

# 基于 Agent 的计算机考试系统的设计与实现

## Design and Implement of Test Systems Based on Agent

徐义峰 徐云青 (衢州学院 浙江衢州 324000)

**摘要:** Agent 技术为分布式开放系统的分析、设计和实现提供了一种新方法。针对原考试系统的局限性,引入 Agent 思想进行重新设计,实现了基于 Agent 的计算机考试系统。通过新的设计,将远程交互转化成本地交互,减少了对网络带宽的占用,提高了系统的响应速度和可靠性。

**关键词:** Agent 分布式计算 考试系统

### 1 引言

计算机考试系统的应用已经十分广泛,但现有系统存在许多局限性,随着网络技术的发展,这些局限性越来越突出。在传统 C/S 模式下的考试系统中,试题库放在服务器端,考生在客户端进行考试。在考试过程中,需要在网络上来回传输包括试题内容在内的许多信息,客户端和服务端要进行很多次的交互,而在交互期间,基本上都要求保持通讯连接。因此,大规模的考试几乎无法进行,特别是在远程考试的情况下。因为现在的网络经常会不通或者经常发生阵发性的拥塞,很难保证所有客户机同时与服务端保持较长时间的持续连接,在现有带宽条件下,是无法做到的,即使能够做到,那也是奢侈的,是对带宽资源的严重浪费。这使得考试系统的可靠性在很大程度上依赖于网络的连通,而这种连通性往往无法得到保证。针对这些问题,我们在原有基于 C/S 模式的计算机考试系统基础上,引入 Agent 技术<sup>[1]</sup>,对原系统中部分进行重新设计和开发,实现了一个基于 Agent 的考试系统,很好地解决了原有系统中的问题。

### 2 基于 Agent 技术的考试模型

系统有三个功能模块:

(1) 题库管理模块。对题库进行管理和维护,主要包括试题的录入、修改、删除、分类等常规数据库管理工作;

(2) 考试相关事务管理模块。包括考试报名注册、成绩管理、合格证书发放等;

(3) 考试服务模块。完成整个考试过程,包括生成试卷、提供考生考试界面使考生完成考试、对考生的考试结果进行评分等。在上述三个功能模块中,前两个模块只在本地进行数据库管理工作,仍然沿用原管理系统,而考试服务功能模块的改进是本文讨论的重点。

由于 Agent 和考试系统各自的特点,使得系统特别适用 Agent 模型来实现。考试系统的特点:考试是在各个考点的很多台机器上同时进行,考试本身具有分布和并行的特点;考试时间较长,因此若采用需要持续连接的计算模式是不现实的,也是不可取的;考试的结果,特别是一些要在磁盘中留下痕迹的操作,分布在各台机器上,采用移动 Agent<sup>[2]</sup>来进行阅卷比较合理;考试工作本身具有严肃性,对考试系统的可靠性、健壮性要求较高。Agent 技术正好具有适合建立高可靠性考试系统的特点:强的远程交互性能;可以实现非连接交互;支持弱客户机;容易分布;可以个体化地为客户机服务等。

因此,我们借助 Agent 的概念,重新构建现有的考试模型,在这种新的模型(如图 1)中,Agent 既是系统的基本构成单位,又是系统的独立运行实体。系统主要有以下四个 Agent:

(1) 试卷 Agent。根据试卷生成算法(采用了离散型随机变量的二项分布函数  $B(n, p)$  建立选题的数学模型,确定题型难度分布,再利用随机函数进行选题),从题库中抽取符合总体难度系数控制要求的试卷,生成考试内容。生成的试卷以加密文本文件的方式提交

给考试 Agent。试卷生成 Agent 的工作可以看成多 Agent 系统中的动态任务分配过程。

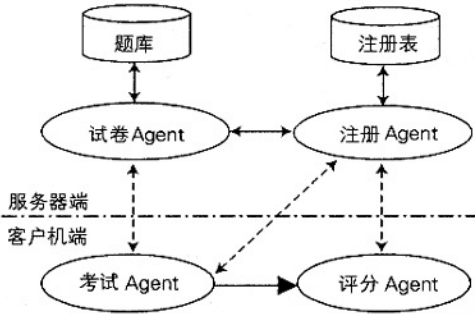


图 1 考试系统模型

(实线箭头表示本地交互,虚线箭头表示远程交互)

