A New Development Architecture for E-Commerce Platform

Longjun Huang 1, Caifying Zhou 2, and Yuanwang Wei 3
1Software of Software, JiangXi Normal University, NanChang, China
Huanglong@jxnu.edu.cn
2Science & Technology Division, JiangXi University of Science and Technology, GanZhou, China
Zhoucaiying_007@mail.jnust.cn
3Department of Computer, JiaXing University, JiaXing, ZeJiang, China
yuanwang_wei@163.com

Abstract—E-commerce is a kind of commercial activity which adopts electronic form under the condition of open Internet. It has become an important life style. The diversity and variability of e-commerce activities means a great challenge to software development. How to design and develop a flexible and reusable e-commerce platform has become a direction that e-commerce industry is heading to. Based on the three-tier architecture, a new e-commerce platform development framework-seven layer architecture has been given in this paper. It describes the design ideas and the concrete realization of each layer. Practice shows that the development architecture proposed here can meet the needs of the diverse and fast-changing e-commerce business. Therefore, it's a feasible solution.

Index Terms—E-commerce; Seven layer architecture; Page structure layer; Bookstore

I. INTRODUCTION

Nowadays, to conduct transactions through e-commerce platform has taken a great role in modern society. E-commerce have entered into all aspects of society, it has a variety of requirements which are easy to change from users, demands different implementation approach, even different deployment platform. Traditional developing methods are not competent. Even if it is realized, the high degree of system coupling, is not easy to change and difficult to maintenance. There are still some other problems which make it far from being able to meet the needs of e-commerce field. It has proposed three-tier architecture which divided the entire business applications into three layer: presentation layer (UI), business logic layer (BLL), Data Access Layer (DAL). The three-tier architecture fully reflects the "high cohesion, low coupling" thinking and it solve many problems encountered in the process of complex system development [1]. Although this architecture improve the encapsulation and maintainability of the project, but the encapsulation still not go far enough, each layer still have some content which could be separated and the integrity of the system is inadequate too. Based on the above analysis, a new development architecture - seven layer architecture has been proposed to respond the e-commerce business features. This paper discusses the system framework, the main layer’s design of this architecture, and gives the implementation of an online bookstore based on this architecture.

II. THREE TIER ARCHITECTURE[2][3]

Layered application designs are extremely popular because they increase application performance, scalability, flexibility, code reuse, and other benefits. In the classic three tier design, applications break down into three major areas of functionality: the data access layer, the business layer and the presentation layer. Inside each tier there may also exists a series of sub-layers that provide an even more granular breaking up the functional areas of the application.

Data access layer (DAL): the data access layer manages the physical storage and retrieval of data. In short words, it is to perform Select, Insert, Update, and Delete of the data table.

Business logic layer (BLL): As the essential part of the whole system, it maintains business rules and logic. The relevant design of business logic tier is associated with the unique logic of e-commerce, such as merchandise inquiries, making orders; adding merchandise to shopping carts etc. If it involves access to databases, then transfer the data access layer.

Presentation layer: the presentation layer houses the user interface and related presentation code. In this layer, ideal system status should not include the business logic. The logic codes in the presentation layer are only relating to the interface elements.

III. SEVEN LAYER ARCHITECTURE

A. Overview

1) Background

Based on three-tier architecture, we make the data access layer and the presentation layer into detailed layers. The DAL is divided into physical data layer, data access layer and entity layer. The UI is divided into page structure layer, logical control layer, and page presentation layer. This layered approach is an optimum approach for three-tier structure. By using this developing model, developers can only concerned with on one layer in the whole system, any layer’s implementation can easily be replaced with a new realization, and the dependence
between layers is reduced. All this features will conduce to standardization and the reuse of each layer.

2) The Definition of Each Layer

Physical data layer: it includes business modeling and database design.

Data access layer: it includes the design and implementation of data access components.

Entity layer: the mapping between objects and data

Business logic layer (BLL): As the essential part of the whole system, it maintains business rules and logic based on user requirements, includes adding, deleting, updating, and selecting methods which are the guarantee of finishing the system.

Page structure layer: it includes page element such as HTML, but not the style which is described in page presentation layer.

