

可视化协同导向钻井虚拟环境

高晓荣¹, 李 琪¹, 徐英卓²

(1. 西安石油大学石油工程学院, 西安 710065; 2. 西安石油大学计算机学院, 西安 710065)

摘要: 针对传统导向钻井技术中信息展示方式和工作方式的不足, 基于虚拟现实、CSCW 和 Agent 等技术构建一个可视化协同导向钻井虚拟环境, 为不同地点的钻井技术人员和专家提供协同工作平台, 提高了钻井效率。描述虚拟环境的体系结构及其主要技术, 包括钻井对象三维可视化、服务器与客户端的通信、协作成员的管理和多用户并发控制。

关键词: 导向钻井; 可视化; 虚拟环境

Virtual Environment for Visualization Cooperative Geosteering Drilling

GAO Xiao-rong¹, LI Qi¹, XU Ying-zhuo²

(1. Institute of Petroleum Engineering, Xi'an Shiyou University, Xi'an 710065; 2. Institute of Computer, Xi'an Shiyou University, Xi'an 710065)

【Abstract】 Aiming at the lack of traditional information revelation manner and working manner, this paper proposes a virtual environment for visualization cooperative geosteering drilling constructed by the technology of virtual reality, CSCW and Agent. It provides a cooperation working platform for drilling workers and experts in different place to improve drilling ratio. The architecture and main technology of the virtual environment are described, including 3D visualization of drilling objects, communication between sever and client, management of cooperative partners and concurrent control of multi-user.

【Key words】 geosteering drilling; visualization; virtual environment

导向钻井是于 20 世纪 90 年代发展起来的一项钻井高新技术。在导向钻井过程中, 由于地质条件复杂多变, 井身穿越的地层地质和油藏参数、井身工程和井眼参数等随钻信息通常是不精确或模糊的, 因此要根据上述信息实现实时随钻控制, 必须先进行实时分析、处理、解释与展示, 并由多方专家协同决策, 以达到优质、快速、安全、准确中靶的钻井目的。传统方式是组织有关专家亲临现场, 依靠单维数据和各种解释图件进行决策指导。单维数据和各种解释图件无法直观地显示真实环境下的三维结构细节, 且钻井作业地域辽阔、复杂多变, 组织各方专家亲临现场难度较大。

1 可视化协同虚拟环境原型系统设计

1.1 系统框架设计

考虑到导向钻井的特殊性与复杂性, 通常在油田基地进行钻井决策指导, 而钻井作业则分布在远离油田基地的现场。如图 1 所示, 根据系统功能、用户级别及其所处工作环境, 系统总体框架分为 2 级: (1)井场级。面向施工现场的钻井技术人员。通过井场实时监测与信息处理子系统对井下随钻测量的实时信息(包括地质参数、井眼轨迹参数、钻进参数和钻井评价参数等)以及地面综合录井和泥浆录井等信息进行实时处理后, 以图片和文字的形式展现给现场技术人员。将上述信息与邻井及区块的现场数据进行综合集成, 通过卫星通信、GPRS 等方式实时传输到油田基地, 使现场技术人员实现对现场作业的监控, 并得到后方基地专家对现场作业的实时指导。将邻井及区块的现场信息实时传回基地, 用于对比分析、施工决策。(2)基地级。面向油田基地的领导、技术专家。通过信息可视化与协同决策虚拟系统综合分析远程实时