(2) 考试 Agent。主要是为考生提供考试服务,该 Agent 在考生所在的客户机上运行,每个考生的考试由一个 Agent 来负责。在整个考试过程中和考生进行交互,在考生输入考试号后,将考试号发送至注册 Agent,由注册 Agent 完成注册和检查工作;从考试开始即进行计时,到考试结束时候,关闭所有与考生的交互;考生结束考试或者考试时间到,则启动评分 Agent 进行评分。在考试 Agent 中主要包括了考试界面和由试卷 Agent 生成的考试题。考试 Agent 的这种构成策略,既使 Agent 本身不会过于庞大,又避免了考试过程中与服务器的持续交互。

(3) 注册 Agent。主要负责接收考生的考试号信息,在考生注册表中进行登记,这样既可以得到参加考试的实考人数和缺考人数,也可以检查考试号的合法性,包括:考试号长度、是否合法注册考试号、有没有重号等。正确注册之后,在考生机器上显示出考生姓名和考试类别,便于监考教师核查考生身份。若发现上述情况中的任何一种不合法情况,则向考生发出警告信息,提醒考生检查考试号是否正确,同时在服务器上发出警告信息,要求监考教师协助检查。考生重新输入考试号后,要重新进行注册检查。对考试号的合法性检查是必须的,这样做不仅可以立刻统计出实考人数,而且可以避免由于误输考试号带来的额外工作。

(4) 评分 Agent。对考生的考试结果进行评判,将评判结果传送回服务器。同时根据考生答题的正确

性,对各考题的难易程度作出评估。

### 3 考试 Agent

#### 3.1 考试 Agent 结构

考虑考试系统的实际需求和实现的简便,本系统采用层次结构来实现各个 Agent<sup>[3]</sup>。以考试 Agent(如图 2)为例来说明 Agent 的结构。

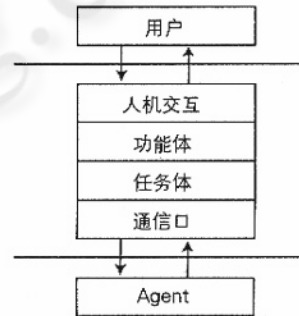


图 2 考试 Agent 的结构

(1) 通信口。是一个单纯的通信接口,处理进程与 Agent 之间通信格式的转换,并将结果提交给任务体,同时接受来自任务体的消息并发送给指定的 Agent。各 Agent 之间完全通过通信口完成彼此的通信,在不同的 Agent 中,通信口的作用是相同的。系统中通过一个独立的 DLL 来完成通信口的功能,有两个通信接口:消息响应函数 OnReceiveInfo(),接受外部传来的消息;消息发送函数 SendInfo(),被任务体调用,发送消息。

(2) 任务体。接收来自通信口的信息,根据自身在系统中的位置,判断信息的恰当性,决定是否提交给功能体,或者转发给其它合适的 Agent,是 Agent 中的协调组织者。同时,任务体接收来自功能体的信息,根据要求发送给指定的 Agent。任务体内部记录其它各个 Agent 的地址和状态,根据需要选择合适的 Agent 发送信息。任务体还是一个感知器,定时感知整个系统的运行情况,以更新内部各种记录和状态。考试 Agent 中任务体的状态有登录和未登录两种状态。其接口包括:与通信口的接口 RecelvelInfo(),从通信口传递消息到任务体;与功能体的接口 LoginRegisterAgent()和 CommitAnswer(),其功能分别为登录注册 Agent 和

提交答案。

(3) 功能体。接收来自任务体经过处理的信息或人机交互提供的信息。功能体分析信息,并将结果返回给任务体或者提交给人机交互展示给用户。考试 Agent 中功能体的状态与考试 Agent 状态保持一致,有等待、考试两种。其接口包括:与任务体的接口 `ReceiveInfo()`,从任务体传递消息到功能体;与人机交互的接口 `LoginRegisterAgent()` 和 `CommitAnswer()`,其功能分别为登录注册 Agent 和提交答案。

