

多用户协同虚拟现实的技能训练系统^①

许爱军

(广州铁路职业技术学院, 广州 510430)

摘要: 针对传统 CBT(计算机支持的教育培训)系统用户单一、沉浸感不强、不支持协同学习等问题, 提出协同虚拟环境支持下技能训练系统的实现方案. 采用分层结构思想, 设计系统的框架结构和工作流程; 提出基于模型复杂度的快速建模策略, 构建协同虚拟训练场景; 利用并发服务器通信技术, 构建多用户协同工作环境; 采用定时采集和发送数据的方法, 降低协同感知过程中服务器的负担; 结合矢量化电子图板和文字消息传递机制, 满足教师实时评注和多用户讨论的需求. 结合应用实例, 验证了本方案的可行性和相比传统 CBT 的优越性.

关键词: 协同学习; 智慧环境; 技能训练

System of Multi-User Collaborative Virtual Reality Assist in Skills Training

XU Ai-Jun

(Guangzhou Institute of Railway Technology, Guangzhou 510430, China)

Abstract: This paper puts forward the realization of collaborative virtual environment assist in skills training system against on the CBT (computer based training) systems, which contains the problems of single-user, lack of immersion and no collaborative learning is assisted. The system designs system framework and workflows base on stratified structure idea. It proposes a rapid-modeling strategy based on model complexity to build collaborative virtual training scenarios. It uses multi-communication server technology to build the multi-user collaborative work environment. It reduces the burden of server in collaborative perception process by timing acquisition and sending data methods and meets the needs of teachers' real-time commentary and multi-user discussions by vector electronic drawing board and text messaging mechanism. It verifies the feasibility and superiority between the new system and traditional CBT by application examples.

Key words: collaborative virtual environment; collaborative learning; intelligent environment; skill training

我国是劳动力资源最丰富的国家, 但高技能人才短缺和教育培训经费投入不足, 一直是制约我国企业持续发展和阻碍产业转型升级的两大瓶颈. 大力发展职业技能培训, 降低培训成本、提高培训效益, 已成为我国当前和今后一段时间面临的紧迫任务. 但是, 当前职业技能培训受培训时间、场地、设备、经费和安全性等因素制约, 通常难以构建真实学习情景, 导致培训质量良莠不齐.

协同虚拟环境(Collaborative virtual environment, CVE)产生并应用到教育领域后, 催生出新型的智慧学

习环境. 这种智慧学习环境将单用户虚拟环境(Virtual Environment, VE)扩展成多用户虚拟空间, 支持多用户之间协同与交互, 让学员在身临其境的虚拟环境中协作学习, 共同完成学习任务, 逐渐掌握操作技能. 与传统 CBT(计算机支持的教育培训, Computer Based Training)系统相比, 协同虚拟环境支持下的技能训练系统, 能打破时空限制、支持协作学习、安全可靠, 能大大降低培训成本, 可以看作是当前技能培训与职业教育教学方式变革急需的支撑条件.

利用虚拟现实技术开展技能训练已有较长时间,

① 基金项目:2014 年广州市哲学社会科学“十二五”规划课题(14G56);广东省教育科学“十二五”规划 2014 年度教育信息技术研究项目(14JXN018);广州市教育科学 2013 年度课题(2013A238)

收稿时间:2015-05-16;收到修改稿时间:2015-06-11

出现一些典型案例。如：荷兰 VSTEP 公司开发了危险化学品救援 Rescuesim 消防演习系统；美国多伦公司、加拿大 CGSD 公司、西班牙 GSC 公司等开发了虚拟汽车驾驶模拟器；瑞典斯德哥尔摩手术科学部研制了虚拟腹腔镜手术模拟训练系统等。在国内，华东师范大学开发了上海航空公司火灾虚拟应急处理训练系统、天津大学开发了盐业职工虚拟技能培训与考核系统、广东工业大学开发了车削加工技能虚拟培训系统等。这些案例大多局限于单用户使用或多用户异步操作，缺乏对多用户协同工作的支持，尤其是对协同环境中的权限控制与协同讨论还鲜有涉及。为此，本文设计并实现了协同虚拟环境支持下的技能训练系统，以下将对其结构、原理和实现技术等详细介绍。

