CONSERVATOIRE NATIONAL DES ARTS ET METIERS
PARIS

MEMOIRE
présenté en vue d'obtenir
le DIPLOME d'INGENIEUR CNAM
SPECIALITE : informatique
OPTION : Architecture et Ingénierie des systèmes et des logiciels

par
QIU Jing

Research and Implementation of Lightweight Database Management Information System base on GWT Framework

Soutenu le 21/06/2011

JURY
PRESIDENT :
MEMBRES :

CONTENTS

1. Introduction 1
1.1. Introduction of project 1
1.2. Research Background 1
1.3. The situation of nowadays 2
1.4. The research of theoretical and practical significance 3
1.5. The Organization of thesis 5
2. Requirement 6
2.1. Business requirement of the system 6
2.1.1. System introduction 6
2.1.2. Basic business requirement 6
2.1.3. Other business requirements 8
2.1.4. Demand explanation 9
2.1.5. Core demand 10
2.1.6. System performance demand 10
2.1.7. Demand of analysis design 11
2.1.8. Technical demand 11
2.2. Introduction of AREVA 11
2.2.1. Context of the Organization 12
2.3. Summary of this chapter 13
3. The design of System framework structure 14
3.1. The structure design of system 14
3.1.1. According to permission to divide modules 14
3.1.2. According to function to divide subsystems 15
3.1.3. The analysis of system use cases 17
3.2. The general framework design of system 18
3.3. Elaborated the correlation technique and selection 21
3.3.1. GWT framework 21
3.3.2. SmartGWT framework 22
3.3.3. Spring framework 22
3.3.4. Hibernate introduction 23
3.4. The choice of Framework 24
3.4.1. Performance layer framework 24
3.4.2. DataSource layer framework 24
3.5. Difficult points of development 25
3.6. Development environment 27
3.7. Summary of this chapter 28
4. The implementation of MIS with agile inquiry 29
4.1. The implements of database framework design 29
4.1.1. The single table inquiry traceability 30
4.1.2. The multi-table inquiry traceability 30
4.1.3. The loop problem of Multi-table inquiry 32
4.2. System model 33
4.3. The design and implementation of performance layer 34
4.4. The communication model under the SmartGWT 34
4.4.1. Asynchronous communication under SmartGWT 34
4.4.2. Asynchronous communication in system use 35
4.4.3. The synchronous communication of SmartGWT framework 36
4.4.4. The usage of synchronous communication in system 36
4.5. Definition the transmission format of communication 36

4.5.1. GWT data component and server communication activity chart 38
4.5.2. TreeGrid component and XML transmission format 39
4.5.3. ListGrid component and JSON transmission format 40
4.6. The implementation of client side 43
4.6.1. Database of client side use inquired mode basically 43
4.6.2. The client side inquired the design of UI to implement nimbly 44
4.6.3. The implementation of user's agile inquiry on the client side 44
4.6.4. The foreign key relations implement in agile inquiry 46
4.6.5. The activity chart of agile inquiry on client side 48
4.6.6. The multi-language i18n design's implementation in performance layer 49
4.6.7. MVC model on the performance layer 49
4.7. The implementation of server side 51
4.7.1. The selection of server side framework 51
4.7.2. The implementation of the layers in server side 52
4.7.3. The function implementation on the server side 56
4.7.4. The sequence chart of agile inquiry 58
4.8. Summary of this chapter 59
5. Deployment testing and user experience 60
5.1. The deployment of testing environment 60
5.2. Unit testing 60
5.3. Integration testing 60
5.4. Test conclusion 61
5.5. Related functions interface demonstration 61
5.6. Summary of this chapter 65
6. **Summary and forecast** 66
6.1. Summary of thesis work 66
6.2. The future work 67
References 69
Acknowledgements 71

Abstract

With the prevalence of networks and a lot of lightweight-database are used in many small and medium enterprises, the management information systems (MIS) need to build a lot of queries to meet the users' requirements, which leads to the duplication of development. When the users' query-needs are not fixed or will change often, (a fixed user interface only meets a fixed need), these systems obviously can not meet the needs of users. Because the End Users cannot interact with the database directly, system developers cannot demand End Users to learn the knowledge and technique of the database.

This thesis studied and implemented a simple-query user interface for an inquiry supported environment. As long as the user knows the name of the database table attributes and the meaning contained, not need to know the knowledge of database query language, they can create the specific-need queries through the MIS. Company SAP provided products such as Business Object for large enterprises. (BO is a system which can make the reports by inquiring large database). This paper focused on the research of lightweight database MIS which can do the agile query operation.

This thesis discussed the system which uses the technique based on GWT (Google Web Toolkit) framework. The presentation layer of the system uses SmartGWT framework to achieve. This system enable that users only need some database configurations to make this system query on the specific database. Make use of the technology Based on the GWT-AJAX which can reduce the pressure of the server. The presentation layer of the system is programmed by Java programming language, then GWT framework can cross-compile JAVA and JavaScript which is optimized, and JavaScript can automatically run on all the browsers. As a result of using SmartGWT framework, the module of data source will obtain data from the server at a time, and the clients can do the data-analysis and query operations to the local data source. A large number of reusable components and user interface module can greatly reduce the amount of codes. With a few codes the system can provide the agile query capabilities.

This system has been implemented in the configuration management database of AREVA (France Nuclear Power Company) Taishan Nuclear-Power Project, and which has good results of satisfying the needs of the users. The inquiry module has some market in small companies.

Key words: GWT; SmartGWT; common database; MIS; Agile Query;

1. Introduction

1.1. Introduction of project

The project is to develop a configuration management system which used in AREVA Taishan nuclear project in Taishan, China. The system is mainly used for managing the configuration information among five parts: projects, software servers, hardware servers, software and workstations. Through the system, the users can quickly get the configuration information. The system also provides traceability function of all kinds of devices and software versions. Each project requires a large amount of software system and tools. The software is installed on different software servers. These software servers will be installed in the hardware servers. Each hardware server can install some software servers. Some different projects may use the same software server, and some different software server will use the same software. But users also have some workstations (PC) which install some software client. Different types of clients may use different software Matrix (which is a list of software configuration).Through this information system, one can obtain management information of the five parts more accurately and more easily.

1.2. Research Background

With the rapid development of Internet, more and more applications are designed based on network. Through network, the user terminal can access the services they need conveniently. Large number of network applications (Queries, resources accessing, peer-to-peer) develop rapidly. Now various types of systems including MIS become network-diversity gradually.

In the architecture aspect, from the previous C/S architecture to modern B/S architecture, both the advantages and disadvantages have been studied for a long time. C/S architecture is suitable to do the inquiry operation in a large number of information. But with this architecture, users need to install the client application locally. The B/S structure is considered to access more conveniently and has an excellent presentation layer to give a friendly user-interface. When users need a lot of analysis operation to the data, the server will undertake enormous pressure on the server. The long deal-time will directly affect the user experience. But AJAX architecture provides a good solution to mix the two framework advantages. The research refers to this part.

In the development aspect, because of the special features of the data query, the databases of different business systems require the different development. The developer need to coding

for each of query need. Repeated development of MIS will happen very often, which brings about wasting of resources. Because of this flexibility demand driven, the research of an easy-query and general solution for the database system is valuable.

In the user requirement aspect, the user's query needs is the most important requirements of MIS. The customers have specific needs to get the necessary data from the database. And sometimes these needs will change often. A design of MIS which can meet all the needs will be very difficult to implement, and the number of developing code will be very large. Therefore, A MIS system which is general-purpose and agile query can reduce the work of repeated development.

This thesis has done some research about a MIS for lightweight database which provides general-purpose and agile inquiry base on GWT: GWT framework has the advantages of those two architectures, the design of heavy client can solve the server's bottleneck problem, and it bases on Java programming. A large number of developers can quickly master the development skill. The framework leads MIS to be Network Oriented.

1.3. The situation of nowadays

In the world, SAP is the greatest ERP provider. The Business Objects series products are the leader of business intelligent solution. They provide a platform for data report, data query and data analysis and data integration. The solution provides data integration, data quality management, meta-data management and life cycle management products, which can make data more reliable. SAP also purchased the company which produces report industry standard product - Crystal products. In 2009, the statistics show that series of BO products of SAP have occupied more than 60% of the world's market of large database.

Crystal Reports is a powerful, dynamic and actionable reporting solution that will help you through the network design, development, visualization and dissemination of reports, or embedded in enterprise applications. With it, the end user will be able to use the visualization features dazzling to present the report, and do the on-report business modeling, and one can do the rapid decisions according to the report itself, which reduces the dependence on IT and developers.

BO series of products is a solution based on the reporting and the query design flexibility and reporting. The middle layer of the Business Object, which hides data of the database，only show the object of business logic to the end user. With these objects, the user can create the query to search information what they need. One also can do data mining to the results data, the result can be showed by Crystal Reports.

These products are aimed at large enterprises with large databases, and very expensive, it is impossible for small business to afford, and such large-scale software is used for some small-scale which is unwise. For small and medium enterprises, there is not a suitable common MIS system which has been developed.

In this paper, the idea agile query of the MIS system refers from Business Object. It can design products to meet the user requirement of the query diversity and variability of demand.

1.4. The research of theoretical and practical significance

In functions, the MIS is organization theory, accounting, statistics, mathematical model and a mixture of economics. It is full use of the computer technology, the network communication technology and database technology. It is also a multidisciplinary edge technology, so it is a technical system. From the social point view of technical systems, MIS has interaction with organizational structure, the introduction of MIS will lead to the generation of new organizational structure, and the existing organizational structure will has impact on the analysis, design of the MIS. The impact of factors includes the organizational environment, organizational strategy, organizational goals, organizational structure, organizational processes and organizational culture. So the MIS is both a technical system, but also the social system. This thesis attempts to make information systems separate from the social system, and in order to make it only a technical system which can implement query versatility.

This paper studies a general MIS which can provides a solution of the agile query. It can be used in a broad scope. The characteristics specifically in the following areas:

1) The general feature of the database platform.

The general feature of the platform is mainly reflected in: the system only need a small amount of database connection configuration to be able to connect to the objective database correctly. It also can access information in the database and provide flexibility query operation to the relevant table.

2) The platform provides the flexibility of the user query interface.

The end-users are business people rather than the database administrators. Querying the database, which will inevitably involve the usage technology of database, for example, the user must be familiar with SQL statements. For the diversity of user requirements and variability, through the help of database administrators to meet the accessing needs of the user, which is obviously not adaptable. So the solution is to package parts of database technology and presented to the user some business entities, the user can select different business entities

related to make the query. Tables of links and parse SQL statements were encapsulated within the system.

The user interface design imitates the UI design of Business Object, which uses the framework of drag and drop. It has a very good user experience, users only need to set the selected content and filter conditions to complete the related queries.

3) The reusability of the components

In order to simplify the code, reusability of the components is very important. With SmartGWT framework, components are packaged into modules. Developer can modular data source on the client side. Modular of data source and UI offer a good reusability, which let the development more easy. This method of modular design greatly reduces the work caused by maintenance procedure. Corresponding to the new requirements, developer only uses a small amount of code to implement.

4) Low requirement of the server hardware

The MIS used the B/S framework will bring great pressure to the server, specifically in inquiries, the traditional B/S architecture is basically using the framework of the MVC structure, for each request, server will return the entire page of data. And since the birth of AJAX, the "only update the part needs to be updated," the design of this idea to become mainstream. GWT framework uses AJAX technique. To the user's request, the server only needs to return data to the client, and no need to update the entire page of data.

In addition, with the framework of the SmartGWT (heavy client-side framework), data will be downloaded to the local to form a local data source, which is for reading local data, if the user do the keyword query or data analysis on the client side, data processing procedure can provide very high performance. The bottleneck of performance has been transferred from the client side to the server side, to reach a loading balancing effect. The server does not need high performance hardware that can afford the access operations to the MIS database. But the framework of offline data sources to bring the data may be inconsistent. It needs to do further research with the data consistency problem. In the framework SmartGWT, because of the source code hidden, developers only can through the JS code to study concrete realization of component.

At the first time visit the Web site, the local browser will download static HTML pages and all the JavaScript from the server to the local, and now PC can save the website cache (HTML of static pages and JS form local cache), so when the users enter the website again,

the client side will access the local cache to get the large amounts of pages and components without to re-download from the server. The loading speed will be greatly increased.

1.5. The Organization of thesis

In this thesis, the content of each chapter is summarized as follows:

In the chapter 2, the thesis analyzed the requirement of 《the French AREVA Taishan nuclear power project configuration management system》. There are four aspects of the analysis: basic requirement, core requirement, performance requirement, requirement of analysis and design, which are as the foundation for the outline design and detailed design.

In the chapter 3, the thesis firstly introduced the sub-modules, subsystems and the framework of the organizational structure of the system. Secondly it analyzed the system's use cases. Thirdly, it introduced the overall hierarchical framework and some related technologies. In the end, it gave a brief description of the study point and implementation difficulties of this system.

In the chapter 4, it analyzed and studied database design, the user UI layer design, communications design and main class of pattern design, and related technical points. In accordance with the client and server design and implementation, it described it in two ways.

The fifth chapter is about the deployment, operation and tests situation. It used screenshots to show the operation situation of the user interface and implementation of innovation functions. In the end, it also introduced the user interface of agile query.

The last chapter is about the conclusion part. Some problem remarks are given for providing direction for further research.

2. Requirement

The research project of this thesis is based on 《AREVA Taishan nuclear power project configuration management system》 to study the implementation process. This chapter analyzes the requirement of the system.

2.1. Business requirement of the system

2.1.1. System introduction

The system is used in the AREVA project in Taishan, China. It mainly used for configuration management of five objects: engineering, software servers, hardware servers, applications and workstations. Using the system, users can quickly access the configuration information. The system also offers the traceability of various versions of software and hardware. Each project requires a large amount of system software and tools. These software are installed on different servers, the server refers to the software server, which will be installed on the hardware server. Each hardware server can installed some software server. Each project will be cross-use of these software servers, which also install some appropriate tools. Users use some computers as workstation client. The workstation clients also install some different software. Different types of workstation clients use different software Matrix (a list of software configuration). Through this MIS, user can obtain more accurate management information for further decision.

2.1.2. Basic business requirement

For most basic MIS, the basic business requirement is to do the insertion, deletion and modification operation on related table. Basic business requirement as follows:

Table 2-1 Business requirement of system

Demand name	Operations	Related operation	Remark
User permission demand	Insertion operation	Does not have	Must have the administrator permission
	Deletion operation	Does not have	Must have the administrator permission
	Modify the permission operation	Does not have	Must have the administrator permission

	Show all user operations	Does not have	Must have the administrator permission
Projects table demand	Add A project operation	Does not have	Must have “write” permission
	Modify the project attribute operation	Does not have	Must have “write” permission, some essential attributes do not permit modifying
	Shutdown a project operation	Insert traces in the record table	Must have “write” permission, joins the traceability table to record the application software which this project used completely
	Inquiry project operation	Does not have	Implements through the agile inquiry
Servers table demand	Add a software server operation	Does not have	Must have “write” permission
	Modify the server attribute operation	Does not have	Must have “write” permission, some essential attributes do not permit the modification
	Shutdown a server operation	Does not have	Must have “write” permission
	Inquiry server table operation	Does not have	Implements through the agile inquiry
Applications table demand	Insert application operation	Does not have	Must have “write” permission
	Modify application attribute operation	Does not have	Must have “write” permission, some essential attributes do not permit the modification
	Shutdown application attribute operation	Insert traces in the record table	Shutdown the application to add record to trace table
	Inquiry applications operation	Does not have	Implements through the agile inquiry
Workstations table demand	Add a workstations operation	Does not have	Must have “write” permission

	Modify the workstation attribute operation	Does not have	Must have "write" permission, some essential attributes do not permit the modification
	Shutdown workstation operation	Does not have	Must have "write" permission
	Inquiry workstation table operation	Does not have	Implements through the agile inquiry
Hardware table demand	Add hardware server operation	Does not have	Must have "write" permission
	Modify the hardware server attribute operation	Does not have	Must have "write" permission, some essential attributes do not permit the modification
	Shutdown hardware server operation	Does not have	Must have "write" permission
	Inquiry hardware server table operation	Does not have	Implements through the agile inquiry
Site table demand	Add the site to operate	Does not have	Must have "write" permission
	Modify the place information operation	Does not have	Must have "write" permission, some essential attributes do not permit the modification
	Shut down the site operation	Does not have	Must have "write" permission
	Inquiry the site table operation	Does not have	Implements through the agile inquiry

2.1.3. Other business requirements

Table 2-2 Other business requirement of system

Demand name	Operations	Remark
Other business requirements	Add a project with software server and application relations	When one adds a new project, relations table, including increase, applies the selection
	Inquiry the server which the project use	-

	Inquired the application which the server uses	-
	Inquired the application which the project uses	-
	Inquiry the servers which are used in shut down project	-
	The inquiry the software which are used in shut down server	-
	Inquired the application which are used in the shut down server	-

2.1.4. Demand explanation

- User permission demand

The system need to provide the basic demand which includes insertion, deletion and modification operations. Administrator carries on the operations in the administrator interface, and he can give the user the "administrator" permission, "reader" permission and "writer" permission (to be possible to modify data).

- Project table demand

The insertion operation, the project does not permit deletion operation, if one needs to delete, you must carry on the deletion through database administrator (users cannot do the delete operation). One can shutdown a project (modify "IsClosed" attribute).

- Software table demand

The system need to provide the basic insertion and modification operation, preserves all data (does not need to delete data). The modification operation only refers to all attributes besides name (Name) and version number (Version Num) and SP number (SPID).

- Software server table demand

The system need to provide the basic insertion and modification operation, preserve all data (does not need to delete data). The modification operation only refers to all attributes besides name (Name) and version number (Version Num) and SP number (SPID). It also provides the shutdown function, when software server no longer uses, then modify "isClosed" attribute. One cannot do deletion operation. It also provides the OS version number traceability function, which records the OS version use the start time and end time attributes.

- Hardware server table demand

The system needs to provide the basic insertion and modification operation, and can only modify all attributes besides the name. It also can provide the shutdown function, when the server no longer uses, one can modify shutdown attribute rather than delete.

- Site table demand

The system needs to provide the basic insertion and modification operation, and can modify attributes besides the name operation.

- Workstation table demand

The system needs to provide the basic insertion and modification operation, and can modify attributes besides the name operation. It also can provide the shutdown function, when the workstation no longer uses, one can modify shutdown attribute rather than delete. It also provides the Matrix version traceability function, which records the Matrix version use the start time and end time attributes.

- Other demands

The system also needs to provide adding the relationship system's demand among project, server and software, and each two relationship between them, for example, some software are installed on the software server.

Because of the complexity of business logic and the system function, it needs to have a system framework to support the performance interface, business logic, the data access and transaction code reused, in order to enhance maintainability of system functions.

2.1.5. Core demand

In the system, the most important requirement is the demand of inquiring. This system mainly provides various users of departments carry on the inquiry related to configuration information. Because the users are from different departments, and the relations of four entity table are complicated, obviously the fixed inquiry function cannot meet the user's needs. Presently the inquiry demand is fixed, but developer must keep the interface to carry on the following development for later inquiry demand. One kind of agile inquiry design provides a solution to this inquiry demand question.

2.1.6. System performance demand

The system must have a good user interface and user experience. It can directly be accessed through the browser, because there aren’t many user to access the system at the same time, therefore does not need to consider the question of concurrency capacity. The server

needs to provide quick speed of response for the inquiry. A good inquiry design can cause the user to obtain more accurate information quickly, which makes the work more efficient and have a better decision-making.

2.1.7. Demand of analysis design

Developers must select a correct method of analysis to design the relational database according to user's demand. The design of relational database is quite important, which directly affects later period's business logic.

