

基于插件技术的试题多样化解决方案和实现^①

王志瑞, 黄 慧, 刘正涛

(三江学院 计算机科学与工程学院, 南京 210012)

摘 要: 针对在线外语学习平台中试题展示的需要, 研究了插件的基本思想, 将插件的思想应用到试题的设计过程中, 提出了一种基于插件的试题设计与展现方法, 并在 ASP.NET MVC 环境下给出了具体的设计与实现, 并将此方法应用到易立方智能化外语学习平台中. 实践证明该方法能够很好的解决题型多样化问题, 为当前在线学习平台的设计提供了一定的参考价值.

关键词: 插件; 题库建设; 在线英语学习; 局部视图; 反射; ASP.NET MVC

Design and Realization of Plug-in-Based Test diversification

WANG Zhi-Rui, HUANG Hui, LIU Zheng-Tao

(Department of Computer and Information Engineering, Sanjiang University, Nanjing 210012, China)

Abstract: For the sake of test presentation on Online language learning platform, in this paper, the idea of plug was studied and applied to the thought of the design process. Furthermore, on the basis of the plug, we propose a test design and presentation method, which was implemented on the ASP.NET MVC environment, and was successfully applied to foreign language learning easy cubic intelligence platform. Practice has proved that this method can solve questions diversification, providing a certain reference value for the design of the current online learning platform.

Key words: plug-in; exam construction; online English learning; PartialView; reflecting; ASP.NET MVC

随着网络技术的不断进步, 在线学习已成为当今一大趋势. 在线学习也渐渐得到大众的认可, 很多在线学习平台也陆续上线, 也得到了大众的一致好评. 例如猿题库, 梯子网等一些优秀网络学习平台都在新形势下产生, 并且很快得到了上千万融资支持.

外语是中国大众经常会接触到的一种语言工具, 可是当前缺少一个很好的在线外语学习与测试平台, 其中一个主要原因是外语的试题种类繁多, 展示方式多样化, 四六级考试时常会出现一些新的题型, 外语教材中也会用到很多不同的题型, 在早期的题库系统研究中, 在试题展示方面只提供几种简单题型的处理, 往往达不到展示所有试题的需求, 针对这个问题, 本文提出了一种基于插件机制的题型设计思想, 针对不同的题型设计不同的题型插件, 在插件中完成特定题型的处理工作, 在试卷展示时, 能够根据试卷所拥有的题型信息, 自动加载对应的题型插件, 通过对应的

题型插件来完成题型的个性化展示设计, 能够很好的应对当前外语试题种类繁多, 展示多样化问题.

1 基于插件的软件框架

随着面向对象程序设计技术的发展, 为了解决软件的灵活性与可扩展性, 提出了基于“框架+插件”的软件设计模型^[1,2].

在“框架+插件”的软件模型中, 框架是软件的主体, 可以独立于插件独立运行, 框架在运行时根据配置信息动态的载入相应的插件, 并在运行过程中负责插件的加载和运行管理, 是插件的宿主和运行容器.

在框架设计阶段要预先定义好框架与插件之间的交互接口规范, 在设计插件时, 插件需要遵循框架与插件之间的交互接口, 框架能够根据交互接口很方便实现插件的加载与调用, 实现框架功能的动态扩充^[5].

插件的定义就是通过实现主框架交互接口, 让主

^① 收稿时间:2014-12-22;收到修改稿时间:2015-03-18

框架可以按照统一的方式来实现对各个不同插件的调用, 这样主框架就可以不必关注各插件的细节, 简化了系统集成的难度, 增加了系统的可扩展性和灵活性^[9].

“框架+插件”的软件设计模型能够很好的适应在线外语学习平台中试题展示的需要, 在框架中实现题库的统一管理和维护工作, 通过插件实现各种题型的个性化展示, 通过添加插件的方式丰富题库系统中支持的题型的种类, 能够很好的应对在线外语学习平台中试题的管理与展示的需求.

2 在线英语学习平台的设计

2.1 系统结构设计

在线英语学习平台是一个综合的外语学习平台, 其中包括套题练习, 语法专项练习, 学生班级管理, 学生成绩管理等很多的功能. 但是其系统框架主要由如图 1 所示结构构成, 可以根据系统的功能需求进行扩展.

