

# 互动答题系统需求

## 1 综述

本系统通课堂教学效果即时反馈系统一改以前老师讲，学生听的“填鸭式”教育模式，为老师提供了一套方便的获得学生反馈的电子教学工具。老师即时提出问题（口头或通过 PPT），学生们通过自己手上的答题器回答，每条问题的即时回应都会总结到一个图表中，不论是教师或学生，对每次作答情况都能一目了然。这使学生对自己的学习有客观评价，课堂辅导更具针对性，教师也可以及时调整教学进度。系统还可以将学生课堂学习进度报表及时提供给家长协助监督学生学习。

互动反馈系统调动了课堂中所有学生的积极性，让每个人都可以直接参与到每一堂课，实现记名或不记名作答方式，提高师生互动，活跃了课堂气氛，激发学生学习兴趣和参与乐趣。

不单是在平常的教学情况下，有许多地方也能用到这样的系统，如：电视台的综艺节目需要观众进行投票，一些单位机构进行评比选举，有些企业需要做社会调查，还有电视上有奖答题与电视机前观众的互动等等。

使用起来又方便又快捷，而且显示全面，分析彻底。

### 1.1 使用场景

#### 1.1.1 无线答题器使用场景设计 1

- 1、教师针对某一问题发问，给出若干选项，屏幕显示倒计时
- 2、学生按下手中答题器选择答案，倒计时结束前允许多次更改答案
- 3、倒计时结束之后显示答题结果，显示可用多种方式：表格、柱状、明细等

- 4、再显示出正确答案，老师给出讲解和分析
- 5、答题结果可以显示出对错人数，对错比例，或者每个人的答案。

### **1.1.2 使用场景设计 2**

IPTV 上实现交互式游戏，比如实现 QQ 游戏中的“大家来找茬”

- 1、进入游戏室，选择准备开始游戏。
- 2、开始游戏启动倒计时。
- 3、启动倒计时完毕游戏开始，启动游戏结束倒计时。
- 4、玩家通过自己的按键器进行游戏。
- 5、游戏倒计时结束，显示玩家成绩。

### **1.1.3 PC 使用场景设计 3**

局域网，网络延迟可以忽略，若是互联网需要考虑延迟问题。课堂测验，30 道题 45 分钟完成

- 1、学生登录客户端
- 2、答题服务器发送试题到客户端，启动答题计时器
- 3、学生开始答题
- 4、答题完毕交卷，结束答题计时器，传送答案到答题服务器
- 5、答题服务器传送成绩到客户端
- 6、学生能看到自己的成绩，以及每道题的答题情况，和正确答案

### **1.1.4 综艺节目使用场景设计 4**

电视节目总需要场内或场外的观众对场上的选手进行投票

- 1、节目主持人宣布投票开始，现场屏幕和电视机屏幕显示倒计时开始计时
- 2、现场观众可用手中的投票器进行投票，只能按一次，如果多按，则以第一次的投票结果作数
- 3、场外的观众可以通过手机短信进行投票，每个手机号只能投票一次，如果多

投则提示错误或作废

- 4、在规定的时间内，收集投票的结果，结束倒计时器
- 5、屏幕上显示出每位选手的柱状投票结果图
- 6、主持人宣布投票产生的结果

### **1.1.5 打分使用场景设计 5**

一些单位机构对每个员工的相互打分

- 1、领导先主持大家对某个员工进行打分
- 2、大家开始用手中的大分器记名或不记名地对这个员工进行打分
- 3、打分完毕，分数汇总
- 4、可以显示总分，平均分，（每个人的打分情况）等等

### **1.1.6 选举使用场景设计 6**

一些单位机构在某项选举大会上进行选举投票

- 1、主持人公布候选人，并宣布选举开始
- 2、大家开始按投票器投票，若有重复按键，则按最后一次按键作数
- 3、宣布投票结束，所有人都不许再按键
- 4、投票结果显示，可以有列表，柱状图，饼状图等等显示方式

### **1.1.7 调查问卷使用场景设计 7**

企业的社会调查

- 1、企业给出调查问卷，可以以群发短信的形式告诉所有被调查者
- 2、被调查者可以通过手机短信的形式对调查问卷作出回答
- 3、服务器接收回答并汇总
- 4、结果可以对每个问题的回答进行分析，显示出比例分布图，或者选项百分比等等

## 1.1.8 有奖答题使用场景设计 8

电视上的有奖答题

- 1、电视上的节目出一些谜语，脑筋急转弯之类的题目让电视机前的观众以短信的形式进行回答或者抢答。

### 若进行回答再抽奖

- 2、当主持人宣布答题开始时倒计时器开始计时
- 3、观众在规定的时间内发短信，只能发一次，若重复发送，则提示错误或者作废，不受理
- 4、结束倒计时器，由服务器收集答案，并整理出答对题目的人，
- 5、再在回答正确的人中抽取幸运观众
- 6、公布结果。

### 若通过抢答来评奖

- 2、当主持人宣布答题开始时用户计时器开始计时
- 3、观众在规定的时间内发短信，只能发一次，若重复发送，则提示错误或者作废，不受理
- 4、当第一个用户发送短信成功并且回答正确时，结束用户计时器
- 5、以这个用户作为获奖观众
- 6、公布结果

## 1.2 任务与目的

我们的任务是：

- 1) 开发交互测验的系统，包括：硬件、软件。完成系统的开发、测试和使用验证。
- 2) 从应用场景中，抽象出通用的软件部件或模块，以及相互间的通信与编码协议。

3) 依据不同的使用环境和系统，能够将软件部件进行配置，形成新的系统。

## 1.3 计划与项目的执行

作为软件工程任务，需要经历需求、设计、构造、测试等过程。但是，作为学生项目，需要尽快迭代出一个结果。

学生要：

- 1) 讨论和定义过程阶段，
- 2) 给出预计的时间和人员分工
- 3) 按计划执行该过程
- 4) 验证计划与实际的是否一致？
- 5) 评审与检查活动：在计划中要明确每个阶段或活动的结束时，如何？何时进行评审？评审中发现的问题如何解决？

## 2 应用环境

考虑几种应用环境：

- 1) 每个学生和老师面前都有一台 PC 机；
- 2) 只有老师有一台 PC 机，每个学生手里有一台答题器
- 3) 双向机顶盒，（远程教育的）每个学生，在观看电视时，可以通过遥控器答题
- 4) 单向广播机顶盒（有外存空间），（远程教育的）每个学生，在观看电视时，可以通过遥控器答题。
- 5) 每个观众观看电视的时候通过手机短信进行答题

