瞬时状态(snapshot). 当数据发生变化时, 用新数据代替旧数据, 系统成为另一个瞬态, 旧数据不复存在, 因而无法对数据变化的历史进行分析, 更无法预测未来的趋势. 目前业务需要对电网的分析

^① 基金项目: 国家电网公司科技项目(SGITG-KJ-JSKF[2015]0012)

收稿时间: 2016-03-24; 收到修改稿时间: 2016-06-06 [doi: 10.15888/j.cnki.csa.005566]

都是对运行态的电网资源做实时分析. 随着业务精细化的发展, 需要针对历史电网进行分析时, 目前电网的地理信息系统(GIS)是无法满足其业务需求. 如当现有的电网资源满足不了用户用电需求的时候, 就会基于现有电网资源做规划, 需要参考在某一个历史时刻的电网运行情况. 再者, 基于历史时刻的电网运行分析、为其他业务提供决策等等, 所有这些都需要有效保存并管理历史变化数据, 以便将来重建历史状态、跟踪变化、预测未来. 这就要求在实现对传统 GIS 中的空间数据进行组织、管理、操作的同时, 还能对空间数据的历史演变过程进行模拟或回溯.

为解决这一系列数据迁移及处理问题, 简化数据准备及访问流程, 提出统一时空数据模型及访问技术. 希望通过简单的配置, 实现不同格式和不同来源的数据的高效利用.

1 统一电网数据模型

统一电网数据模型是一种元数据模型^[6,7], 通过对原始数据及地理对象的元数据描述, 达到抽象电力设备, 屏蔽底层数据不同结构的目的.

对于描述同一电力设备的不同数据, 如设备的空间表、设备的属性表等, 可以抽象的视为描述这一设备不同方面的原始数据(元件). 这些原始数据之间通过唯一 id 或名称存在关联关系(元件之间的关系). 在此基础上通过定义设备的逻辑描述(对象), 可以将实际需要的空间或属性数据从原始数据中提取出来, 组成新的设备数据. 整个空间数据模型的思想类似于数据库的表与视图的关系.

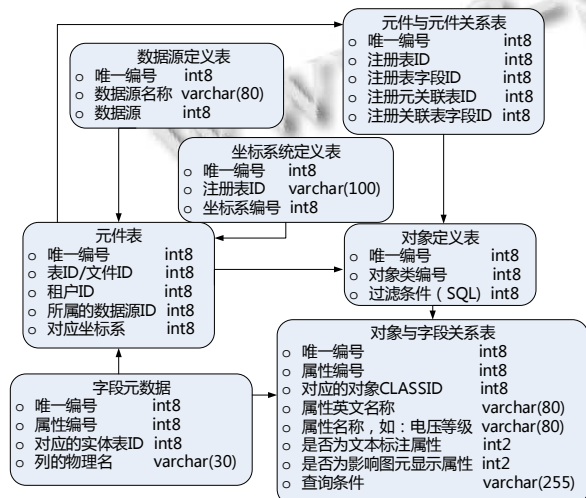


图1 空间数据元数据模型

统一数据模型由一系列元数据描述表组成, 包括坐标系统定义表、数据源定义表、元件表、元件与元件关系表、字段元数据表、对象定义表、对象与字段关系表. 坐标系统定义表定义了数据来源可能存在的坐标系统, 每一个元件都会对应一种坐标系统. 数据源定义表定义了各种数据来源的连接参数及其它信息, 供访问各个元件的数据时连接数据源使用. 元件表定义了指向的原始数据及数据的一些基本信息, 包括坐标系, 可否编辑、空间字段名称、空间字段几何类型、空间数据范围等. 元件与元件关系表定义了同一对象的元件之间如何关联. 字段元数据表定义了一个元件包含原始数据哪些字段及字段的描述信息. 对象表定义对象的基本信息: 名称、过滤条件等. 对象与字段关系表定义了对象包含的元件及这些元件的字段、元件数据提取条件、是否是显示字段等信息.

依据这一基本思路, 建立统一空间数据元数据模型表. 定义设备时, 首先根据已有的数据表定义对应的设备元件、设备元件字段元数据、数据表坐标系及数据表数据源; 然后根据各个设备元件表的连接关系定义元件与元件关系; 最后定义设备对象, 根据设备使用需要的信息定义设备对象与元件字段关系, 将设备对象与设备元件关联, 最终形成完整的设备定义. 使用时根据设备对象, 找到对应的设备元件字段, 根据元件的数据源及数据表信息连接设备数据实体表再抽取相应数据, 组成完整的设备对象.