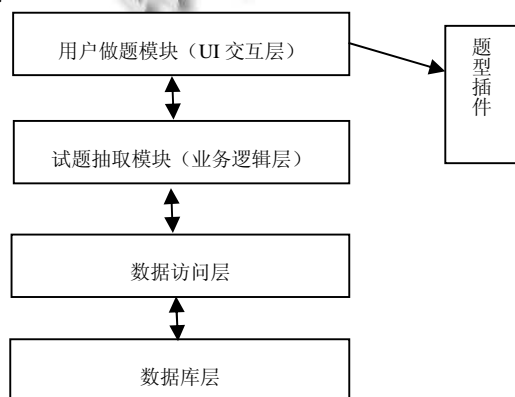


图 1 系统结构图

1) 数据库层是学习平台的底层, 存储系统正常运行所需要的各类数据(主要有用户信息, 试题相关信息, 学习记录相关信息等).

2) 数据访问层: 主要完成对系统试题库的访问, 为上层的业务逻辑层提供统一的数据存储接口;

3) 试题抽取模块也称为业务逻辑层: 主要完成试题的获取与保存, 根据系统功能需求, 可以分为抽取整套试卷, 随机抽取试题, 随机抽取专项练习, 获取历史错题等相关的业务逻辑, 为 UI 交互层提供业务数据的获取.

4) 用户做题模块也称为 UI 交互层, 是与用户交互的接口, 为了解决各种题型展示的多样性, 此层引

入了题型插件机制, 利用试题的类型自动加载对应的题型插件, 实现了试题展示的多样化.

2.2 系统数据库设计

在线外语学习平台中外语试题是系统的核心数据, 为了能够合理方便的存储各种外语试题, 需要对系统数据库进行设计, 设计的基本目标是能够适应各种层次, 各种题型和套题的保存, 便于业务逻辑层能够很方便的抽取套题或者随机抽取试题^[6-8]. 基于上述目标, 我们设计数据库的原则是数据库尽量简单, 便于存储, 对于试题内容的解析交由业务逻辑层处理. 系统数据库关系结构如图 2 所示.

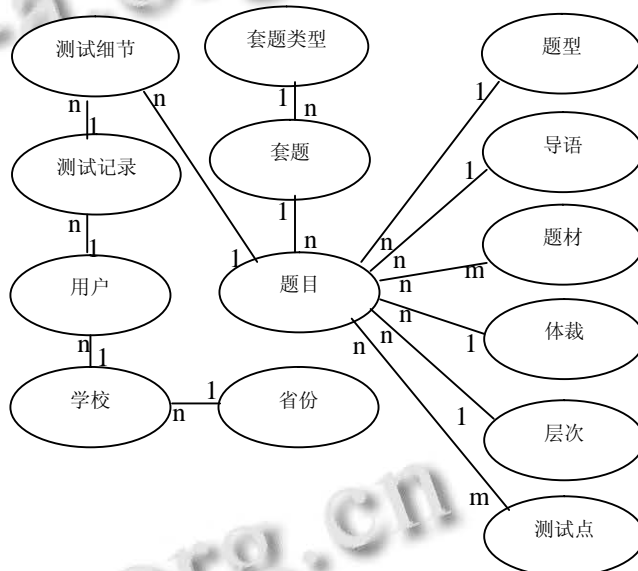


图 2 系统数据库关系图

从图 2 可以看出, 系统数据库主要由用户库, 试题库, 测试记录库组成. 用户库用来保存用户的基本信息, 便于后期对平台内的用户进行分析; 试题库用来保存系统中相关的试题信息, 并从题型、层次、测试点、题材、体裁等多个角度对试题进行标注, 便于在试题抽取过程中能够更加方便的完成试题的多角度筛选任务; 测试记录库用来保存用户的学习记录, 通过测试记录能够完成对用户的能力水平, 学习情况进行分析, 并可根据分析结果来为用户提供更加适合其学习的学习内容.

3 在线英语学习平台的实现

本平台采用 ASP.NET MVC 4.0+SqlServer 实现,

MVC^[4]全名是 Model View Controller, 是一种软件设计模型, 是一种将业务逻辑、数据、界面显示逻辑分离的方法组织代码, 将业务逻辑聚集到一个部件里面, 在改进和个性化定制界面及用户交互的同时, 不需要重新编写业务逻辑, 广泛应用与有 UI 交互的程序设计过程中^[4].

在 MVC 中 Model 称为数据模型, 要求将系统所用的数据库转化为模型数据, 即进行 O/R Mapping, 就是对象关系映射, 本文采用 EntityFramework^[3]框架实现数据模型层.

在 MVC 中 Controller 是应用程序中处理用户请求的部分, 通过 Action 方法处理用户的每次请求, 并获取对应的模型数据来渲染对应的 View 或者 PartialView^[4]来回应用户的请求.

