基于.Net Remoting 的分布式综合录井图的设计与实现①

Design and Implementation of Distributed Logging Diagram System Based on .Net Remoting

沈疆海　杨刚林　(长江大学 计算机科学学院 湖北 荆州 434023)

摘　要：本文提出了一种基于.Net Remoting 远程技术的分布式综合录井图的设计方案。采用该设计方案有利于提高数据的传输效率，并使综合录井图系统的扩展性和兼容性更好。通过油田的实际应用表明，该方案是可行的和有效的。

关键词：.Net Remoting 分布式技术 录井图 设计模式

近年来，随着计算机网络技术的飞速发展，分布式应用，能更有效的利用网络资源，降低企业应用成本，越来越被企业所青睐。综合录井图是石油行业中广泛使用的一种基础图形，从绘图的基础角度看，综合录井图由一些符号、曲线、文字等组成[1]，表示地层在垂向上的变化情况。综合录井图集中测井、录井、地化、地质数据以及储层评价等信息，直观反映地层含油气情况，深受专业人员的喜好。传统的综合录井图绘图软件需由人工繁琐地添加数据，一个图件一个图件地手工做图。随着油田信息化的建设，目前都有较完整的专业数据库，如何利用现有数据快速绘图成了当前亟需解决的问题。本文所讨论的，基于.Net Remoting 的分布式综合录井图就是在此背景下设计与实现的。

1　.Net Remoting技术简介

1.1 .Net Remoting 技术概述

微软的.Net Remoting 是一种允许对象通过应用程序域与另一对象进行交互的框架[2]。该框架提供了多种服务，包括激活和生存期支持，以及负责与远程应用程序进行消息传输的通讯通道,框架如图 1 所示。它提供格式化程序，为消息能在信道上得以传输对其进行编码和解码。

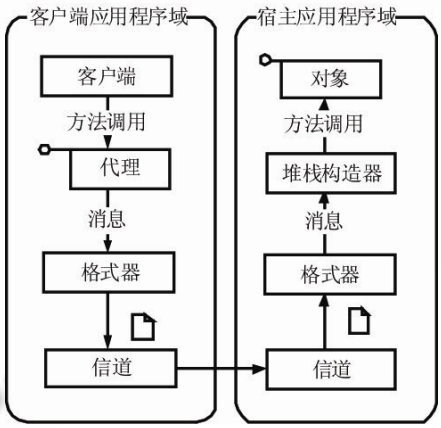

图 1　.Net Remoting 架构

1.2 .Net Remoting 远程对象类型

.Net 对于跨越程序域边界的访问提供了两种类型，值传递和引用传递。通过值传递访问一个对象时，客户端得到的对象是宿主对象的一个克隆拷贝，一旦一个拷贝被传送到远程客户端，那么两个对象是完全不同的并且可以独立改变各自的状态。通过引用传递访问一个对象时，远程客户端只获得宿主对象的一个引用，为一个代理，类似于 COM。

① 基金项目:湖北省自然科学基金项目(2007ABA007)
收稿时间:2008-10-06

1.3 .Net Remoting 引用封送激活模式

.Net Remoting 支持两种类型的引用封送对象，客户端激活对象和服务器端激活对象。客户端激活对象受基于租用的生存期管理器的控制，这种管理器确保了租用期满时对象可被回收。而对于服务器端激活对象，可以选择“SingleCall”模式或“Singleton”模式。

服务器端 SingleCall 激活对象，远程服务器类型总是为每个客户端请求设置一个实例。下一个方法调用将改由其他实例进行服务。从设计角度看，SingleCall 类型提供的功能非常简单,不提供状态管理，如果需要状态管理，这将是一个不利之处；如果不需要，这种机制将非常理想。

服务器端 Singleton 激活对象，该方式遵循传统的 Singleton 设计模式，内存中只有一个实例。由于远程对象的实例只被创建一次，服务于所有的客户程序请求，其生命周期一直保持，直到宿主应用程序被关闭，所以可以在远程对象中保存应用程序的状态信息，缺点是不能在多个服务器上部署，具有可伸缩性问题。