传输的井场信息, 实现钻井过程动态可视化, 并提供多方协同钻井的虚拟环境, 使分散在不同地点的多方专家和领导通过网络对现场钻井作业进行实时监控、协同决策与指导。

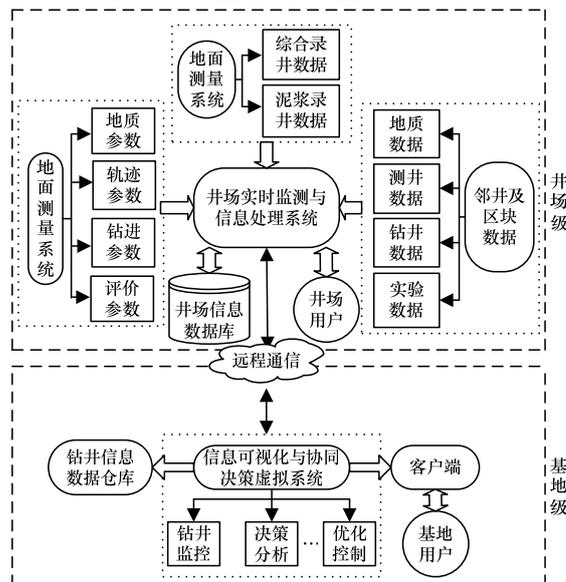


图 1 可视化协同导向钻井虚拟环境原型系统框架

基金项目: 国家科技支撑计划基金资助项目(2007BAB17B04); 西安石油大学中青年创新基金资助项目(Z07070)

作者简介: 高晓荣(1975—), 女, 讲师, 主研方向: 钻井工程, 科学计算可视化; 李 琪, 教授、博士; 徐英卓, 教授

收稿日期: 2008-11-27 E-mail: gxr_0501@126.com

1.2 系统体系结构设计

基于上述系统框架,将系统体系结构设计为4层,即用户层、应用层、控制层和数据层,如图2所示。

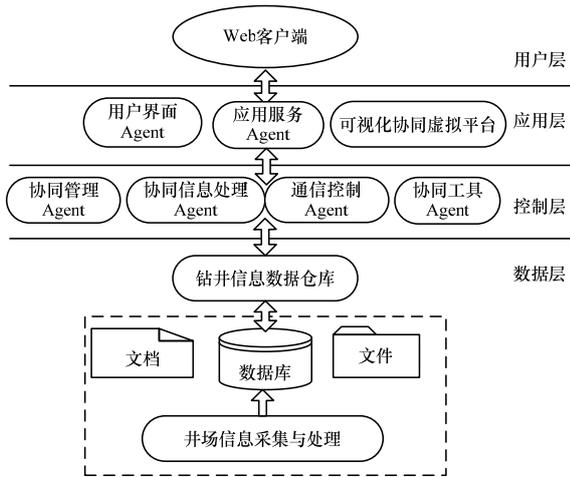


图2 可视化协同导向钻井虚拟环境原型系统体系结构

应用层由Web服务器和应用服务器组成。Web服务器用于管理客户端Web页面。应用服务器承担信息可视化与虚拟环境生成工作,负责启动控制层的协作服务器、接收客户端的参数以及传递可视化结果等通信过程的控制。该层主要包括用户界面Agent、应用服务Agent和可视化协同虚拟平台,它面向导向钻井中各种复杂问题,通过浏览器为用户提供一个可视化协同钻井虚拟环境。

(1)用户界面Agent。负责用户登录系统时身份、账号验证等,以及前端与后台连接的初始化认证,并根据用户登录时选定的身份定制用户界面,提供人机交互接口。

(2)应用服务Agent。提供导向钻井中各种复杂问题(包括钻井设计、钻井监控、钻井事故诊断与处理以及钻后分析等)的服务功能。

(3)可视化协同虚拟平台。是一个虚拟现实系统,实现井下钻井对象和动态作业过程的三维可视化,为钻井人员营造一种身临其境的现场钻井虚拟环境,可以直观地显示地层分层、储层、井眼轨迹、靶区等,并能进行地层解释、钻井轨迹调整等,使基地的多方专家能对现场钻井过程进行可视化分析和随钻控制。