The implementation of version traceability function makes database table relationship more complex. Considered that the users are not too many, moreover, this main function of system are storing the historical data and agile inquiry. The data total quantity isn't very large. Through estimating, entire database data number are approximately 30,000 to 50,000 records. Therefore, the access performance doesn't affect too much with the user experience, the main work of project places on the inquiry design.

2.1.8. Technical demand

The technical demand is: The development language is JAVA, and uses Google Web Toolkit (GWT) which is very popular now as the client side. The database uses MySql database.

2.2. Introduction of AREVA

All over the world, AREVA provides a response adapted to the needs of electricity generators in both nuclear energy and renewable energies. The range of solutions is comprised of an array of capabilities and technologies enabling competitive production that is safe and respectful of the environment.

AREVA supplies its customers with solutions for carbon-free power generation and electricity transmission and distribution. With its knowledge and expertise in these fields, the group has a leading role to play in meeting the world's energy needs.

2.2.1. Context of the Organization

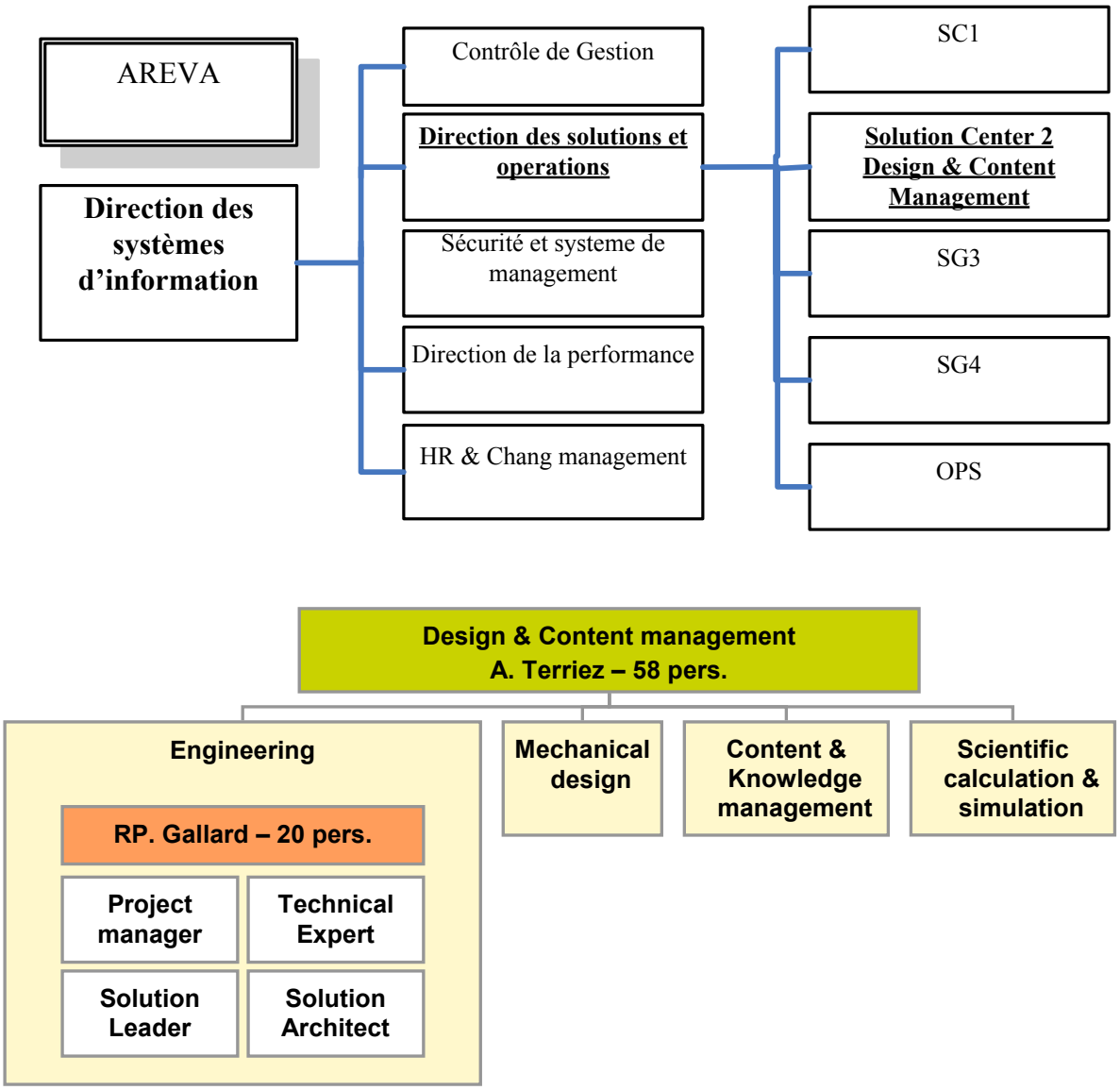


Figure 2-1: the organization of AREVA

In AREVA, we belong to the "Solution Center 2 Design & Content Management", and this department is divided into 4 parts: Engineering, Mechanical design, Content & Knowledge management and scientific calculation & simulation. We belong to the "Engineering" group. In this thesis, all the systems are for Taishan project.

Taishan project is located in JiangMen, Guangdong province, China. The first phase of construction will build two 1750 Megawatt PWR nuclear power units; it uses the third generation technology of Europe. It is the largest nuclear power unit in the world. This project is partly joint development and construction program between AREVA and China Guangdong Nuclear Power Company (CGNPC).

This simulation project belongs to Taishan Project and will be done by IMS department of Taishan Project (Information Management System).

2.3. Summary of this chapter

From three aspects: the system function, performance, the analysis design demand, this chapter explained the characteristic of system framework. The framework should support the code reuse in performance level, business logic level, the data access and transaction aspects, in order to enhance the maintainability of system function. In the end, this chapter introduced AREVA and its department organization.

3. The design of System framework structure

This chapter mainly discussed the frameworks and technical selection, and described the organizational structure design of entire MIS. The system needs to implement the business requirement (English) of 《France AREVA Taishan Project Configuration Management System Requirement》.

3.1. The structure design of system

3.1.1. According to permission to divide modules

The design of the system mainly is divided into the administrator module, read permission and write permission module:

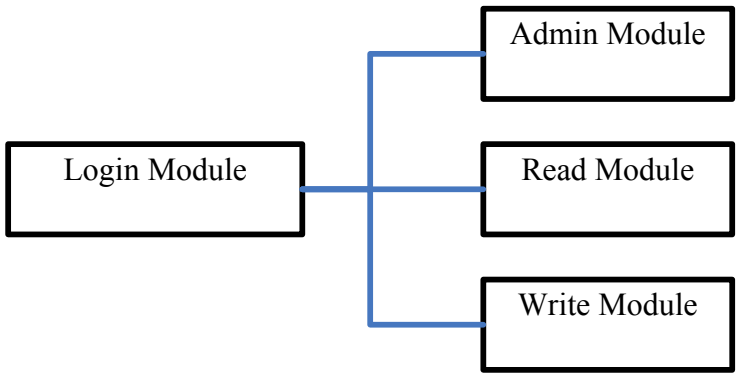


Figure 3-1 permission module chart

Because the permission part is not the key point of this article, but this part can be research more deeply, for example: the permission refers to tables and attributes. This article implemented simple permission module, according to user's permission to open different user UI. The basic statement of permission is as follows:

- Administrators: The administrators are mainly responsible for controlling the user information: add, delete and modify user's information, which is mainly used to modify the user's permission.
- Read permission user: Read permission users only have the permission of accessing functions, which includes inquiry, saving inquiry function and exporting the result to excel file.
- Write permission user: Write permission users have the permission of accessing and writing functions, which includes inserting a new record, modification, deletion operation to the information.

3.1.2. According to function to divide subsystems

This method is according to the functions to divide subsystems. Because the Read permission user module and the Write permission user module can use the same UI module, which only needed to judge and to select the different view in the corresponding permission .Different UI view can provides different functions. The function subsystem includes:

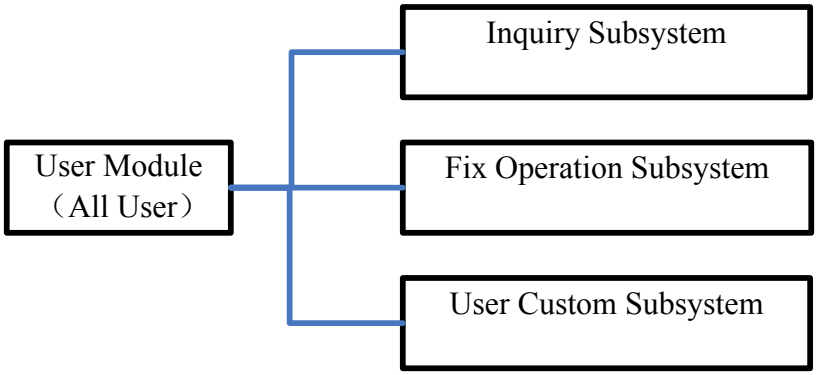


Figure 3-2 function subsystem

Inquired subsystem is the important part of research. The user interface used the design which mainly imitated the multipurpose inquiry interface of Business Object. Three sub-systems can be selected by the left tree menu of the User Interface. Each sub-system interface can developed with much the reusable code, and can use the object-oriented method to design.

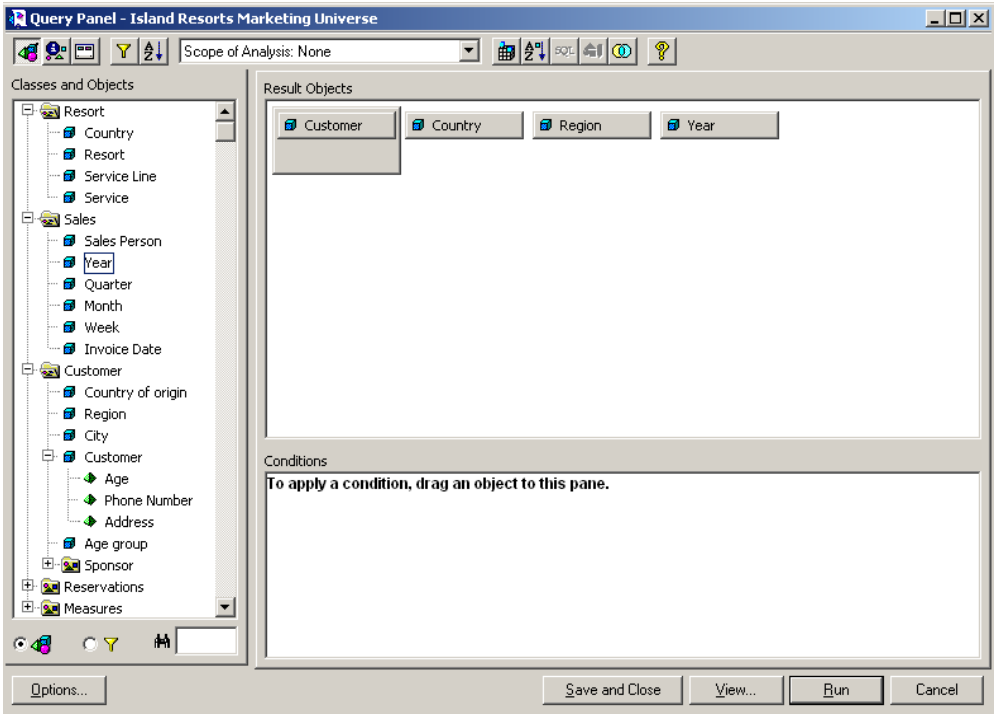


Figure 3-3 the inquired UI of Business Object diagram

The simple introduction of subsystem is as follows:

- **Inquiry subsystem:** Inquired subsystem mainly to manifest the research of agile inquiry in this article. The general operational procedure was: The users select the item which you want to know from the left side of UI, and drag it to the Select space on the right side of UI. The system will search the name and add the table name to the Where space automatically on the right side. One also can add some conditions to the inquiry to filter the result. The save query to user space or save query to all user space function are provided in the toolbar.
- **Fixed Function Subsystem:** This subsystem provided the functions for business requirement, which is according to 《France AREVA Taishan Project Configuration Management System Requirement》.
- **User Custom Subsystem:** The subsystem can display the query sentences which are saved by user. Users can get the data directly by the record. User also can import the SQL sentence to get the data you want, and save this query to user space and all user space. The result grid of this part can reuse the standard result UI which can show the data nimbly.

3.1.3. The analysis of system use cases

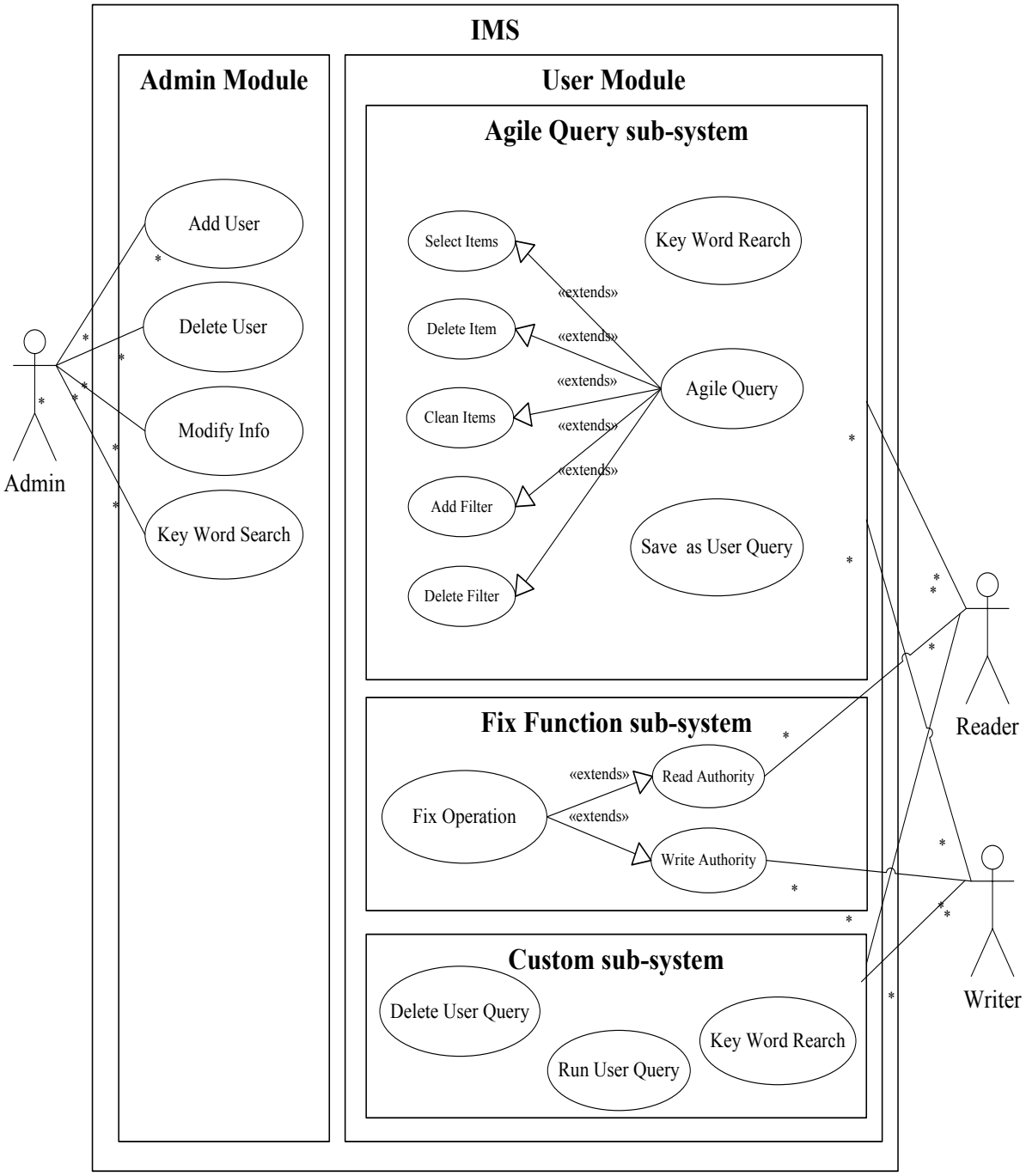


Figure 3-4 System use case

Figure 4 showed the use case of all the system. The main part of the design is agile inquiry subsystem (Agile Query Sub-System). But inside the user module, the there subsystem can have function communication. They may share the same run inquiry module. The fixed function subsystem (Fix Function Sub-System) has many functions which are based on the inquiry module. And user custom subsystem (Custom Sub-System) mainly store the information of inquiry, from which user can obtain the result directly. Therefore, the inquiry

module's code can be packaged to a service for reusing, with which it enhances the maintainability of system function.

3.2. The general framework design of system

The lamination comes from the experience and practice, which can be widely used in current MIS (Management Information System. This system also uses the lamination structure, sees Table 3-1, it is divided to the Presentation Layer, the Service Layer, the Domain Layer and Data Source Layer.

Table 3-1 the description of system layers

Ply	Description
Presentation Layer	The performance layer which is mainly used to interact with users, displays information and responses user's operation. This layer mainly uses the SmartGWT (GWT) framework to implement. This framework also requires implement the local data source in this layer.
Service Layer	This layer mainly provides the external service interface. For the server, it provides the request-dispatcher function; analyzing parameters function and method invoke function.
Domain Layer	This layer mainly manifests the service logic, which sends the data request to the data layer. After obtaining the result data, it analyzes the result and return to the client side.
Data Source Layer	This layer provides accessing database and database transaction processing, system logging and so on. After receiving the request from the Domain Layer, it accesses the database and returns the result.

Table 3-1, because the system used GWT framework to design, the Presentation Layer data will download to local when user use browser to access this system at the first time. Therefore, it is enables to separate from other three layers, which can be called "under B/S model's C/S pattern". Users can use the browser to access the system from anywhere with network. About the introduction of GWT framework, sees the following sections. Data Source Layer is mainly responsible for connecting to the data pool (database) and does the interaction, it also includes other technique part which hasn't relationship with the business. For example: system logging, database transaction processing and so on. Therefore, Craig

Larman also called it the Technique Layer. Domain Layer includes all the service logic and service objects, which uses object-oriented to implement.

The advantage of lamination, which is that few concept need to be processed in one layer, it is easy to classify. The classification may have the standard, which can make few classification matters need to be processed, and it is not easy to make a mistake. Moreover, the lamination can make it easy to deploy the package to different machine. But its disadvantageous is also very obvious: the addition or modification will need to be done in all the layers, and multi-layer brings more execution time. But the requirement of basis efficiency is enough quick, but is not quickest. From the view of Object-Oriented, such lamination separated the business layer, which pulled distance closer between OOA (Object-Oriented Analysis) and OOD (Object-Oriented Design).Additionally, it used OOPL (Object-Oriented Programming Languages), so it was easy to implement, maintenance and testing.