经过验证, 此模型可以很好的适配不同的设备数据表且可以根据应用程序需要高效的组织成为不同的设备对象, 能够简单容易的响应设备数据表的变化. 对于设备对象需求的变化也可以比较容易的予以响应. 可以应对设备数据来源复杂、冗余, 设备对象变更的情况.

2 时空电网模型

根据时空数据模型方法以及电网实际业务需要, 在电网资源模型中加入时态化、多级版本^[8,9], 支持电网模型的全生命周期, 满足运行、设计、规划应用需求.

电网运行版本下面分为多个规划版本和多个设计版本. 规划版本面向电网规划, 主要内容是输变配电的主网架结构, 细节的线路和设备内容可以被抽象省略. 设计版本面向电网工程初步设计, 设计版本下面

包含多次的设计变更, 每次设计变更都是叠加在设计版本和前面各次的设计变更之上。

电力资源在电网运行以后数据变更很频繁. 对于电网运行态中电力资源数据变动情况, 将运行版本下的电网状态沿时间轴的变动划分为若干个连续的时间断面(即时间断面链), 在每个时间断面中以“增量”的形式记录该断面中新建、修改和删除的电网模型的图形拓扑和属性. 电网模型的修改历史过程从初始状态断面 1 开始, 每次对电网模型的修改都被记录在一个状态断面中, 依次为状态断面 2、3、...*i*-1、*i*、*j*、*k* 至电网模型的最新版本. 电网状态的修改是采用增量记录的形式保存在电网模型数据中。

时空电网模型中, 利用面向对象抽象方法将每个电力资源数据对象分为设备属性对象、空间图元对象、拓扑连接关系对象, 同时在每个对象上增加 timestamp(时间戳)、flags(修改标识)两个属性: timestamp 是进行数据编辑发布时如果存在时态数据变更则需要分裂产生一个新历史对象时, 对其标识的分裂时间; flags 是用来标识电力资源对象是否被删除, 当该对象删除时, 通过该字段进行标识存储, 同时分裂一个历史对象. 在时态查询电力资源数据时, 默认情况下查询最新时间下的最新数据, 如需要查询历史某一个时刻、某一个时间范围的电力资源数据时, 则通过记录时间戳 timestamp、修改标识字段 flags 来索引进行过滤查询其子图元对象、设备属性对象然后封装成电力资源对象集合进行返回. 数据状态和版本的变化包括以下 4 种^[10]:

- ① 基版数据: 电网运行中某一时刻相对稳定不变的对应电网数据;
- ② 编辑态编辑: 当电网运行态发生变更时, 需要对其电网基版数据进行编辑, 系统提供工单机制, 所有数据编辑都是基于工单进行编辑, 编辑未发布时, 对所有非本工单用户权限是非可见的, 是一种临时数据, 用户可以撤销自己的编辑数据, 同时可将其使用“发布”功能将其编辑数据发布为正式数据;
- ③ 编辑态发布: 对编辑态编辑的临时数据, 发布为正式数据, 将电网运行态记录到一个新的基版状态下, 其变更数据对所有用户可见, 发布的同时, 比对新基版状态与原基版状态的差异, 并对其差异内容按照时间戳 timestamp 进行备份, 形成可以根据时间戳进行查询的历史断面数据;
- ④ 历史态查看: 每次基版数据编辑发布后, 对其原来的基版数据则以历史断面方式进行存储, 支持可以按照时间戳、时间戳范围对其电网运行历史态数据

进行查询、分析。

通过以上所描述的方法, 实现对电网过去、现在、将来的可视化展示与对比分析, 包括以下四个方面. 规划设计版本查询: 根据时间和版本信息查询规划和设计数据, 与现在的电网叠加现实, 动态展示差异和变化情况. 电网空间信息时态查询: 根据时间条件查询过去电网的建设情况, 过滤掉大于时间条件后面的数据, 动态叠加显示规划数据展示未来电网的建设情况. 电网状态信息横向时态查询: 通过时间段查询发生状态变化的数据, 已经变化的情况. 电网状态信息纵向时态查询: 查询设备发生状态变化的时间以及详细情况。

3 统一空间数据访问技术

统一空间数据访问技术是一套具有访问不同数据源能力的空间数据引擎模型, 解决不同数据源间适配问题, 支持物理数据库, 内存数据库、文件数据等不同模式数据源。

统一空间数据访问技术将空间数据(Geometry)和属性数据(int、double、bool、string、blob.....)抽象为统一的数据表示; 将一个数据库或一个数据文件抽象成一个数据源, 通过数据源能力描述信息定义数据源可以支持的操作(支持的操作命令、多线程、支持的空间操作、事务.....); 数据库中的每个表或文件中的每个图层抽象成一个属性表, 辅以属性表描述信息及属性描述信息组成完整的数据属性表描述. 它们之间的联系如图 2 所示。