### 2.1 学生和老师都有一台 PC 机

这种情况下，假定每个学生都已登录系统中。系统的示意图 2-1 如下：

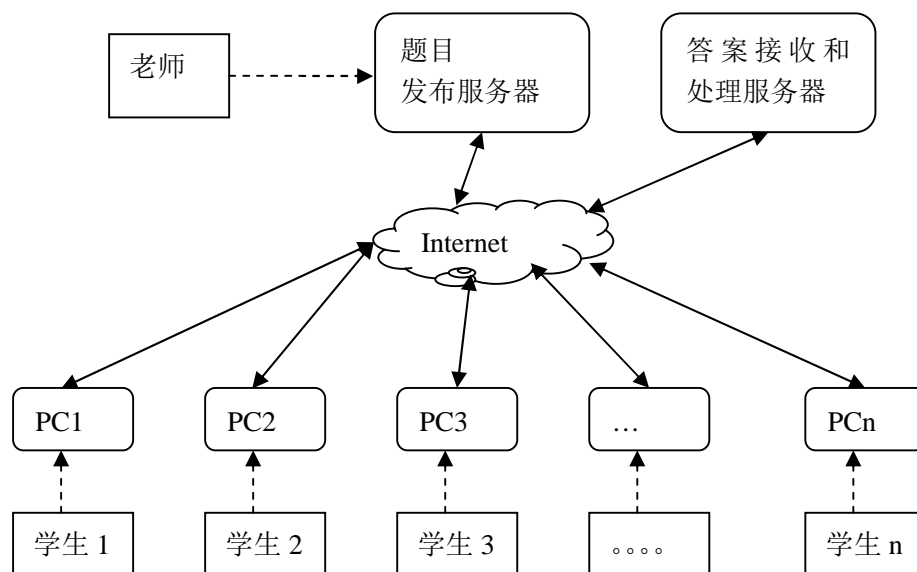


图 2-1 PC+ Internet 的交互答题系统

- 1) 老师将题目先发送至 Internet 网上
- 2) 每个学生在自己面前的 PC 上登录到可以答题的服务器上
- 3) 学生用自己的账号回答老师提出的问题
- 4) 学生将自己的答案提交到 Internet 网上
- 5) 答案接收和处理服务器从 Internet 网上获取学生的答题信息
- 6) 答案接收和处理服务器按照老师需要的结果和分析进行显示

## 2.2 教室中每个学生有一个答题器

在具有多媒体设备（PC 机+投影仪）的教室中。每个学生有一个答题器（无线或有线）。

无线答题器（votingmachine）：用于学生选择答案编号，其示意图如下。



图 2-2 无线答题器

无线接收主控基站：接收学生选择的编号，并将获得的数据传送给系统守护进程。工作示意图如图 2-3 所示。



图 2-3 具有无线答题器的教学环境示意图

- (1) 教师在自己的 PC 机上显示出题目
- (2) 由投影仪将 PC 机上的屏幕显示放大显示到投影布上
- (3) 学生看投影布上的题目并思考
- (4) 每个学生通过自己手中的答题器进行答题
- (5) 无线接收主控基站接收学生选择的编号，并将获得的数据传送给系统守护进程

(6) 系统守护进程处理学生的答案并把结果和分析显示到 PC 机屏幕上

(7) 投影仪再将这些结果显示到大型投影布上

## 2.3 双向机顶盒

在广播环境中，采用双向机顶盒（STB），意味着题目可以经广播通道传送到每个机顶盒上。机顶盒也可以接收学生的答案（按键），通过回传通道，将结果发送给“答案接收和处理服务器”。

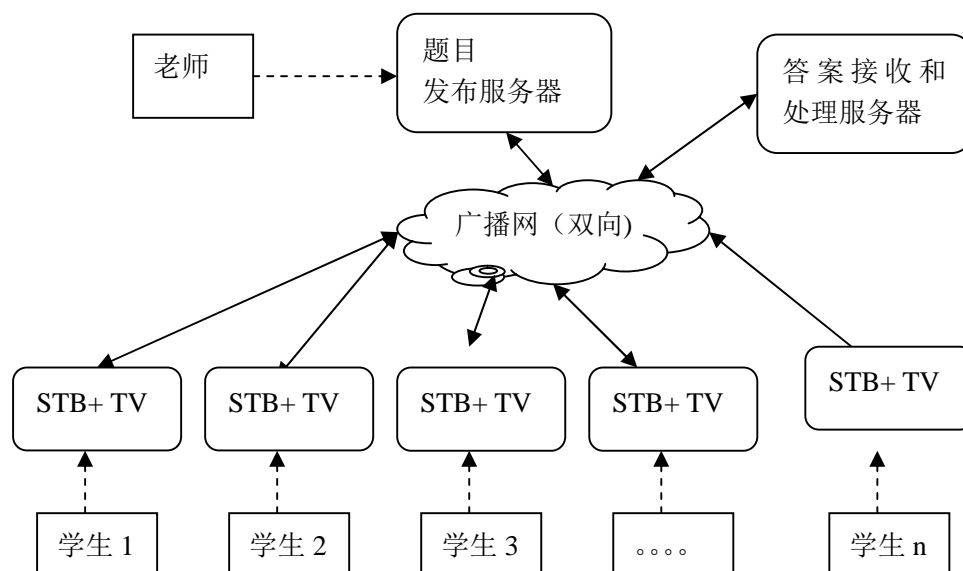


图 2-4 双向机顶盒答题系统示意图

(1) 老师将题目放置发布服务器上

(2) 题目经广播网通道传送到每个双向机顶盒上

(3) 通过双向机顶盒（STB）和电视机（TV）学生可以看到题目

(4) 每个学生通过自己手中的按键器进行回答



- (5) 双向机顶盒接收学生的答案（按键），传送到广播网
- (6) 通过回传通道，结果被发送给“答案接收和处理服务器”
- (7) 答案接收和处理服务器对答案进行处理和分析，并将结果反馈或显示

## 2.4 单向机顶盒

在广播环境中，采用单向机顶盒（STB），意味着题目可以经广播通道传送到每个机顶盒上。机顶盒也可以接收学生的答案（按键），但没有直接上传答案的回传通道，因此，不能将结果立即发送和处理。

这种情况下，有两种处理模式：

- 1) 在所有的题目答完后，广播端将答案广播到每个机顶盒上，有机顶盒自动校对答案，并呈现结果。
- 2) STB 将答案记录到外存，例如，一个 U 盘中。用户用其它方式将 U 盘答案，批量发送到“离线答案接收和处理服务器”。以便与广播端的老师获得统计数据。