控制层是协作服务器,负责管理与控制整个协同钻井过程,主要包括协作管理Agent、协同信息处理Agent、通信控制Agent和协作工具Agent。

(1)协作管理Agent^[1]。是整个系统的监控程序,实现协作成员的注册与账户管理、成员的动态加入与撤离管理以及成员分组管理。在系统运行中,可以跟踪每个成员的状态,并为客户端提供查询其他成员当前状态的服务,从而实现成员之间的协同感知。

(2)协同信息处理Agent。实现协作任务的信息发布和同步管理,提供共享信息的访问与查询服务。

(3)通信控制Agent。负责对信息路由和服务质量(QoS)的管理和带宽分配、存储转发等。

(4)协作工具Agent。提供协作过程所需的协作工具,主要包括应用程序共享、多媒体交互、电子白板、电子邮件服务等工具。

数据层位于数据库服务器端,通过井场信息采集与处理系统对井场(可能同时有多口井)各种信息进行综合集成,构

成面向导向钻井的钻井信息数据库,为上层应用提供数据资源。

2 系统实现的主要技术

2.1 服务器与客户端的通信

服务器与客户端之间的通信机制较复杂,包括Web服务器与Web浏览器之间最基本的通信,以及Web浏览器、应用服务器和协作服务器之间参数和图形图像的传递机制。后者由应用服务器上开发的一个应用服务管理程序承担,其通信利用Java提供的Socket通信接口来实现^[2]。Socket指定了客户端和应用服务器之间进行双向数据传输的网络通信端点。应用服务器端利用ServerSocket类来建立,具体描述如下:

```
ServerSocket serverSocket=new ServerSocket(80); //绑定至//80 端口
```

应用服务器不主动与客户端建立连接,而是被动地监听客户端的连接请求,并为其提供服务。它能同时接收多个连接请求(见图3),但每次只能处理一个,所有已建立并开始监听的连接按顺序存放监听堆(监听堆是一个无应答的连接请求队列)。accept()方法负责取出堆中的连接,返回一个用来与来访者对话的客户端连接,并建立输入和输出流,其代码如下:

```
Socket clientSocket=serverSocket.accept();
DataInputStream inbound = new DataInputStream(clientSocket.
getInputStream());
DataOutputStream outbound = new DataOutputStream(client
Socket.getOutputStream());
```

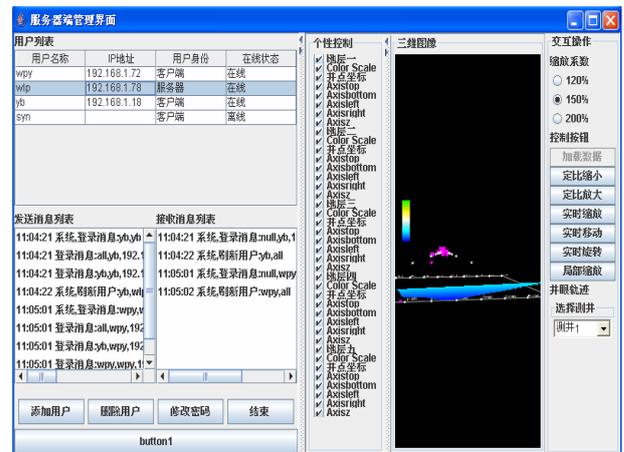


图3 多用户请求连接时的服务器端界面

一般的I/O操作可以在新建的流中运用,在应用服务器回应前,它等待客户端发送一个空白行,当会话结束时服务器关闭流和客户端clientSocket。若队列中没有请求,则accept()方法将阻塞应用服务器线程直到请求到来。当所有监听堆的连接处理完毕时,应用服务器退出。

应用服务器上设计了3个线程^[3]:RequestT, ListenT 和 Exec,用于实现多任务的实时操作。当ListenT进程监听到客户端有一个用户请求时,启动一个RequestT线程与其对应,启动一个Exec线程来启动信息可视化及钻井服务程序,并连接协作服务器对其进行管理与控制。RequestT和Exec是Socket通道中一组唯一的流连接。一个Socket通道中有多组流,它们互不相干,各自传送各自用户的请求和结果,即每个用户有一套自己的RequestT, Exec和一组DataInputStream/DataOutputStream来完成对用户要求的一次处理。