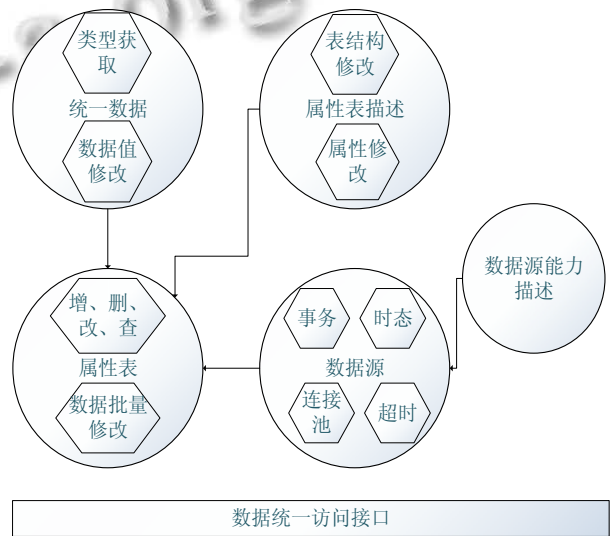


图 2 统一空间数据访问模型

对于外部应用,数据源和属性表使用统一访问接口操作数据及获取数据源和数据的支撑信息,屏蔽各种数据源的不同实现,使不同的数据源对外拥有同样的行为,外部应用访问不同源的数据时采用相同的方式。

对于每种不同的数据源实现我们称之为对应数据源的 Provider。每个 Provider 按照统一空间数据访问的抽象原则,具体针对本数据源的数据特点,提供统一访问要求的行为。只要安装了对应的 Provider,应用程序就可以通过调用统一接口来直接访问相应的数据存储,而不再需要进行格式转换。

按照这一思路,使用 C++定义统一空间数据访问接口并分别实现 oracle、postgresql、内存库、shape 格式的支持。

统一空间数据访问技术实现包括 Value(数据)、Data(要素)、Config(连接信息)、AttributeDescription(属性)、ClassDefinition(属性表描述)、DataClass(属性表)、DataSource(数据源)、DataSourceCapabilities(数据源能力)、Expression(表达式)、QueryFilter(过滤器)、DataCommandRespond(结果集)、DataReader(读取器)、Transaction(事务)、SpatialOperation(空间关系)。

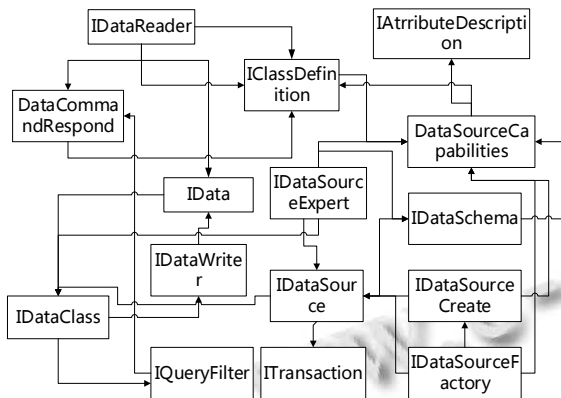


图3 统一空间数据访问实现框架

应用程序使用 C++实现的 oracle、postgresql、内存库 Provider 可以对各个数据源的数据进行增删改查等操作,且效率上无明显损失。

4 应用效果

在电力 GIS 系统实现中,应用统一空间数据模型作为电力数据模型,使用统一空间数据访问技术作为系统数据访问层。使用统一空间数据模型和统一空间

数据访问技术比传统方式有明显优势。表 1 展示了系统建设中关键流程两种方式需要的工作对比,可以看出使用统一数据模型及访问技术能够明显减少地工作量。

表 1 与传统方式的工作量对比

	统一空间数据模型 与访问技术	传统空间数据模型与访问技术
空间 数据	不需额外工作	需要将多个数据源的数据合并到一个数据源(数据库迁移)
		不足 <ol style="list-style-type: none"> 1、数据迁移可能因数据源原因导致部分数据精度丢失 2、不同数据源支持存储方法、数据类型不同,需要解决 3、数据迁移需要时间较长
数据 访问	如果增加之前未支持的新数据源,需要开发新的访问插件	需要修改切换数据源后不兼容的地方
		不足 <ol style="list-style-type: none"> 4、切换不同数据源对使用可能影响较大,会出现原有访问方式不兼容问题(如 sql 语法不可用,文件要转换为数据库存储等)
原有 业务 的支持	仍然支持原有程序运行	使用扩展后的模型
		不足 <ol style="list-style-type: none"> 5、应用要使用扩展的数据模型,会出现之前程序不适用的问题(如描述信息被合并、类型改变等) 6、需要做很多修改,开发周期加长