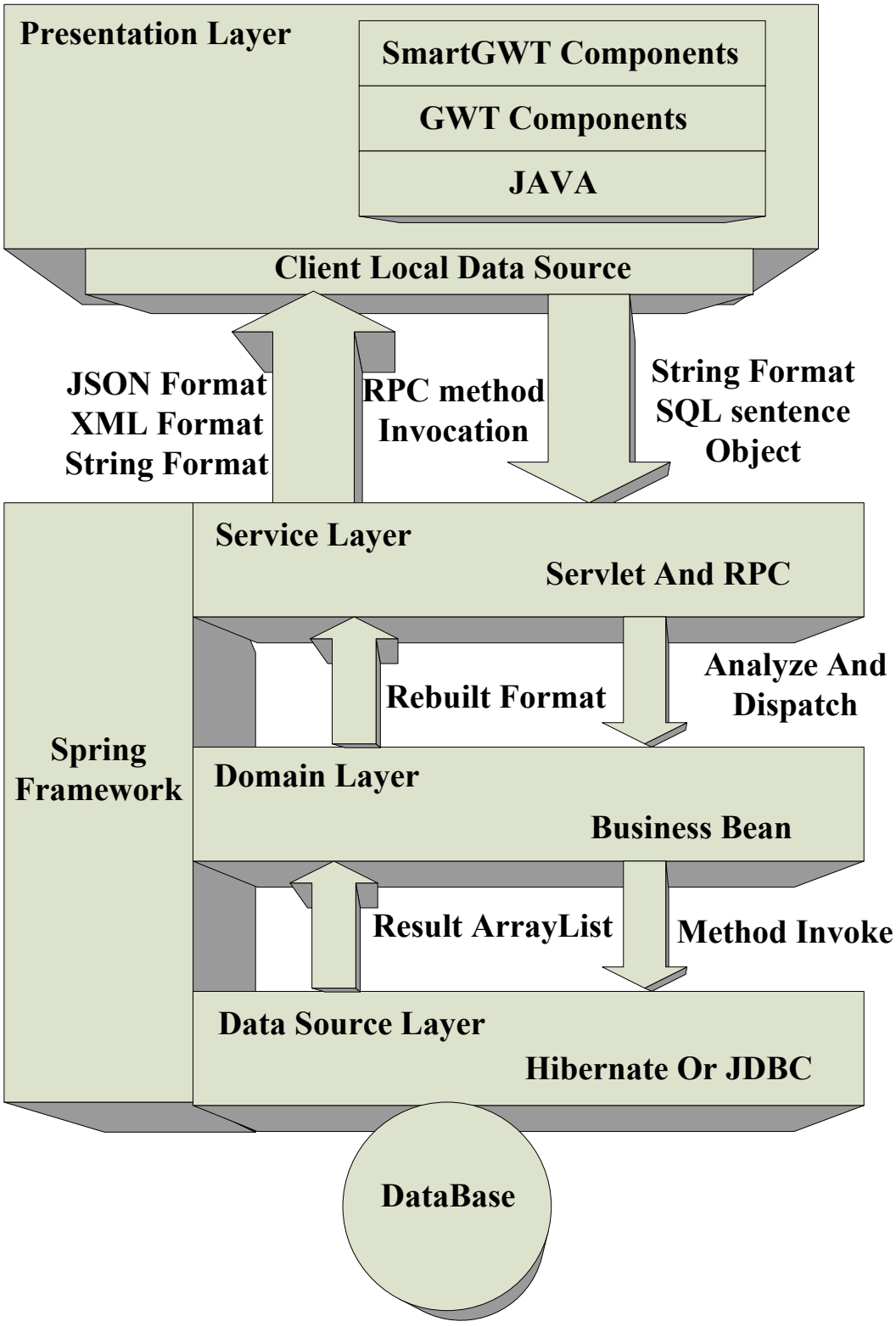


Figure 3-5 information system's structure framework

The Performance Layer mainly used SmartGWT framework Components to develop. The components obtain data mainly through the client side local data source, which is bound with a fixed URL. From this fixed address, it can connect to server. The main format of the

transmission data is string of character. After receiving parameter string, the Server Layer analyzes and processes it, and then retransmits to the related business logic methods on the Domain Layer. The logic will invoke the methods on DataSource Layer to access database. After getting back the result data, the Domain Layer logic will rebuild the result format to various format types of strings (XML and JSON). Through the Service Layer, the result string return to the client side, Presentation Layer analyzes and displays the results.

3.3. Elaborated the correlation technique and selection

3.3.1. GWT framework

GWT (Google Web Toolkit) is the framework which is used for developing the Ajax application recently released by Google. It supports Java development and the Ajax application debug. In the majority situations, the program developed by GWT can support IE, Firefox, Mozilla, Safari and Opera Brower. The programmers don't need to worry about browser's compatible problem. GWT provides the components which enables user to develop beautiful UI very easily. If the API provided by GWT cannot meet the need, developer can use JSNI make JavaScript embed the Java code.

The Ajax technology is a very popular technology current to develop the web application. It also is important constituents of Web 2.0 important constituents. However, if one uses the traditional method JavaScript to develop Ajax application, which will be hard to carry on the debugging and reduce the production efficiency. Google releases GWT to solve this difficult problem. GWT is the framework which used to develop the Ajax application. It let the programmers to use the Java simultaneous to develop client side and server side. The GWT compiler the client side the Java code to JavaScript and Html files, but programmers don't need to care about this process. Programmers like to develop Ajax application inside Java IDE. Because in the phase of exploitation system don't need to generate the HTML section, in fact, the code of the development is runs on JVM, so user can use the traditional debugging method to debug Java code, which speeds up the debugging process, reduces time of the software development.

The goodness of Ajax application is:

- Through the asynchronous model, it improves the user experience.
- It optimizes the transmission between the browser and server, reduces the data transmission, and reduces the bandwidth taken.
- The Ajax engine runs in the client side, reduces the undertaken of the server and reduces the load of a great volume of users.

3.3.2. SmartGWT framework

Since Google Web Toolkit has released, it has attracted the Java developers to join in the Web development. Massive expansions based on GWT appear. SmartGWT recently released undoubtedly is outstanding. It has included GWT APIs of the mature Ajax framework SmartClient. SmartGWT encompassed many Web components. Which makes it outstanding is the data integration with server. Users can develop the Web application of operating business data easily, which enables it to become the sharp weapon to the development of enterprise Web2.0 system.

SmartGWT encompasses all the SmartClient GWT APIs. SmartClient is an OpenSource Ajax enterprise development framework. The release of Google Web Toolkit greatly reduced the difficulty of Java development for the beginners. However, GWT itself provides components which are not enough, so many third party OpenSource expansion presented base on GWT. They provided many available API to GWT Groups. SmartGWT Group is outstanding inside them. Its lower level uses SmartClient this mature Ajax framework. SmartGWT is developed from 2008, which is in very active state. In December 2009, it has released 2.0 editions formally, which has become a mature Web 2.0 development framework.

SmartGWT is not only a “another Web component base”. Its biggest characteristic which is providing the client side and server side data framework. Because in lower level, SmartClient has implemented the Ajax’s MVC model, which causes in SmartGWT, the components can integrated with backend data, not need users to write code to integrate. Process many kinds of types and complex business data is the difficulty of the enterprise Web development. SmartGWT offers concern over the data operations, which simplifies the enter condition of enterprise Web2.0 application development. This point lets SmartGWT be outstanding.

3.3.3. Spring framework

Spring is an OpenSource framework. At present, it is very hot in the OpenSource community. It is considered that is one of most promising OpenSource framework. It created by Rod Johnson, the objective is to simplify the enterprise system’s development. Developers think Spring is as the substitute of EJB, it is one kind of lightweight's vessel and to solve the complexity of enterprise application development. Spring uses basic JavaBean to finish the work which is done by EJB before. However, Spring framework is not only restricted in the development of server side. From the simplicity, testable and loose coupling's view, we can say that any Java application may profit from Spring framework.

Spring framework's function is very formidable. Uses Spring framework, it is very convenient to decouple module and simplify development. Through the IOC vessel which Spring provides, we may hand over the control of dependence among objects to Spring, in order to avoids the excessive program coupling caused by coding hardly. If have Spring, the users do not needed to code for the single example model class and attribute file analysis and so on. They can more dedicate in the upper formation application. Through the AOP function which provided by Spring, developers can easy to carry on to the programming. Many functions not easy to implement with traditional OOP now can be deal with through AOP with ease. In Spring framework, we can use annotation to carry on the manage transactions nimbly, raises the development efficiency and quality. Spring framework also can carry on all the testing work by using the non-vessel rely-on programming method. Spring does not repel various kind of outstanding OpenSource framework, in contrast, Spring reduces the usage difficulties of various kind of frameworks, Spring has provided support for various kind of outstanding frames (for example Struts, Hibernate, Hession and Quartz). Spring provides a light layer to Java EE API which is difficult (for example JDBC, Java Mail, RPC and so on), through the layer of Spring providing, the usage difficulty of these Java EE API greatly reduces.

3.3.4. Hibernate introduction

Hibernate is an OpenSource object-relational mapping (ORM) framework, it encapsulate lightweight object to JDBC, which enables the Java programmer use the Object-Oriented method for manage the database operations. It has not only provided the mapping from the Java class to the data table, and also provided the data inquiry and recovers mechanism. Compared with using JDBC and SQL, which needs to manage operation database by manual, Hibernate greatly reduces the work to operate with database. Moreover, Hibernate can through proxy-mode to simply the process of class loading, which will greatly reduces the quantity of code extract data from the database by using HQL, and then reduce the development time and development cost. Hibernate can be integrated with many kinds of web servers or application servers well. Now it has supported almost all popular database servers.

3.4. The choice of Framework

3.4.1. Performance layer framework

In GWT framework, for JAVA programmers, there are three mature frameworks which include: GWT-EXT framework, GXT framework and SmartGWT framework. The GWT-EXT framework is SmartGWT framework's predecessor. The GWT component base provides basic UI component base. Components of client side use RPC method to communicate with server asynchronously. SmartGWT is a commercial data framework for enterprise, which integrate with data more closely. It simultaneously also provides massive data processing API. SmartGWT has two versions: the enterprise version and personal development free version. Enterprise version can use the high-level function of SmartGWT to develop the complex enterprise Web application. Free version has also provided the basic data communication framework. This thesis mainly research the system based on the free version.

This thesis chose SmartGWT framework to implement the client side, because it has provided the integrated data framework to connect client side and server. The core of an MIS is data, 80% operations of MIS are the operations of data, which include the creation, modification, deletion and inquiry operations. Each operation involves the operation of data. This framework binds data pool at the client side, and put the return data from the server to local data source, with which, the programmers can carry on the data programming on local, it can filter data and search data very efficiently. This framework has also integrated many complex operations of data to SmartGWT APIs, with which programmers can control data with ease.

3.4.2. DataSource layer framework

Because this system's core is the agile inquiry, therefore, if the structure of table is not fixed, it will be very difficult to develop by using Hibernate framework. When the user carries on cross multi-tables inquiry on multi-dimension, the object-relational mapping of Hibernate framework doesn't seriously suitable for flexible design. Hibernate provides the object-oriented data inquiry mechanism, but when the complex inquiry involves multi-tables attribute, it cannot use the object to present results, which has low performance. Therefore, on the DataSource layer, it only can use the base technique JDBC to implement the multi-table inquiry. Because JDBC can through SQL statement to select the attribute sequence to get the result and present them, so it can meet the agile inquiry requirement perfectly.

In the database tables which don't involve agile inquiry needs, they still can use Hibernate to implement in order to enhance the development speed. The mix-use of two will

not affect the efficiency and cause incompatible problem. The tables which mainly use Hibernate to implement are user information table, custom query table. In the tables with certain user's needs, Hibernate framework is also a good choice, for example: project-application traceability table which is reserve the history date. It has not inquiry-relation with other tables.

3.5. Difficult points of development

The difficult points of system design mainly are centralized in agile inquiry subsystem and user fixed function subsystem. The difficulties of development mainly are in following several aspects:

- Database design

Because some subsystems don't relate to business logical function's table, for example: user administration subsystem and custom inquiry subsystem, therefore, the design of these subsystems and the design of database structure can be carried in parallel. But the design of database structure's good or not directly affects the inquiry the efficiency and the implementation of user fixed demand. It needs to communicate with the user repeatedly to definite business requirements. The difficulty lies in the uncertainty of user demand and easy to appear misunderstanding, because of the communication difficult.

- User interface (UI) design

The user interface design immediate influent user efficiency of agile inquiry. The operation need to be simply and easy. In the vision aspect, the UI must let the user feel comfortable. In the control aspect, it is better to design a drag module to set the inquiry operation just like Business Object.

- Data transmission between client side and server side

Because the GWT framework's design based on AJAX technique, each page can carry on part-refreshing. Therefore, after one operation of date is done (filtration, insertion and modification), the system need to refresh the current data. But with this framework, it only needs the server to pass on the data which is related. The server not needs to transmit data with entire page logic code. In the transmission process, transmits the specific objects is very unwise, which seriously restricts the data inquiry's versatility and flexibility. Therefore, it is better to select the base class of data --"data string of characters" to carry on the transmission. GWT framework supports XML and JSON data format, and before the transmission, all the data must transfer to a right format.

- The research of SmartGWT data components

SmartGWT provided a data components base, which is used for all kind of display module with the operations to data. But the operational mechanism of component still need to discuss thoroughly, for example, the data grid links to the local data pool, it can through invoke the method automatically to control the data pool, The data pool can through setting URL with the specific parameters connect to the server to execute operations and get back the data. Because the code of components is based on the JavaScript compilation, it is very inconveniently to the pure JAVA programmer.

- The design of server business logic layer

For each MIS, it has its particularity of requirement. Therefore, the complex business logic should be design more clearly in order to enhance the maintainability. For example, the user may modify some tables' attributes, which will implicate a series of operations of other tables. General and reusable code can reduce complexity of development, which is also very easy for maintenance.

- The foreign key problem

In the design of agile inquiry operation, how to processes foreign keys of database tables is the difficulty point, because the database table has "foreign key relations". One cannot isolate the table which includes "foreign key relations". In the inquiry process, one need to judge the inquiry attributes' "Foreign key relations" to make the relation, with which he can get the correct result data.

- The design of cascade insertion data

In the data tables, it is possible to exist the cascade insertion and cascade deleted operations, for example, one adds a project, simultaneously, it also want to add certain servers to relate with this project, but if the servers are too many, in the user selection UI design will be very difficult, for example, uses are to pull the comboBox to display available server. If the servers are more than 100, that selection server is a painful operation. Here one wants to use the component of SmartGWT to show the available data, and through the key words search function to select, which can greatly enhance user's use experience.

- The compatibility of three frameworks

Because SmartGWT framework located on performance layer, its framework, Spring and Hibernate framework have not conflictions. The client side of this system selects SmartGWT framework to implement, the reason please see chapters 3.3, which separates the client side's performance layer and server side's business layer and persistence layer.

The business layer situated between the performance layer and persistence layer, it carries on processing the business logic, calling the persistence layer method to implement data persistence and inquiry work. Simultaneously, it must let the performance layer call RPC to invoke the methods on the service side. Using Spring framework can be very good to solve the program coupling problem.

The persistence layer is a layer which can communicate with database directly, the Hibernate framework through ORM solve the persistence layer's common problem: testing difficult and controlling difficult. But because Hibernate framework is limited by agile inquires, therefore, the part of agile inquiry is not advocated to use Hibernate framework.

Integrate these three frameworks in development, which can greatly reduce the cost of development, implement the loose coupling, good extendibility and high maintainability. Simultaneously it also may reduce project development cycle time, avoids the question of duplicating development.

Integrate Spring and Hibernate is a very mature technology, but SmartGWT framework as the performance layer based on the GWT to integrate with them which research is quite few.

3.6. Development environment

The development environment is mainly built based on the Eclipse environment:

Table 3-2 The development environment

Object	Version
JDK	JavaSDK 1.6
Eclipse	Eclipse JavaEE IDE (Helios) version 3.6
GWT Plug-in	Cypal Studio 2.0
SmartGWT	Version 2.1
Mysql-connector	version 5.1.7
Spring	version 2.5.6
Hibernate	version 3.3.0
Testing environment	Eclipse JUnit
Deploys the condition	Tomcat 6.0

3.7. Summary of this chapter

This chapter elaborated on the design of system's framework structure and the laminarization design of system. This chapter also introduced several related technology frameworks, and explained the reason of selecting these frameworks. Meanwhile, it analyzed the main design difficulty point from each layer, which shows the further research direction.

4. The implementation of MIS with agile inquiry

This chapter mainly elaborates the implementation of overall system framework, from the database design to the project detailed analysis. Simultaneously, it will research and implement the solutions to the design difficulty.

4.1. The implements of database framework design

The design of database is mainly about the four entity object table and their specific relations. “Projects”, “Servers” and “Applications” have a closer the relations. They are many to many relations. Moreover, the system must implement the traceability which includes OS version of “Servers”, version number and SP number of “Applications”. “Workstation” to “Project table” is one-to-many relation and to “Applications” is many-to-many relation.

Through the communication with client about the specific demand analysis, the relationship can be showed as the following database constitution table:

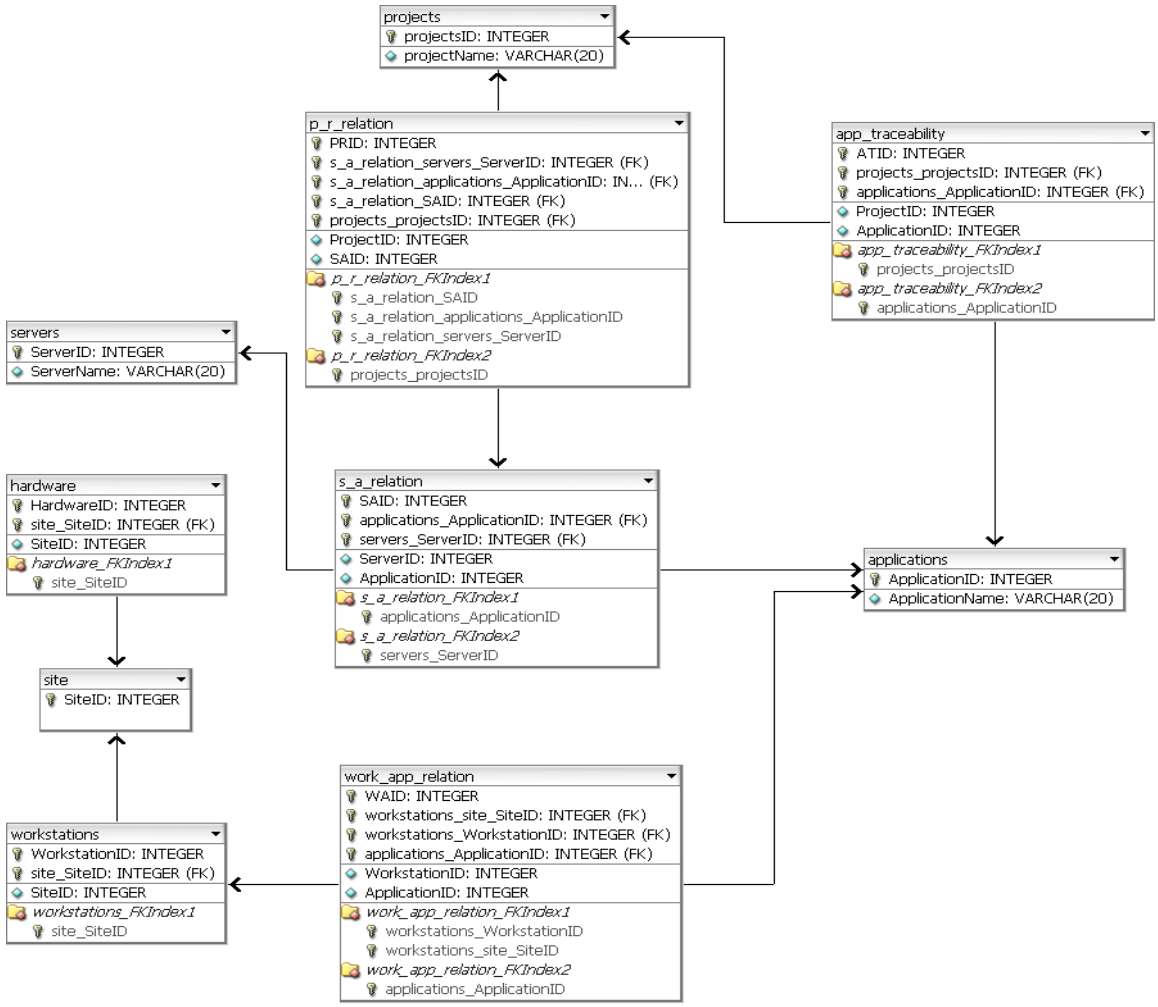


Figure 4-1 Configuration management MIS database table relationship

Because of the many-to-many relationship among these three tables, it must use two relation tables to the link these three. Because the applications mainly install on software server, therefore, the first binding relation table is relate to these two tables ("Application" and "Server"). Which can be regarded as is a installation entity. Each Project has many installation entities like this which need a relation table to link them. But when the inquiry is related to two relation tables, it needs to filter the redundancy result. For relation of "Workstation" table and "Applications" table, one relation table can link them.

