

# The Study of Reusable WEB Information Inquiry System Design Framework

Yang Ling-xing Qujing Normal University  
College of Physics and Electronic Engineering  
Qujing Normal University  
Qujing, China  
e-mail: 84759653@qq.com

Xia Wen-xian Qujing Normal University  
School of Humanities  
Qujing Normal University  
Qujing, China  
e-mail: 549065819@qq.com

**Abstract**—As the WEB information inquiry system application is getting wider and its requirements are getting more complicated, this paper proposes a design framework for reusable WEB information inquiry system. It adopts the advanced and mature design methods and technologies sufficiently, such as XML, XSLT, Web Service and Enterprise Library. This framework is able to realize the inquiry of heterogeneous data, the automatic analysis of business types, the unified access to data, the sharing of condition search interface and the display of search results interface. Based on this, the developers are able to exploit the WEB information inquiry system that can satisfy the clients' need with some simple configurations and a few code. With high reusability, When the frame is applied to inquiry system, it can improve the development efficiency and decrease the maintain cost. Due to its high expansibility and transportability, the development and maintain work of the WEB information inquiry system would be sharply reduced.

**Keywords**—component; Design Mode; Heterogeneous Data Access; WEB Services; Business Analysis Components; Storage Procedure

## I. INTRODUCTION

All With the development of network technology, all sectors of society establish their own information systems. In these systems, the requirements are becoming larger and complicated, and the requirement contents are constantly changing. Therefore, the system design quality affects the user satisfaction directly. It is essential for designers to design the information system framework and satisfy users diversity and variety requirements without transforming the system structure. Meanwhile, based on this framework, developers can establish the different inquiry systems rapidly, reduce the difficulty of system development, and improve the system development efficiency. In addition, the developed system is easy to extend and maintain, and the software components can be made full use of, these are important contents of the software engineering study[1]. Given this, it takes advantage of mature design methods and advanced technology, and base on .NET Framework, we present a common and reusable inquiry search framework, and it has been successfully applied to the population

information inquiry system in Yunnan province . It has been proved that it is convenient to apply this framework to other information inquiry system and the developers can satisfy the clients' needs with some simple configurations and a few code.

## II. FRAMEWORK STRUCTURE

The designers adopt the concept of distributed query so that the framework system has high reusability, and can meet the heterogeneous data inquiry, the characteristics of diversity and variety of inquiry and distributed query. The designed framework has high cohesion, low coupling between each module, and it's easy to modify and expand, without interaction between the layers. It follows that the framework structure is shown in Figure 1.

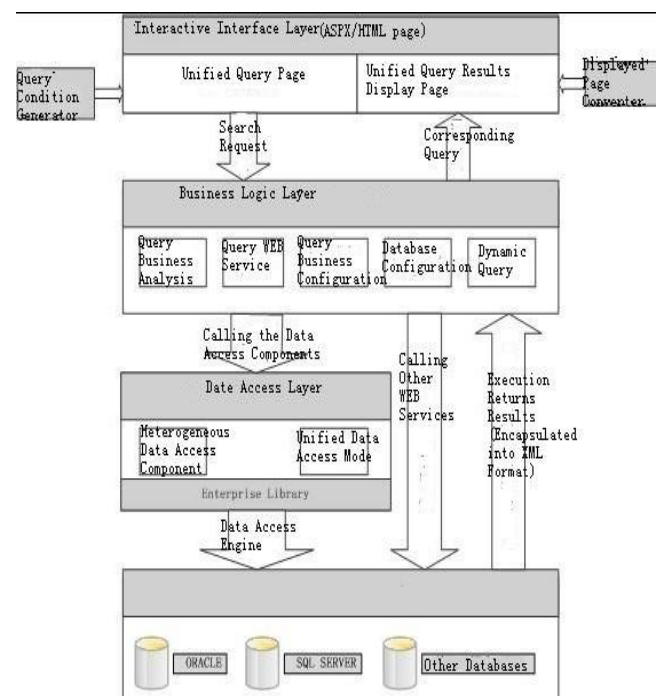


Figure 1. Structure Figure of Query System Framework System

From Figure 1, we can perceive that the hierarchical design of framework structure is divided into interactive

interface layer, business logic layer, data access layer and database layer. The call relationship and approach between the designed layers are different from the traditional ones. In order to improve the reusability, every layer takes the particular design mode. The details are as follows.