1.4 .Net Remoting 远程对象类型

.Net Remoting 在客户端和宿主程序间对消息进行格式化以便传输。.NET 为远程调用提供了三种协议:TCP、HTTP 和 IPC,统称为传输信道。IPC 是在.Net 2.0 Framework 中新增的，只能在同一机器的跨进程调用时使用，较 TCP 与 HTTP 效率更高，更安全。.Net Remoting 中 TCP 和 IPC 通道默认使用二进制格式化消息，基于原始套接字通信，能有效向远程对象传送消息，能够达到最好的性能，更适合于部署在一个受控制的安全环境中，如企业内部网中的远程对象。

2 基于.Net Remoting的综合录井图系统

2.1 系统需求分析

为能科学、方便、快捷的查询日积月累的勘探开发录井资料数据，综合录井图系统主要解决的问题是，充分利用现有勘探数据资源，有效快速查询并绘制出科研人员所需的专业图形，进行评价分析等。

目前的勘探开发日趋信息化，各种勘探数据有其电子数据，如测井数据，分析化验数据，岩心图像，以及储层的评价信息等，而把这些信息如何集中反映到录井图中，构成了本系统的核心需求。

2.2 系统架构设计

综合录井图系统，涉及数据库、GDI+绘图、Web、.Net Remoting 以及客户端技术等，而本文探讨其中的关键技术是.Net Remoting，依靠它来实现分布式的综合录井图系统，系统结构见图 2。

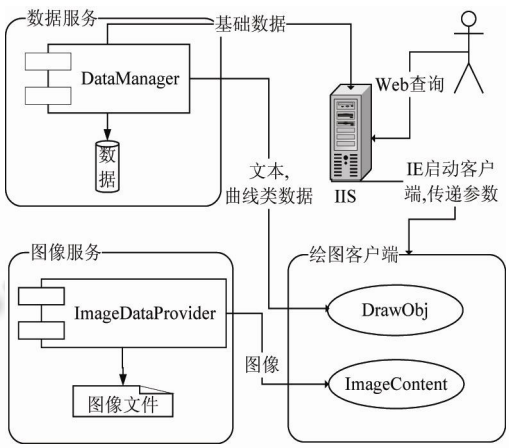

图 2　系统架构

总体设计上，首先抽象出整个系统的功能模块，然后从功能模块中抽象接口及基类，采用面向对象的设计方法，降低各模块间的耦合性。从功能上可分为远程业务逻辑组件，绘图客户端，以及 web 启动模块三大部分。而从数据流来看，综合录井图只包含三类数据：文字性描述、曲线和图像(其中地质符号可归纳为图像)。基于松耦合、部署及扩展的考虑，服务端的宿主程序分为两个：数据服务和图像服务。数据服务宿主程序提供文字描述和曲线类数据，图像服务宿主程序提供图像数据。远程业务逻辑 DataManger 组件负责从数据库中提取数据，供绘图客户端调用。ImageDataProvider 为绘图客户端提供图像数据。而 web 端则按用户选择传递参数启动绘图客户端。

系统的运行流程为 Web 端从网页启动客户端绘图程序，客户端绘图程序根据所定义的模版，加载配置信息，根据配置信息调用远程服务组件的对象。DrawObj 调用远程数据服务组件 DataManger 的提供的文本和曲线类对象；ImageContent 调用远程图像对象 ImageDataProvider 提供的图像类对象,最后由客户端绘图程序，把所取回的远程对象在视图中进行绘制。