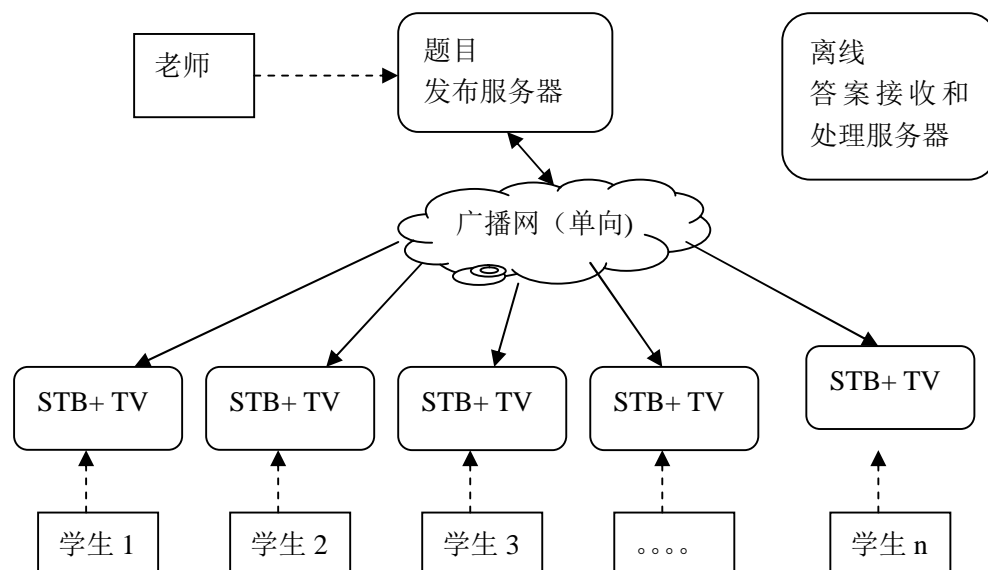


图 2-5 单向机顶盒答题系统示意图

- (1) 老师将题目放置发布服务器上
- (2) 题目经广播网通道传送到每个单向机顶盒上
- (3) 通过单向机顶盒 (STB) 和电视机 (TV) 学生可以看到题目
- (4) 每个学生通过自己手中的按键器进行回答
- (5) 当学生把所有的题目回答完毕后, 有两种处理模式:

第一种: 单向广播端将答案广播到每个单向机顶盒上, 由单向机顶盒自动校对答案

第二种: STB 将答案记录到外存, 例如, 一个 U 盘中。用户用其它方式将 U 盘答案, 批量发送到“离线答案接收和处理服务器”。以便与广播端的老师获得统计数据。

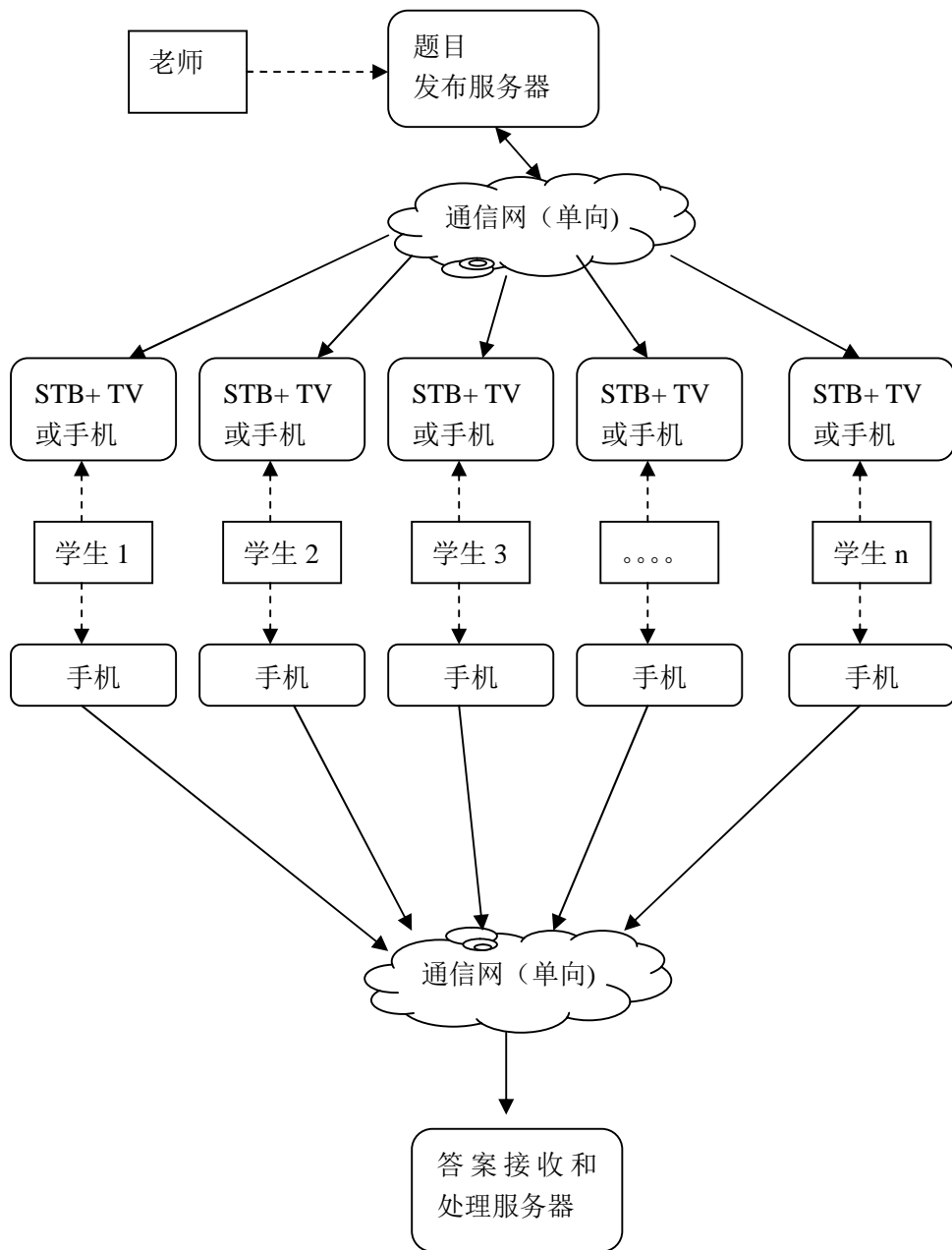
- (6) 因而之后的显示方式也有两种:

第一种: 单向机顶盒自动校对完答案后, 在学生面前的 TV 上显示学生的答题结果, 老师不能直接得到学生答题的结果。

第二种: 离线答案接收和处理服务器保存学生回答问题情况的数据, 并且与广播端的老师联系, 使其获得统计数据。

## 2.5 通过手机短信答题

在通信环境中, 题目可以经通信通道以短信的形式传送到每个观众所拿的手机上或者通过单向机顶盒显示在观众面前的电视上。通过回传通道, 观众在把答案以手机短信的形式将结果发送给“答案接收和处理服务器”。



- (1) 老师将题目放置发布服务器上
- (2) 题目经单向通信网通道传送到每个单向机顶盒上或者每个手机上
- (3) 通过单向机顶盒 (STB) 和电视机 (TV) 或者手机短信学生可以看到题目
- (4) 每个学生通过自己手中的手机以短信的形式回答问题