1 协同虚拟环境下技能训练需求

在协同虚拟环境中，用户替身(avatar)是不可缺少的元素。共同处于同一个虚拟空间的师生，通过用户替身来标识，替身的状态位置和方向标识了该用户在场景中的视点(Viewpoint)^[1]，在计算机支持的协同工作(CSCW)机制下，实现你见即我见(WYSIWIS)。为此，协同虚拟环境支持下的技能训练应满足如下需求：

(1)能逼真地模拟技能训练场景。可以逼真地模拟技能训练的外部环境(如：实训场所的结构、大小、环境、布局等)和实训设备的三维构造(如：设备的结构、组成、材质等)；

(2)能感知其它学员的存在和行为。可以感知其它学员的工作状态(如：正在完成哪项任务、进展到什么程度等)，并能通过观察代表用户标识的替身行为，获知其它学员的当前行为。教师能在全景终端上观察总体情况；

(3)能与其它学员进行交互和沟通。可以利用输入设备与环境或其它学员进行交互，能提供便捷的交流手段(如：电子白板、文字消息传递)实现师生之间、学员与学员之间的相互沟通。教师还应能对学员的训练情况进行批注和评阅。

2 技能训练系统总体设计

2.1 系统整体框架设计

基于以上对技能训练需求分析，采用分层结构思想构建了系统的体系结构(如图1所示)。其中，学生端主要显示技能训练场景和用户替身，支持学生开展技

能训练、利用电子白板与师生沟通等。交互层提供用户身份和替身管理、训练空间一致性维护、并发控制和协同讨论等。数据层包含各种应用数据库和 VRML 模型。分层结构使系统具有良好的扩展性、灵活性和可维护性^[2]。

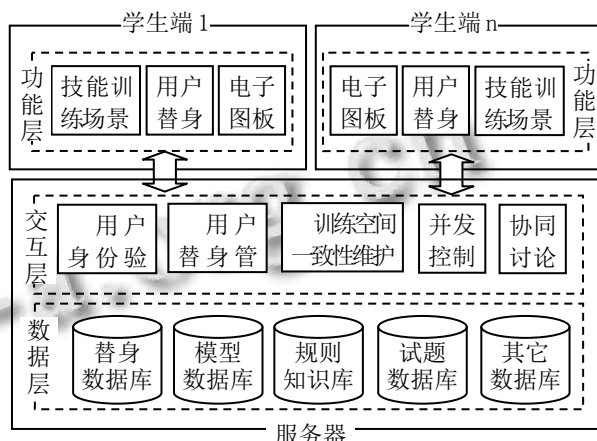


图1 技能训练系统框架结构

2.2 系统工作原理与流程

系统采用 C/S(Client/Server)架构，分为学生端和服务端两部分。其中，服务器端由应用服务器和数据服务器组成。应用服务器是实现协同支撑的枢纽，负责数据通信、同步控制和消息传递。服务器端利用 Java Socket 机制与学生端连接，监听和响应学生请求。通信协议采用 TCP/UDP 相结合的方式，学生登录信息采用 TCP 传输，学生之间的即时通信采用 UDP 进行发送和接收。这种方式能保证通信质量的同时实现通信的实时性。系统工作原理如图2所示。