每种数据源所用的数据库往往都有自己的优点,导致最初的开发者使用了不同的数据库。传统迁移合并的方式,会牺牲掉一些数据库的优势。表 2 展示了在 500 万数据的数据表上执行 150 个线程操作时单一数据源与多数据源方式平均消耗时间的对比。由此体现了通过统一空间数据访问,可以发挥每种数据库的长处,在整体性能上达到优化平衡。

表 2 不同数据源的访问性能对比

	PG	Oracle	统一数据访问
空间查询平均响应时间(秒)	0.0045	0.0035	0.0038
属性查询平均响应时间(秒)	0.0012	0.0017	0.0014
数据插入速度(条/秒)	587	582	584

对于不同网省数据结构的差异化,通过统一空间数据模型定义相同的数据模型表现层,在外部使用数

据时视为一整套相同结构的数据,可以大大简化外部业务的实现逻辑.图 4 展示了统一空间数据模型的管理界面,通过管理功能,可以方便的以对象化方式构建空间数据模型或进行调整.



图 4 对象元件管理实现

不同网省或者同一网省不同业务系统使用的数据存储方式存在差异,如 oracle、shape、sqlserver 等,通过统一空间数据访问技术,可以屏蔽这些差异,使外部应用不需要关心数据具体的存储技术.图 5 展示了使用统一数据模型实现的 GIS 系统显示效果.

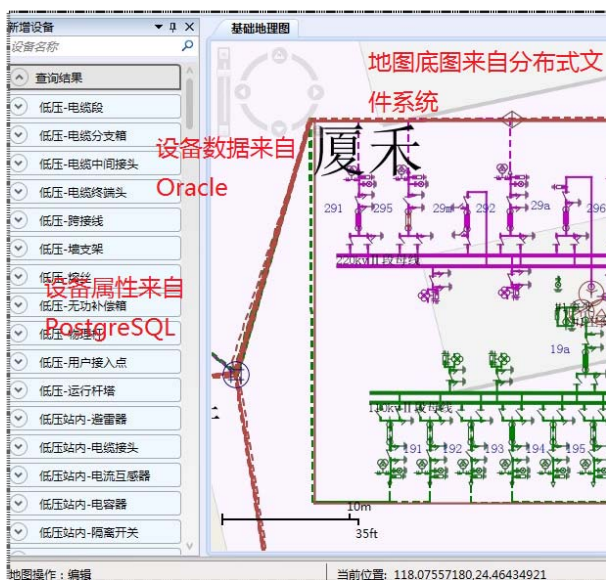


图 5 多数据源数据访问示例

5 结语

实践证明使用统一空间数据模型和统一空间数据访问技术配合,可以从源头上解决地理数据的异构问题.在使用接口及表现方式上对外部应用程序屏蔽数据结构异构或者数据存储技术异构,使应用程序使用数据时采用相同的数据访问及操作行为,能够最大限度的降低数据异构对外部业务逻辑的干扰.在应用中可以简化系统数据处理实现流程,缩短系统开发周期.具有极好的灵活性,对老旧数据系统或新的数据系统接入具有很好的兼容性.对不兼容格式响应迅速,新格式支持以独立插件方式加入系统的数据访问模块,不需要上层修改代码也不会对现有功能造成任何额外影响.

参考文献

- 1 温振龙,许今中,李莉红,赵宏萍.基于全生命周期管理理念的固定资产地理信息系统建设.陕西电力,2009,37(4):69-72.
- 2 曹海涛,贾博,张波.移动 GIS 切片地图展现技术.计算机系统应用,2013,22(12):215-218.
- 3 俞亮.基于 GIS 的电网规划设计系统架构设计及系统运行案例分析.科技创新导报,2015,(17):64-65.
- 4 刘春波.地理信息系统 GIS 在电网企业的应用分析.信息化建设,2016(1):109.
- 5 张凯.一种异构数据源协调器的设计与实现[硕士学位论文].长春:吉林大学,2005.
- 6 赵汗青,郭刚.地理空间信息元数据模型.测绘通报,2002(z1):19-23.
- 7 李建花,曲守宁.面向信息集成的元数据模型的研究和应用.计算机与数字工程,2010,38(12).
- 8 龚健雅.GIS 中面向对象时空数据模型.测绘学报,1997,(4):289-298.
- 9 陈荣权,蒋锦霞.基于版本化模型的配电网辅助设计系统的实现.电力信息与通信技术,2011,9(8):24-27.
- 10 吴淑玮,闫训超,郭宝,等.多时态统一电网模型.计算机系统应用,2015,24(2):244-247.