在 MVC 中 View 是应用程序用来处理数据展示的部分, ASP.NET MVC4 中的视图支持 Razor^[4]表达式, 能够在 View 层嵌入后台的代码, 能够更加方便的控制模型数据的展示, 通过 Razor 表达式能够在 View 中嵌入 PartialView, 能够减少代码重复, 提高页面代码重用率. 本文在 ASP.NET MVC 框架下借助 PartialView 实现插件机制, 具体的工作原理示意图如图 3 所示.

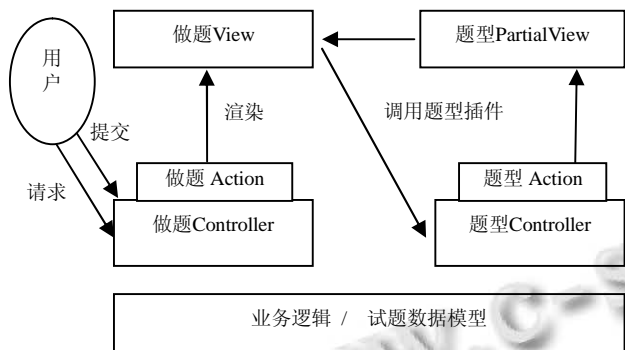


图 3 插件工作原理示意图

具体含义如下: 1) 用户通过浏览器请求“做题 Controller”内的“做题 Action”方法, 此方法根据用户做题的类型, 利用“业务逻辑”和“试题数据模型”查询出需要练习的试题数据, 并将试题数据渲染到“做题 View”中.

2) 在“做题 View”中通过 Razor 表达式, 构建试题数据的遍历处理, 对试题数据内每一道试题进行处理, 根据每个试题的类型信息去调用“题型 Controller”内对应的“题型 Action”方法.

3) “题型 Controller”内的“题型 Action”方法根据具体的题型和传递进来的试题数据, 对试题数据进行相关的数据解析处理, 处理结束后利用处理结果渲染“题型 PartialView”, 并将“题型 PartialView”的渲染结果嵌入到“做题 View”中. 所有试题遍历结束后, 即可完成“做题 View”的渲染工作.

4) 在做题结果提交阶段, 由于不同题型的展示方式不同, 为了能够统一处理“做题 View”的数据提交工作, 本文在“题型 PartialView”中采用 JQuery 代码获取对应题型的用户作答结果, 并保存到指定的隐藏表单中, 在试题提交时, 获取指定的隐藏表单信息, 即可获取试题的作答结果.

3.1 试题数据模型的创建

在 ASP.NET MVC 中 View 层支持强类型视图, 能够将后台的业务对象声明为某个 View 的视图模型, 在 View 层的 Razor 代码中支持模型属性的代码提示, 提高编码效率和减少出错几率. 本文所创建的数据模型采用微软新一代数据访问组件 EntityFramework^[3]实现, 此框架能够根据数据库内表结构信息, 自动创建对应的数据模型类, 根据表字段信息, 生成类的成员变量, 根据表之间的关系, 生成类之间的关联关系. .Net 框架下提供了 Linq 查询表达式, 能够对所生成的模型数据进行相关的数据查询操作, 以集合的方式返回查询结果, 并提供了以操作对象的方式操作数据库数据的途径, 简化了数据库的相关操作.

3.2 题型插件的实现

根据插件的思想, 在 ASP.NET MVC 环境下, 本文借助局部视图^[4]的方式来实现插件机制, 在“做题 View”中通过动态的加载局部视图的方式来实现插件机制.

局部视图是 ASP.NET MVC 中页面设计所提供的一项视图技术, 它允许程序员将视图的一部分分离出来, 在某些需要的页面引用, 能够减少代码重复, 提高页面代码重用率.

局部视图有两种方式, 本文采用 Controller, Action 和 PartialView 的方式, 首先创建一个题型插件 Controller, 在 Controller 内为每一种试题类型添加 Action 方法(方法名建议与题型标示之间有一定联系, 能够利用题型标示生成方法名, 便于题型与局部视图之间的对应), 并为此 Action 方法添加对应的 PartialView, 一个 Action 方法和 PartialView 负责一种

题型的处理, 在 Action 内为对应题型添加所需要的相应的数据处理代码, 在 PartialView 页面内完成对应题型所需要的 UI 展示设计工作。

3.3 做题模块的实现

题型插件构建完毕后, 就可以构建做题模块, 做题分为套题练习、专项练习以及其它相关的练习, 无论哪一种练习, 区别在于业务逻辑中试题的获取方法, 试题的展示部分基本相同。本文以套题练习为例介绍实现方法。

① 添加做题模块所需要的 Controller, 在 Controller 内添加获取套题试题所需的 Action 方法, 在 Action 方法通过 Linq 查询表达式获取所要练习的某套题内所有的试题信息, 并以试题集合的方式返回; 在 Controller 内增加表单提交处理 Action 方法, 方法内获取所有试题的标识和作答结果。