#### *A. Interactive Interface Layer*

The interaction with end users is implemented by interactive interface layer. Users input the query conditions and check the query results satisfying the conditions. There are some ASPX and HTML files in this layer and the data operation and storage is encapsulated. It is transparent to users. This design pays attention to the mass traffic and various inquiry conditions. Providing that, we design an inquiry page each inquiry, there will be a large number of pages. With bad reusability, this kind of design mode is hard to manage. The solution is to share the same inquiry condition page, and different conditions are created by generators that dynamically inject them into query condition pages. If there are new query needs or changes, just amend the configuration files of query conditions[2]. In this way, the reusability of query condition generation is realized.

Similarly, this layer shows the query results to users using the same concepts as query condition generating. As the query results are encapsulated into the customized XML format, the displayed page converter transforms the concerning contents into the needed page files and show them. According to users' specific needs, it can generate the display style of theme. Only one page file is needed. The display content is generated by page converter. Input query result display page dynamically and realize the reusability of display query results.

The reusability of interactive interface layer has no relationship with specific query system and inquiry business. Whatever the system and inquiry need are, the query conditions generators and displayed page converter can be reused. In this way, the similar information system can be constructed fast and the efficiency can be improved. It is very easy to maintain.

#### *B. Business Logic Layer*

Business logic layer fulfills the request of the specific inquiry business. It releases the logic function by components or web service. To any information system, business logic layer is the most complex and most difficult to reuse. By abstracting the information system, we can see that it can be reused to some extent. Data operation of information system is nothing more than the addition, modification, deletion and query statistics of data. Although the operating objects are diverse, the operating modes are similar. Therefore, with object-oriented programming idea, some operation base classes are designed[3]. The specific business operation is inherited from these base classes. They only need to realize their peculiar operation function and put the common data operation into the base classes, and solve the problem of the reusability of business logic layer.

In this design mode of query system, base class is devised aiming to every operation. The operational specific business is abstracted into an object. In order to invoke easily, all SQL sentence are stored in one side of data server in the form of storage procedure or function. The validity of data is verified uniformly in interactive interface layer and background. One query system merely needs to make a classification validation. It is unnecessary for every specific business data to verify solely. Logic business, database and user interface interaction are relatively separated, thus, they are easy to modify and expand[4].

#### *C. Data Access Layer*

Data access layer realizes the access operation of database. In practical operation system, the types of database may be different and they are distributed in different places. In order to access to these data uniformly, the access component of heterogeneous data is devised. The component masks all the access details and interaction of database, ensuring the transparency of data access. Moreover, we comprehensively integrate related systems or information application. Providing that other information systems provide interface or web services, the query can be completed by calling these service when establishing the query system. It is unnecessary to do the repetitive work. Hence, the framework indicates this demand. The goal is to implement the query system efficiently, rapidly and to establish a strong system.

The operation of data access is shown as previously mentioned. The SQL sentences operated by specific business data are put into the data server. The specific operation of data access only demands to call storage procedure, and input or output the relatively data set. Of course, parameter numbers and types of every specific query requirement are different. Passing parameter dynamically can be realized by .NET object-oriented technology. Without considering the parameter numbers and types, the process of calling storage is abstracted to unified operation and be designed as a base class. The specific data access inherits this base class completing the needed functions. Every storage procedure name and parameter type of specific query are configured in a XML file. In implementing, the relative data is read in configuration files. What's more, the specific query demands would be gained from the business analysis in business logic layer and they can be employed during the whole query procedure[5].

