第34卷 第23期 计 算 机 工 程 2008年12月
Vol.34 No.23 Computer Engineering December 2008
·多媒体技术及应用· 文章编号：1000—3428(2008)23—0237—02 文献标识码：A 中图分类号：N945

VegaGIS可视化系统的设计和实现

郎 兵[1,2]，方金云[1]，韩承德[1]
(1. 中国科学院计算技术研究所空间信息处理技术实验室，北京 100190；2. 中国科学院研究生院，北京 100049)

摘 要：针对跨平台多层体系结构的GIS系统要求，设计一种GIS可视化系统架构。该构架实现了GIS可视化系统常规功能和跨操作系统平台，具有动态使用多种绘制引擎、动态组装绘制算法、带有自适应输出设备等特性，能适应需求变更和图形技术的发展。在VegaGIS平台上得到实现，已被成功应用于多个领域。

关键词：可视化系统；VegaGIS平台；地理信息系统

Design and Implementation of VegaGIS Visualization System

LANG Bing[1,2], FANG Jin-yun[1], HAN Cheng-de[1]
(1. Laboratory for Spatial Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190;
2. Graduate University of Chinese Academy of Sciences, Beijing 100049)

【Abstract】 Aiming at the request of Geographical Information System(GIS) system for cross platform multilayer structure, this paper designs VegaGIS visualization system architecture. The architecture realizes the general function of GIS visualization system and the cross platform for OS. It has several characteristics including self-adapting to multiple operating system, dynamic rendering engines, dynamic rendering algorithms, and self-adapting to output devices. The design is implemented on the VegaGIS platform, and is successfully used in several fields.

【Key words】 visualization system; VegaGIS; Geographical Information System(GIS)

1 概述

地理信息系统(Geographical Information System, GIS)是在计算机软、硬件系统支持下，对现实世界各类空间数据及描述这些空间数据特性的属性数据进行采集、存储、管理、运算、分析、显示的系统[1]。地理空间数据的显示和表达是地理信息系统最重要的功能之一，因此，矢量图形可视化系统是GIS系统的重要组成部分。

矢量图形系统被广泛研究并应用，文献[2]在实现InterViews系统时提出了一套基于C++的图形绘制框架Unidraw，定义了components, tools, commands, external 4套基本抽象组件，它们分别定义图形对象的外观和行为、操作、命令以及映射。上述4套组件抽象了图形绘制的工作流，简化了图形系统的应用。文献[3]描述了HotDraw图形系统，HotDraw定义了10种图形系统模式，包括绘制、交互、存储等过程，这些模式被以后的图形系统大量使用。

美国环境研究所的ArcGIS[4]是技术领先的GIS厂商，其可视化系统速度快、显示美观，得到了广泛应用，但存在以下不足：(1)二次开发接口不能实现具体算法级的动态组装和拆分；(2)其客户绘制系统在Windows下只支持GDI方式，无法利用其他图形绘制引擎。

地理信息的表达具有数据量大、数据多尺度、需要制图与出版等特点，基于上述特点以及地理数据表达的特殊性，VegaGIS可视化系统主要实现了以下功能：

(1)可视化系统的跨平台性(操作系统无关性)；

(2)可定制的绘制引擎系统(GDI, GDI+, OpenGL, DirectX)；

(3)不同绘制算法的动态组装；

(4)绘制算法对绘制设备(显示器、打印机)的自适应。

2 可视化系统的设计与实现

2.1 系统的模块划分

地理信息的可视化系统包括：绘制工具，绘制的过程以及地理信息独有的符号表达3个部分。因此，VegaGIS可视化系统也相应地按逻辑划分为3个模块：绘制资源模块，绘制模块和符号化模块。上述3个模块共同构成整个VegaGIS可视化系统。绘制资源模块生成绘制过程中需要的资源，包括画笔、画刷等。绘制模块实现对各种地理要素的绘制表达。符号化模块实现对地理要素符号化的功能。因为整个系统设计要求必须是跨平台的，所以可视化系统在实现3个模块的过程中，使用抽象、分层、设计模式的方法，隔离平台相关性与实现，以满足跨平台要求。