(1)RequestT线程。对客户端,它发出操作请求,并发送

传递参数。一个客户端的 Applet 可以启动多个 RquestT 线程进行查询,以解决对 Socket 端口的共享,从而在客户端实现多任务的实时查询。对应用服务器端,该线程负责接收用户请求和通过 Socket 传来的参数。

(2)ListenT 线程。用于监听 Socket 端口的用户请求,连接和启动信息可视化及钻井服务程序以及协调服务器,启动后一直运行并制止管理员关闭服务,监听到用户请求时,启动一个 Exec 线程。

(3)Exec 线程。用于启动信息可视化及钻井服务程序、协作服务器的连接和参数传递等操作。它由 ListenT 启动,当 ListenT 监听到一个用户请求时,创建一个 Socket 对象,把它作为一个参数传给 Exec 并启动。Exec 将参数传入信息可视化及钻井服务程序中进行处理,并返回结果给 Socket 输出流,以传回客户端。ListenT 可以启动多个 Exec 线程,以实现可视化信息的操作共享。

2.2 钻井对象的三维可视化

钻井对象(包括地层、井眼轨迹、靶点、设计轨道、实钻轨迹等)的三维可视化采用可视化工具 Java 3D 来实现,其虚拟世界根据场景图来创建。场景图^[2]是一种 DAG(Directed Acyclic Graph)图,包含节点和数据对象,其中,节点分为组节点和叶节点。组节点用于控制其下的子节点,叶节点为末端节点,包含构成场景图的核心语义元素,如物体几何形状。数据对象包含叶节点所需的数据信息,叶节点通过对数据对象的引用提供其语义所要表达的数据信息,它与数据对象配合可构造三维对象。根据所包含信息的不同,场景图分为内容分支和视分支。图中每个节点均为对象,节点间为父子或引用关系。场景图定义了几何体、灯光、位置、方向、外观等可见对象,由 Java 3D 类的实例来创建。其实现过程如下:

- (1)创建一个用于绘制三维图形区域的 Canvas3D 对象,该对象是一个包含场景中物体视图的矩形。
- (2)建立 Virtual Universe 对象,用于容纳所建场景。
- (3)创建一个用于放置一组物体的数据结构场景 Locale 对象,并将 Locale 对象关联到 VirtualUniverse 对象。一个 Local 对象是一个局部坐标系,它提供了虚拟世界中一个点的引用,决定了可视物体在虚拟世界中方位的路标。
- (4)创建 View 子图,构造虚拟场景的视平台,以观察整个虚拟环境。观察动态场景时,由于视平台有着复杂的运动状态,因此需要视分支能在运行过程中根据实时获得的控制数据进行运动,本文将转移节点置于视平台上,结合运动节点和 Alpha 生成视图的运动来实现。创建 View 子图时,先创建一个 ViewPlatform, PhysicalBody 和 PhysicalEnvironment 对象,再创建一个 View 对象,并将 ViewPlatform, PhysicalBody, Physicalvironment 和 Canvas3D 对象关联到 View 对象上。
- (5)构造内容子图并编译子图。
- (6)将上述 2 个子图插入到 Locale 节点。

导向钻井场景图的构建是将井下各种虚拟对象在不同对象坐标系中生成的各节点作为完整场景图的不同组成部分,并按钻井过程中各对象的空间关系、结构关系和设计的参数,将上述节点集成到世界坐标系中,生成最终所要求的场景。图 4 给出了钻井三维场景结构,主要包括 3 个部分:三维钻井对象的外观(灯光、材质、纹理等),三维钻井对象形体和观察者视点。图 5 是生成的单井井眼轨迹及其穿越地层的三维可视化视图。根据钻井参数及其解释结果实现近井带地层的三维可视化,并将井眼轨迹嵌入地质结构的三维场景中,有助于直观显示井眼穿越的地层是何种类型。