2.3 数据远程对象设计

数据远程对象，生成客户端绘图所需的 TextObj

和 CurveObj 对象，通过.Net Remoting 传递给客户端。数据远程对象的类结构见图 3。DataManager 从类 DataProviderDef 派生。DataProviderDef 定义了文本和曲线类数据的传递方法，当然其必须从 MarshalByRefObject 派生，因为只有继承自 MarshalByRefObject 的类才可以跨越应用程序域边界被引用及调用，同时必须实现 ISerializable 接口或具有 Serializable 属性，以便传递的对象能够在远程计算机之间克隆。远程调用时，将产生一个远程对象在本地的透明代理，通过此代理来进行远程调用。DataProviderDef 实现了接口 IDataProvider，接口 IDataProvider 定义了系统主要业务逻辑的数据传输行为，见随后的程序片段。其中 GetTextData 为文本类数据业务逻辑，其中 jh 为传入参数，list 为引用对象链表，查询返回的文本类 TextObj(后面客户端设计中有介绍)对象保存到 list 中，然后通过.Net Remoting 传递给远程客户端，供客户端绘制文本类数据时调用。GetCurveData 为曲线类数据业务逻辑，GetJhData 获取井的开钻和完钻井深，综合录井图的绘制范围以此为上下界限。

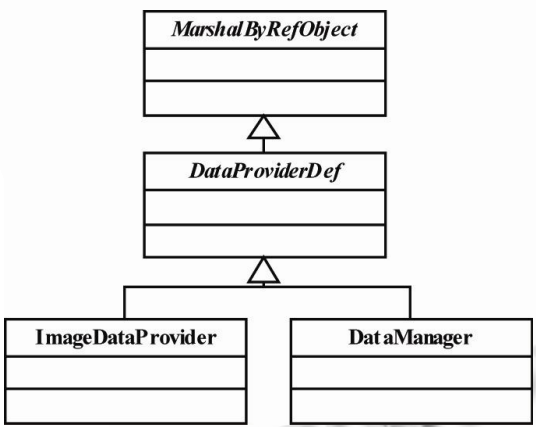

图 3　远程对象类图

```
public interface IDataProvider
{
    bool GetTextData(string jh, ref System.Collections.ArrayList list);
        bool GetImageData(string jh, ref System.Collections.ArrayList list);
        bool GetCurveData(string jh, ref System.Collections.ArrayList list);
        bool GetJhData(string jh, ref int beginDepth, ref int endDepth);
    }
```

2.4 图像远程对象设计

远程图像数据服务，生成客户端绘图所需的 ImageObj 对象，通过.Net Remoting 传递给绘图客户端。数据远程对象的类结构见图 3，实现了 IDataProvider 接口。数据库中存储的图像，实际为图像文件名，当绘制综合图时候，传入图像文件名，从远程图像服务获取图像对象，即 GetImageData 图像类数据业务逻辑，返回的 ImageObj(后面客户端设计中有介绍)对象保存到 list 中，供远程客户端绘制图像。

数据远程对象的宿主程序和图像远程对象的宿主程序均为 Windows 数据服务，基于 windows 的服务程序具有与登录用户无关，系统启动后就连续运行等优点，能更好的承担宿主程序的职责。服务启动时，首先读取.Net Remoting 配置文件，URL 地址设置，端口设置等。前面阐述的数据服务读取数据库配置以及各数据体对应的数据表或视图配置信息，图像服务读取图像存放的文件路径，然后注册用来接受远程客户端调用的信道，最后采用服务器“SingleCall”方式激活。

2.5 客户端设计

客户端从功能上分，属于系统中显示综合录井图部分。采用面向对象设计方式[3]，虽然综合录井图由多种数据类型图单元组成，均有背景颜色，线条粗细，Size 大小等共性，可抽象出基类 DrawObj， 所有绘图对象均派生自基类 DrawObj。客户端绘图程序主要类关系见图 4，由 Well、WellHeader、Track、ImageContent、CurveContent 和 TextContent 组成。其中 Well 为井，WellHeader 为井标题头。一个 Well 包含多个 Track 道。Content 为内容基类，派生类有 TextContent、CurveContent 和 ImageContent。Track 可包含多个具体的 Content，而具体的 TextContent 包含 TextObj 对象，ImageContent 包含 ImageObj 对象。Well 再包含 Ruler 刻度尺对象，Ruler 派生于 Track，简单而言，整个综合录井图客户端就是由一个树形结构组成，根节点为 Well，分支为多个 Track，Track 再包含具体的内容，对象之间的包含关系采用 ArrayList 保存。

