基于 Java 拓扑模型和 RCP 的 GIS 平台研究

高 昂[1,2],陈荣国[1],卫文学[2],孙 剑[1,2],郎玲玲[1,3]
GAO Ang[1,2],CHEN Rong-guo[1],WEI Wen-xue[2],SUN Jian[1,2],LANG Ling-ling[1,3]

1.中国科学院 地理科学与资源研究所 资源与环境信息系统国家重点实验室,北京 100101
2.山东科技大学 信息科学与工程学院,山东 青岛 266510
3.北京师范大学 地理学与遥感科学学院,北京 100875
1.LERIS,Institute of Geographic Sciences and Natural Resources Research,CAS,Beijing 100101,China
2.CISE,Shandong University of Science and Technology,Qingdao,Shandong 266510,China
3.School of Geography,Beijing Normal University,Beijing 100875,China
E-mail:gaoang@lreis.ac.cn

GAO Ang,CHEN Rong-guo,WEI Wen-xue,et al.Research on Java GIS platform with Eclipse RCP and JTS framework.Computer Engineering and Applications,2007,43(5):106-109.

Abstract: To solve the problems such as poor expansibility and long period development in building GIS application,a best practice of how to combine and deploy JTS and Eclipse framework to build high performance Geographic Information System has been provided.The combination of using JTS and RCP frameworks could construct an extensible and standard multi-platform GIS desktop application as well as improving the development efficiency greatly.
Key words: JTS;GIS;Eclipse;RCP;rich client platform

摘 要:基于传统地理信息系统开发周期长、拓展不灵活、应用平台单一等问题,给出使用 Java 拓扑模型与 Eclipse 富客户端平台整合开发的实例,同时研究利用 Java 拓扑模型提供的空间分析算法和空间对象的处理功能,简化系统的复杂程度并且规范化开发流程,并且通过 Eclipse 富客户端平台组织和管理各部分组件,快速构建易于拓展的产品,以及如何充分发挥两者优势进行系统开发,构建结构清晰、具备强大扩展性和稳定性的跨平台地理信息系统。
关键词:Java 拓扑模型;地理信息系统;Eclipse;富客户端平台
文章编号:1002-8331(2007)05-0106-04 **文献标识码**:A **中图分类号**:TP311.52

1 引言

软件框架一般是可复用的设计架构,它规定了应用程序的体系结构,阐明了协作构件之间的依赖关系、责任分配和控制流程,提供一组可以直接调用的抽象类,同时规范类的各实例间的协作方法。随着软件工业的发展,软件开发的方式已经从使用通用类库进化到了面向领域的开源代码及应用框架的复用。基于优秀软件框架进行研究开发,不但可以迅速构建应用系统,更可以借用优秀的软件设计思想及程序架构,并且根据实际的要求进行改进与创新。

Java 拓扑模型(Java Topology Suite)是服务于地理信息的优秀开源框架,通过调用模型提供的空间分析算法与空间数据处理功能,可以实现基于 Java 语言的高效跨平台地理信息系统。Eclipse 是由纯 Java 技术打造的新一代开源集成开发环境,使用 Eclipse 的富客户端平台框架(Rich Client Platform)可以便捷地重用 Eclipse 的各种功能,集成到富客户端产品之中。基于 Eclipse 完善的架构创建富客户端应用程序,不但可以规范界面风格,保证程序的健壮性和拓展性,同时也使软件设计者集中精力在业务逻辑实现中,不被繁琐重复性的劳动所羁绊。

两种优秀的 Java 框架整合进行软件产品开发,可以缩短开发周期、标准化软件开发过程,更重要的是可以构建一个易用性及拓展性强,并且跨越各种操作系统平台,支持多种语言的桌面 GIS 应用。