For the data in each table, which is not permit to delete, therefore, developers don't need to consider the cascade deletes problem. In the modification operation, it only can make the modification to the non-essential attribute. For example, one cannot modify the OS version number attribute of "Servers". Once to replace the version data, one need to add a new record and this attribute is called the essential attribute. ID as primary key must be the essential attribute.

4.1.1. The single table inquiry traceability

When one need to update OS version, the old version attribute "isClosed" is set to true, simultaneously system will record the usage closure time for OS version (OS version's start time is already created when the record created). The traceability operation can be implemented by filtering "isClosed" attribute which is "false" through the key words inquiry. This kind of design base on the table data cannot be deleted, once uses can do the deletion, developers need to build a separately traceability table to trace the shut down record.

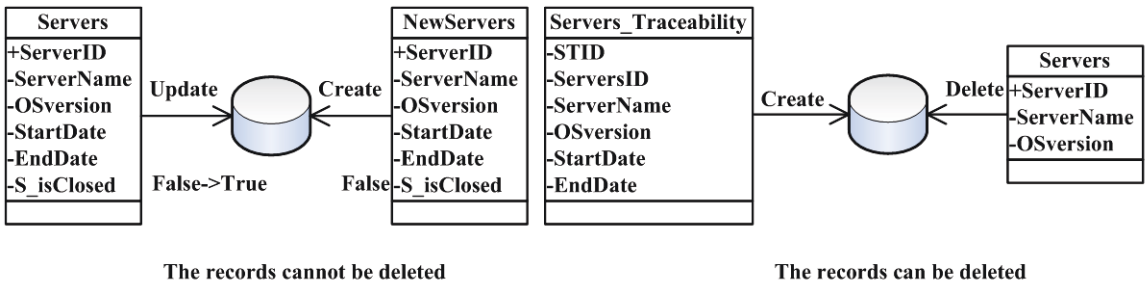


Figure 4-2 the design of single table inquiry traceability

4.1.2. The multi-table inquiry traceability

This kind of design only can satisfy the single table inquiry traceability demand. For designs the multi-table inquiry traceability, obviously this design method cannot satisfy the users' demand. For example, if the user need to trace that some project uses which version of the application, then only joined the related start time and closure time in "Applications" table

is not enough, because it is possible that many projects use the same version number application system. The attribute of Applications cannot relate to the “Project” table, because there are two relation tables between them. Therefore, in the multi-table traceability inquiry, in order to achieve the target, it must add one relation table which has the traceability attributes to connect “Projects” table and “Applications” table.

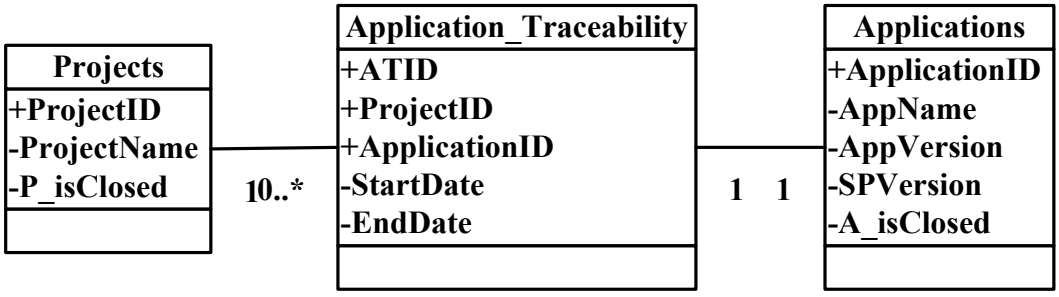


Figure 4-3 multi-table inquiry traceability design

The application version's traceability demand is: system can trace the application version number some project uses. In other words: Record this application version all servers use, the start time and closure time in certain project. When all servers did not use this version of application, one can define it as the closure time.

Because of the relationship between “Projects” table and “Applications” table, through the relation table, whether one can put the traceability attribute to store in the relational table? The answer is no. For example: When if some application software is installed on two different servers, but a certain project simultaneously uses the same version application software on the different server, this will have two records in “Pro-ReL” table (because there are two server), because each record will have one Application’s “StartDate” and “EndDate”, which will make the inquiry redundancy. In fact, when an application version updates, all the servers who use this application versions also need to update (It is possible to filter redundancy records through the DISTINCT key words in SQL sentence). Therefore, the relation table which includes traceability attributes is needed in order to confirm the traceability relations of “Projects” and “Applications” table.

Except for the tables which meet fixed demand, several system tables are also needed to establish (non-business logic table). They are “UserInfo” table, “UserRecords” table and “Fixoperation” Table. “UserInfo” table is used to store users’ information. “UserRecords”

table is used to save the inquiry information made by users themselves. “Fixoperation” Table is used for saving the inquiry information for all the users.

4.1.3. The loop problem of Multi-table inquiry

To do the agile inquiry in the system, it will involve the loop problem. In relational database, among tables there are several different relation paths, which form a loop. In the database model, the closed path is considered as a ring circuit, as shown in Figure 4-1, “Projects” table, “Relation” table, “Applications” table, “Workstation” table and “Work-App” table compose a ring circuit. If an inquiry encompasses many paths to relate, then the result mostly was wrong. Return result is an intersection of sets of two path result. Moreover, from the result it is very difficult to observe.

In some kind of stratification plane, this system lacked one layer like universe layer which is in the Business Objects product. The universe layer is used to connect the relations of business logic and database logic, and it is impossible to change the database structure, when the inquiry involves the loop. In the Business Objects procedure, it adds a universe layer. It can change the database structure mapping on the universe layer, and then do the corresponding inquiry operations on the universe layer. There are two methods to solve the inquiry loop problem:

♦ Use table alias

Table alias is a table substitution name, but on the universe layer, these two tables share the same point to the real table in the same database table. Then it can break the relations of that kind of inquiry loop. This is a very good solution, which can solve the database structural inquiry problem which caused by the limiting of SQL.

For a simple example: one user wants to order ticket, involves the city table, it will connect to three foreign key: the location attribute of user, the airplane ticket’s departure city and arrival city. Then among them there is a loop route surely. The method to break the loop is: make it as two tables which are called the user location table and ticket city table, but these two tables point to the same database table (city table). This is very good way to break the ring circuit.

♦ Use context path method

The context path was designed to one kind of rule: when there are many paths in the inquiry operation, the system will carry on the match operation to a special table, which includes the information for the path to carry on the inquiry, and then users can obtain the

inquiry result according to the context rule. But when the user selection inquiry involves two paths, the system will make a wrong inquiry. But using the alias will not have such problem.

Based on implementation of system, insert an intermediate layer also need further research, this thesis doesn't involve such demand. Without this intermediate layer, it has the possibility to make the wrong inquiry, which is inevitable. Therefore, when the users setup the inquiry, which involves loop inquire, he must check the result by manual, which is inevitable.

4.2. System model

For a MIS based on Web, the most important thing is to build the model of data system, which decides the data format in transmutation process. For application based on Web, data is transmitted on Internet between the Server and the Brower. The Client Side and the Server Side need to build a unified data interface.

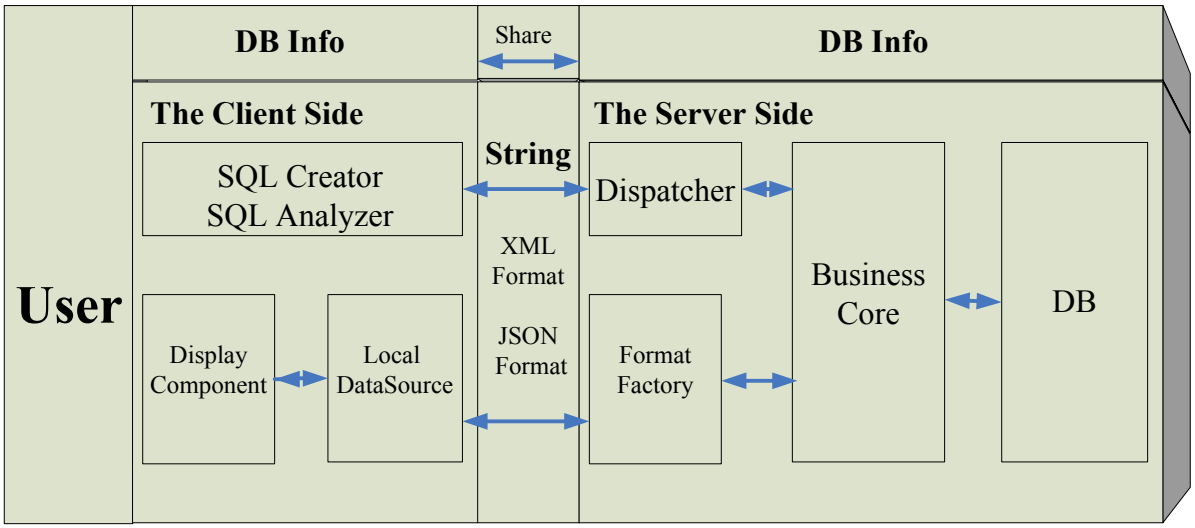


Figure 4-4 System model

In order to implement the agile inquiry, the important data core is SQL sentence. The Client Side need to provide the selected items name and the SQL sentence String to for the Server Side. Through unifying Client Side and Server Side interface, the Server can obtain the enough information to rebuild the result format for returning to the Client Side. Through different SQL sentence string, the Business Core can provide different result after accessing to Database. The Client Side needs to Share the DB information with the Server Side to control the unified record names.

4.3. The design and implementation of performance layer

The design of client side performance layer is quite simple, it only needs to transform from two kinds of user interfaces and user login interface. When the user enters the system, system generates user login interface instance. Using user name and password one can login, then the system generates user interface instance, and it will not destroy the login instance (not cancel thoroughly, just hide it). When user log out, the hided UI instance can be set to visible. This kind of model makes the user UI transform very rapid, does not need to generate the new instance. But with this way, the initialization method still needs to call before loading, otherwise, when the UI is showed again, it will hold a state which before logout, therefore, before activating UI, it must firstly carry on the initialization method.

4.4. The communication model under the SmartGWT

There aren't many requirements about user's permission. The administrator module is not a key module in this system. But regardless of login module and administrator module, it uses the communication framework of SmartGWT (bottom layer with the GWT framework) to carry on the communication between client and server. As the AJAX application development package, GWT causes the implementation of the communication becomes very simple.

4.4.1. Asynchronous communication under SmartGWT

Because RPC communication in the GWT framework is asynchronous call-back, this kind of asynchronous setting's advantage and disadvantage are also very obvious:

- **The advantage of RPC asynchronous communication in SmartGWT**

Asynchronous communication's advantages are very obvious. The client side only needs to define a service interface and its service asynchronous interface. The server side defines the method of the relevant service implements. After the client side carries on the server RPC long-distance call, it will send call information to the server, and the server calls the implemented method, after that it sends-back the results to the asynchronous interface method of the client side. This process to the user is transparent. At the same time, when the client side send call request to server, it can divide the multi-thread to deal with this operation, after that the main thread not need to wait until the server return back the data and keep always activate. This method can provide very good user experience.

Moreover, in the GWT RPC mechanism, it has provided transmits the java object operations between the client side and server side through the HTTP protocol. For the

developers, they only can use the data type or the object which supported by GWT as the parameter and return value. This communication mode is also mature, it can deal with polymorphism class hierarchical structure and the object chart loop, even can throw the exception step across the network.

- **The disadvantage of RPC asynchronous communication in SmartGWT**

The disadvantage of asynchronous communication is very easy to throw "Null Exception". When the server hasn't return the data, system carries on the setting or the update operation to the components, it is very easy to present the exception of the components are empty (Null) or the content is empty (Null), developers need to write massive codes to guarantee after obtaining the result data and then carries on the operation in the call-back result method, the asynchronous model always totally different with the synchronization mode.

Generally, GWT supports all the JAVA basic data type which is serializes. If user defines a special data object for transmission, which must be serialize through implementing the com.google.gwt.user.client.rpc.IsSerializable interface. GWT doesn't support the array to carry on the transmission, but can support the object type like ArrayList. The developers need to do the data transform to meet the demand.

At the same time, this communication mode must obtain the support of components, which make it easy to implement connect to server. Because GWT framework is actually a components integration package framework, its RPC communication mechanism can support integration with the data components. But in local DataSource framework of the SmartGWT, local DataSource don't support RPC directly, but it supports HttpRequest, RequestBuilder these types of calls, which are bind through URL. A service interface can only set to one URL, and it cannot map to the implemented method on the server, which make that use RPC communication to DataSource is very difficult.

4.4.2. Asynchronous communication in system use

In SmartGWT framework, components provide many methods to carry on HttpRequest and RequestBuilder calls, which can bind the data, components and communication methods. For example:

```
DSRequest requestProperties = new DSRequest();
requestProperties.setActionURL(URL);
editorForm.saveData (new callback(), requestProperties);
```

The above simple code implements to save data to URL, it creates a instance of DSRequest, and sets parameters, then it uses the component (EditorForm) to carry on the call procedure. The above method provides a call-back function and transmits a request, with which one can bind the transmission parameters.

The asynchronous communication mainly uses on system's data display components, which are used for display, modification, deletion and other operations to the data, but the different operation method of component will supplement the different parameter to carry on the transmission. It provides the parameter recognition (mainly on server side framework of SmartGWT enterprise payment version) in implementation of server side. The free personal version needs developers to define parameter recognition module.

4.4.3. The synchronous communication of SmartGWT framework

In SmartGWT framework, it is the same with GWT framework. it provides all the communication methods including HttpRequest, RequestBuilder and RPC which are asynchronous. To implement synchronize communication. It can use the mode of “Form” to submit Action. Through “FormElement” which is get from HTML Document to create a (URL) link to connect to server side. Then, it submits this Form to the Server side. Thus, the client side will wait for the server side until it get the response corresponding. The example code is as follows:

```
String url=”/”;
FormElement formElement = Document.get().createFormElement();
formElement.setAction(url);
formElement.setName (“ThisActionName”);
formElement.setMethod (“post”);
Document.get().appendChild(formElement);
formElement.submit();
```

4.4.4. The usage of synchronous communication in system

Synchronous communication here mainly used to the communication flow which must execute, for example: the communication of login operation, involves the communication of judgment operation (e.g.: Communication of permission judgment). If the login operation uses asynchronous communication to design, the UI still needs to the mask the operation components, when server response slowly, and with no mask to the components, the user may carry on the operation again, which will causes the problem of redundant communication, but the synchronous communication does not need to consider this exception.

4.5. Definition the transmission format of communication

When design communication transmission format, developers must consider the whole system demand and the whole framework demand. In the system requirements, it must implement the agile inquiry. The return result must be one kind of general format, but non-fixed transmission object. In the framework demand, the communication only allows transmitting serialize objects. Simultaneously it must consider that the client side's display component support this format.

In SmartGWT components, they mainly accept two kinds' formats of data: XML format and JSON format, and they can analysis through reading string of characters.

4.5.1. GWT data component and server communication activity chart

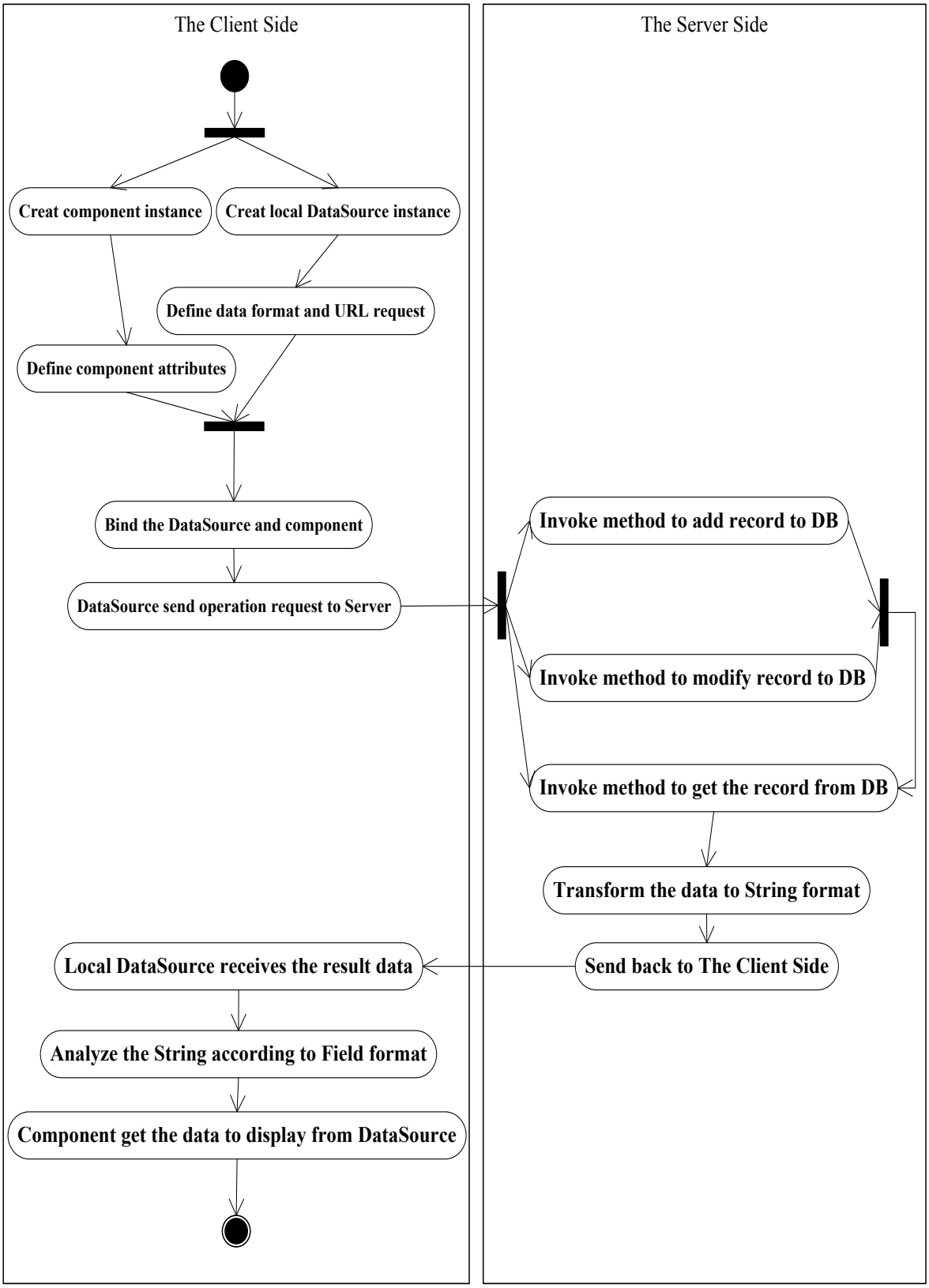


Figure 4-5 communication activity chart of component and server

4.5.2. TreeGrid component and XML transmission format

♦ **XML format**

XML is the abbreviation of eXtensible Markup Language. Expansion marker language XML is a simple data storage language, which uses a series of simple tag to describe data, and these tags be established with a convenient mode. Although XML takes more the space than the binary data, XML is very easy to understand and to use. It makes the computer very simple to produce and read information, and ensures that the data is precise. XML has avoided the omissions of general language design: It is expandable, platform independent and can support the internationalization format. It has not the Unicode incompatible problem.