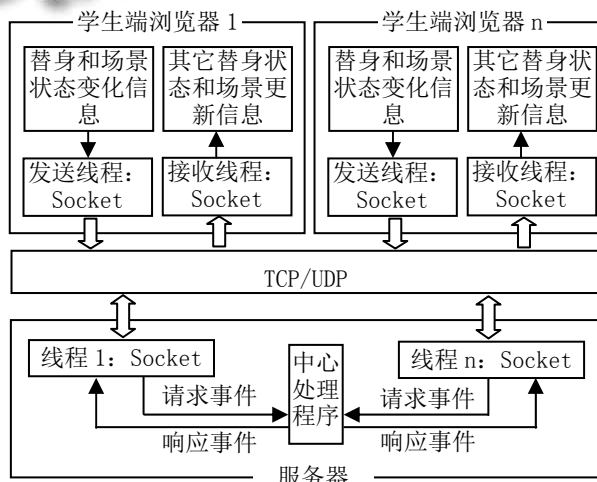


图2 技能训练系统工作原理

在协同虚拟环境中, 学生端的操作和向服务器端发出请求都是随机的, 这样就可能会出现多个请求同时到达服务器的情况. 为解决这个问题, 服务器端采用并发服务器(Concurrent Server)工作方案, 即随系统启动产生一个守护进程, 在无学生请求时处于等待状态, 当学生请求到达后则立即为该学生产生一个子进程, 由子进程响应和处理学生请求, 守护进程回到等待状态. 这个方案中, 并发服务器是主服务器(master), 子进程为从服务器(slave), 一个进程的阻塞不影响其他进程的运行, 运行可靠性有所保证.

当学生端需要发送场景和替身的状态变化时, 首先生成一个 StudentPost 线程, 由服务器的守护进程调用, 将学生端的信息发送到服务器. 当有其它学生端更新信息达到时, 则生成一个 StudentReceive 线程, 由服务器的守护进程调用, 将其它学生端的更新信息接收下来, 更新学生端的状态场景. StudentPost 类的描述方法为:

```
Public class StudentPost extend Script { //从 Script 派生
    Variable Define(...); //变量初始化
    Initialize(...); //类初始化方法
    ProcessEvent(Event e); //场景事件响应方法
    ControlAvatar(...); //控制替身方法
    UpdateAvatarStatus(); //学生替身状态更新方法
    UpdateSeceneEntity(); //场景中实体状态更新方法
}
```

系统的工作流程(如图 3 所示)为:

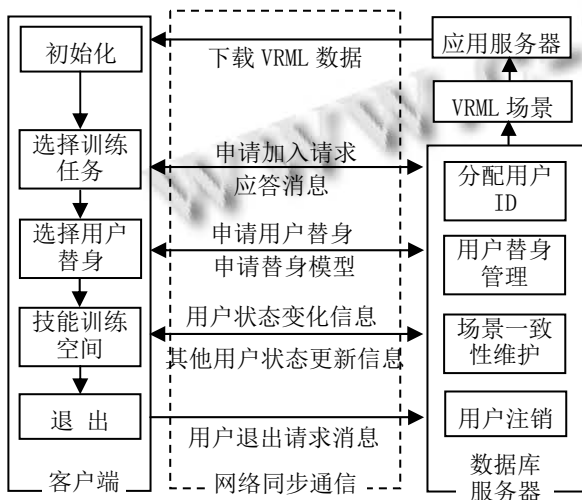


图 3 技能训练系统工作流程

(1) 学生用兼容 VRML 的浏览器登陆系统前, 先进行初始化工作(执行 Script 类的 initialize 方法), 与服务器端建立连接. 应用服务器提供登陆界面.

(2) 学生输入用户名和密码后, 根据训练内容选择对应的训练任务, 选择结果作为服务发送到应用服务器.

(3) 应用服务器收到请求后, 与数据库服务器交互, 数据库服务器为学生分配 ID, 并把 VRML 场景文件发送到应用服务器, 应用服务器生成模型描述图后, 再发送给学生端显示, 学生端生成操作线程.

(4) 学生选择用户替身后, 在应用服务器的协同控制下开展技能训练. 训练任务完成后, 学生退出, 服务器调用 shutdown 方法删除监听服务器和学生端的操作线程^[3].

3 协同虚拟环境的关键支撑技术

3.1 虚拟训练场景的建模方法与策略

技能训练系统中存在许多三维虚拟模型, 模型的复杂度和精细度决定了虚拟场景的逼真度. 建模的基本流程为: 数据获取与处理(实训场地数据、设备尺寸数据等)、专业三维软件建模(如: 用 Pro/E、3DMax 建模)、可视化编辑软件修正(如: 用 Cosmo Wrolds 修正误差、添加材质和交互)、文本编辑软件完善(如: 用 Vrmlpad 优化结构、减少代码、压缩文件)等阶段.

为提高建模效率, 可以采用基于模型复杂度的快速建模策略^[4], 基本思想是在建模前, 先分析模型在虚拟场景中的作用, 来决定采用何种建模方式. 若模型为形状不规则、结构复杂, 在虚拟场景中处于主要地位的模型(如: 实训设备、组装零配件等), 则利用专业三维软件建模, 采用一般的建模开发流程; 若模型为形状规则、结构简单, 在虚拟场景中处于次要作用的模型(如: 实训室内部空间、墙壁装饰等), 则采用 VRML 直接建模, 通过 Appearance 节点的 material 域调用材质或 Texture 域引用贴图来实现.

