

基于 Aglet 平台的网上考试系统的实现

王丽侠

(浙江师范大学行知学院, 浙江 金华 321004)

摘 要:针对目前网上考试系统在效率及安全性上的不足,提出了基于 Aglet 的网上考试系统的实现方法。该方法利用移动 Agent 代码移动的特点,考前将考生 Agent 移动到考生机上,考完后,自动消除,从而提高了网上考试系统的安全性、效率和易用性。

关键词:移动 Agent; Aglet; 网上考试

中图分类号: TP311; G434

文献标识码: A

文章编号: 1673-629X(2006)06-0051-03

Realization of Network Test System Based on Aglet

WANG Li-xia

(Xingzhi College of Zhejiang Normal University, Jinhua 321004, China)

Abstract: To solve the problem of current network test system in the aspect of efficiency and security. The paper puts forward a realization method of network test system based on Aglet. The method takes advantage of code mobility of mobile agent. Test agents are dispatched to student computers before test, and are disposed after test. So safety, efficiency and availability of the network test system are improved.

Key words: mobile Agent; Aglet; network test

0 引言

网上考试系统以其方便、快捷、安全、公正等优势,得到了广泛的应用,其实现多采用 B/S 模式或传统的 C/S 模式,两种模式各有特点,并且都有许多成功的应用实例^[1,2]。相对而言, B/S 模式的考试系统易于维护和升级,用户操作简便易学,但交互性差、安全性不好、数据处理能力弱,难以实现客户端的考试控制。相反传统的 C/S 模式的考试系统,数据处理能力、安全性和交互性都较强,但客户端安装和升级工作量大,可扩展性差。针对这些问题,文中将移动 Agent 技术应用到网上考试中,借助移动 Agent 代码移动和分布计算能力,使系统既具有较强的交互性、安全性,又具有较强的自动发布能力和扩展性。

Agent 是指模拟人类行为与关系、具有一定智能并能够自主运行和提供相应服务的程序。与现在流行的软件实体(如对象、构件)相比, Agent 的粒度更大,智能化程度更高^[3]。网络技术的发展,可以让 Agent 在网络中移动并执行,这就是移动 Agent(Mobile Agent)的思想。移动 Agent 具有很多优点,主要表现在^[4~8]:

① Agent 技术通过将 Agent 动态地移到服务器端执行,使得 Agent 较少依赖网络传输这一中间环节而直接面

对要访问的服务器资源,从而避免了大量数据的网络传送,降低了系统对网络带宽的依赖,提高了系统效率。

② 移动 Agent 不需要统一的调度,由用户创建的 Agent 可以异步地在不同节点上运行,待任务完成后再将结果传送给用户。

③ 为了完成某项任务,用户可以创建多个 Agent,同时在一个或者若干个节点上运行,形成并行求解的能力。

④ 移动 Agent 具有自治性和智能路由等特性。

总之,移动 Agent 适合于解决传统方法中要么代价过于昂贵,要么解决不了的问题。

1 基于 Aglet 网上考试系统的设计

系统结构见图 1。系统创建了两类 Agent:一类是主控 Agent,主控 Agent 运行在中心服务器上,主要功能为:

(1) 考试管理:为用户提供考试有关参数设置的界面,并根据用户的设置情况,以及考试主机 IP 信息创建、派送、消除考生 Agent,以及与考生 Agent 进行通信以获取考生考试成绩。

(2) 试题管理:提供增加、删除、修改试题和试卷信息等功能。

(3) 成绩管理:对学生成绩进行管理和统计。

(4) 考生管理:完成对考生信息的维护,包括对 IP 表的维护。

另一类是考生 Agent。考生 Agent 由主控 Agent 在考

收稿日期:2005-10-17

基金项目:浙江省教育厅科研项目资助(20030717)

作者简介:王丽侠(1970-),女,黑龙江兰西人,副教授,研究方向为网络和数据挖掘。

试时创建,并携带试卷信息移动到考生机,为考生提供考试界面,可以让考生登陆、答题,以及对考试的控制,在考生提交试卷后,批阅试卷,得到考生成绩,并以消息(message)的形式发送给主控端 Agent,得到主控 Agent 的 dispose 消息后自行 dispose。

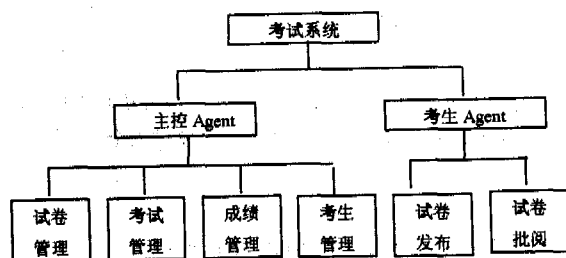


图 1 基于 Aglet 网上考试系统模型

系统的整个工作流程见图 2。首先教师利用主控 Agent,设置考试相关的参数,主控 Agent 利用相关信息(考试编号、IP、考生编号、校验码)建立考生 Agent,考生 Agent 利用考试编码查询后台数据库,获得考试的试题信息,建立试卷,根据 IP 表中的考生信息,派送考生 Agent。考生 Agent 移动到考生端后,显示登陆界面,考生登陆后,考生 Agent 进行身份认证,当校验成功后显示试卷内容,并开始计时。答卷完成或时间到后,考生 Agent 自动结束考试,进行试卷批阅,并通过 AgletProxy 接口与主控 Agent 通信,将成绩以消息(message)形式发送给主控 Agent,主控 Agent 接到成绩后,进行成绩的入库,若要结束考试,向考生 Agent 发送 dispose 消息,考生 Agent 接收到主控 Agent 的 dispose,自行 dispose。