客户端启动时，传入的参数包含数据远程对象以及图像远程对象的 URL，通过 Activator.GetObject

方法，从指定的 URL 来获取远程对象，完成分布式的操作。

对于用户的鼠标事件，系统抽象出了 ICommand 接口，改接口定义了撤销与重做功能即 Redo 和 Undo。对于需要撤销重做的鼠标事件，实现该接口。如 TrackMove 就实现了 Track 道的移动操作，每一步 Track 道的移动，都保存到一个链表中，采用命令设计模式[4]，供撤销与重做，使绘客户端具有良好的恢复重做机制。

利用基于.Net Remoting 的远程技术后，客户端就仅仅只专注于图形的绘制工作，而复杂的业务逻辑及复杂的对象生成都交给远程服务去完成，功能划分更明确，内聚性也更高。

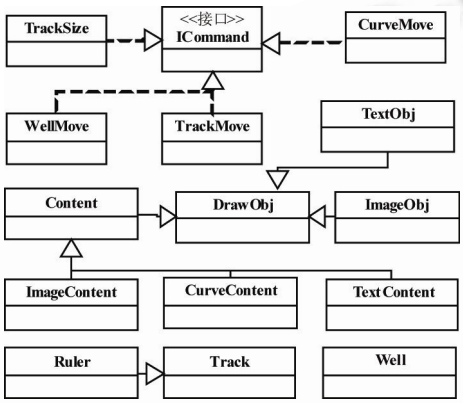

图 4　绘图客户端主要类图

2.6 web 端的设计

本系统 web 端负责系统用户身份登录的验证，提供模糊查询，传递井号以及远程服务 URL 地址和端口给客户端程序，启动客户端绘图。另外，还还发布最新公告消息等。传统的 C/S 架构是直接从本地启动客户端程序，而本系统采用 web 程序启动客户端，能更好的集成系统，增强安全性能，也符合统一系统入口的原则。

从网页启动本地客户端程序，目前可选的方案有两种：一种是通过 ActiveX 控件来启动客户端程序，该方式经常会因浏览器的安全设置等问题而引起不允许下载安装 ActiveX 控件，导致系统不能正常运行；另外一种是通过安装客户端软件时，往注册表中写入自定 URL 协议信息，注册应用程序以处理自定义 URL 协议的方式来启动客户端程序，该方案能更有效启动客户端，不会因为客户端升级或者浏览器的安全设置等引发启动失败的问题，是一种比较理想的解决方案。

3　结束语

本文简要介绍了.Net Remoting 技术，并详细介绍了基于.Net Remoting 技术的分布式综合录井图系统的设计与实现方案。综合录井图基于模板的特性，使特定图形与具体数据项进行分离，当需要绘制特定图形时，只需调用相应的模板，动态加载数据，更加灵活方便。

最后，通过某录井公司的实际录井数据验证和压力测试，以及用户的反馈，采用该.Net Remoting 技术的综合录井图系统，运行稳定，响应及时。证明基于.NetRemoting 的分布式的综合录图系统，能更好利用油田企业现有的企业内部网络与计算机资源，降低用户计算机硬件要求，扩展性和兼容性更好。另外，由于客户端和宿主服务程序是松耦合关系，当需求发生变化时，只需要单独更改服务宿主程序或客户端即可以满足要求，避免了因系统升级等引起的繁琐部署及系统结构调整问题，为系统的完善升级奠定了良好的设计基础。

参考文献

1 金泽兰.地质图编绘法.北京:地质出版社, 1982:34－78.

2 Löwy J.刘如鸿译..Net 组件程序设计.北京:电子工业出版社, 2007:342－370.

3 Larman C.李洋等译.UML 和模式应用.北京:机械工业出版社, 2007:5－8.

4 Cooper JW. 张志华,刘云鹏,等译.C#设计模式.北京:电子工业出版社, 2003:176－185.