♦ **Analysis XML format of TreeGrid component**

TreeGrid component is the same as other communication components in SmartGWT. Through the URL set in local DataSource to access server, it can reads the XML format data. After extracts the data in local DataSource, component analyzes the internal attributes, extracts two essential attribute values: “ID” value and “ReportsTo” value. The following are some codes of database table XML for using in TreeGrid:

```
<info>
<Database>
  <Block>
      <tid>100</tid>
      <ReportsTo>1</ReportsTo>
      <title>studentInfo</title>
      <icon>icon_table.gif</icon>
      <type>table</type>
      <parent>1</parent>
  </Block>
  <Block>
      <tid>101</tid>
      <ReportsTo>100</ReportsTo>
      <title>studentID</title>
      <icon>icon_field.gif</icon>
      <type>field</type>
      <parent>studentInfo</parent>
  </Block>
</Database>
</info>
```

Here tid is correspondent of “ID” value, represents for markup leaf ID attribute. The value of ReportsTo is correspondent of “ReportsTo” value, which mainly representes father

node ID of current node. Usually root node's ID was set to 1, node ID cannot be the same, which must be primary key.

The standard definition of data format and data attribute in DataSource is as follows:

```
DataSourceIntegerField IdField = new DataSourceIntegerField ("tid", "tid");
IdField.setPrimaryKey(true);
IdField.setRequired(true);

DataSourceIntegerField reportsToField = new DataSourceIntegerField ("ReportsTo",
"ReportsTo");
reportsToField.setRequired(true);
reportsToField.setForeignKey (id + ".tid");
reportsToField.setRootValue ("1");
```

What is the most essential is to set "ForeignKey" and "RootValue", and the TreeGrid can display the related nodes serialize. TreeGrid doesn't need to set parameters for displaying data, only needs to define display name and the attribute name of data Record. Moreover, through configuring DataSource attributes, DataSource can read data asynchronously (After the user clicks open the node, TreeGrid requests sub node's value from the server side. It follows Figure 4-3 the transmission diagram to carry on the communication with server.

4.5.3. ListGrid component and JSON transmission format

♦ JSON format

JSON (JavaScript Object Notation) is one kind of lightweight's data exchange format, which is very easy to read and to compile. Simultaneously, it is also easy for computer's generation and analysis. It based on JavaScript programming language, Standard ECMA-262 3rd Edition-december 1999 subsets. JSON uses an independent language text format, but also uses the C language family's custom (including C, C++, C#, Java, JavaScript, Perl, Python). These characteristics cause JSON to become the ideal data exchange language.

Table 4-1 two kinds of constitutions of JSON

Name/value (A collection of name/value pairs)	In the different language, it was considered as object, records, structure, dictionary, the Hash table, keyed list and associative array.

An ordered list of values	In the majority of languages, it was considered as array.

These are the common structure of data. In fact, the majority of modern machine language supports them in some form, which causes that it is possible to transfer this kind of data format among these constitutions programming languages.

JSON has many forms as follows:

The object is a disorder “name/value” set. An object by “{” start, “}” finished. Each “name: value” as a set. Among “name/value” it uses “,” to separate.

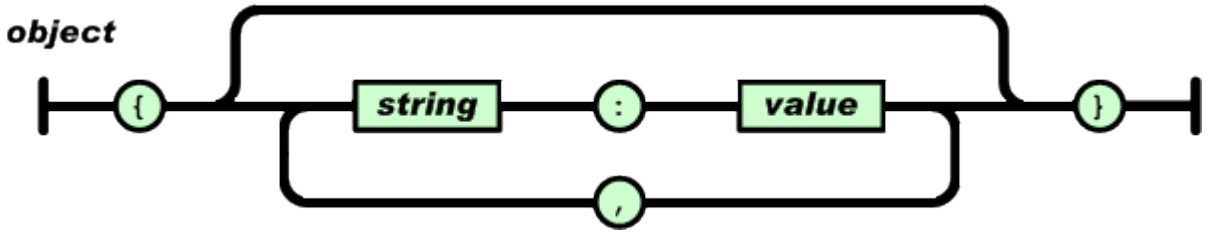


Figure 4-6 JSON format - 1

The array is the values’ ordered set. An array start with “[”, ends with “]”. Among the values it uses “,” to separate.

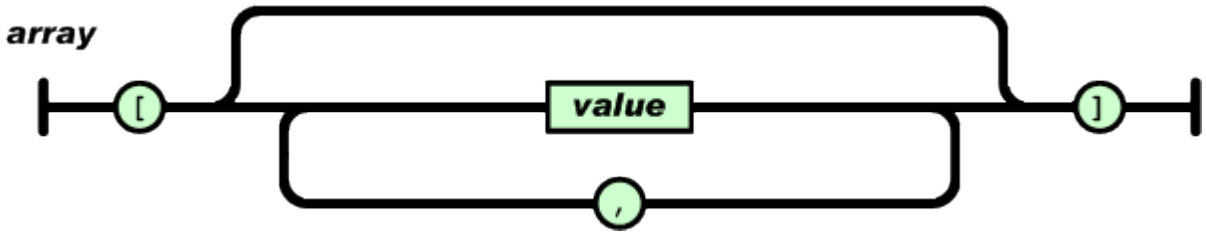


Figure 4-7 JSON format - 2

Compares with XML, the JSON’s superiority is being succinct of the format. The generation data size changes little, and it is easy to process. Especially in processing massive complex data, this superiority appears very obviously. From the supports of various browsers, XML needs DOM to analyze, and different browser manufacturer's support's different XML-DOM, which gives the WEB developers more difficulties, especially after AJAX is widely used, many developers did not return XML results, but return HTML piece or serialized strings format, when the product will only limited by use IE then uses XML to transmit. JSON does not have this problem in this aspect. It doesn’t need other interpreter, which obtains the support of majority languages and browser, it used widely.

- ListGrid component and JSON format

ListGrid component mainly is used to display data. It must through the local DataSource to communicate with the server. Because of using local DataSource, it supports XML and

JSON format. In local DataSource, one can define data's display format. After binding component and DataSource, the DataSource will extract data from server through the URL which has been set. After analyzing data format, the result data is displayed on the component according to the defined format.

The definition of data format's method is the same:

```
setDataFormat (DSDataFormat.JSON);
DataSourceField userNameField = new DataSourceField ("userName",
FieldType.TEXT, "Name");
userNameField.setPrimaryKey(true);
userNameField.setCanEdit(false);
DataSourceField userPasswordField = new DataSourceField ("password",
FieldType.TEXT, "Password");
DataSourceField authorityField = new DataSourceEnumField ("authority",
"Authority", 50);
```

When setting DataSource, it must set the data format first. Simultaneously, it needs to set data type data length.

In ListGrid component, it has defined many data methods for various operations. Developers can carry on "updataData" and "removeData" operation to ListGrid directly, in data insertion operation, one also can make use of DynamicForm which is binding local DataSource to invoke "saveData" operation. Simultaneously, ListGrid also provides one kind of modification operation in order to modify data immediately, which by using saveAllEdits method to submit the operation. Through binding with local DataSource, these methods of ListGrid can call the related methods of DataSource to request to server directly. Through analyzing the request information, server retransmits the parameter separately to call the different methods to response.

After ListGrid carries on the operation, the local DataSource transmits the information to server side (only for reference):

- **Delete**

authority=Admin&password=12345&userName=JING&%2429a=null&%2429a=null

- **Add**

userName=tt1&password=tt2&authority=Admin

From the comparison of above two kinds of operations in SmartGWT free version, it shows that it only provided a transference parameter interface, hasn't defined about the parameter of operation. Therefore, the parameter needs to add by manual. Through binding the DSRequest object, one can add the URL parameter before the transmission. All the components which can interact with server support the DSRequest object.

The definition is as follows:

```
DSRequest requestProperties = new DSRequest();
requestProperties.setActionURL(URL);
```

One can define the system internal operation parameters in a URL String. After receiving the parameters, the server analyzes it and invokes related method call.

4.6. The implementation of client side

The client side completely uses the SmartGWT framework to implement. The version of SmartGWT is LGPL version (personal development free version, which omits server side many data functions). The version number is 2.1.

4.6.1. Database of client side use inquired mode basically

The agile inquiry is after user choosing the fields and inquiry conditions, the system can automatically accurately get the data results from database. For one inquiry statement sentence:

```
Select fieldl, field2, field3, field4
From tabl, tab2, tab3
where condition 1 and condition 2 and conditions 3
```

In the table above, field "fieldl", "field2", "field3" and "field4" are the fields which selected by user. Table "tabl", "tab2"and "tab3" are the selection fields' correspondence table, the condition following "where" like "tabl.ID = tab2.ID and tab2.ID = tab3.ID", which is given according to the relationship of the tables. The method search the foreign key of the selected table and find the relationship with others tables. It will add to the condition part automatically. The condition 2 is the conditions which user gives. The condition 3 is the expressions of key words or the fields in same table formed, for example: tabl.ID= '1' or tab2.field3=888. Any complex inquiry may implement through the above statement (the results of one statement can be the parameter of another statement), therefore, so long as one

can describe the inquiry expression accurately, he can obtain the correct inquiry results. For different structure's database, all fields of inquiry expression above and condition may change, but the inquiry model will not be changed.

4.6.2. The client side inquired the design of UI to implement nimbly

The agile inquiry design of client side mainly refers to Business Object system's inquiry interface, which uses Drag & Drop mode to implement. The users only need to select the table item and drag them to the selection area, and then they can obtain the corresponding inquiry result. The UI design as follows:

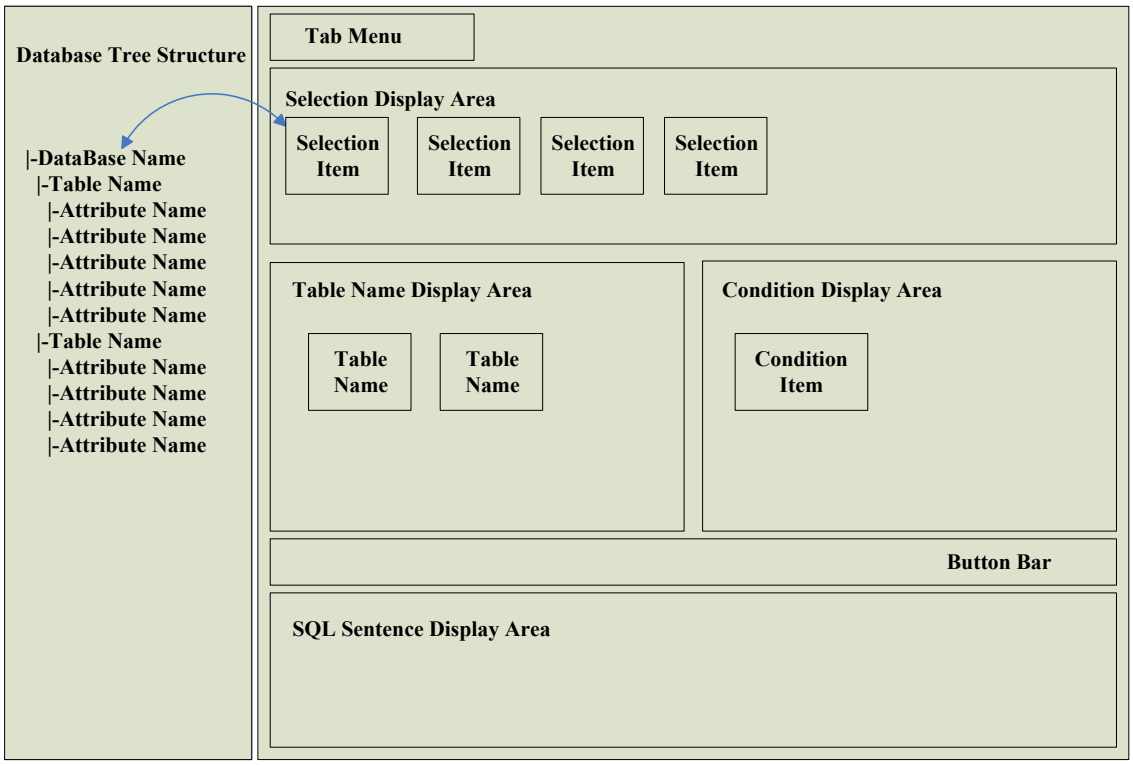


Figure 4-8 the UI of the agile inquiry

In the chart above, the database tree structure and selection display area and condition display area are the interactive sector for user. Table display area and SQL statement area are generated according to the selections.

4.6.3. The implementation of user's agile inquiry on the client side

The client side uses Drag-Drop mode to implement the user operation. From the left side's database tree structure, users drag the items to the right side "selection display area" to select the data which has the information they want to view. The left tree uses TreeGrid component, which mainly through the XML format reads from server side to save to local DataSource. The interface of right side uses TileGrid component, it needs to set several

DataViewField items to setup the display layout. At the same time, the receiving component TileGrid need to set a data space receives the storage. Because it doesn't obtain the data from the server, but from others components, therefore, developers can give a new RecordList array to store data.

The example of TileGrid data display side's setup as follows:

```
DetailViewerField iconField = new DetailViewerField ("icon");
iconField.setType ("image");
DetailViewerField parentField = new DetailViewerField ("parent");
parentField.setDetailFormatter (new DetailFormatter() {
        public String format (Object value, Record record,
                              DetailViewerField field) {
            return "-" + value + "-";
        }
      });
DetailViewerField nameField = new DetailViewerField ("title");
nameField.setDetailFormatter (new DetailFormatter() {
        public String format (Object value, Record record,
                              DetailViewerField field) {
            return "<font color= \ "read \ "><b>" + value + "</b></font>";
        }
      });
selectGrid.setFields(parentField, iconField, nameField);
```

For TreeGrid component, it is also through the local DataSource to access the server, which needs to define the related attributes of the XML format string. At the time of definition, developers need to set the same loading attributes for TreeGrid component and TileGrid component. Even if there are some attributes which not need in TreeGrid, it also can load first (the component can save the attributes and not to show them). When one drops the items, TileGrid will preserve all attributes, and some parts of TileGrid component display attributes can be got from the object dragged. The object makes a bridge for two components.

For TreeGrid component, the "tid" attribute and "ReportsTo" attribute are needed. The name can be different, but developers need to through setForeignKey() and setRootValue() methods to relate to "tid" attribute. These two attributes can be read automatically by component. The "tid" stands for the node identity, and the "ReportsTo" attribute stands for father node identity. This kind of represented method is simple.

```
DataSourceIntegerField IdField = new DataSourceIntegerField ("tid", "tid");
IdField.setPrimaryKey(true);
IdField.setRequired(true);

DataSourceIntegerField reportsToField = new DataSourceIntegerField ("ReportsTo", "ReportsTo");
```

```
reportsToField.setRequired(true);
reportsToField.setForeignKey (id + ".tid");
reportsToField.setRootValue ("1");

DataSourceTextField titleField = new DataSourceTextField ("title", "title", 128);
DataSourceTextField iconField = new DataSourceTextField ("icon", "icon", 128);
DataSourceTextField iconpField = new DataSourceTextField ("iconp", "iconp", 128);
DataSourceTextField typeField = new DataSourceTextField ("type", "type", 128);
DataSourceTextField parentField = new DataSourceTextField ("parent", "parent", 128);
```

For the Drag-Drop method in SmartGWT framework, the object dragged for the developers is transparent. Developers only need to set the interface for these two components, and configure the corresponding parameter values alternately. At the same time, it is not all the components can make the data interactive, generally only limits by the data components, for example: TreeGrid component, TileGrid component, ListGrid component, HStack component and so on, they can carry on the interactive operation of data, meanwhile they must have the display function. Otherwise, after receiving data, they cannot show the results.

4.6.4. The foreign key relations implement in agile inquiry

In tables of database, in order to relate two related tables, it uses "foreign key" to relate. The official concept of "Foreign key" is: If one public key word is the primary key of the related table, then this public key word is called the foreign key in its table. "Foreign key" represents relations between two relational tables. The table who takes the "foreign key" as the primary key is called the main table. The table that has this "foreign key" is called Sub-table. "Foreign key" is also called "Foreign primary key".

In agile inquiry, the processing of "foreign key" is quite important. If one don't process the foreign key, then after the selection operation, the SQL sentence will lack the relation constraint of the foreign key, then the result inquired makes the cross connection to these two tables directly, which is called "Cartesian product". Such result is meaningless. Therefore, before the addition and the deletion operation to the "From" table, developers can carries on the foreign key filtration. In the java base SQL class package, there are some methods for database information. Through getMetadata method, one can extract all the database information of system table.

The related code is as follows:

```
Connection con= DBConnection.getConnection();
dbMetaData = con.getMetaData();
   public void getAllExportedKeys (String schemaName, String tableName)
   {
      try
      {
         ResultSet rs = dbMetaData.getExportedKeys (null, schemaName, tableName);
```

```
        while (rs.next())
        {
           String pkTableCat = rs.getString ("PKTABLE_CAT");
           String pkTableSchem = rs.getString ("PKTABLE_SCHEM");
           String pkTableName = rs.getString ("PKTABLE_NAME");
           String pkColumnName = rs.getString ("PKCOLUMN_NAME");
           String fkTableCat = rs.getString ("FKTABLE_CAT");
           String fkTableSchem = rs.getString ("FKTABLE_SCHEM");
           String fkTableName = rs.getString ("FKTABLE_NAME");
           String fkColumnName = rs.getString ("FKCOLUMN_NAME");
           short keySeq = rs.getShort ("KEY_SEQ");
           short updateRule = rs.getShort ("UPDATE_RULE");
           short delRule = rs.getShort ("DELETE_RULE");
           String fkName = rs.getString ("FK_NAME");
           String pkName = rs.getString ("PK_NAME");
           short deferRability = rs.getShort ("DEFERRABILITY");
           System.out.println (pkTableCat + "-" + pkTableSchem + "-" + pkTableName + "-" +
pkColumnName + "-"
                 + fkTableCat + "-" + fkTableSchem + "-" + fkTableName + "-" + fkColumnName + "-
" + keySeq + "-"
                 + updateRule + "-" + delRule + "-" + fkName + "-" + pkName + "-" + deferRability);
        }
     } Catch (SQLException e)
     {
        e.printStackTrace();
     }
  }
```

The methods above can be used for reading complete "foreign key" attributes. For getting the relations, developers only need to obtain "foreign key" table name. Before the addition of table, system need to read the database information to obtain "foreign key" attributes' list, and then compare the name and existed table names, if has the same table, adds the corresponding condition string in the SQL sentence, which guarantees SQL statement links successfully.