为提高网络传输效率, 对建模生成的 wrl 文件进行优化和压缩是非常必要的^[5]. 常用的方法有: 用文本编辑工具(如: Vrmlpad)进行压缩、用 DEF/USE, PORTO 和 EXTERNPROTO 语句引用相同类型的实例模型、优化数据(数据取整或固定精度)、用专门压缩软件(如: wingz, Gzip)压缩等. 压缩后的 wrl 文件更小, 更适合在网络上传输.

3.2 协同工作原理与实现技术

3.2.1 两种主要的交互方式

VRML 作为虚拟环境的三维描述格式规范,本身不具备网络通信能力,对协同虚拟环境缺乏有效的支持(如不支持协同虚拟环境所必需的共享行为和环境状态的连续性),但它提供了内部 Script 节点(Script Authoring Interface, SAI)和外部编程接口(External Authoring Interface,EAI),这2种方式可以与Java结合起来实现网络通信和分布式控制。

利用SAI的基本方法是,在VRML的每个节点代码中设计一个Java脚本程序,通过路径(Route)将传感器事件的输出,作为Java脚本程序的事件输入,将需要实现的网络通信和分布式控制功能,交给Java程序来处理。SAI为扩展节点行为状态提供了强大的灵活性,但对于与外界的信息交流(包括信息的输入和输出)却显得力不从心。利用EAI的基本方法是,将Java Applet与VRML场景嵌入同一个页面中,利用Java Applet与VRML浏览器插件互相通信,来实现外部对虚拟场景节点的控制。一般而言,大部分交互功能都可通过这两种交互方式实现,但在使用过程中,较为单一的VRML交互通常采用SAI,而复杂环境的网络通讯则使用EAI来实现。

3.2.2 协同感知的实现

系统的协同感知是通过VRML的临近检测器节点(ProximitySensor)和视点节点(Viewpoint)来实现的。利用ProximitySensor节点,可以感知当前学生对虚拟物体(VRML模型)的操作并输出位置和方向的变化值。利用Viewpoint节点,可以根据接收到的其它学生操作信息调整虚拟物体(VRML模型)的空间位置和方向。

当学生对虚拟物体进行操作时,ProximitySensor节点检测到虚拟物体空间方位的变化,position_changed域和orientation_changed域分别获取虚拟物体的位置和方向值,通过SAI接口启动输出线程StudentPost,向服务器发送信息。服务器的守护进程生成子进程后,调用其它学生端的接收线程StudentReceive,更新其它学生端的Viewpoint节点,从而实现了虚拟物体的协同感知。

在协同虚拟环境中,学生利用鼠标连续不断地操纵虚拟物体,如果对鼠标事件实时采集并发送,容易造成服务器负担过重甚至广播风暴。为了解决这个问题,系统采用定时采集和发送数据的方法,每隔一个时间

间隔再获取虚拟物体的方位值,并启动发送机制。时间间隔的设置采用VRML中节点的时间戳。当Script节点收到EventIn后,从Event对象的Name、Value和Timestamp域中,用getStamp()方法得到时间戳。这个方法大大降低了数据的传输频率,并能较好地保证虚拟物体协同感知的连续性。

3.2.3 协同并行控制方法

学生在虚拟环境中的替身方位和行为是通过Script节点控制TimeSensor传感器,进而控制OrientationInterpolator和PositionInterpolator传感器实现的。当多个学生同时操作某个虚拟物体时,可能产生操作冲突。系统采用对象锁的方法,当学生需要操作某个虚拟物体时,先向服务器发送一个操作请求,服务器收到请求后检查该物体是否已锁。如果未锁,服务器向该学生端发出“可操作”的响应,同时向其他学生端发送“该物体已锁”的消息。学生端只有完成对一个物体的解锁后,才能申请对另一个物体的操作请求。