Logical control layer: it control the page’s content by dominating the page structure layer, it access the business logic layer.

Page presentation layer: this layer use div+css to meet the user’s requirements of the page surface, it separate from page structure layer.

The overall structure and relations between each layer are showed at figure 1:

![Figure 1. The Overall Structure of Seven Layer Architecture](image)

3) Signification

By using the seven layer architecture, if errors occurred in the system, we can easily find the layer where the error occurred, and can quickly find the exact location of the error in the layer and then amend it, because the encapsulation degree is very high in our architecture. The flexibility of seven layer architecture is reflected in the reusability, scalability and maintainability.

B. Physics Data Layer

In this layer, it mainly includes the establishment of basic data tables, the realization of data integrity, the encapsulation of complex business logic. This layer is competed by systems analysts, database administrators, and experts in the field together.

C. Data Access Layer

A Data Access Layer (DAL) is a layer of a computer program which provides simplified access to data stored in persistent storage of such kind, such as an entity-relational database. For example, instead of using commands such as insert, delete, and update to access a specific table in a database, a class and a few stored procedures could be created in the database. The procedures would be called from a method inside the class, which would return an object containing the requested values. Or, the insert, delete and update commands could be executed within simple functions like register user or login user stored within the data access layer [4].

D. Entity Layer

Object-oriented thinking is now generally used for development, while the data is often stored in a relational database. The physical layer’s main purpose is to resolve this contradiction, it maple relational date into objects. Often the entities are generated by code generator. It is a mistake that the entities are just the mapping tables. If the domain model is considered, the entities will different from tables and will be solved by mapping. As domain model, entities should consider more about domain itself.

E. Business Logic Layer

As the essential part of the whole system, business logic tier maintains business rules and logic. The relevant design of business logic tier is associated with the unique logic of e-commerce, such as merchandise inquiries, making orders; adding merchandise to shopping carts etc. If it involves access to databases, then transfer the data access layer.

F. Page Structure Layer

The traditional page structure and appearance are mixed together. If you want to modify the appearance of the page, the workload will become very large. This page is not in conformity with maintainability. So in the page structure layer we just define the page level, put all the required data elements on the page by <div> tags and without considering the decoration of the page. Although the page appears very stiff and monotonous, however, this way makes it very easy to produce and change page’s appearance by using style.

G. Logic Control Layer

Logic control layer is about the nexus. On one hand, it controls the page structure layer which aim is to collect data; on the other hand, it uses the business methods provided by BLL and makes the logic embodied in page structure layer, aims to load data.

H. Page Presentation Layer

The advantages of Div+css layout are being able to greatly improve the maintainability of the page by separating the page structure layer and the page presentation layer; this is the benefits of xhtml too. When the customer requirements for the appearance of the page change, you can easily make changes to the css part, thus changing the overall effect of the page, but the page itself need not to be changed.

IV. ADVANTAGES

Both seven-tier and three-tier architecture are based on hierarchical design ideas. They fully reflect the "high
cohesion, low coupling” thinking, solve many problems encountered in the process of complex system development and improve the encapsulation and maintainability of the project. Compared to three-tier architecture, seven-tier architecture divides the data access layer and the presentation layer into more detailed layers. In this architecture, the page presentation designer and the business designer are separated; the DBA and the programmer are separated too. By reducing the dependence between layers, it will conducive to standardization and the reuse of each layer and so improve the efficiency of system development and maintenance.

V. APPLICATION

An online bookstore is developed based on seven layer architecture mentioned above; it has following functions [5]:

- Book Management: This module mainly includes book category management; book adding, modifying, deleting, inquiring; book inventory management, book comment browser as well as other management functions.
- Order Management: This module mainly includes the purchase of books, order generation, automatic delivery and orders modification. Here the order generation is the kernel module of this system, which put the information acquisition and processed data into database after every order instantiation. It aims at tracking each transaction.
- Customer Management: It is the pivotal step in designing an online shopping website, and mainly used to manage the general information as well as the status of the vast number of registered members by putting their information into database. After registration, users can log in to shopping under their name and password on websites.
- Forum Management: It is very important for a good e-commerce website to provide customers with effective ways of communication. This book store online system has designed a full-fledged forum for users to communicate.
- Decision-making support: Most of e-commerce websites only provide online transactions functions. However, in this system, users are provided with a wealth of functions assisting decision-making, and much useful information useful to making decisions can be achieved through the analysis of many sales databases, such as the charts of sales of books, the shopping tendencies of specific customer base etc.
- System Maintenance: This module is mainly used to do the maintenance of some general information of the users and manage users’ privileges and database security in order to ensure the safety of the system by preventing unauthorized users’ from causing damage to the program intentionally or unintentionally[6].