2.2 绘制资源的抽象与封装

绘制资源的抽象与封装主要有2个目的，即屏蔽平台相关性以及使地理要素与其可视化参数实现分离。

绘制资源是较典型的操作系统相关实现，Windows环境下一般都使用GDI或GDI+来实现绘制显示，而Linux环境下常用QT, KDE等框架实现。为了实现跨平台，系统提出了绘制资源抽象层，使GIS系统的其他模块只要使用抽象的绘制资源，而不用依赖具体平台相关的绘制资源。具体的平台实现可以在Runtime阶段进行动态组装。

绘制资源的抽象与封装可以使地理要素与其可视化参数

基金项目：国家“863”计划基金资助项目(2002AA114020, 2001AA135210)；中国科学院知识创新基金资助项目(20036020)

作者简介：郎 兵(1981－)，男，博士研究生，主研方向：虚拟现实，GIS可视化，软件工程；方金云，副研究员、博士；韩承德，研究员、博士生导师

收稿日期：2008-07-03 **E-mail：**langb@ict.ac.cn

分离，常规做法是将所有表现信息集成在地理要素类中，并在地理要素类里集成大量功能参数。进行绘制时，需要对各种参数进行识别和组织才能完成整个绘制过程，破坏了地理要素类的逻辑独立性，并导致代码冗长。

本文方法是在系统中将所有绘制资源封装成描述类的体系，而地理要素类只要看到整个体系的抽象接口，无须考虑资源分配和使用的具体实现，可以在 Runtime 阶段进行组装和分派，从而解决了上述问题。所有对参数的使用、组装等操作都封装在绘制资源体系内部，可以根据需要适时扩展，且实现了对客户使用的透明。

在绘制资源体系里，先设计资源边界接口，再分出画笔和画刷的边界接口，然后衍生出各种实际取用的画笔和画刷类型。它们存储自身表达所需的各种描述参数，并通过 initial 方法来根据参数形成。

使用上述绘制资源体系以后，地理要素类只依赖资源的边界接口，边界接口提供了 get 和 set 方法，可以动态改变关联的资源类型，实现了绘制资源与地理要素的分离。

2.3 多引擎绘制模块架构的设计

设计系统的绘制模块时，要解决多种绘制引擎的适应问题，以及几个相关模块的衔接与参数传递问题。因为绘制过程涉及多个模块，如符号模块、要素模块、资源模块以及最终的绘制算法实现模块，所以在进行此部分架构设计时，要抽象绘制引擎的接口，减少模块的相互依赖和关联，尽可能分离它们的职责，并使整个过程可以在 Runtime 阶段确定。绘制体系的架构见图 1。

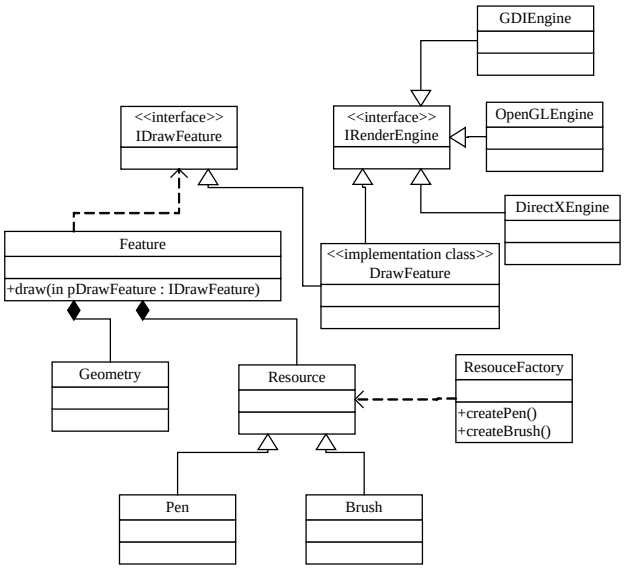

图 1　绘制体系的架构

在设计和实现绘制模块架构时，系统设计了资源工厂类 SicResFactory，使用工厂方法来生成资源。工厂会调用上文所述抽象资源接口，并在 Runtime 根据客户的传入参数生成具体绘制资源，从而通过工厂屏蔽绘制资源模块与绘制模块之间的依赖关系。