2 系统关键技术实现方式

2.1 Java 拓扑模型中间件

Java 拓扑模型(Java Topology Suite)提供了实现二维空间分析功能的 Java 应用程序接口。Java 拓扑模型利用精确的模型和成熟的几何算法,为二维几何数据提供了完整一致的空间核心算法函数。同时 Java 拓扑模型实现了 OpenGIS 标准规定的简单特征规范,即通过一组符合 OpenGIS 标准的通用组件,实现在网络环境中对不同种类地理数据和处理方法透明访问。

作为中间件的 Java 拓扑模型,处于应用软件和系统软件之间的软件层。中间件屏蔽了环境底层的复杂性,提供给软件设计人员功能统一的应用程序接口。通过 Java 拓扑模型实现

基金项目:国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2004AA132020)。
作者简介:高昂(1982-),男,硕士研究生,IEEE-CS、CCF 学生会员,主要研究方向:空间数据库、地理信息系统、网格 GIS;陈荣国(1963-),男,博士,研究员,主要研究方向:地理信息系统、空间数据库;卫文学(1967-),男,博士,副教授,主要研究方向:计算机网络、软件工程。

应用之间的互操作，屏蔽信息访问的底层细节，并向程序调用提供标准接口。程序通过接口与中间件进行通信，保持客户应用的相对独立性。当信息发生变化时，只要在中间件层作相应的更新即可，应用程序不做任何修改就可以继续应用于新的系统。

由于 Java 拓扑模型封装了地理几何基础算法以及空间数据处理通用操作，同时提供的程序接口层次清晰、结构一致、易于程序调用及后期维护升级。作为遵循 LGPL 开源协议的成熟模型，可以将 JTS 作为 Java 地理信息系统底层支持环境，应用于产品级软件的开发。

2.2　基于 Java 拓扑模型的拓展

Java 拓扑模型提供了基于内存的空间索引，使用 JTS 模型可以保证算法的健壮性以及性能优异的计算表现。同时 Java 拓扑模型用作构建其他 GIS 应用的中间件，在许多开源项目中得到了充分的利用，代表性较强的是基于 JTS 模型框架设计的 GeoTools 和 GeoServer 中间件。

建立在 Java 拓扑模型基础上的 Geotools 是采用纯 Java 代码实现的 GIS 中间件，代码包中提供大量应用于地理信息处理的 Java API 类库，包含了几何属性、要素类、协同参照等内容的数据模型，数据处理和过虑功能的查询模型，以及符号化、格网、地理标记语言和XML 文件读取的支持功能，架构设计层次清晰、易于扩展。使用 Geotools 提供的各种地理数据处理功能进行产品开发，可以为软件基础设计提供优秀规范的解决方案、简化软件的设计与实现过程，并且可以在其基础上充分拓展已有功能，打造稳定健壮的软件产品。

另一个以 Java 拓扑模型来提供空间算法底层基础的中间件是 GeoServer，其提供 OpenGIS 关于 Web 服务器标准的 J2EE 实现。作为发布地理数据的 J2EE 服务器，GeoServer 包含了对 Java Servlets 以及 Java Server Pages(JSP)的支持，可以运行在 Tomcat、Resin、JBoss 或 WebLogic 等 Servlet 或 J2EE 容器中，响应客户端发出的地理数据请求，并且可以对访问请求进行并发控制。GeoServer 工作时，客户端向 Web 容器发出数据服务请求，Web 容器解析请求后将地理数据的传送要求转发给 GeoServer 中间件，GeoServer 响应客户端请求，同时把请求转化成空间数据库能够接受的数据查询格式，向数据库检索相应的空间数据和属性数据，并把数据传回客户端显示。整个请求响应过程由 GeoServer 中间件和 Web 容器配合完成，相互协调工作，完成对地理数据请求的信息反馈。

2.3　Eclipse RCP 富客户端插件体系

Eclipse 是 Java 开源领域中优秀的集成开发框架。富客户端平台 Eclipse RCP(Rich Client Platform)是 Eclipse 基于 OSGi 插件体系的最新拓展，同时也是构建 Eclipse 软件产品最为高效可靠的方式。