若某个学生获得虚拟物体的操作权后,由于网络断开或发生异常,其他学生将长时间不能获得该物体的解锁信息而处于等待状态,系统可能出现死锁。为防止死锁的发生,系统采取异常处理机制^[6]:服务器端每隔一个时间间隔获取虚拟物体的方位值时,先检查该物体与授权学生的链接状态,若检测到该物体失去链接达到预定的时间,服务器负责将该物体收回,向所有学生端发送该物体“可操作”的消息,并按照先来先服务的原则,响应最先申请操作权限的学生请求。物体的“锁定”与“解锁”,是服务器对VRML场景节点的(ExposedField SFBBool)touchsensorenabled域值的控制来实现的。

3.3 协同讨论环境的实现

在协同虚拟环境中,协同工作机制能很好地实现虚拟物体的协同感知,但对协作过程中的探讨和交流仍有不足,需要建立便捷的讨论环境。

协同讨论涉及到的鼠标消息包括标记、圈阅、文字和交流文本消息传递等服务^[7],每种服务都是一个多线程的Java ServerSocket。系统采用矢量化电子图板和文本框控件实现学生之间、师生之间的协同讨论。

3.3.1 交流文本消息传递

学生之间的相互交流是通过类似聊天室的文本框控件实现的。当输入交流信息后,点击“发送”按钮时,

学生端生成 StudentPost 线程, 构建包含学生 ID 信息的请求数据包, 向服务器启动消息传递机制, 并把这个消息同步到其他学生端. 由于数据包中含有学生的 ID 信息, 所以学生端都能显示发送信息的学生姓名. 当然, 学生也可以选择只与某个或某部分学生进行交流, 服务器则只向这些特定的学生端发送消息. 将消息发送给服务器的关键代码如下:

```
Public Boolean handleEvent(Event event)
{...
String b=m_textfield.getTex();,
```

```
if((event.target==m_textfield)&&(event.id==Event.ACTION_EVENT))
```

```
{ m_textfield.setText("");
try{
m_out.writeUTF(b); } //将消息发送给 ChatServer
catch(IOException e) {e.printStackTrace(); }
Return true;
}
```

其它客户端监听服务线程, 将收到的消息显示到聊天窗口中, 关键代码如下:

```
Public void run() {
try {
while(true) //监听对应服务线程发来的消息
{String s=m_in.readUTF();//读 UTF 格式字符串
if(s!=null) //将消息显示在信息显示窗口中
m_textarea.append(s+"\n");
}
catch(Exception e)
{m_textarea.append("出错提示.\n");
m_textfield.setVisible(false);}
}
```

3.3.2 教师批注与评阅

对学生训练情况进行批注和评阅是教师端需要实现的关键功能. 系统在矢量化电子白板上设置了标记、圈阅、文字说明等图标, 对每种图标的操作都设置一个类来响应教师的鼠标消息. 当选择窗体上的图标后, 教师端检测鼠标的 MouseUp 事件, 生成 TeacherPost 线程, 等待鼠标 MouseOut 事件启动, VRML 控制器进行截图, 将三维模型转化为二维图像存入矢量化电子图板中, 同时构建包含图元参数的数

据包, 向服务器启动消息传递机制, 把这个消息同步到其他学生端. 学生端收到数据包后, 将电子图板同步区的图形重绘, 从而实现了教师在线批注和评阅. 由于校园网环境稳定, 教师端产生的批注和评阅图片都上传到服务器端保存.

4 应用实例及分析

4.1 应用实例描述

以 VRML 可视化编辑器 Cosmo worlds 为开发工具, 用 Pro/E 和 3Dmax 建立虚拟训练环境和设备模型, 以 Cortvrml VRML 浏览器插件为图形引擎, 结合 Java 编程语言, 实现了电脑组装技能训练系统. 如图 4 所示.



图 4 电脑组装技能训练系统学生端界面

上方是 Cortvrml VRML 浏览器插件显示的学生工作空间. 学生登录系统后, 选择训练任务和用户替身. 当学生需要安装电脑配件(如: CPU、内存条、键盘接头)时, 只需要点击相应的配件, 用户替身自动将其抓取. 此时, 电脑配件的位姿与手的位姿进行绑定, 电脑配件随着虚拟手的位姿变换而移动, 位姿采集的时间戳设为 250 毫秒, 能较好保证位姿移动的连续性. 学生只有释放这个虚拟配件后, 才能申请其他虚拟配件的操作权限. 当配件移动到安装位置附近时, 系统出现精确装配位置提示, 学生单击对应的位置后, 配件安装成功, 此信息发送到其他学生端. 当系统检测到所有配件都成功安装后, 协同工作任务完成. 教师端可以观察整个场景的安装进程.