This design introduces Microsoft's Enterprise Library according to the aspects of system performance and scalability, and the solution to enterprise-class applications. It is quite useful and helpful for enterprise application development. There is no need to introduce the knowledge of Microsoft enterprise library in details. Microsoft's development document, <http://msdn.microsoft.com/en-us/library/ff648951.aspx>, can serve as references to destination and application details of enterprise library.

### III. THE INTRODUCTION OF THE GORDIAN TECHNIQUES

#### A. The Query Condition Generator

The objective to design this component is to improve development efficiency. The analysis reveals that the query system has large query demand, frequent use, large data of query business and complex query condition. In consideration of it, the framework designs query condition generator specifically. The component uses C# language, which supports the current common data types. The control type that the query condition needed is configured in an XML file in a certain format. When parsing a query to a specific business, it can read related data and call the query condition generator to generate page controls which meets the criteria of HTML or ASPX. In this way, all the condition condition page controls of the inquiry business can be realized with ease. What the developers need to do is designing the inquiry table and hand the rest of the work to the query condition generator. Whether adding new inquiry or changing the query condition, it can be realized only by modifying the configuration files without editing codes. All the .NET based inquiry system can reuse the component on that basis. The construction of the inquiry system is very easy, and someone who doesn't know programming can also develop the component. The biggest advantage is that the user doesn't need to worry about mistakes because if the program goes wrong, it must be the inner components. They can check from the components.

For instance, if querying the door number plates in a certain inquiry information system by the name of roads and streets, the configuration file would be:

```
<?xml version="1.0" encoding="gb2312"?>
<CONFIG><CONDITION ElementText="Jlx"
ElementName="Jxmc"ElementType="TextBox"
SYSTEM="SecondID System" ElementDataSource=""
ElementFieldData=""OperateName="JxmcQueryType"
OperateType="RadioButtonList"OperateDataSource=""
OperateFieldText="OK"OperateFieldName="Jlx"
OperateFieldData=""OperateAffixFlag=""DefaultValue=""><QUERYACTION ActionName="mlpxxcx" />
</CONFIG>
```

The last part of the configuration files above declares that the inquiry business is door number plates inquiry. It can be English characters. In general, the business of signal query is simplified by Chinese. <CONDITION> node identifies the query condition. If there are other query conditions, the programmer can configure similarly and just add <CONDITION> node. The system use a XSL file to transfer the XML format into ASPX page. Condition configuration files include displayed text, name, type, system name, data source, dictionary text, dictionary code, option name, option type, option data source, dictionary text and default, etc. The most critical of them are the name and type. All the types C# supports and the type customized for user's specific demand are listed in the XSL file. The realization ideas of showing page converter are similar to

the query condition generator, which also encapsulates the query results into certain format of XML file and transfers it to the required display interface.

#### B. General Query Base Class

The framework design focuses on the generality of query operations rather than specific query business. Especially this query component can be used in other query systems. In order to enhance the ratio of reuse and easy use, the query realizing SQL is placed on the database server. General query base class solves the problem of abstract operation by regarding the results returned by SQL as an object, and using generic technology provided by .NET to boils down any business data query to an abstract query. And then the query results would be encapsulate into an XML file, which then be switched into the display format users need. In this way, one displayed page can realize all the query business display, and one query method can realize all the query business query. Here are some code examples which implemented abstract query:

```
namespace General.SearchLayer
{
    public class RequestBase< DA, BE> where DA: new()
    where BE: new() { DA _D_B = new DA (); public DA D_B
    { get { return _D_B; } } List< BE > BList = new List<
    BE >(); BE Build(IDataReader Dr) { BE D = new BE ();
    BuildEntity (D, Dr); return D; }
        public List< BE > BuildList(IDataReader Dr)
        { while (Dr.Read()) { BList.Add(Build(Dr));}
    Dr.Close(); return BList;}
        public virtual void BEntity(BE Busns, IDataReader Dr)
        { //Call Related Procedure
        } }
    Finally, the returned data is encapsulated and then
    disposed by the page transformer to realize data display .
    The query method also can be released as Web Service for
    the use of other application programs.
```

#### C. Heterogeneous Data Access Components

The types of database the query system adopting are not necessarily the same, they may have several databases and even be located in different geographic positions. So the data access layer in the framework needs to solve the problem of heterogeneous data access. The heterogeneous data access component is specially designed for this. The basic idea also starts from facilitating the developers and tries to meet the present using type of database access needs.