众多功能强大的插件支持，使得 Elipse 拥有其他结构相对固定的 IDE 开发环境很难具有的灵活性和拓展性。Eclipse 采用基于 Java 窗口组件技术的 SWT 和 JFace 定制用户界面，较之传统的 AWT 和 Swing 窗口组件更加便捷。同时基于 Java 开发的 Eclipse 具有跨平台特性，Eclipse 可以将 Java 代码编译运行在不同的操作系统平台上，并且操作模式不会因为平台转换而有所不同。Eclipse 平台的各部分组成结构如图 1 所示：

由图 1 可以看出，Eclipse 平台核心包含运行时库和 Eclipse 工作平台，其他的各项功能都作为插件加载到 Eclipse 体系当中，遵循统一的 OSGi(Open Services Gateway Initiative)规范对 Eclipse 提供的开放平台进行拓展。OSGi 框架是一个通用、安全、可管理的 Java 框架，在轻量级服务架构应用方面被广泛地支持。OSGi 标准的采用提高了 Eclipse 向多个硬件和操作系统的移植能力，并且提供了对管理的执行环境的兼容性。

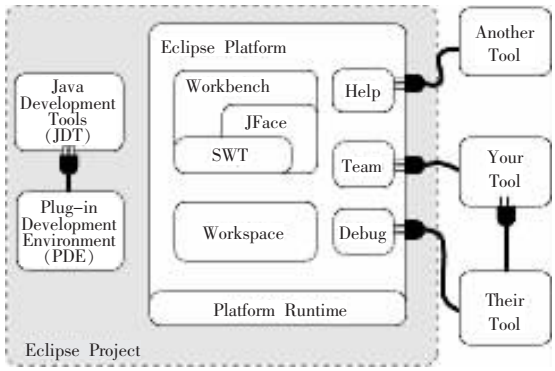

图 1　Eclipse 平台体系结构

Eclipse 拥有基于插件的可扩展体系结构。这个平台允许任何人构建与环境和其它工具无缝集成的工具。除了轻量级的运行时内核之外，Eclipse 中的所有功能插件都是建立在 OSGi 规范基础之上的。OSGi 规范中将插件称为 Bundle，Bundle 作为整个插件的生命周期管理对象，负责插件的启动和停止动作，具体描述工作通过标准的 mainfest.mf 文件来完成，其内容主要包括 Bundle 的名称、版本、对外引用的包、提供的服务、依赖的插件、动态引用类库等。其他插件可通过 Bundle 对象获取插件的定义信息，在 Eclipse 平台处理插件信息时，认为插件是一个 Bundle 外加上定义插件扩展点的 plugin.xml 配置文件所组成。

Eclipse RCP 富客户端框架重构了 Eclipse 核心包，将 IDE 集成开发环境尽可能地从 Eclipse Platform 包中分离出来，这样 Eclipse 就可以作为一个纯粹的不带有 IDE 特征的应用软件运行平台，开发者可以将 Eclipse 提供的 GEF、EMF、SWT 等应用于不同领域的资源包加入到符合 OSGi 规范的插件中，然后将其导出为 Java 富客户端软件产品。

3　系统组件管理配置与空间信息基本功能实现

3.1　Eclipse 富客户端平台的插件拓展配置

从非功能性角度来讲，插件结构需要保持不同版本间插件的一致性，并且要求插件的开发便捷及易管理性强。插件结构可以保证系统有一个统一平稳的体系，所有的交互、扩展都通过插件进行。同时各个插件部分具有相对独立性，整个系统基于一种拼装式的松耦合结构组成，每个插件个体对于外部而言都是一个黑盒，只是通过接口调用来使用黑盒所提供的各种功能。在 Eclipse 富客户端平台中配置插件，需要在功能部件清单 feature.xml 部分定义系统的各部分组成模块，具体定义如下：

```
<? xml version="1.0" encoding="UTF-8"? >
<feature
      id="ac.lreis.ustudio"
      label="ac.lreis.ustudio-feature"
      version="1.1.0"
      provider-name="Lreis uStudio"
      plugin="ac.lreis.ustudio">
   <includes
      id="ac.lreis.ustudio_platform"
```

```
    version="0.0.0"/>
  <includes
    id="ac.lreis.ustudio_application"
    version="0.0.0"/>
  <includes
    id="ac.lreis.ustudio_language"
    version="0.0.0"
    optional="true"/>
  <plugin
    id="ac.lreis.ustudio"
    download-size="0"
    vinstall-size="0"
    version="1.1.0"
    unpack="false"/>
  </feature>
```