② 为获取套题试题的 Action 方法添加强类型 View(以试题模型作为其强类型数据模型)。

//声明强类型视图模型

@model List<ylfLib.Models.TestLibrary>

③ 在 View 内, 利用 Razor 表达式构建表单, 设定表单的提交处理方法, 并在表单中构建套题试题的遍历处理, 在遍历过程中根据试题的题型信息去加载对应的题型插件, 加载插件的代码如下:

//构建表单, 并设定提交信息

//Exam 控制器内的 submit 方法

```
@using (Html.BeginForm("submit", "Exam")) {
    //遍历所有试题信息
```

```
@foreach (TestLibrary item in Model)
```

```
{
```

```
    //利用 Html.RenderAction 方法去加载试题
```

```
    //所对应的题型插件, 并传递试题数据
```

```
<div>@{Html.RenderAction(item.TestType,
    "QuestionType", new { model = item });}</div>
```

```
}
```

```
}
```

4 结语

本文所提出的设计方案现已应用与易立方公司开发的校园版在线外语学习平台中, 现在该系统已处于初步销售阶段, 开始在部分高校内进行试售阶段, 该平台中包括了组卷, 布置作业, 做作业, 练习等与大学外语教学有关的相关功能, 为老师的教学和学生的学习提供帮助, 具体运行效果如图 4 所示。现在正在将此设计方案应用与互联网版本中, 利用此方案来完成互联网环境下的在线外语自主学习平台的设计与实现, 为互联网用户提供自主的外语学习与测试平台。

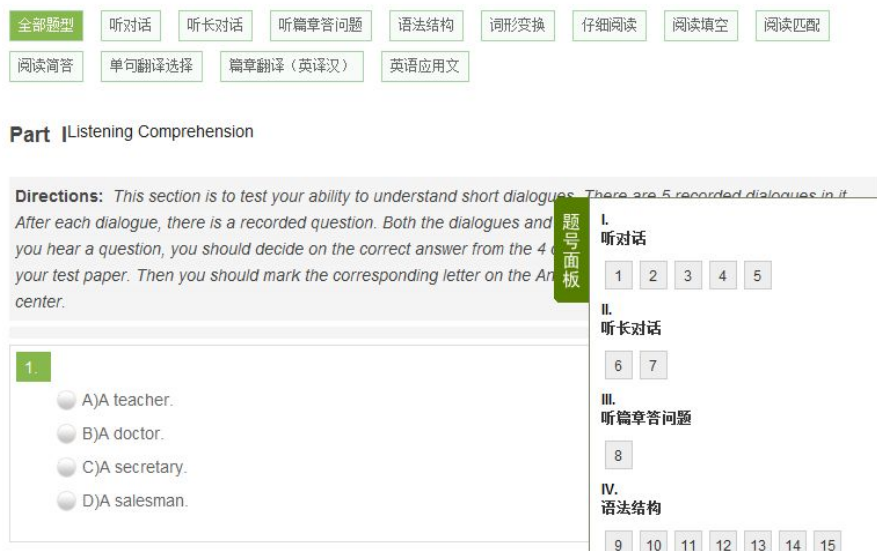


图 4 套题做题界面效果

参考文献

- 1 陈云云, 马捷中, 蒋泽军. 基于插件的组态软件设计与实现. 计算机测量与控制, 2013, 21(2): 436-438.
- 2 刘晓丽, 孙伟, 赵占杰, 等. 一种基于插件结构的可扩展 WPS

- 服务平台.测绘科学,2013,38(3):187-189.
- 3 Jennings R. Professional ADO.NET 3.5 With LINQ and Entity Framework. Wiley Press, 2009.
- 4 蒋金楠.ASP.NET MVC 4 框架揭秘.北京:电子工业出版社, 2013.
- 5 王向前.基于插件的应用集成框架研究.科技信息,2014,(4): 154-155.
- 6 李观发,卢来发,李佳林.基于.NET 的计算机课程在线测试系统的设计与研究.智能计算机与应用,2012,2(6): 80-82.
- 7 吴敏宁.在线考试题库管理子系统的设计与实现.电子设计工程,2013,21(8):4-5,9.
- 8 何文广,王耀民,曾华.基于题库共享的在线考试系统设计与实现.软件导刊,2012,11(6):70-72.
- 9 郑响萍.基于插件的分布式监控平台的设计.硅谷,2012,(17): 54-55.

WWW.C-S-A.ORG.CN

WWW.C-S-A.ORG.CN