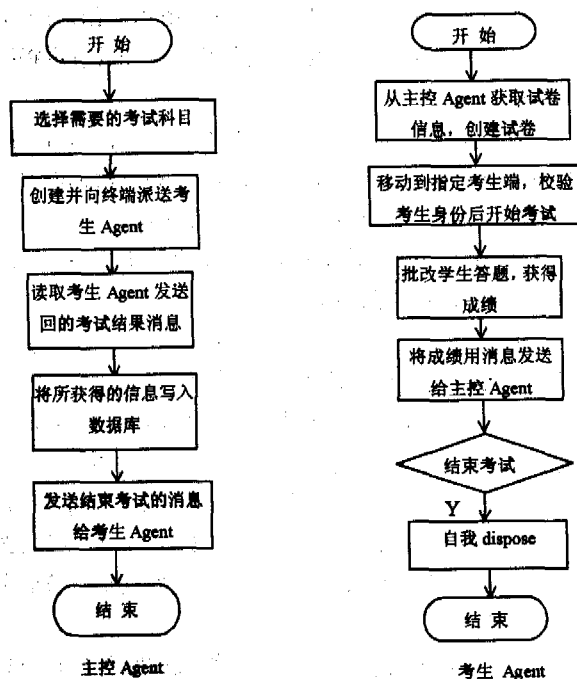


图 2 基于 Aglet 网上考试系统工作流程图

2 网上考试系统实现

2.1 Aglet 程序基本框架和方法

在 Aglet 环境下实现考试系统,首先要理解 Aglet 程

序的框架结构和 Aglet 的运行机制,特别是实现 Aglet 的创建(create)、复制(clone)、派送(dispatch)、召回(retract)、暂停(deactive)、唤醒(active)以及消除(dispose)的方法。

Aglet 程序的框架结构如下:

```
public class tab extends Aglet{
doJob(); //移动到目的端的处理函数
public boolean handleMessage(Message msg){
doGetMsg(); //相应各种消息的处理函数
public void onCreate(Object args){
doOnCreation(); //创建时的处理函数
public void run(){
doRun(); //程序的入口函数}
```

应用程序的主类继承 Aglets-2.0.1 包中的 Aglet 类,doJob()是 Agent 在移动到目的地时要执行的事件处理函数,handleMessage()是 Agent 收到各种消息时的处理函数,onCreation 方法是在 Agent 创建时所要处理的事件,run()是程序的入口函数。

Agent 常用的法有:

1) Agent 创建方法。

格式: getAgletContext().createAglet(getCodeBase(), "AgletName", args)。

说明: AgletName 表示希望创建新的 Agent 的名字, args 是一个 Object 变量,用来存放主控 Agent 与新建立 Agent 之间传递的消息,新建的 Agent 可以在 onCreate 函数中获得此信息。

2) Agent 分派方法。

格式: dispatch(url)。

说明: url 是 Agent 所要前往的目的地的 URL。它应该是符合如下所示: atp://aspirin:5002 的字符串, "atp" 表示其所用的网络协议 Aglet Transport Protocol, "5002" 表示其端口。

3) Agent 间消息传送的方法。

首先应在主控端建立一个 AgletProxy 类的对象,如 controllerProxy,在主控端创建新 Agent 时,将 controllerProxy 赋予创建方法中所使用的 args 参数,然后在新创建的 Agent 中也建立一个 AgletProxy 类的实例,在新创建的 Agent 的 onCreate 方法中接收传送过来的 controllerProxy,创建 Agent 和被创建的 Agent 就有了统一的 AgletProxy,利用它可以实现双向之间的消息传送。发送消息可用如下方法: controllerProxy.sendMessage(new Message("Result")),接收消息则可以使用前面提到的 handleMessage 函数。

4) Agent 消除方法。

使用 dispose() 就可以销毁 Agent。

2.2 主控 Agent 创建考生 Agent 的实现

创建考生 Agent 的代码如下:

```
for(int i=0;i<ipNO;i++){
stuNO=stuInfo[i][0]; //考生考号
stuPwd=stuInfo[i][1]; //考生口令
```

```
String Destination = atpArray[i]; //考生机 IP
args = new Object[] { controllerURL, controllerProxy, subjectNO,
stuNO, stuPwd, Destination };
getAgletContext().createAglet(getCodeBase(), "testgui", args);
//创建 agent
```