在功能部件清单文件中,需要指定使用统一的 Unicode 编码方式。系统基础运行插件的名称遵循 Java 包命名规范定义为 ac.lreis.ustudio,同时定义插件依赖项包含基础平台插件 ac.lreis.ustudio_platform、功能拓展插件 ac.lreis.ustudio_application 以及多国语言支持插件 ac.lreis.ustudio_language 三个部分,其中通过 optional 选项设定多国语言支持插件为可选组件。

针对各个插件部分,在插件配置文件 plugin.xml 中定义相对应的拓展信息和拓展点标记。标准的 plugin.xml 插件内容定义包含插件名、ID 号、版本号等信息,以及插件的依赖关系、插件的运行支持库和插件各部分扩展点标记。由于遵循 OSGi 插件体系,插件基本信息和依赖关系在专门的配置文件 MENIFEST.MF 中定义,所以只需将插件的扩展点标记定义在配置文件 plugin.xml 中,具体代码如下:

```
<plugin>
  <extension-point id="featureType" name="ac.lreis.ustudio.catalog.ui" schema="schema/featureType.exsd"/>
  <extension-point id="fileFormat" name="fileFormat" schema="schema/fileFormat.exsd"/>
  <extension
      point="org.Eclipse.help.contexts">
    <contexts
        file="contexts.xml"
        plugin="ac.lreis.ustudio.catalog.ui">
    </contexts>
  </extension>
  <extension
      point="org.Eclipse.ui.views">
    <view
        name="%Catalog"
        allowMultiple="true"
        category="catalog"
        class="ac.lreis.ustudio.catalog.internal.ui.CatalogView"
        id="ac.lreis.ustudio.catalog.ui.CatalogView">
    </view>
  </extension>
  <extension
      name="catalogPerspectiveContributions"
      point="org.Eclipse.ui.perspectiveExtensions">
    <perspectiveExtension targetID="ac.lreis.ustudio.ui.mapPerspective">
      <viewShortcut id="ac.lreis.ustudio.catalog.ui.CatalogView"/>
      <viewShortcut id="ac.lreis.ustudio.catalog.ui.Search"/>
      <view
          standalone="true"
          relative="bottom"
          relationship="stack"
          id="ac.lreis.ustudio.catalog.ui.CatalogView"/>
    </perspectiveExtension>
  </extension>
</plugin>
```

在上述性配置文件中,所有的拓展都是定义在闭合的 extension 标签之间。插件的扩展点标记使用 XML Schema 子集描述完成某种抽象功能的插件,定义中需要包含简单的属性以及实现具体接口的类名称。配置文件描述了软件产品中视图 View 和透视图 Perspective 的加载位置,以及各部分插件加载时的属性配置,并且加入 exsd 文件中定义的 Eclipse 内置拓展点标记,引用模式 featureType 和 fileFormat 来减少重复性的配置编写工作。

3.2 Java 拓扑模型接口实现地理信息基本功能

作为地理信息系统中的基本功能部件,地图绘制、浏览和放缩等标准功能利用 Java 拓扑模型以及建立在其上的 GeoTools 模型提供的程序调用接口实现。具体的图形处理操作借用开源模型 Java 高级绘图组件 JAI(Java Advanced Image)完成,JAI 提供的程序接口可以方便地实现图像卷积、旋转、扭曲、更换颜色空间、自定义样本分量等功能,应用中集成 JAI 绘图组件模型,简化了 GIS 系统实现过程中图形处理方面的编码工作。同时,现有的 Java 开源模型在功能上已经加入了支持地理标记语言(GML)的 GMLReader 类,用来读取符合 GML 规范的数据,实现系统对符合 GML 规范的空间数据支持。