- (5) 学生的回答短信发送到另一个单向通信网
- (6) 答案接收和处理服务器通过通信网接收学生的回答信息
- (7) 答案接收和处理服务器对这些信息进行处理并按照要求显示

## 3. 需求的抽象

对上面的几种情况，需要抽象出共性部件。显然，这是一个典型的服务/客户（service/client）模型。我们看到：

### 3.1 服务端（Service）

服务器端应具有：

- 1) 题目的编排：从题目库中选择，或老师临时出题，形成统一的发布格式。
- 2) 题目的发布：将题目转换成特定的数据格式（用广播协议、还是 PPT 的 Slide、Web 网页等），发布到网上。
- 3) **答案的接收**：实时或离线（非实时），接收学生的答案。
  - 实时接收时：同时能接收多少个学生的答案？
    - 时间延迟是多少？
    - 如何解决交通阻塞？
    - 如何解决重复按键？
  - 离线接收时**：如何导入到外存中？
    - 如何将外存中的数据批量传送至处理服务器？
    - 如何解决校验问题来确保传输中不会造成数据丢失和错误？
    - 如何解决交通阻塞？
- 4) **判卷和试卷统计**：与标准答案比较，给每个学生打出分数；统计和分析试卷情况，
  - 例如，答错最多的题目是哪一个？
  - 谁最先答对？
  - 每个题目的正确率是多少？
  - 达到标准的学生比例是多少？
  - 等等。
- 5) 学生身份认证：只有认证过的学生，才能答题。
  - 如何认证？

是人为检验还是登录验证？

- 6) 计时器：有时需要评判学生的答题速度和回答问题的优先次序  
在规定的多少时间内需要作答？超过时间答题不作数  
谁最先回答出问题？

## 3.2 客户端（Client）

客户端应具有：

- 1) 接收和解析题目数据：接收题目的数据格式是啥样的结构？  
将数据格式解析出来。
- 2) 题目呈现：将解出的题目数据以何种方式呈现出来？（PPT？ Web？ 其它？）
- 3) 接收答案并编码：接收学生输入的按键，转成上传的编码。
- 4) 实时发送答案到服务器：编码是否冲突？  
编码校验后是否准确？  
**或离线发送答案到服务器：**编码是否冲突？
- 5) 学生身份的认证：学生必须登录才能进行答题。
- 6) 显示答题结果：学生可以得到答题结果的显示，  
成绩是多少？  
每个题的正确答案是什么？  
谁回答的最快？
- 7) 计时器的显示或提示：在倒计时规定的时间内作答  
或者如果时间到则作出提示并且不能再答题
- 8) 学生的错题整理：**对学生回答的错题进行归纳总结，以便学生日后复习

## 3.3 客户端与服务器的通信协议

客户端和服务端之间的数据交换，构成了两者之间的通信协议。定义一个这样的协议需要解决：

- 1) 题目的编码：用 ASCII 形式？  
用 Text 形式行吗？  
还需要什么？  
是传到学生面前的终端 PC 上还是用投影仪？
- 2) 答案的编码：直接用字符(A、B、C、D)行吗？  
还是用 ASCII 码？  
如何区分题目？  
如何区分学生的身份？
- 3) 编码后传输的手段是：TCP/IP？  
套接字？

端距离无线协议？  
广播协议？

### 3.4 问题

依据上述的需求抽象，你能否给出一个基本的子系统或模块的划分框图？

## 4. 软件工程需求分析

### 4.1 服务器端的软件需求分析

#### 4.1.1 功能分析

##### 4.1.1.1 出题功能分析

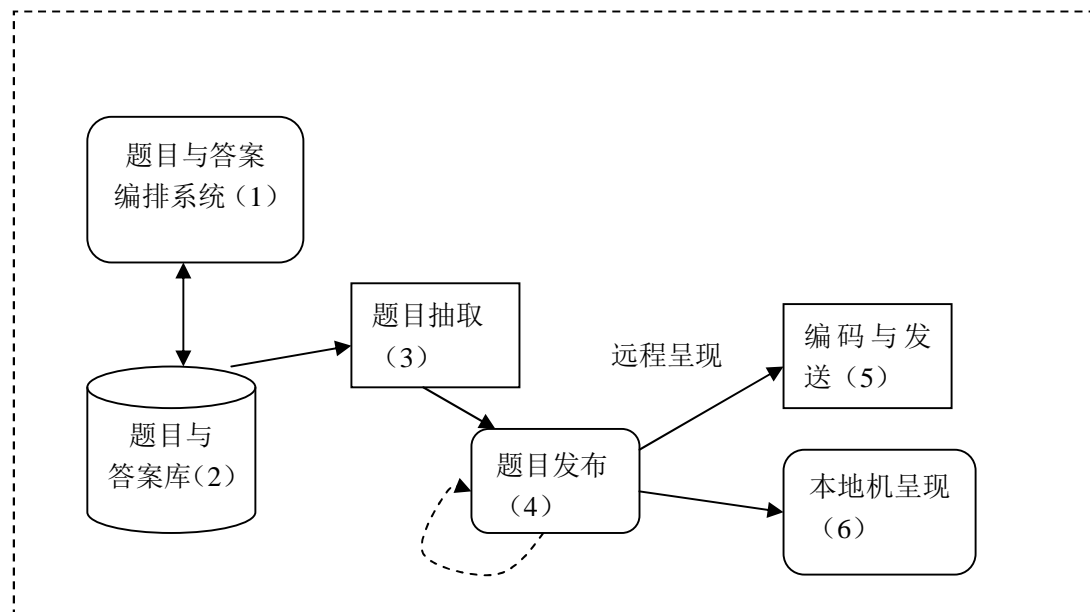


图4-1 题目编排与题目发布数据流示意图

- （1） 题目与答案编排系统首先对老师所要提问的题目及相应的答案进行编排，题目及答案的顺序均可随机的或固定的
- （2） 题目与答案编排系统将编排好的题目放置在题目与答案库。
- （3） 题目与答案编排系统可将题目与答案库中已经编排好的题目及答案

进行重排，重排后继续放置题目与答案库中，并且替换掉之前在题目与答案库中的题目

- (4) 随机或手动抽取已存在在题目与答案库的题目
- (5) 将抽取的题目进行发布
- (6) 可以只发布一次，也可重复发布
- (7) 将题目进行编码，通过远程呈现，将题目发送至网络
- (8) 通过网络传输，学生获取题目
- (9) 或者将题目用本地机呈现
- (10) 若须继续出题，则重复上述过程