人机交互:提供一个人机交互。接收用户信息提交给功能体;接收功能体返回的结果展示给用户。

需要说明的是,任务体与功能体的接口很相似,但两者的功能是不同的。功能体决定登录或提交答案是否可行,如果可行则由任务体来处理数据的格式化,即组织成标准的通信格式进行发送。

### 3.2 考试 Agent 状态图

考试 Agent 有两种状态:等待状态和考试状态。等待状态:考试 Agent 启动后,进入等待状态,这时没有与任何其它 Agent 发生关系;考试状态:考试 Agent 发送考试号到注册 Agent,要求进行考试行为,如果成功登录,则进入考试状态,若登录失败,则考试 Agent 处于等待状态不变,考试结束后,考试 Agent 回到等待状态。其状态图如图 3 所示。

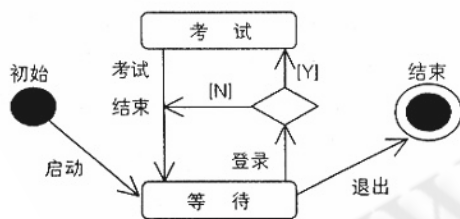


图 3 考试 Agent 状态图

### 3.3 考试 Agent 序列图

考试 Agent 各个层次的函数调用和消息传递过程如图 4 所示。

## 4 系统工作流程及 Agent 之间的通信

系统工作流程(如图 1)。

(1) 考生首先必须要下载并启动考试程序(考试

Agent),由考生在其界面上输入考试号,考试 Agent 启动一个线程,将考试号发送到注册 Agent。若是合法的考试号,则由注册 Agent 将考试号发送给试卷 Agent,后者收到后,发送一份试卷给考试 Agent,考试 Agent 收到试卷后,开始计时,考生开始考试。

(2) 注册 Agent 收到考生的考试号后,启动一个线程,在注册表中注册。若是合法的考试号,则返回考生的姓名和考试类别,显示在考生所在的客户机上。若不合法,则对考生提出警告。

(3) 考生考试期间,试卷 Agent 还负责把评分 Agent 发送到考生所在机器的指定目录。考试时间到,考试 Agent 停止与考生的交互,并启动评分 Agent 对考试结果进行评判。

(4) 评分 Agent 完成评判后,将结果发送到注册 Agent,由后者将各项结果登记到注册表中。考试结束。

在考试过程中,考试 Agent 需要与注册 Agent 和试卷 Agent 之间进行通信和数据传输。考试 Agent 与注册 Agent 进行交互时,才需要在网上传递信息,而且传递的信息量很小,故实现时采用 socket 技术中面向无连接的数据报通信协议(UDP)来传递信息。考试 Agent 与试卷 Agent 之间需要传递试卷和评分 Agent,数据量相对较大,为了保证数据传输的可靠性,故实现时采用 socket 技术中面向连接的字节流通信协议(TCP)来传输。试卷 Agent 和注册 Agent 将与多台机器上的考试 Agent 之间传递数据,为了提高系统的效率,实现时都使用了缓冲队列。试卷 Agent 提前生成多份试卷及相应的评分程序并存放在缓冲队列中,这样就可以直接从缓冲队列中发送试卷。

## 5 系统的优势分析

基于 Agent 考试系统的实现方式是一种全新的分布式计算模式,与 C/S 模式有着本质的不同<sup>[4]</sup>。考试 Agent 要求注册 Agent 完成注册检查的工作,从形式上看,与 C/S 模式的请求/应答方式差不多,但在实现过程中,考试 Agent 仅需要发送一个包含考试号的报文给注册 Agent,发送过后并不要求网络一定要保持畅通,也不需要保持连接以等待注册工作的完成,注册 Agent 在完成工作后,将结果发送给考试 Agent。在整个过程中,只需在传送时网络的“瞬时”连通即可。同

样,考试 Agent 与试卷 Agent 之间的关系也是如此。这种基于 Agent 技术的设计和实现方式有着明显的优点:

分散到客户机,减轻了服务器的负担。同时,现在的客户机一般都拥有较强的处理能力,这样的结构还能够充分利用客户机的处理能力和资源。

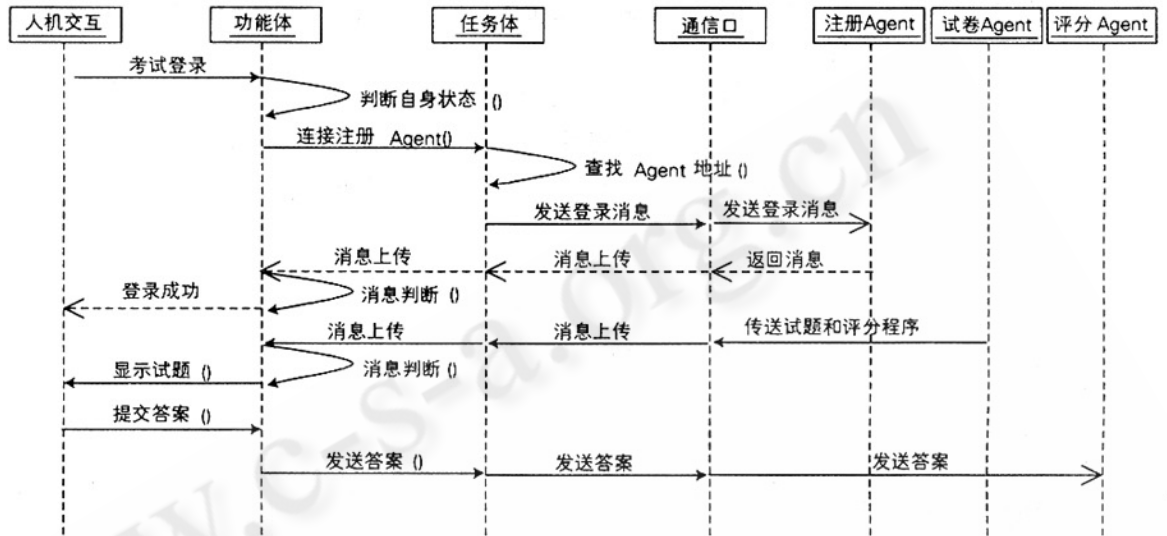


图 4 考试 Agent 序列图

(1) 通过 Agent 化的封装,将远程交互转变为本地交互。考试工作的 Agent 化,使客户端的考试过程完全在本地进行,不再需要与服务器端的题库进行持续的交互,这种远程交互本地化的实现本质上是通过计算向资源移动来实现的(将考生所在机器看作资源,将考试 Agent 作为计算实体);同样评分 Agent 移动到考生所在的机器运行,实现了评分过程的本地化,也是通过计算向资源移动来实现。注册工作的 Agent 化,使得原本在客户端与服务器上注册表之间的远程交互变成服务器端的本地交互。

(2) 远程交互的本地化,使得 Agent 能够和本地资源进行交互而不需要通过网络传递中间数据,不仅有效地节约了带宽,提高了响应速度,减少了网络数据的传输量,而且使得考试系统的可靠性对网络的依赖程度大大降低,也保证了考试的公平性,不会再出现不同的考生因网络延迟的时间不同使得实际的考试时间也不同。这样的考试系统甚至可以在低带宽和高延迟,不可靠的网络中应用。

(3) 每个考生的考试任务由考试 Agent 在考生所在机器上相对独立完成,将原本集中在服务器的处理

(4) 这种考试模式不需要事先在客户端安装客户端程序,可以随到随考,具有较高的灵活性。从理论上讲,随便一台可以上网的机器,都能用来进行考试。

## 6 结束语

通过对本系统的设计与实现,展示 Agent 技术在传统程序设计领域中的作用,在已有的应用系统中通过引入 Agent 技术来改进系统的性能。在开发过程中,建议坚持三个观点:注重 Agent 的实用性;网络环境下计算与资源相结合;系统灵活性与开放性。

### 参考文献

- 何炎祥, Agent 和多 Agent 系统的设计与应用[M], 武汉大学出版社, 2001。
- 张云勇, 移动 Agent 及其应用[M], 北京清华大学出版社, 2002。
- 乔善平, 基于智能移动 Agent 的 C 语言考试系统[J], 计算机工程与科学, 2004(4):101-105。