其中, ipNO 是整型变量, 用于存放本次考试考生机的数量; Destination 用于存放考生机 IP 地址; controllerURL 是 URL 类的对象, 用于存放主控 Agent 的 URL; controllerProxy 是 AgletProxy 类的对象, 用于存放 AgletProxy, 使得主控端 Aglet 和客户端 Aglet 可以有相同的 AgletProxy 以进行通信; args 是一个 Object 类的数组, 用于存放创建考生 Agent 时的相关参数, getAgletContext().createAglet(getCodeBase(), "testgui", args) 方法用于建立新的 Agent, 其中参数 testgui 为新 Aglet 的代码名。

2.3 考生 Agent 移动到考生机的实现

考生 Agent 完成初始化工作后, 按照主控 Agent 所提供的参数移动到指定的考生机上。方法为:

```
if(! move){
try{ URL url = new URL(destination);
dispatch(url); }
catch(Exception e){ }
else{ frame.setVisible(true); }
```

用于记录 Agent 是否移动过的变量 move 初始值为 false, 移动过则值为 true, 当值为 false 时, 考生 Agent 开始向考生机移动, 当为 true 时, 显示图形界面, dispatch 方法将 Agent 派送到考生机上。

2.4 Agent 之间通信的实现

要实现考生 Agent 与主控 Agent 之间的通信, 需要在两端设置 AgletProxy(代理), 然后通过 AgletProxy 来发送和接收消息。

首先当主控 Agent 创建时, 得到一个新的 AgletProxy, 方法为:

```
AgletProxy controllerProxy = getProxy();
```

其中 AgletProxy 是 Aglet 提供的类, getProxy 方法用于从 Agent 上下文环境中获取 AgletProxy。主控端将 controllerProxy 作为建立新考生 Agent 的参数, 传递给考生 Agent, 考生 Agent 在 onCreate() 事件函数中获取主控 Agent 传来的 AgletProxy。这样主控 Agent 与考生 Agent 都获得了 AgletProxy 对象, 利用该对象发送消息, 发送消息的方法为:

```
controllerProxy.sendMessage(new Message("dispose", text));
```

(上接第 50 页)

- [3] Dr Jose P, MSc Rodolfo M, MSc Victor S. DSP - Based Integral Variable Structure Control for Permanent Magnet AC and DC Motor[J]. Proc IEEE, 2000, 2: 724 - 729.
- [4] Grenier D, Dessaint L A, Akhrif O, et al. Experimental Nonlinear Torque Control of a Permanent Magnet Synchronous Mo-

tor Using Saliency[J]. IEEE Trans IndElectronics, 1997, 44 (5): 680 - 687.

sendMessage 是 AgletProxy 类提供的用于发送消息的方法, 后面所跟参数应为 Message 型, "dispose" 是所发消息的标识, text 是用于存放消息内容的字符串。

当主控 Agent 或考生 Agent 接收到对方发送来的消息时, 可根据对消息标识的判断, 并根据消息的类别, 执行不同的事件处理函数。方法为:

```
public boolean handleMessage(Message msg)
{ if (msg.sameKind("finish"))
{operationCode //事件处理代码} }
```

handleMessage 方法用于处理消息。msg 是一个 Message 型变量, 用于存放接收到的消息; sameKind 方法用于判断收到的消息标识是否与参数 finish 相符, 相符则执行 operationCode 代码。

3 结束语

介绍了基于 Aglet 的网上考试系统的实现方法。该方法利用移动 Agent 代码移动的特点, 使系统具有较高的效率, 较好的安全性和易用性, 因此, 基于 Aglet 的网上考试系统具有较好的应用前景。当然 Aglet 本身是一项新兴的技术, 其运行平台还没有完全普及, 因此, 基于 Aglet 的网上考试系统要达到实用程度还有许多工作要做。

参考文献:

- [1] 王丽侠, 王 新, 衣治安. 基于 ASP 技术的网上考试系统的设计与实现[J]. 大庆石油学院学报, 2001, 25(4): 54 - 57.
- [2] 李 琦. 基于 C/S 模式的计算机等级考试上机考试系统[J]. 重庆大学学报, 2004, 27(7): 143 - 146.
- [3] 张友生. 基于 agent 的体系结构[J]. 计算机工程与设计, 2005, 26(2): 342 - 343.
- [4] 张云勇. 移动 agent 及其应用[M]. 北京: 清华大学出版社, 2002.
- [5] Lange D B, Oshima M. Seven Good Reasons for Mobile Agents[J]. Comm ACM, 1999, 42(3): 88 - 89.
- [6] Papaioannou T, Edwards J M. Using Mobile Agents to Improve the Alignment Between Manufacturing and its IT Support Systems[J]. Journal of Robotics and Autonomous Systems, 1999, 27: 45 - 57.
- [7] 张 谨, 张德贤. IBM Aglets 系统研究与应用[J]. 河南大学学报, 2002, 32(1): 40 - 43.
- [8] 刘锦德, 张云勇. 一个实用的移动 Agent 系统 (Aglet) 的综述[J]. 计算机应用, 2001, 21(8): 1 - 3.

tor Using Saliency[J]. IEEE Trans IndElectronics, 1997, 44 (5): 680 - 687.

- [5] Lorentz R D, Lipo T A, Novotny D W. Motion Control With Induction Motor[J]. Proc IEEE, 1994, 82: 1215 - 1240.