#### .4.1.1.2 答题功能分析

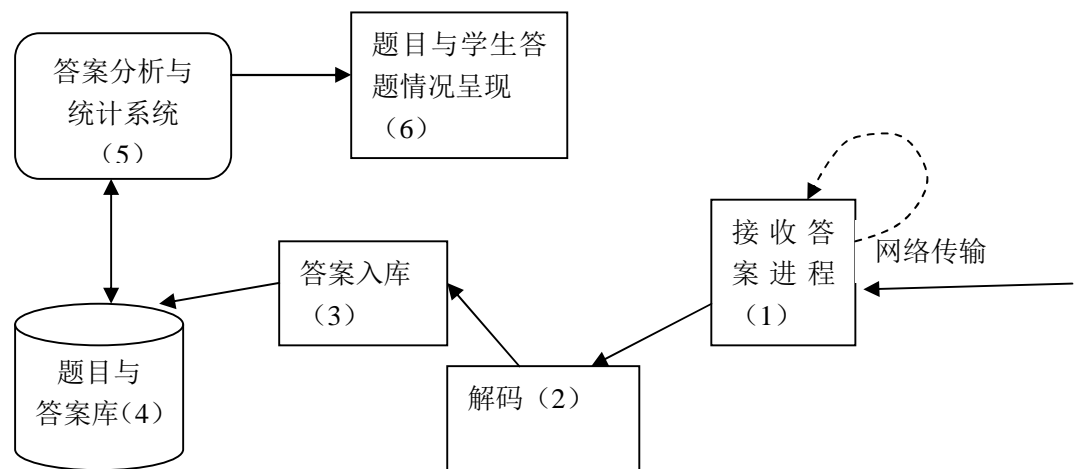


图4-答案接收、分析与呈现的数据流图示意图

- (1) 学生对提出问题进行回答
- (2) 回答之后编码系统对答案编码，上传网络（客户端的终端过程）
- (3) 通过网络传输，接收答案进程接收学生答案，并且可直接接收许多同学的答案
- (4) 接收答案进程接收答案后，对答案进行解码

- (5) 解码后，答案入库
- (6) 答案入库后，与题目与答案库联系并合作，先找出答案所对应的题目，  
再在题目与答案库中关联该题目对应的正确答案，然后与答案库中学生的答案进行对比，看是否正确
- (7) 答案分析与统计系统将所有学生的答题信息都整理出来，并进行归类整理，分析总结
- (8) 题目与学生答题情况呈现，将每个学生的答题情况和总体情况，根据需要进行呈现，

### 4.1.2 非功能分析

- **可靠性**

- 即使硬件出现故障，系统也可以可靠运行吗？
- 复制和故障转移方案是什么？
- 需要手动干预，还是系统可以自动进行故障转移？
- 实现可靠性会对性能造成负面影响吗？
- 实现可靠性的成本有多高？

可靠性需要考虑的一些具体方面是：

- **安全性：**假设攻击者就在外面。  
如何知道系统用户就是他们所声称的，并只让他们访问经过授权的功能？  
如何保护我的系统不受攻击？  
考虑到网络攻击、机器攻击，甚至从您自己的系统内部发起的攻击。
- **事务性：**如何设计系统来保存工作单元的 **ACID** 属性？  
如果在设计中涉及多个独立的子系统(**Web** 服务和 **SOA** 就是这种情况)，则这一点就显得特别重要。不要假设始终可以进行两阶段提交 (**two phase commit**)。

- **可用性**

- 您是否为用户带来不适当的负担(例如，需要特殊的浏览器版本)？
- 系统是否根据模型-视图-控制器 (**Model-View-Controller**) 体系结构设计以使多用户界面成为可能？如果是这样，如何将它们绑定在一起？
- 是否界面本来就有状态而功能无状态（反之亦然）？



- 有效性

- **性能：**这个系统的运行情况有多好？  
它只是平稳缓慢地运行吗？  
系统可以达到其响应时间目标吗？  
应用程序的设计是否符合性能要求？  
您利用缓存了吗？
- **可伸缩性：**如果系统在小范围内运行看起来相当快，那么当扩展至每秒、每分钟或者每小时几千或成千上万个活动的时候呢？  
它的设计是否达到吞吐量目标？  
可以复制系统来实现线性扩展吗？  
是否存在瓶颈（例如公共数据库）？

- 可维护性

您如何配置它？  
如何监视它？  
如果您一件事情需要执行很多次（例如，安装许多应用程序），那么会怎么做呢？  
您是否有一个可复制的部署流程呢？  
您是否可以使重复的任务自动化，使之在大范围内可行呢？

- 可移植性

虽然列在最后，但它并非最不重要。例如：

如何采用标准来提供某种形式的平台中立性呢？  
是否计划将应用程序迁移到您的最新和最高版本的应用服务器上呢？  
如果不打算这样做，则当供应商撤消对该版本的支持时您要怎么做呢？  
如果您的项目基于开放源代码，则也有类似的问题。如果每当某人有个更好的捕鼠器 (mousetrap) 您就必须重写整个应用程序，则没有人会问津。

还有一些分析：

## 1. 系统的完整性

系统的完整性指为完成业务需求和系统正常运行本身要求而必须具有的功能，这些功能往往是用户不能提出的，典型的功能包括联机帮助、数据管理、用户管理、软件发布管理和在线升级等。

## **2. 系统的可扩充性与可维护性**

指系统对技术和业务需求变化的支持能力。当技术变化或业务变化时，不可避免将带来系统的改变。不仅要进行设计实现的修改，甚至要进行产品定义的修改。好的软件设计应在系统架构上考虑能以尽量少的代价适应这种变化，常用的技术有面向对象的分析与设计及设计模式。

## **3. 技术适应性与应用适应性**

系统的适应性与系统的可扩充性和可维护性的概念相似，也表现产品的一种应变能力，但适应性强调的是在不进行系统设计修改的前提下对技术与应用需求的适应能力，软件产品的适应性通常表现为产品的可配置能力。好的产品可能要考虑到运行条件的变化，包括技术条件（网络条件、硬件条件和软件系统平台条件等）的变化和应用方式的变化，如在具体应用中界面的变化、功能的剪裁、不同用户的职责分配和组合等。