4.6.5. The activity chart of agile inquiry on client side

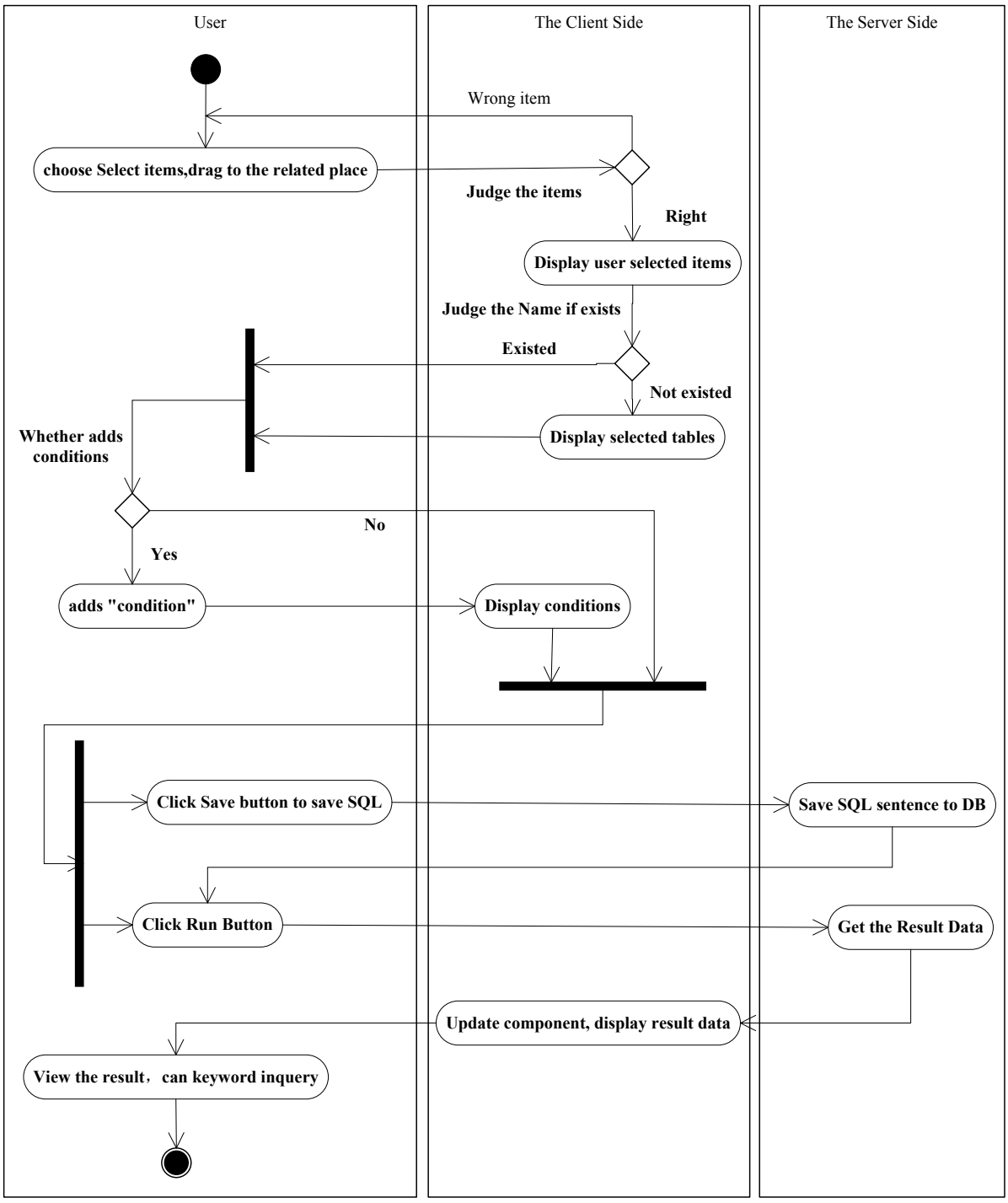


Figure 4-9 the activity chart of agile inquiry client side

According to figure 4-9, users only need to add inquiry items “selection Items” to finish the inquiry. The client side exams whether the content is valid before an operation is done. If the content is not permit by the principle, it will not be added. After adding inquiry items “selection items”, the client side will judge the name of table by selection items, and add to the corresponding place. If the table exists, it cannot be added. The system also will check

that whether the tables have foreign key relationship. If exists, then it needs to add the corresponding foreign key condition to the SQL sentence.

4.6.6. The multi-language i18n design's implementation in performance layer

The multi-language design is the demand proposed by more and more cross international business. One can through inherit to the Message class of GWT i18n package to implement this requirement. The detailed design please sees the next section. For the performance layer, developers can set many configuration language files to define the names, the contents and the text of related module. They can through a unified interface to call, and the method will insert into to component automatically. If one needs to change the language of system, he only needs to add the corresponding configuration to implement configuring language.

4.6.7. MVC model on the performance layer

The performance layer's code-package strategy uses the page-package strategy. For each page, it has a code package. There are four modules for each package, which is: the page module, controller module, commands module and communication's call-back interface, if needed, one may add the model module (for instance: To selects the box controls data processing and local DataSource), mainly implementation as follows:

- Class chart of one module (takes login as example)

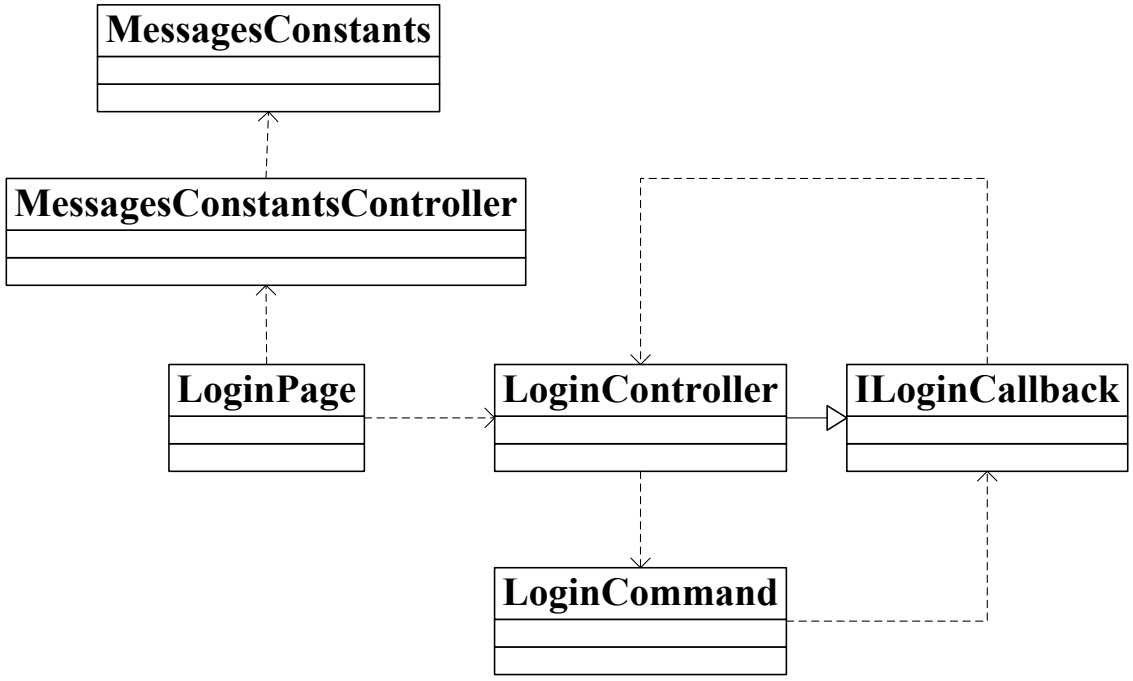


Figure 4-10 class chart of MVC architecture design on client side

In Figure 4-10, MessagesConstantsController class and the messagesConstants class separately are the multi-language control class and its interface. There also have many

language configuration files. Inside the configuration files, it made the definitions for the marked name (marked name = language content). The markup name methods are defined in the interface, and system can invoke the method from the control class directly.

For another four classes, the "LoginPage" class is the UI file, which includes many UI components. In the UI file, developers don't allow carrying on any operation control and logic control, the event listener places in UI class, but doesn't carry on the processing operation to the data. It transmits data to "LoginController" to carry on logical processing. In "LoginController" class, data transmitted need to be packed, analyzed, validated and assembled. Then it starts the communication Action. It has implemented the "ILoginCallback" call-back interface, and transmits itself as an interface through Action method to "LoginCommand" class. In "LoginController" class, developer need to instantiate a "LoginCommand" instance. In "LoginCommand" class, it starts the asynchronous communication to server through calling "LoginService" and its asynchronous interface "LoginServiceAsync", it also need to instantiate call-back function in the command method, which implements the success call-back and the failure call-back method. Because "LoginController" has implemented the "ILoginCallback" interface, therefore, in the call-back method of "LoginCommand", developers can invoke the methods in "LoginController" class through the interface instance which is passed by "LoginController". And the method can do the update UI component operations.

- The sequence chart of MVC module on client side

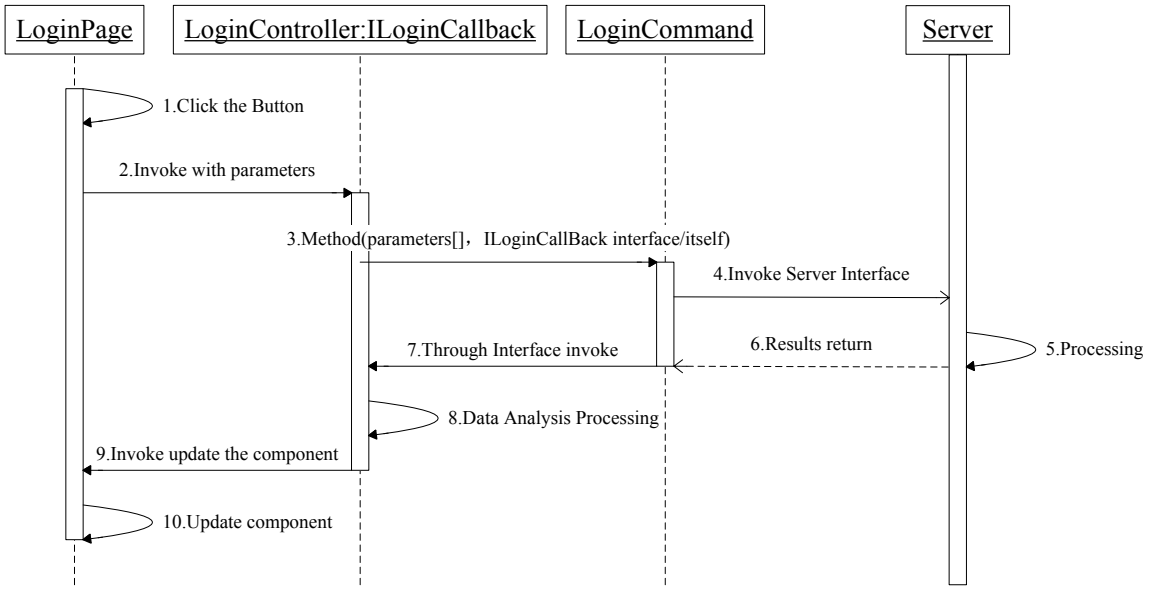


Figure 4-11 the sequence chart of MVC module

From Figure 4-6, one may see that layer level very clearly. “LoginPage” is for the UI design and component response, the “Controller” class mainly uses for transmit the Action command after processing data. “LoginCommand” is responsible for executing the command and communicating with server. It returns the result back to the “Controller” class, after “Controller” processing the format. It calls the update methods of “LoginPage” to update the component. This kind of structure is advantageous to enhance the readability and maintainability of code.

This module based on components integrated well with RPC, the level clear and it has high maintainability. Other pages can be designed using the same design model.

4.7. The implementation of server side

4.7.1. The selection of server side framework

The server side communication interface's framework may can use simple Servlet or the STRUTS2 framework to implement, which is mainly used for controlling the request to the server. Because the performance layer uses GWT to implement, the model layer used Hibernate and JDBC, and the system doesn’t need the interceptor function of STRUTS2, moreover, agile is the important thing in the design demand, therefore, it need to reduce the use of framework as few as possible in the agile inquiry module, which avoids the framework to limit the flexibility.

But in the database aspect, developers can use Hibernate and the custom DAO method to implement the database layer. Hibernate framework maps the database table data to PO (Persistent Object) layer. After operating the database, server obtained the result set need to converse the type to PO. When the client side carries on the agile inquired, the result set may not be a PO object. Therefore, it only needs to use the most basic SQL sentence statement to implement. DataSource layer transforms the data to ResultSet object, and Business layer reads the data and assembles to the JSON format string or XML format string which the client side can analyze and distinguish.

From design to the framework integration and the flexibility’s feasibility research still required the massive time. This thesis has not done further study to the framework integration research. The feasibility of using the framework to implement the agile inquiry, the thesis has the reserved attitude. This design can also provide the basis for the future framework implementation of agile inquiry study.

4.7.2. The implementation of the layers in server side

The server side can divide into five layers on details: the dispatcher layer, analysis layer DAO layer, PO layer and database layer, the database layer and PO layer are fixed which generate automatically. Other three layer need to build by manual. In DAO layer read-write database method can be reused, through transferring parameters the method in analysis layer can invoke the common methods in DAO layer directly, which reduces redundant development of massive database operations.

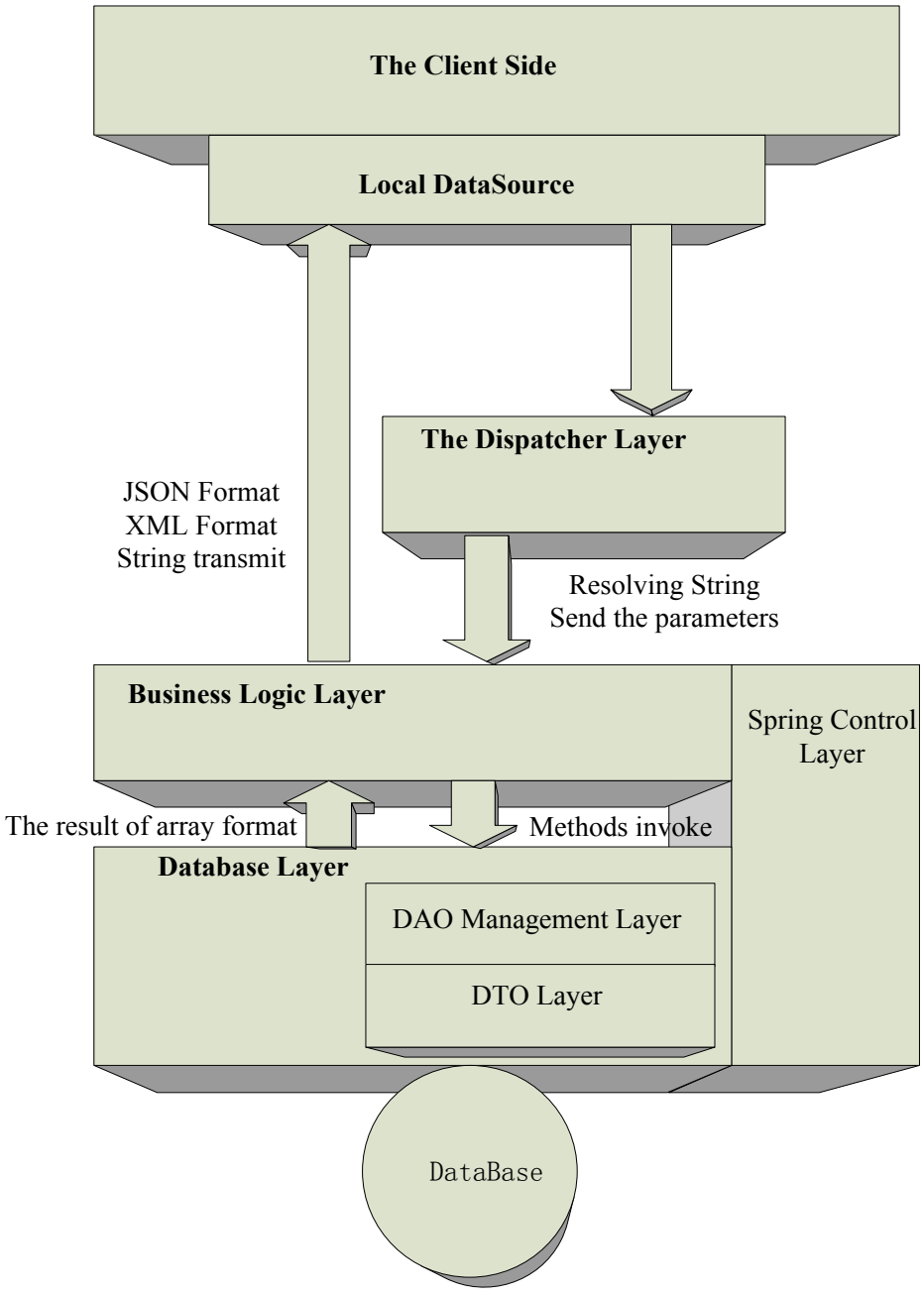


Figure 4-12 server side lamination chart

In Figure 4-4, the dispatcher layer is the service layer of general framework, analyze layer is Domain layer of the general framework. The DAO management layer and PO layer belong to the database layer which is linked by database. In the figure, analyze layer can return result data directly to the client side and doesn't return back to the dispatcher layer again, which reduces the times of method call.

- **Dispatcher layer of server side**

The dispatcher layer mainly uses Servlet to implement, which is because the local DataSource on client side must through URL address to access the server. On the server side, one can uses getWriter to get the Writer, and invoke println() method to return the string result to the client side.

In dispatcher layer, one needs to analyze the user's request. In the URL, developers can define the parameters for the operations to make the server side know the objective of the user request. Dispatcher layer can resolve these parameters and invoke different methods, here is the example:

Table 4-2 URL fixed parameter lists

Parameter name	Parameter value	Note
op	create	New a record operation
	modify	Modifies a record operation
	delete	Deletes a record operation
	query	Inquiry operation
table	Shows the name	The operation name of table's, is mainly used in the user fixed demand
	null	Used for agile inquiry, does not need to designate the table the name
sqlsentence	sql statement	Used for agile inquiry
selectNum	select integer	Used for agile inquiry

The server dispatcher layer through resolving the request from the user, according to the different parameters, invokes different methods access the database with the related parameters. In the design part, it can be divided into four different Sevlet to accept the different requests.

♦ **Business logic layer in server side**

The business logic layer calls the DAO layer methods to carry on the database operation directly. There are mainly two methods: uses sql statement through JDBC to link to database, and uses Hibernate framework, which uses Hibernate query language (HQL) to operate the POs.

For the first method, firstly, it needs to analyze the SQL sentence to get the array list of "Selected items". Then through the SQL sentence it uses JDBC to connect to database and get the result set. After receiving the result data, the Business Layer transforms the result to the JSON format or the XML format, and through the respons.getWriter ().println () method to send back the format string to client side.

For the second method, the Business Layer can call the methods in DAO Layer directly to get result set Iterator, and then converts it to the PO object. The same with the first method, it need to transforms to JSON format and XML format string for transmission. For the transform operations, we can use another tool package to implement, for example one can use JACKSON package to transform the PO array to a JSON string. Hibernate framework mainly is used in the user the fixed demand, for example: the addition, deletion and modification operations of each table. With it developer can use very few Hibernate code to implement the corresponding function.

♦ **Spring control layer on server side**

Through using the Spring framework to integrate Hibernate framework, it can make it very easy to configure the relationship among the classes. Spring can through some configuration code to define and bind the dependant relations in the XML configuration files, and doesn't need to indicate beans' attribute and constructor. The Bean attribute can through attribute name and attribute type to implements binding automatically. Spring also can detect automatically to set the binding mode, which can choose a kind of suitable operational mechanism for itself. Spring transforms the layers' objects from the loose coupling's mode to a tight mode. The Dispatcher object does not need to care about the implementation of Business object. The Business object does not need to care about the Persistent Layer's specific implementation. All the instances of DAO are invoked by DAOFactory completely. When the module needs to refactor, the code rewriting quantity will reduce greatly.