标准空间数据处理功能的实现部分中,需要调用 Java 拓扑模型中对栅格及矢量数据的处理接口,如果需要集成矢量数据与栅格数据之间的格式转换功能,需要借用 Java 拓扑模型完成两种数据格式之间的转换,接口定义的代码示例如下:

```
interface GridCoverageExchange {
  void dispose();
  Format[] getFormats()
  GridCoverageReader getReader(Object source)
  GridCoverageWriter getWriter(Object destination,Format format)
}
interface Format {
  String getDescription()
  String getDocURL()
  ParameterValueGroup getReadParameters()
  String getVendor()
  ParameterValueGroup getWriteParameters()
}
interface GridCoverageReader {
  String getCurrentSubname()
  Format getFormat()
  GridCoverage read(GeneralParameterValue[] parameters)
}
interface GridCoverageWriter {
  Object getDestination()
  Format getFormat()
  void write(GridCoverage coverage,GeneralParameterValue[] pa-
```

rameters)
}

在上述功能类的定义中提供了 Grid 和 Coverage 数据格式的读写操作接口，可以实现对接口 GridCoverageReader 以及 GridCoverageWriter 的继承与调用。同时在 Reader 和 Writer 接口中分别定义读和写函数对数据进行处理的操作,分别调用程序其他接口类中定义的读写函数完成各自相应的功能。

3.3　构建富客户端插件体系结构

富客户端产品的插件导出为 JAR 包时，要求 MANIFEST.MF 文件与其相应的 JAR 包关联在一起,插件各部分遵循 OSGi 规范的定义。位于 META-INF 目录下的 MANIFEST.MF 文件对 JAR 包中各个依赖项进行配置,每个插件包(Bundle)都需要通过该文件描述其自身信息。符合 OSGi 定义规范的 MANIFEST.MF 文件示例代码如下:

```
Manifest-Version:1.0
Bundle-ManifestVersion:2
Bundle-Name:%plugin.name
Bundle-SymbolicName:ac.lreis.ustudio.catalog.ui;singleton:=true
Bundle-Activator:ac.lreis.ustudio.catalog.ui.CatalogUIPlugin
Bundle-Vendor:ustudio.lreis.ac
Bundle-Localization:plugin
Export-Package:ac.lreis.ustudio.catalog,ac.lreis.ustudio.catalog.ui
Require-Bundle:org.Eclipse.core.runtime,org.Eclipse.ui,
org.Eclipse.jface,ac.lreis.ustudio.ui,ac.lreis.ustudio.libs
Eclipse-AutoStart:true
```

如上述代码所示,OSGi 框架中定义了一组标准的 Header 头信息,每个 Header 都有其特定的含义。符合 OSGi 标准的配置文件 MANIFEST.MF 中包含一组 Header 与 Value 的映射关系，示例代码中 Bundle-SymbolicName 对应的包名为 ac.lreis.ustudio_base;Bundle-Activator 项指定启动和停止 Bundle 的类名;Bundle-Vendor 定义插件所属的包名。Export-Package 表示可以被其它 Bundle 导入并使用的包名,Require-Bundle 中定义插件运行时所依赖和引用的其他包名称。

4　软件系统产品化实现

4.1　基于 RCP 平台的国际化方法

发布成熟的软件产品，良好的多国语言支持是必备要求。基于 Eclipse 提供的开放平台,国际化 RCP 富客户端产品非常方便，使用 Eclipse 的本地语言支持 NLS (National Language Support)可以确保产品界面国际化并且可以处理本地语言的数据格式文件而不出现乱码。