（以上非功能分析参考网络上的分析）

## **界面的美观性和人性化**

界面的美观性和人性化也是非功能分析的重要方面。

### **美观性和人性化：**

首先，界面的美观设计和人性化设计可使用户的视觉舒适感增强

其次，很好的视觉效果和人性化设计使用户对软件的喜爱度增高

最后，界面的美观性和人性化设计可以提升产品的品味和档次

## **4.2 客户端的软件需求分析**

客户端作为答题方，用户需要通过服务器端获得所需要的题目，然后再通过客户端进行答题，最后将对问题的回答提交到服务器或在本地显示。

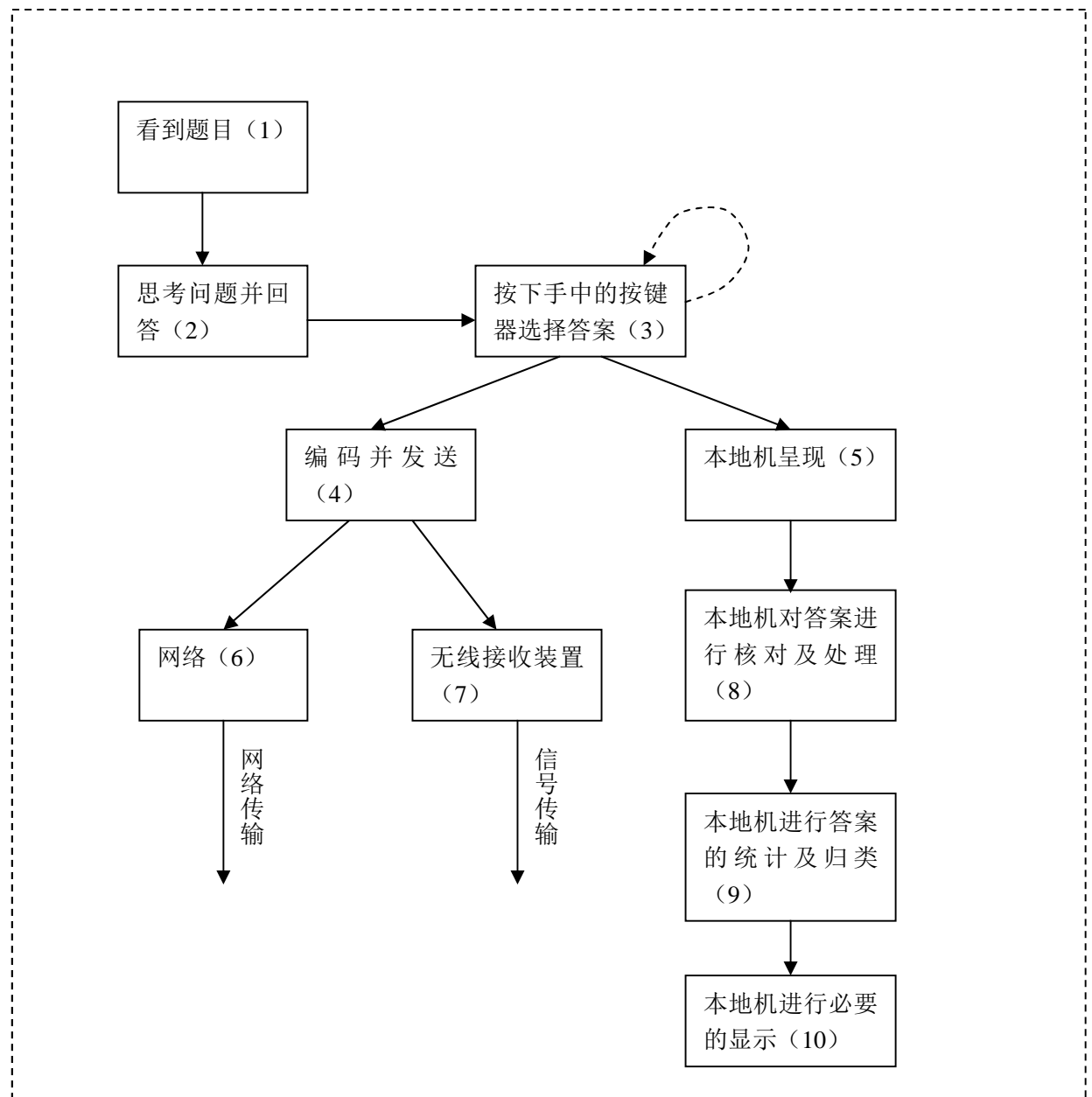
客户端分为：

- 1) 不接收题目，只回答和上传答案

- 2) 接收题目、呈现，并回答和上传答案
- 3) 接收题目、呈现，只回答，不实时上传答案

## 4.2.1 不接收题目，只回答和上传答案

### 4.2.1.1 功能分析



具体步骤:

- (1) 学生看到大屏幕上显示的题目
- (2) 学生对问题进行思考并回答

- (3) 学生按下手中的按键器选择答案
- (4) 客户端将答案编码并发送
- (5) 编码被发送至互联网络或无线接收装置，互联网络对其进行网络传输，无线接收装置对其进行信号传播
- (6) 上传答案的同时也可在本地机上呈现
- (7) 本地机对学生的答案进行核对及处理
- (8) 本地机进行答案的统计及归类
- (9) 本地机对其进行一些必要的显示

#### 4.2.1.2 非功能分析

- 可靠性

事务性：

- 可用性

- 有效性

性能：可伸缩性：

- 可维护性

- 可移植性

还有一些分析：

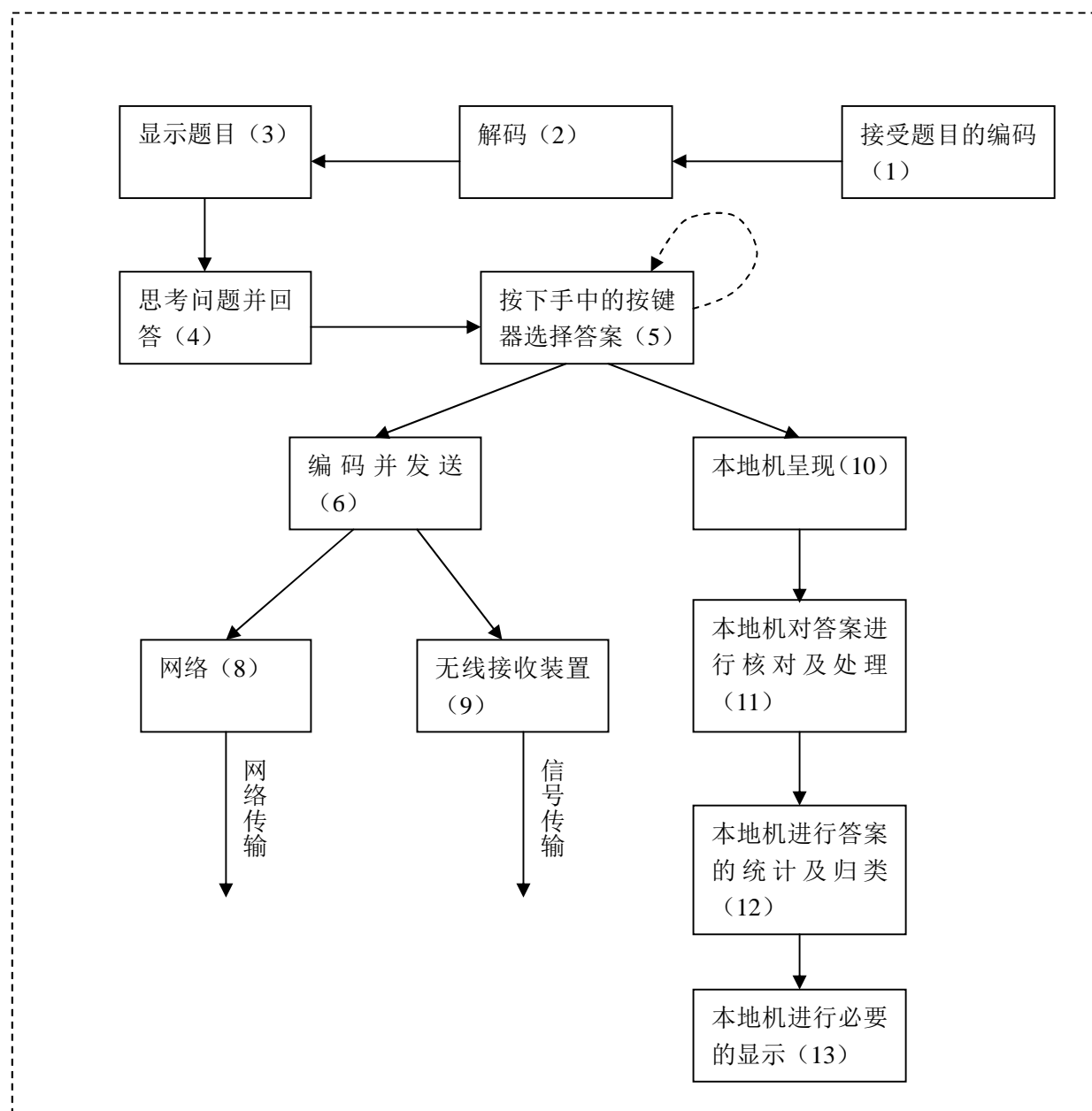
1. 系统的完整性
2. 系统的可扩充性与可维护性
3. 技术适应性与应用适应性