For the Business logic layer, Spring controls the dependence relations, which has implemented well the loose coupling design mentality. Through a XML configuration file, it can controls all the POs and DAO beans, the basic code is as follows:

```
<beans>

<bean id= " myNamingStrategy" class= " org.hibernate.cfg.DefaultComponentSafeNamingStrategy"/>
//defines the default the name strategy
<bean id= " sessionFactory" class= "
org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean " >
                <property name= " dataSource" ref= " dataSourceHibernate"/>
                <property name= " namingStrategy" ref= " myNamingStrategy"/>
                <property name= " annotatedClasses " >
                        <list>
                                <! -- Metamodel Hibernate beans -->
                                <value> com.areva.cm.se rver.dto. Userinfo</value>
                 //sets the corresponding PO class
                        </list>
                </property>

                <property name= " configurationClass " >
                        <value>org.hibernate.cfg.AnnotationConfiguration</value>
                </property>
                <property name= " hibernateProperties " >
                        <props>
                                <prop key= " hibernate.dialect " >$ {hibernate.dialect} </prop>
                                <prop key= " hibernate.show_sql " >false</prop>
                        </props>
                </property>
        </bean>

        <bean id= " transactionManager" class= "
org.springframework.orm.hibernate3.HibernateTransactionManager" >
                <property name= " sessionFactory" ref= " sessionFactory"/>
        </bean>

        <bean id= " MyAbstractDao" abstract= " true " >
                <property name= " sessionFactory" ref= " sessionFactory"/>
        </bean>

        <bean id= " UserinfoDAO" parent= " MyAbstractDao" class= " com.areva.cm.se
rver.dao.UserinfoDAO"/>
      //defines the DAO beans
</beans>
```

In DAOFactory, it resolves the XML file, and then it defines each DAO's extraction method.

```
private static ApplicationContext springContext = new ClassPathXmlApplicationContext
("applicationContext-DAO-Hibernate.xml");
        //defines the Spring configuration files, is mainly used in Hibernate
public static UserinfoDAO getUserInfoDAO() {
        return (UserinfoDAO) springContext.getBean ("UserinfoDAO");
} //extracts DAO through the name
```

The basic concept Inversion of Control model (which is also called as Dependency Injection) is: Don't create the instance, but describe the way to create. In the code, it doesn't need to link to the instances and services directly, but describes which instance uses which service in the XML configuration files. The vessel (in the Spring framework, it is the IOC vessel) is responsible for relating these dependence relations together.

In the model IOC, the vessel creates all the objects and sets essential attributes to link them together, and decides when to invoke the method. The code above implemented the factory class to extract the UserinfoDAO instance through the DAO name in the XML file directly. It doesn't need to create a UserinfoDAO instance on the Business logic layer. Other DAO instances are the same.

♦ **DAO management on the Database Layer**

The DAO management layer is mainly to do the database JDBC operations and Hibernate operations. Business layer can deal with the PO objects. The DAO management layer need to access database through HQL (Hibernate) and SQL (JDBC) and return data Iterator to the Business layer. For Hibernate operations, developers can extract massive reusable methods, which can greatly reduce the quantity of code. For JDBC operations, the result sets will return to the Business layer to resolve.

♦ **PO Layer (DTO Layer)**

The PO layer stores all the PO objects which are generated automatically by Hibernate Tools. Developers don't need to write these codes. If the POs need to transform to JSON format with the third-part frameworks such as JACKSON, which must declare to the object with:

```
@JsonIgnoreProperties (value= {"hibernateLazyInitializer"})
```

Otherwise, the transform processing will throw a Hibernate Exception. The POs is very inconvenient to the agile inquiry, therefore, the agile inquiry module only uses the JDBC to create the connection to database. For another fixed operations to the table, one can directly operate the PO to implement.

4.7.3. The function implementation on the server side

In the dispatcher layer of the server, we use three different Servlets to carry on the dispatcher work. Each Servlet has instanced corresponding Business layer's objects. For example, the ReaderAction instanced the ReaderAPI instance and QueryAPI instance in order

to carry on the Reader user's operations and agile inquiry operations. The WriterAction instanced QueryAPI and WriterAPI object, WriterAPI was inherited from ReaderAPI, because Writer has all functions of Reader user, and he also has his own writing operation.

For agile inquiry, QueryAPI needs instance DAOFactory to carry on operations to the database. Developers can use DBConnection and DBConfigHelper to simplify the database configuration work. After obtaining the data, it needs to analyze the SQL sentence in the QueryAPI instance in order to obtain the fields to assemble the format string.

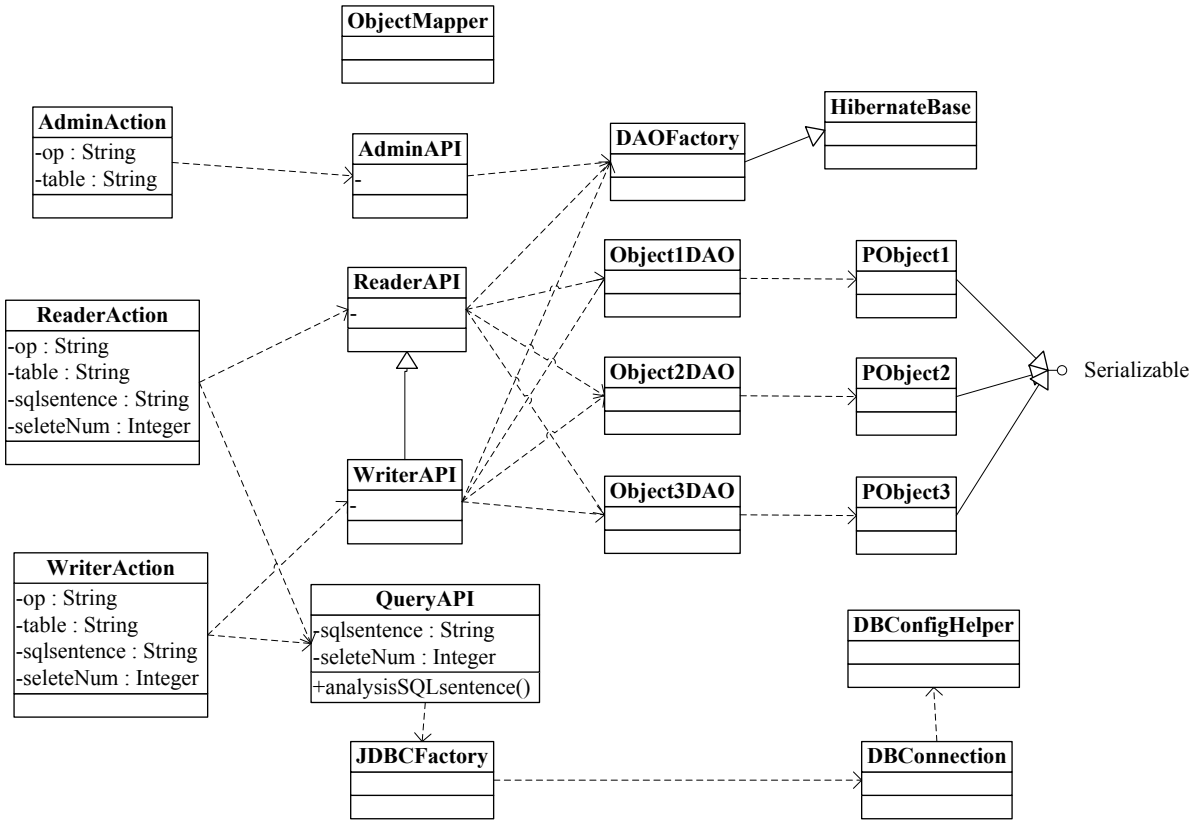


Figure 4-13 server end class chart

For another fixed database operations, it selects Hibernate framework to build the Database layer, and uses Spring to control the dependant relations, each DAO class maps to an DAOBean, which includes one data object and the related operations. Each DAO through inheriting HibernateBase class (which inherits in the Spring frame HibernateTemplate) to manage transaction, for example: Creates SessionFactory, opens the session and starts and closes the transaction operations.

In business layer, AdminAPI, ReaderAPI and WriterAPI may call the extraction methods directly in DAOFactory class to get back corresponding DAO instances, which can carry on the different operations to the POs or Database. Through transferring related parameters to

invoke related method and then extract result sets' iterator. One can through the static method of ObjectMapper class to transform Object ArrayList to JSON string.

Because it needs to implement the keywords inquiry on the client side, therefore, the paging function doesn't be used in the system. If one needs paging function, he can only transfer the "start" and "limit" parameters through the components of the client side. These parameters define the quantity of records at one time. When system uses paging function, the keywords inquiry is only for current page's inquiry. If need to implement the keyword inquiry to the complete data, it must download all the data to the local DataSource at one time.

4.7.4. The sequence chart of agile inquiry

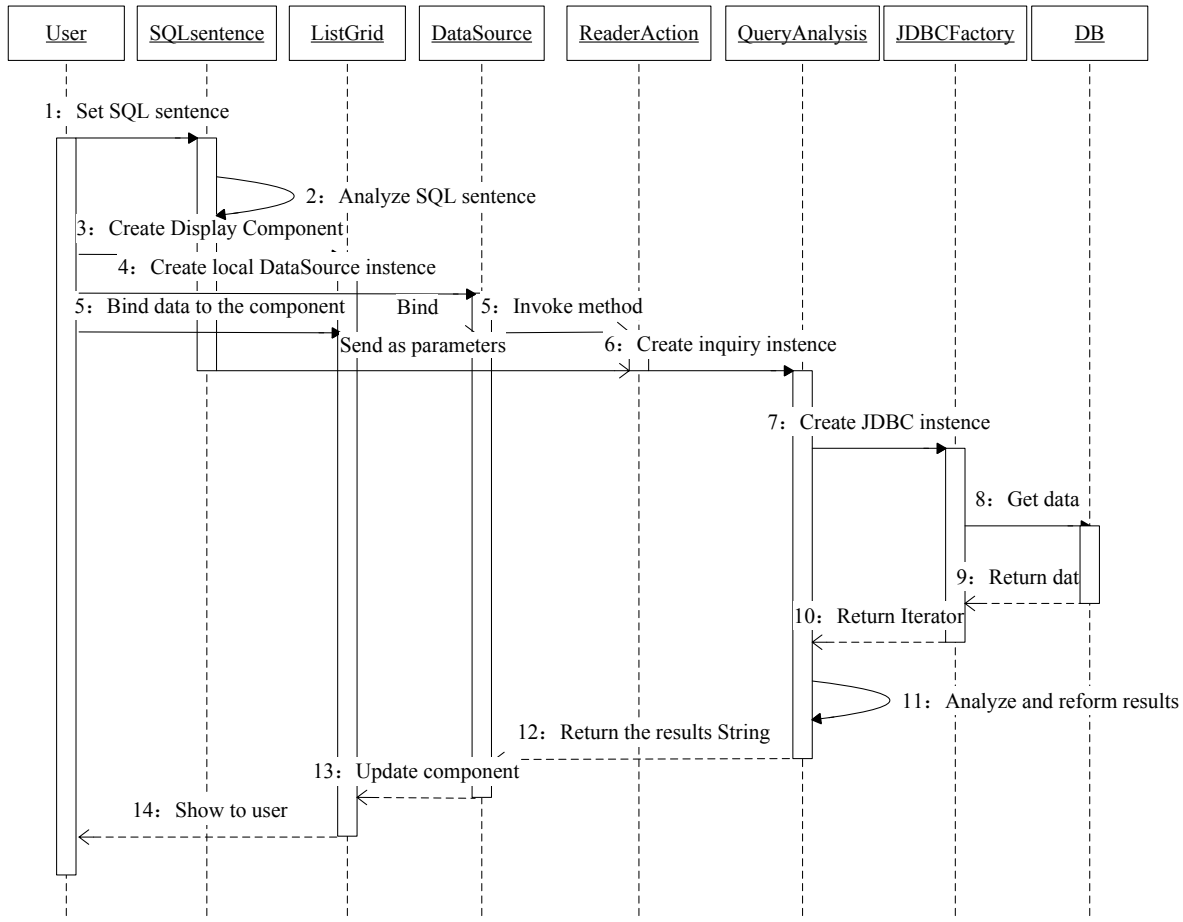


Figure 4-14 the sequence chart of agile inquiry

The communication mode of inquiry is asynchronous, which will not affect user's experience. In the fifth step, communication with server is the sub- thread operation. It builds a new thread to communicate with server, and then transmits the SQL sentence instance to the server side, after analyzing and assembling the data results. The server returns them to the client side and then displays. Server side's pressure lies in the accessing and inquiring data.

4.8. Summary of this chapter

This chapter was from implementation aspect to elaborate building database structure and the implementation of client side and server side. It also described the implementation of entire agile inquiry module, which includes design UI from the user view, the data communication between the client side and server. From the difficulty of design, it analyzed and solved the problems, and proposed the feasible implementation solutions. After satisfying under "The requirement of AREVA Taishan Project Configuration Management system", this system adds the design of agile inquiry module, which can satisfy the inquiry requirement well.

5. Deployment testing and user experience

For testing the system, the main target is to confirm whether system's functions satisfy the user's need, they also need to satisfy the system software demand specification definition, according to the different test, which maybe divided: Software unit testing, software integration test, software stress test and user acceptance test. In this thesis don't involve the user acceptance test.

5.1. The deployment of testing environment

Testing environment of deployment is Tomcat 6.0. The database platform uses MySQL. The test data is generated by the user.

5.2. Unit testing

The unit testing of JUnit mainly aims Business layer and DAO layer on the server site. Each function points need integrated unit test, which is carried by the developers. JUnit belongs to the test of program rank, therefore, it can detect various kinds of errors when the code compiled, which reduces the human cost enormously.

For example:

```
public class TestDbConnection extends TestCase {

        private static Logger logger = Logger.getLogger (TestDbConnection.class);

        public void testHibernate() {
                if (logger.isDebugEnabled()) {
                        logger.debug ("bye bye");
                }
                WorkstationAPI.getSingleton().createWorkstation ("test");
        }
}
```

Through the simple code, developer can implement the basic function of unit test.

5.3. Integration testing

The target of testing is mainly framed by two aspects: The first is the testing of the technical integration, which is to test the accuracy of the system function integration. The second kinds of testing are about the system's usability, business function's extendibility, operationally, user interface friendliness and other aspects.

The mainly test methods are the black box testing, the fault-tolerant testing and performance testing. The performance testing uses the software Loadrunner to carry on. Testing member installed the tool in the front workstation, through compiling test script to do

the performance stress testing, which is to imitate multi-user concurrent access environment. According to the result, developers need to modify the number of the connection in the connection pool to make the server in a high efficiency state. The Bugs which are detected in the testing need to carry on the regression testing confirmation after the modification.

5.4. Test conclusion

From the stress testing result, the test member analyzed and obtained request response time, request success ratio and other quotas under the existing server hardware source condition, which satisfy the user's demand. Because there are different hardware resource between the testing server and production server's environment, the data quantity are also different, considered that the data quantity of production database is small, and the system is applied to use a lightweight database Mysql, the test result show that the system can satisfy the user the demand.

5.5. Related functions interface demonstration

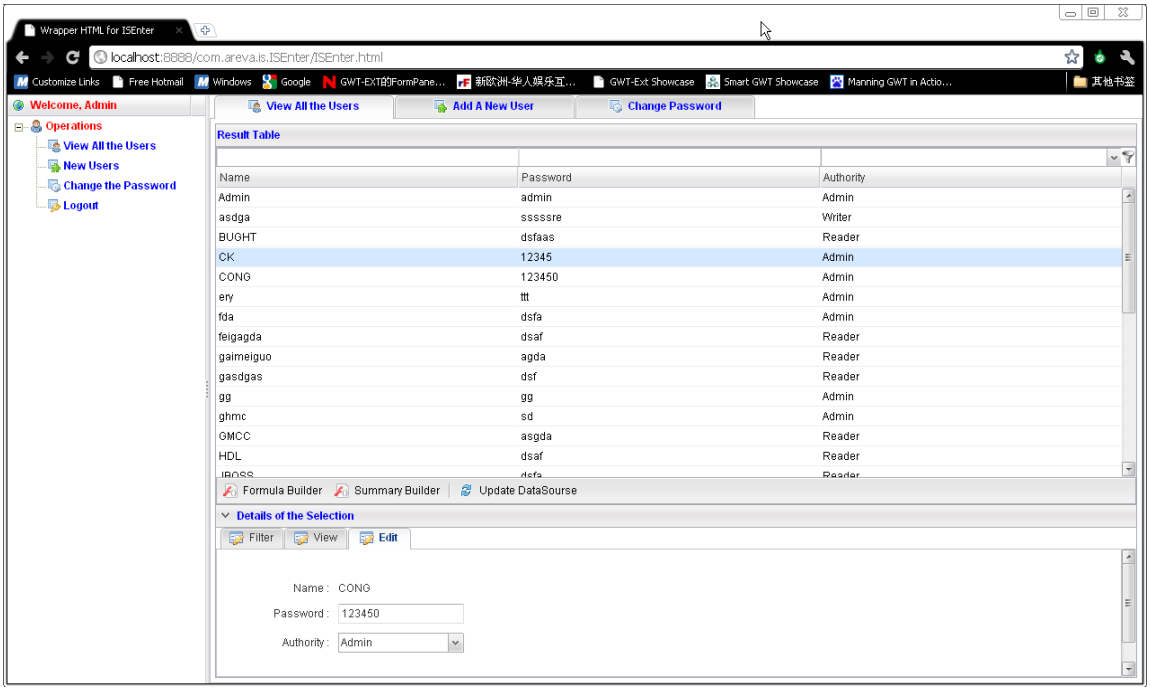


Figure 5-1 is the administrator interface

The administrator UI implements simple functions of user management, it provided related key words inquiry function.

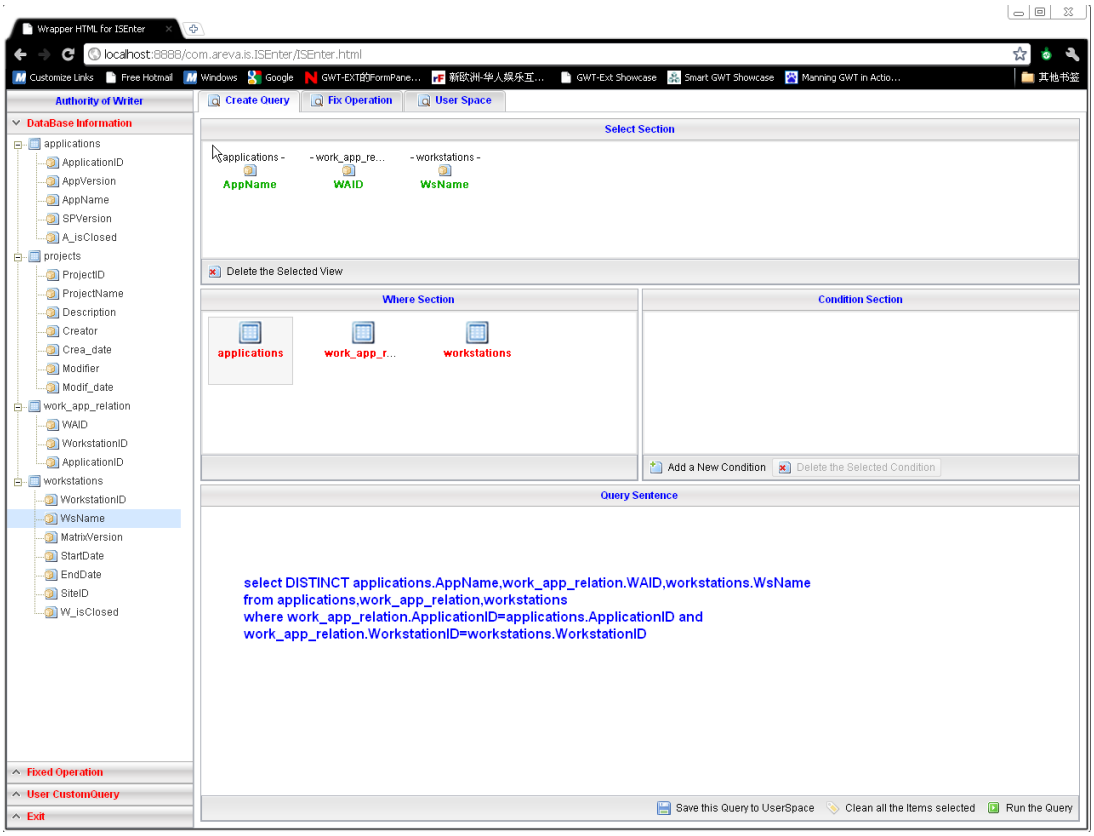


Figure 5-2 The agile inquiry interface

Through selecting the items of tree structure to the right corresponding selection area, other areas' items (including Where Section and Query Sentence Section) will generate automatically, and then users can click on the run button to run the inquiry.