The implement is relatively simple because it is based on the Enterprise Library of Microsoft. Many details are disposed by the Enterprise Library. The component just does one layer abstraction and puts present using types of database into one XML file[6]. The heterogeneous data access component realizes the access to these database respectively in merely one method. Regardless of what database the developers access, they all apply the same access pattern. So, when constructing other query systems, to realize data access of fitting specific demand, the developers only need to configurate related connection string.

The design idea is to make the common database types and corresponding engine into one-one mapping and protocol the set of operation of common database. The operation of all the types of database are invoked in the same mode[7]. This will shield the specific database access details. No matter what type of database, developers can use conveniently. When the using database type changes, developers can implement database steering only by modifying related configuration files rather than the code. The example of the partial code of database connection is as follow. The realization of other function is similar to this:

```
public DataAccess(string SysName, string MName)
{try{DataRow r=this.ConfigRow(SysName, MName);
string vServer = r["DataSource"].ToString();
string vUserName = r["User"].ToString();
string vPassword = r["Password"].ToString();
string vDataBase = r["DBName"].ToString();
string vDataType = r["DBType"].ToString().ToUpper();
.....
dProvider =CfgDoc.SelectSingleNode("Dprovider") ;
dbConn = new CfgDo.SelectSingleNode("DEngine")
(string.Format("Data Source={0};User
Id={1};Password={2}; vServer, vUserName, vPassword,
vDataBase));
dataAda = new CfDoc.SelectSingleNode("DAdapter");
throw new Exception("DataType Error");
}} catch (Exception vError)
{ throw new Exception(vError.Message); }.....}
```

It realizes unified access to different databases, so do the other operations of different databases. It is not only easy to use, but also conducive to enhance the ratio of reuse. All the .NET based information system can reuse the heterogeneous data access components. The other technicals are used in the framework design, such as unified database access mode, the stored procedure code technology, paging background methods and the unified data validation techniques. They cannot be introduced in details due to the limited length space.

#### IV. CONCLUSION

The query framework constructing in layering design idea is relatively independent between each layer and

irrelevant to specific query business, so it has high ratio of reuse and can be used in any information query system. The frame designs heterogeneous data access components which admit developers to use their database type without modify code in the second development. The access of any system based on .NET to database is transparent and easy to be used. In addition, the system shares a query interface and display interface, reducing the work of development and increasing the efficiency. The unified verification of the data validity and query paging method can also be used for reference in the database system. Practices have proven that the framework applied to some police integrated information platform construction shows good effect in operation. The successful application of this design mode has theoretical reference value and practical application on the construction of information systems. Certainly, if the design can make development tool and platform be portable, it will be better. Developers are working on it.

#### REFERENCES

- [1] Jiawei Han ,Micheline Kamber, Translators: Fanming, MengXiaofeng, Concepts and Technology of Data Mining [M].Beijing: China Machine Press,2007.
- [2] Zhang Yi. Essentials and Modes of Software Design [M].Beijing: Publishing House of Electronics Industry,2007.
- [3] Wen Yu. Software Framework Design [M]. .Beijing: Publishing House of Electronics Industry,2007.
- [4] Guan Quanlong, etc. The Design and Implementation of Establishing a New Dynamic Web Query System Platform [J]. Computer Applications and Software,2008,25(9):125-126.
- [5] Yang Lingsxing. The Design and Implementation of Reusable Query Component Based on .NET [J]. Computer Age,2010,6:20-21.
- [6] He Liren, etc. The Design and Implementation of Common Query System Platform Based on Web [J]. Microcomputer & Its Applications,2009,19:7-9.
- [7] Chen Yiqun, etc. The Design and Implementation of Highly Adoptable Enterprise Information Query Expand System [J]. Computer Engineering and Applications,2011,47(2):227-231.