为了实现对各种绘制引擎的支持，系统设计了抽象引擎 IrenderEngine, IRenderEngine 向系统提供公共统一的绘制方法。当系统需要绘制基础图形时，会向 IRenderEngine 查询相应绘制方法，而不会去直接调用某个绘制引擎的方法。系统将继承实现各种绘制引擎的方法，向绘制系统提供绘制的底层功能。但绘制系统看不到具体实现方法，具体使用哪种引擎由客户在 Runtime 阶段装入决定。

本文系统设计了抽象绘制接口类 IdrawFeature。IDrawFeature 是封装了地理要素绘制方法的接口，要素类只看到绘制接口类，而完全看不到绘制的实现方法和算法。具体实现绘制方法和算法的绘制类 SicDrawFeature 则从 IDrawFeature 继承，并组合 IRenderEngine，实现其内部封装的所有绘制方法的接口，实现了要素体系和具体绘制方法及算法的解耦合，极大增加了系统的扩展性。上述设计方法实现了绘制模块中的要素类、绘制资源、绘制方法和绘制引擎的解耦合，减少了它们之间的依赖关系，使这 4 个模块几乎完全相互独立，只要保持接口不变，更改一个模块不会影响其他模块。对资源和绘制方法以及绘制引擎的组装可以实现成动态行为，即都推迟到 Runtime 阶段，通过各自的组装接口就实现了透明的组装绘制过程。

如图 1 所示，资源与要素通过组合方式进行设计，要素与绘制实现通过接口方式进行设计。资源对客户使用工厂的方法，实现了对绘制中模块的彻底解耦合，因此，可以动态组合地进行工作。

2.4 符号化绘制接口的设计和实现

符号化绘制体系如图 2 所示。

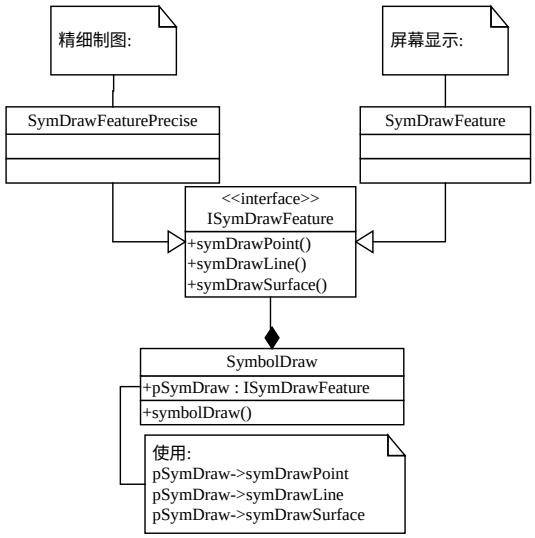

图 2　符号化绘制体系

地理要素的符号化是 GIS 的重要功能，即为各种地理要素铺设设计好的符号。符号化接口是 GIS 可视化系统独有的组成部分。符号化接口要求简单、方便调用，它将算法和实现都放到符号模块内完成，GIS 客户端只要看到使用的方法和参数即可。符号化模块既有对数字媒体设备显示的需要，也有为制作纸质地图的要求，为了使系统能适应算法的不断更迭、减少变更带来的影响，从而使算法的替换和更改对接口完全没有影响，本文系统使用了模板方式。系统在模板中定义算法的骨架，而实现则延迟到子类中，使子类可以动态更改算法的实现。

结合模板方法和上述客户端接口设计，系统可以实现良好的组织，满足以上要求。此系统具有以下优点：(1)可以固定对外的使用接口，使用户在使用时不必考虑接口的实现和算法，只要传递正确的参数即可；(2)在符号系统的实现端，要考虑各种算法的演进和更迭，使系统在稳定接口的同时可以对算法进行动态组装和替换，把实现的具体工作放在 Runtime 阶段进行确定和组装。 （下转第 246 页）