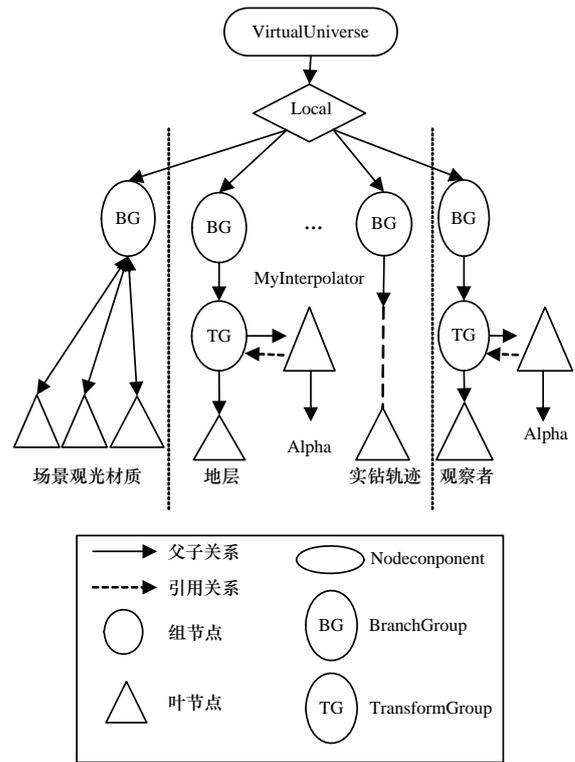


图 4 钻井三维场景结构

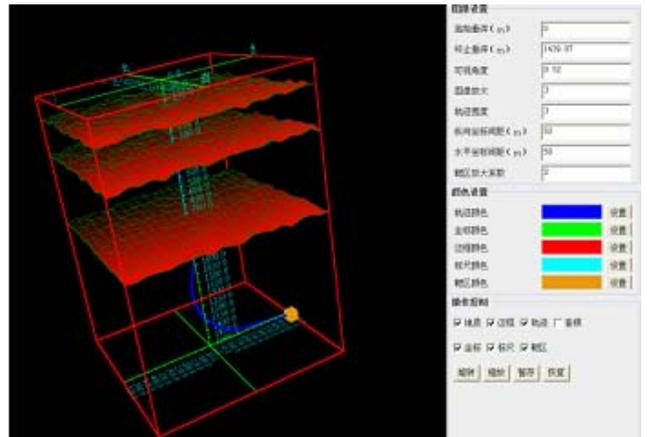


图 5 地层与井眼轨迹三维可视化视图

2.3 协作成员的管理

由于参与协同钻井的人员来自不同部门或学科,协作时系统可根据需要动态地将多个成员组成一个工作组。同一个成员在不同工作组内可以担任不同角色,因此,进行成员管理时,必须考虑账号、工作组、角色和权限等。为保证系统安全性,本文系统采用基于成员-角色的授权管理^[4]方式,不同角色能发挥的作用通过相应权限来体现。权限被定义为一个有序对 (O, F) ,其中, O 表示系统的一个对象; F 是对 O 访问方式的集合。角色是已命名的权限集合,它是一个有序对 $(name, pset)$,其中, $name$ 为角色名; $pset$ 是权限集合。本系统有如下 5 种对象与权限:

- (1)当 O =应用服务时, F ={任务安排, 任务监控, 执行操作};
- (2)当 O =数据库时, F ={创建, 读, 插入, 删除, 修改};
- (3)当 O =协作工具时, F ={允许使用, 不能使用};
- (4)当 O =协同工作小组时, F ={总工程师, 工程师, 技术员};
- (5)当 O =虚拟对象时, F ={控制, 操作, 接收}。

(下转第 251 页)