（以上非功能分析参考网络上的分析）

界面的美观性和人性化

## 4.2.2 接收题目、呈现，并回答和上传答案

### 4.2.2.1 功能分析



具体步骤:

- (1) 客户端接收题目的编码
- (2) 客户端对答案进行解码
- (3) 客户端显示题目
- (4) 学生对问题进行思考并回答
- (5) 学生按下手中的按键器选择答案
- (6) 客户端将答案编码并发送
- (7) 编码被发送至互联网络或无线接收装置，互联网络对其进行网络传输，无线接收装置对其进行信号传播
- (8) 上传答案的同时也可在本地机上呈现
- (9) 本地机对学生的答案进行核对及处理
- (10) 本地机进行答案的统计及归类

(11) 本地机对其进行一些必要的显示

#### 4.2.2.2 非功能分析

- 可靠性

可靠性需要考虑的一些具体方面是：

安全性：

事务性：

- 可用性

- 有效性

性能：

可伸缩性：

- 可维护性

- 可移植性

#### 1. 系统的完整性

#### 2. 系统的可扩充性与可维护性

#### 3. 技术适应性与应用适应性

#### 界面的美观性和人性化

界面的美观性和人性化也是非功能分析的重要方面，

##### 美观性

首先，界面的美观设计可使用户的视觉舒适感增强

其次，很好的视觉效果使用户对软件的喜爱度增高

最后，界面的美观性可以提升产品的品味和档次

##### 人性化

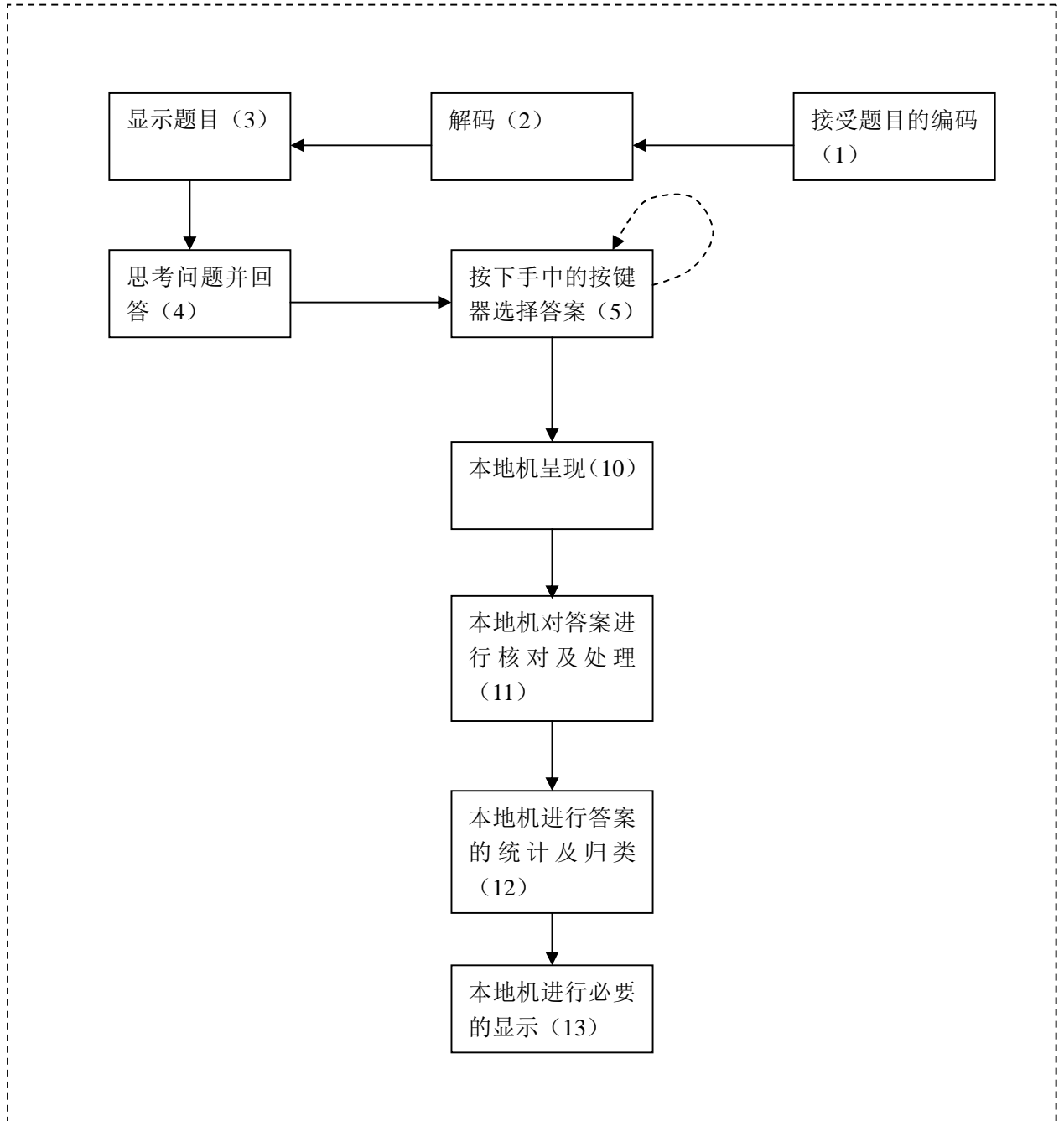
首先，人性化设计可使用户的使用舒适感增强

其次，人性化设计同样能使用户对软件的喜爱度增高

最后，界面的人性化设计同样可以提升产品的品味和档次

### **4.2.3 接收题目、呈现，只回答，不实时上传答案**

#### **4.2.3.1 功能分析**



具体步骤:

- (1) 客户端接收题目的编码
- (2) 客户端对答案进行解码
- (3) 客户端显示题目
- (4) 学生对问题进行思考并回答
- (5) 学生按下手中的按键器选择答案
- (6) 上传答案的在本地机上呈现
- (7) 本地机对学生的答案进行核对及处理
- (8) 本地机进行答案的统计及归类
- (9) 本地机对其进行一些必要的显示



#### 4.2.3.2 非功能分析

- 可靠性

可靠性需要考虑的一些具体方面是：

{ 安全性：  
事务性：

- 可用性

- 有效性

{ 性能：  
可伸缩性：

- 可维护性

- 可移植性

#### 1. 系统的完整性

#### 2. 系统的可扩充性与可维护性

#### 3. 技术适应性与应用适应性

### 界面的美观性和人性化

界面的美观性和人性化也是非功能分析的重要方面

#### 美观性

首先，界面的美观设计可使用户的视觉舒适感增强

其次，很好的视觉效果使用户对软件的喜爱度增高

最后，界面的美观性可以提升产品的品味和档次

#### 人性化

首先，人性化设计可使用户的使用舒适感增强

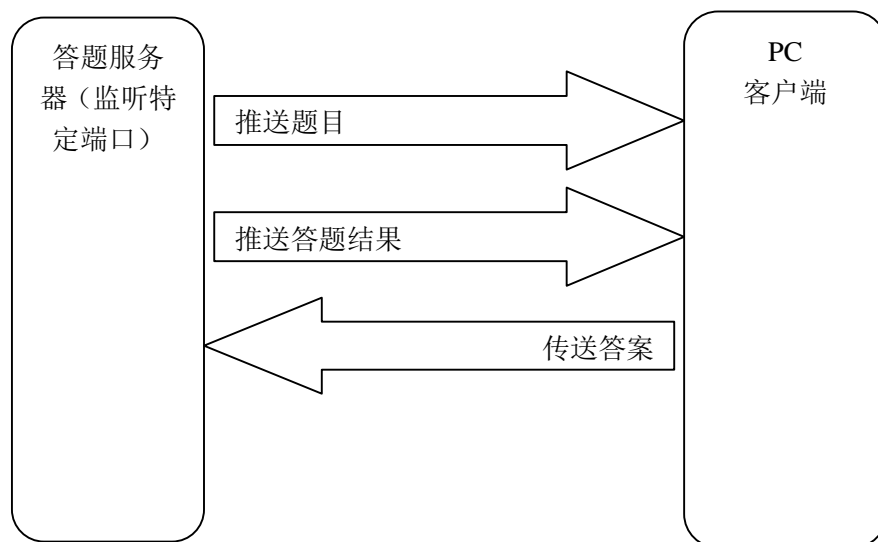
其次，人性化设计同样能使用户对软件的喜爱度增高

最后，界面的人性化设计同样可以提升产品的品味和档次

## 4.3 客户端与服务器端的通信协议

客户端和服务端之间的数据交换，构成了两者之间的通信协议。定义一个这样的协议需要解决：

- 1) 题目的编码的形式：用 ASCII 形式？  
用 Text 形式行吗？  
是传到学生面前的终端 PC 上还是用投影仪？
- 2) 答案的编码：直接用字符(A、B、C、D)行吗？  
还是用 ASCII 码？  
如何区分题目？  
如何区分学生的身份？
- 3) 编码后传输的手段是：TCP/IP？  
套接字？  
端距离无线协议？  
广播协议？



PC 版工作原理

答题终端为 PC 机，该 PC 机装有答题系统的客户端软件，客户端软件负责显示题目和备选答案、采集学生答题信息、传送答题信息、显示答题结果。客户端软件和答题服务器通过 Socket 协议进行通讯。

- (1) 答题服务器（监听特定端口）将题目编码
- (2) 编码后，题目被推送至 PC 客户端
- (3) PC 客户端对题目进行解码
- (4) 用户回答问题
- (5) PC 客户端再将答案编码，传送至答题服务器（监听特定端口）
- (6) 答题服务器（监听特定端口）再对答案进行解码
- (7) 答题服务器（监听特定端口）对答案进行统计及处理
- (8) 答题服务器（监听特定端口）再向 PC 客户端推送答题结果

## 5. 需要注意的问题

**同步：**为了公平公正，凡是带有测试需要给出成绩的情况需要考虑答题时间同步的问题，为此可能需要答题服务器负责客户端何时开始显示题目并接收答案，针对不同网络环境可能需要评估各个客户端到服务器的网络延迟，这种方式在网络游戏中有相关案例。

**身份认证：**特定的场合比如考试需要进一步得知每个答题者的答题情况，便于考试给出成绩。

**重复按键：**在题目提出后，学生因为犹豫不决，因为按错键，因为改正之前的答案，会反复对题目的答案进行按键，应该有一个规定，或者采用学生最初的答案，或者采用学生最后选择的答案，或者采用规定时间内学生回答的最后一个答案。

**信号干扰：**若是选用无线传输方式，或者广播传输方式，则对信号的要求相对较高，但信号容易受到干扰，或者发生静电屏蔽的现象，从而

导致传输出错或传输受阻，所以如何尽最大可能避免信号的干扰也是一个重要问题。

## 6. 约束条件

约束条件规定了实现本系统时需要遵循的原则或其它要求。

- 1) 服务器端用 Windows 平台；
- 2) 数据库：并不意味着用 DBMS，而是希望先用文件系统实现。这样可以摆脱本系统对 DBMS 的要求。也容易将本系统与现有的各种题库系统进行联接。
- 3) 开发环境：服务器端用 VC 6.0，客户端用 C。开发时，尽可能用 ANSI C。客户端也用 C。其目的是便于系统的移植。
- 4) 系统结构采用 C/S 结构。避免用 Web-Service，例如，Tomacat、We-Logic 等软件。其目的是，希望抽象出专门用于答题的服务器端软件。
- 5) 通信协议要自己自订，必须用 C 的 struct 给出数据结构。用 MSC 给出通信过程描述。并要求抽象出通信协议常见的几个操作。通信部分，必须用 ANSI C。

## 参考资料

1. 百度或者 google 搜索关键词：

表决器、votingmachine，非功能分析

- 2.