下方左侧是在线用户列表, 显示登录学生的姓名. 下方右侧是协作者之间交流和讨论的文本消息传递区.

当协同工作完成后,教师可以对学生训练情况进行批注和评阅,如图5所示。

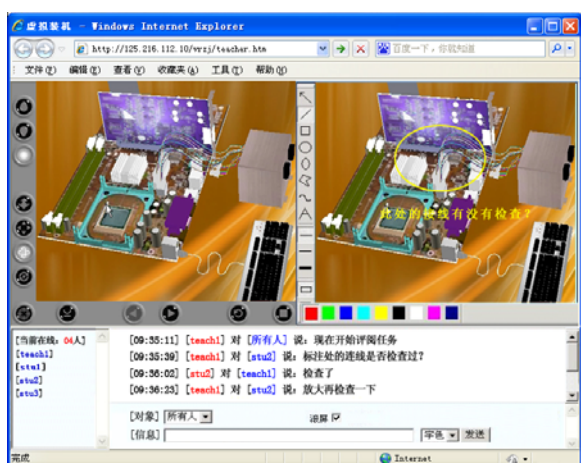


图5 教师批注与评阅界面

4.2 与传统 CBT 的对比分析

与传统的 CBT 相比,协同虚拟环境支持下的电脑组装技能训练系统,在参与人数、沉浸程度、协同感知、教学手段、师生关系、开放程度等方面,具有诸多优势,见表1所示。

表1 协同虚拟现实的技能训练系统与传统 CBT 对比

类型	参与人数	沉浸程度	协同感知	教学手段	师生关系	开放程度
多用户协同虚拟现实的技能训练系统	多用户	强	支持	多种体验式互动教学	平等多向的交往关系,主动学习	开放,可扩展
传统 CBT	单用户	一般	不支持	传统教学手段	不平等单向教学关系,被动学习	相对封闭

由此可知,协同虚拟环境支持下的技能训练系统,能充分调动学生学习的积极性和主动性。在教学应用中,

学生可在教师的指导下进入系统进行自主学习和交流讨论。在任务完成后,教师总结和提高,从而实现分布式学习和群体学习。

5 结语

协同虚拟环境对推进教学改革、降低教育投入、提高训练效率具有重要意义,已成为支撑智慧学习的关键技术,近年来得到广泛关注。协同虚拟环境支持下的技能训练系统,能弥补传统 CBT 用户单一、沉浸感不强、不支持协同学习等不足^[8],在教育培训领域具有广阔的发展前景。为了满足大规模、多用户培训的需求,还需要在协同业务规则、替身行为机制、多用户虚拟场景共享的实时性等方面进一步研究,这也是下一阶段研究的重点。

参考文献

- 1 许爱军,张文金,黄正午.支持协同工作的 VRML 网络虚拟现实系统.计算机仿真,2009,(11):287-290.
- 2 行开新,田凌.支持异地协同设计的异构 CAD 虚拟装配系统.清华大学学报(自然科学版),2009,(2):226-231.
- 3 沈江玮,陈天滋.基于多用户环境下的 VRML 系统框架.计算机工程与设计,2005,26(10):2654-2656.
- 4 许爱军,张文金,易丹.基于 VRML 的虚拟现实技术及应用.计算机与数字工程,2009,(4):186-189.
- 5 李文航,龚建华,周洁萍.协同虚拟地理教学环境的设计与实现.武汉大学学报信息科学版,2008,(3):289-292.
- 6 陈家新,米晓红.协同环境中 3D 替身交互行为动态控制的实现.工程图学学报,2005,(2):50-54.
- 7 许爱军,张文金,易丹.基于虚拟现实技术的远程教育平台研究与实现.计算机系统应用,2007,16(8): 23-26.
- 8 蔡新,钱楚.基于虚拟现实的协同训练系统研究与实现.现代教育技术,2010,(4):120-122.