针对产品中的每个插件包,建立带有“nl1”名称的插件片段 Fragments，在插件片段包中可以编写中文以及其他多国语言支持文件。在 Eclipse 中,不必修改程序中的源代码来支持本地语言,只需将代码中的常量字符串外部化到遵循统一规格的 Properties 属性文件中,再针对每种语言,遵循 Unicode 编码规则撰写不同版本的 Properties 属性文件，然后部署到 JAR 文件中即可。包含中文字符串常量文件 Messages_cn.properties 的结构如下:

```
#Created by Lreis uStudio Team
Catalog=\u7C7B\u522B
Provider=Lreis uStudio.
Search=\u641C\u7D22
dataImportWizard.name=\u6570\u636E
newFeatureType=\u521B\u5EFA\u7279\u5F81\u7C7B\u578B
plugin.name="Catalog UI Plug-in"
resetService.label=\u91CD\u7F6E
wizard.filechooser.name=\u9009\u62E9\u6587\u4EF6
wizard.fileopen.name=\u6587\u4EF6
```

在上述 Properties 属性文件中定义了程序中常量与中文 Unicode 编码文字的对照映射关系。使用 Eclipse 外部化字符提取工具将源代码中的字符串提取到 Properties 属性配置文件当中，然后利用 Unicode 编辑器将字符串映射为 Unicode 编码格式的中文,这样当软件产品启动时,加载中文界面只需在启动参数中加上相应的编码标志符号“ZH_CN”,即可实现软件产品的国际化支持。

4.2　软件系统打包与版本发布

遵循富客户端平台规范的 Eclipse 应用可以直接发布,建立产品描述文件 uStudio.product 负责管理产品配置信息和启动参数。最终产品导出时依据插件配置文件 MENIFEST.MF 以及特征描述文件 feature.xml,将每个插件项目依赖的 Eclipse 自身类库及其他插件以 JAR 包形式导出到产品的插件目录 plugins 中,整个产品生成过程可以配置后通过 Eclipse 自动完成。基于 Java 拓扑模型和 RCP 平台实现的地理信息系统运行界面如图 2 所示:

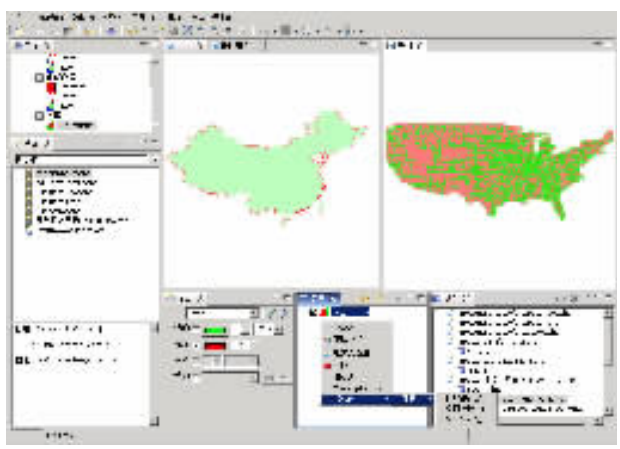

图 2　基于 Eclipse 富客户端平台发布的最终产品

由图 2 可以看出,产品的菜单、编辑窗口以及透视图都符合 Eclipse 统一的界面风格。基于 Eclipse 的 RCP 富客户端规范发布的产品可以运行在不同的操作系统平台之上，由于纯 Java 语言程序运行与操作系统平台无关,只要目标操作系统有 Java 运行环境(Java Runtime Environment)的支持即可,如果为了保证产品良好的通用性，也可以将 Java 运行时环境打包与 RCP 富客户端产品同时发布。

5　结语

利用 Java 拓扑模型成熟的空间分析算法与空间数据处理功能,嵌入到地理信息系统应用中,加快了开发过程,也增加了程序的健壮性和拓展性。Eclipse 富客户端平台的使用,在重用 Eclipse 标准软件界面、功能的同时,也规范了产品开发的流程与架构,使产品具有与 Eclipse 相同的可拓展结构。

优秀的开源软件框架由于其开放性和技术先进性,逐渐成

(下转 142 页)