VI. THE IMPLEMENT OF EACH LAYER

A. The Realization of Physical Data Layer

Our system design a total of 21 tables, including user, books, sales, rights management etc, in addition to a large number of stored procedures, views, and so on.

B. The Achievement of Data Access Layer

This layer is aimed to encapsulate the access to the database. According to the interface-oriented programming ideas, we define the data access interface at data access layer, the upper layers program to interface and transparency to physical database. The project at this stage just uses SQL server2005, so it achieved data access interface class only for SQL server as showed at fig 2.

Figure 2. The design of Data Access Layer

Data providers which are encapsulated in this project are much simpler to native data providers-ADO.Net. A large number of database operations are encapsulated in the base class. The programmer just call method in this class, this practice reflects factory pattern of design pattern. Encapsulation not only reduces the number of codes, but also reduces the probability of making errors.

C. The Implement of Entity Layer

As the bridge between relational database and objects, entity plays as a data transmission carrier at each layer. In the entity’s constructor function, we use PropertyInfo technology in reflection to construct entities. When constructing, it receive a parameter (DataTable), then traverse the columns in the table and the attributes in the entity, if them equals, give the column’s value to the attribute automatically, so the duplication of code is reduced. This model not only improves the working efficiency, but also reduces the chance of error. The following is the main code for base class[7].

```c
public Base(System.Data.DataTable table, int rowindex)
{
    // Getting the attributes of the object
    System.Reflection.PropertyInfo[] pinfo = this.GetType().GetProperties();
    //Looping the columns
    for (int i = 0; i < table.Columns.Count; i++)
    {
        // Finding the attribute which has same name and type with the columns
        for (int j = 0; j < pinfo.Length; j++)
        {
            if (table.Columns[i].ColumnName == pinfo[j].Name & table.Columns[i].DataType == pinfo[j].PropertyType)
            {
                if(Convert.IsDBNull(table.Rows[rowindex][i]) == false)
                {

```
Giving the column’s value to the attribute which has same name and type.

... pinfo[j].SetValue(this, table.Rows[rowindex][i], null);

} -----

D. The Implement of BLL

Business Logic Layer is undoubtedly the core part in the system architecture. It mainly focuses on the system design such as the formulation of business rules, the implement of business flows and so on which are related to business requirement. In other words, it is concerned with the system’s domain logic. In this system, business logic layer in every module is encapsulated in a subproject- Components. The web layer calls the approach in business logic layer to achieve the appropriate action including basic adding, modifying, deleting, and a variety of query operations.

E. The Achievement of User Interface Layer

In this project, the user interface layer is divided into page structure layer, page, page presentation layer and logical control layer. By the separation between presenting layer and structure layer, the project have scalability, maintainability. Page structure layer is described by html, logic control layer is realized by the asp.net file and the page presentation layer is achieved by CSS.

VII. CONCLUSION

E-commerce is booming and more attention will be paid to the response speed of website, the security of customers’ data, the stability and cost of system operating by future e-commerce enterprise class. It is very important to choose a flexible developing architecture when set up e-commerce platform.

By applying the tiered technical design system, the seven layer architecture under discussion in this paper enjoys a rigid structure, clear logic and small coupling between layers. This architecture can well adapt to system maintenance, scalability. An online bookstore has been developed by using the seven layer architecture and works well. Practice has proved that it is a workable architecture.

ACKNOWLEDGMENT

We are grateful to Professor Huang Minghe for his help to our work. We especially want to express our gratitude to the authors of the referenced papers. In many respects, this paper represents the blending of insights gained from their research work.

REFERENCES