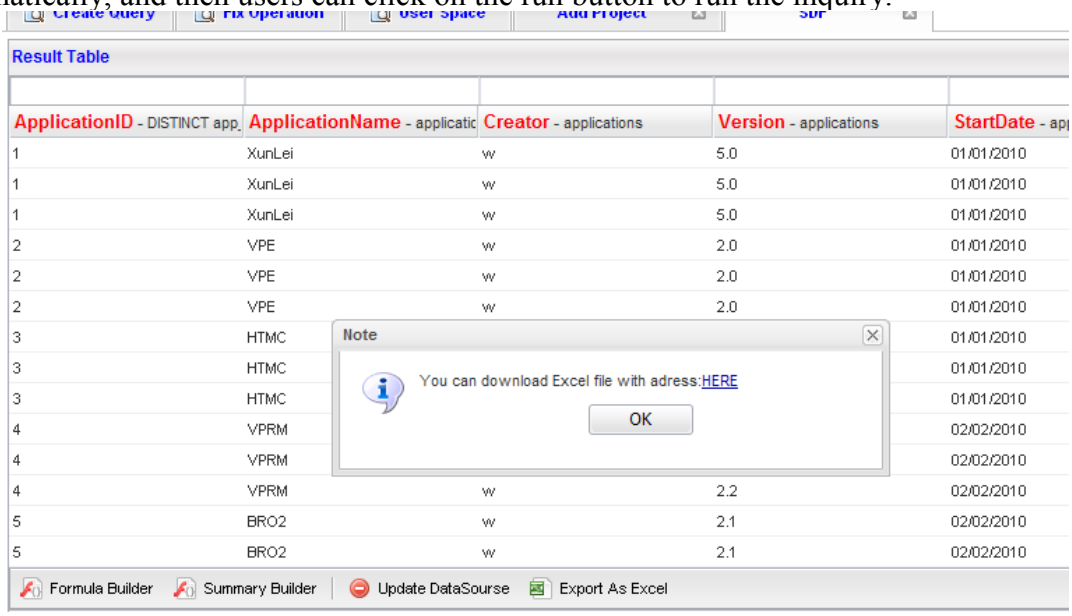


Figure 5-3 Export results to Excel file

Query results can be flexible exported to Excel format file, which can be downloaded from the server and print the document. According to the results field's type to build the Formula (Formula Builder), which is used for integer type, and show the connection string

display field (Summary Builder), which is only for string data type. For all the query data result, it provided flexible keyword search features.

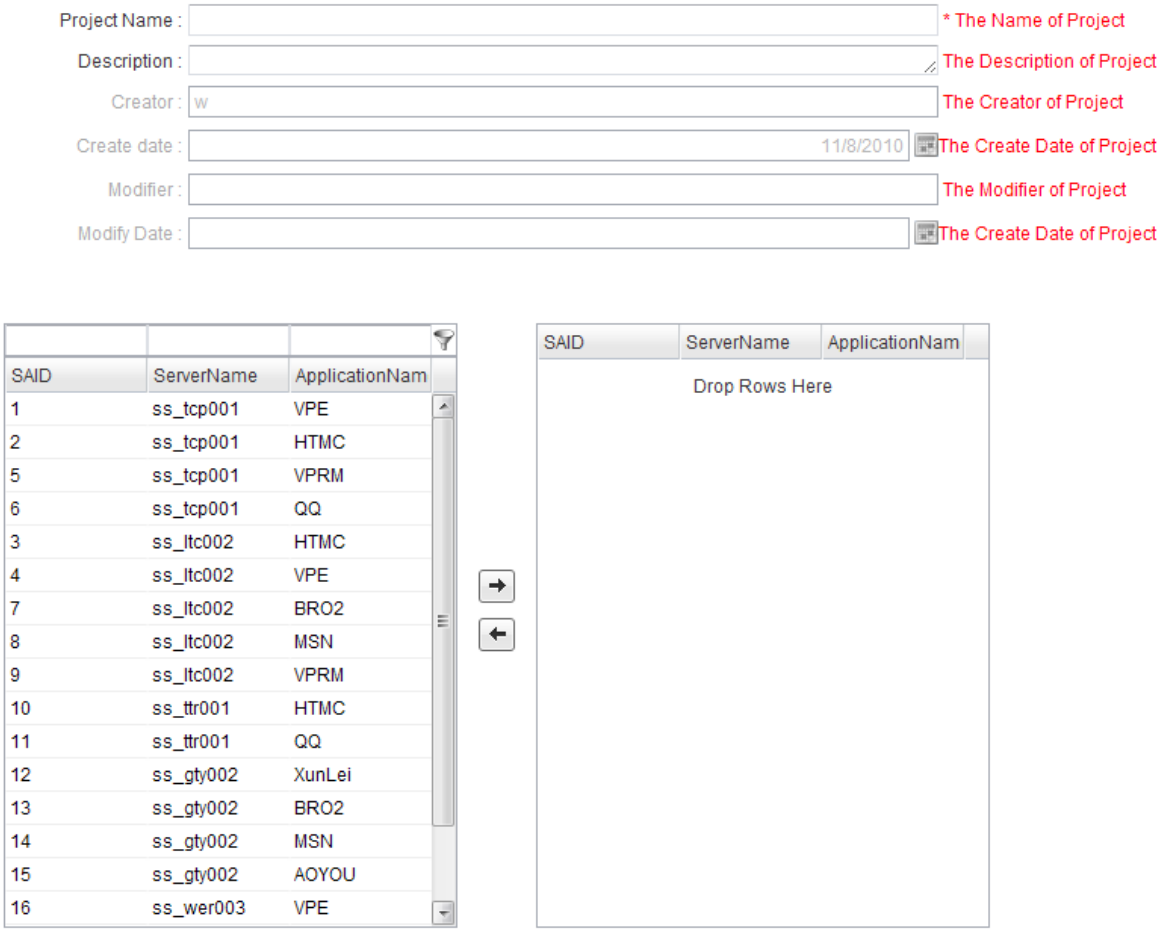


Figure 5-4 Add project UI

The figure above shows the UI of adding a new project, one can choose to only create a new project or create a project with the relationships of software installations (the gird left shows server name with its associated software). One can use the drag mode to choose, or also can choose by clicking the right button. The gird left provided the keyword query function, which simplifies the selection operation. The objects which have selected will not be displayed in the original table.

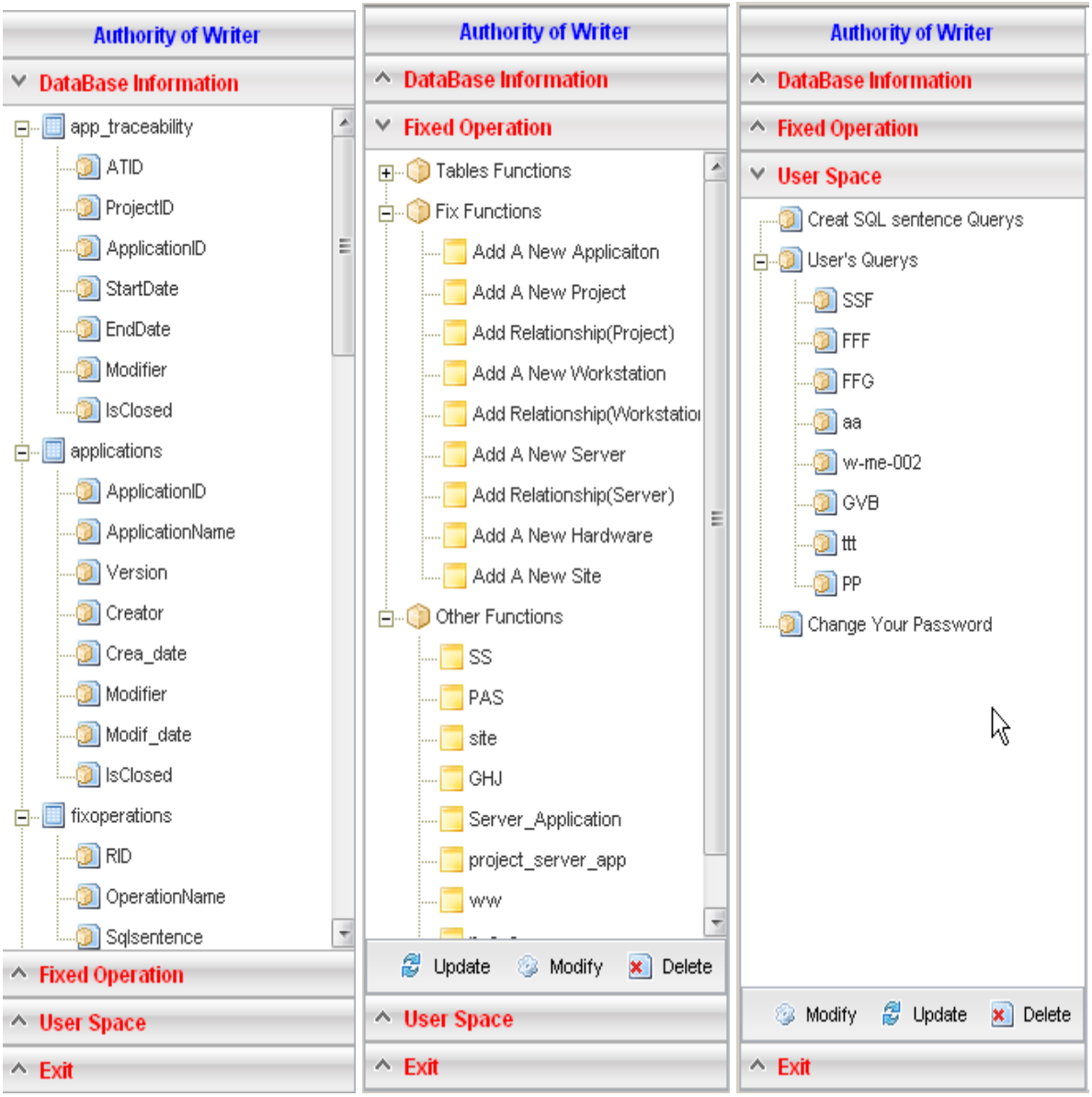


Figure 5-5 Writer's function menu UI

The first menu shows the information of the database, which includes the table name and attribute name. It is used for carrying on the agile inquiry. The second menu shows the basic requirement of users, which includes three parts: Tables Functions (show the data in the table, and can modify related attributes, can close object), Fix Functions (the user requirements) and Other Functions (this part shows the query created by user for all the users). The third menu is User Space. Users can import its own SQL sentence to get results, change the password, and use the queries which are created by themselves (only show for themselves).

5.6. Summary of this chapter

This chapter described the testing deployment environment which is deployed successfully in the hardware server and the system has been tested. This chapter also showed the mainly UI design of the system which provides a good user experience.

6. Summary and forecast

This kind of general agile inquiry can be used for the various database management systems. One only needs to do some configuration work to make the system connect to the database. It has high reusability. It can meet some specific area, some specific people or some specific demand to provide the data search service. The characteristic of the system is “more general and more flexible”. It has the generality, and through expanding and implementing the user fixed demand then can support the MIS of all domains. Compares to the traditional inquiry, it satisfies changeable inquiry requirement of user. Unifies with data binding components of SmartGWT framework, it implemented the components integrating with data, which greatly enhances the user’s inquiry efficiency and inquiry precision. This chapter has summarized the tasks what the thesis has done and given the forecast to future work.

6.1. Summary of thesis work

The tasks of this thesis can be summarized to the following several aspects:

- It proposed one kind of general agile inquiry MIS design solution, which only needs to configure a XML configuration file to simply related to a small lightweight database.
- In inquiry aspect, the UI imitated the Business Object Report Inquiry System’s interface to design, which has a wonderful friendly UI in inquiry (The Drag-Drop mode likes Business Object). The uses don’t need to grasp the SQL statement knowledge to use this system.
- It implemented the save operation to the queries which are created by users. Users can save it to all users’ function space and also to their own space. After the SQL sentence analyzing, system reused the display class and local DataSource, and then got results from server to display. Simultaneously it provides user’s queries management.
- It implemented the integration of SmartGWT + Spring + Hibernate frameworks. The integration of Spring and Hibernate are quite mature technique. SmartGWT is only the framework of presentation layer, and then the integration process becomes easier.
- Base on "AREVA Taishan Project Configuration Management System Demand", it provided the solution from the overall architecture design, database design the detailed subsystem design, and implemented this system according to the design.
- It researched deeply SmartGWT framework, it is mainly in the part of the communication between the client side components and server side. Meanwhile, it provides a solution for the presentation layer design pattern.

6.2. The future work

Along with the development of various kinds of frameworks, more and more frameworks can be integrated together. The OpenSource provides a big platform for the development of technique. But framework also has its restriction, which is very difficult to implement the system generality. This thesis carried on the client side programming based on SmartGWT, used the general agile inquiry design thought, which provides a solution based on Web environment. It also provided for the user who has inquiry requirements changeable. But there are some aspects insufficient existed in the system. One can continue to do the deeper research, which is in several aspects:

- **More deeper research of user permission**

Because this project doesn't need make a complicated user permission sub-system, it only needs to divide into three types of users to implement three parts of sub-system. But if the system is used for major industry's project, because of the broad span users from different departments, the permission design appeared more important. In this aspect, one may also to the refined user group permission, the table permission and operation permission and so on.

- **The big quantity of data inquiry to a big database**

For the agile inquiry, it only used for the small database, when the data quantity and table quantity increases progressively, because the system architecture dependent on the connection between local DataSource and server, the big quantity of data will slow down the performance obviously. One need to do more work on about optimizing database, build the database index, optimizing the connection pool, use multi-thread pool to implement and so on.

- **The research of Hibernate to used for multi-table constitution's agile inquiry**

Hibernate framework is a very mature framework on persistence layer, it through a simple object-relation to map to entity object. The operations for the database mapped to the operations to the related entity object. But in Business logic layer, the inquiry results are not formed an object, which hindered the agile design. Although developer can use Spring to reduce the coupling for the module or layer, but to rebuild the results format according to the changeable request is not a convenient matter. Whether to conveniently carry on agile inquiry in the object layer, which still needs deeper research.

- **The research of division the end-user inquiry logic and database**

This system displays database table information directly. The inquiry selection operation is carried on by users. This inquiry mode needs the end-user to be familiar with the database

structure. Usually many managers would also want to do the decision-making by inquiring data through the system. The mode of agile inquiry is not easy to accept for them. In this situation, he need to ask a system specialist to set the query and save to the all user query space and then he can get the results easily. For this problem, Business Object produce will add a middle layer to divide the database object and business object. The business object can include many database objects. The end-users can use their professional knowledge to carry on the inquiry through the business object. The research of this part is valuable to the middle-and-small database.

- **The research to SmartGWT component**

SmartGWT framework has done much work on the data integration of server side, it is implements based on GWT, and it has very high extension and probability. In presentation layer components, developers can design the special-purpose component through rewrite the methods and also can bind the components to a big module. After that, developers can release and share the re-development products. Because the framework is not tout an OpenSource framework and all the components are packaged by JavaScript, the developers need to master some knowledge to carry on the re-developing work. SmartGWT is an outstanding presentation layer framework. It has been widely used in the world.

In recent years, along with the development of data analysis and data mining, the data appeared more and more important in the enterprise. Database is also getting more and more complex. The requirement of inquiring data to make decision for users is much higher than before. The complex inquiry logic transferred from database layer to business layer, and transferred from business layer to the client side, which makes database only emphasize on the simple access operation, the burden of database server and application server transferred to the client side, which is really implement the load.

As a result of the technique similarity in the enterprise application system, this MIS structure may be used to the other enterprise systems, and provides the reference for the maintainable and expandable system design.

References

[1] 王立新．论计算机网络的应用与发展 [OL]．http://www.lunwentianxia.com/product.free.10005390.1/，2009

[2] MIS [OL]．http://baike.baidu.com/view/64355.html，2008

[3] Crystal Reports Introduction [OL]．http://baike.baidu.com/view/1717253.htm#1， 2010

[4] Grant K. Slender，Developing with Ext GWT Enterprise RIA Development，Apress， 2009

[5] David Geary with Rob Gordon，Google Web Toolkit Solutions，Prentice Hall，2007

[6] 冯浩．基于 B/S 架构的员工管理信息系统的设计 [D] ．广州:中山大学，2006

[7] 赵东波．基于 B/S 结构的管理信息系统的设计与实现 [D] ．西安:西北工业大学，2007

[8] Jeff Dwyer，Pro Web 2.0 Application Development with GWT，2008

[9] 高昂，卫文学．基于 Hibernate 与 Struts 框架的数据持久化应用研究 [N] ，计算机应用，2005 (12) ：2818-2819

[10] 蔡雪焘．Hibernate 开发及整合应用大全 [M] ，北京：清华大学出版社，2006

[11] 刘丰、王泽．数据库应用查询的报表设计 [J] ，万方数据库，2005- 41（24）

[12] 周平平 基于通用关系模式的 XML 数据存储模型研究 [D] ，兰州:兰州大学，2009

[13] 蒋外文，熊东平，张肖霞．基于多维数据库的 MOLAP 存储及查询技术研究 [N] ．计算机工程与应用，2005．41(24)．166～168

[14] Prabhakar Chaganti．Google Web Toolkit: GWT Java AJAX 编程 [M] ，北京:人民邮电出版社，2009

[15] Crane D，Bibeault B，Sonneveld J，et al．Ajax 实战实例详解 [M] ，北京:人民邮电出版社，2008

[16] Vipul Gupta，Accelerated GWT Building Enterprise Google Web Toolkit Applications[M], Apress，2008

[17] 骆开华．GWT 应用程序与服务器间数据传递技术的研究 [EB] ，http://www.14edu.com/ligong/jisuanji/0I1D34H010.html，2008

[18] David Geary Rob Gordon．Google Web Toolkit Solutions: More Cool Useful Stuff [M] ，北京: 机械工业出版社, 2008

[19] 苏珏， 曾贤龙，蓝天．Ajax 在 Web2.0 中的应用与研究 [N] ， 网络与通信, 2007

[20] 骆开华．GWT 应用程序与服务器间数据传递技术的研究 [D] ，广州:华南师范大学，2010

[21] 徐六五．基于 GWT-Spring-Hibernate 保险业务系统的设计与实现 [D] ，广州:中山大学，2008

[22] 陈凌，王建东．基于 GWT 技术开发 AJAX 应用的分析和研究 [N] ，计算机技术与发展, 2009

[23] 赵威．基于 XML 数据库的查询优化研究 [D] ，哈尔滨:哈尔滨理工大学，2008

[24] 张亚林．基于 XML 的数据库查询架构的设计与应用 [D] ，大连:大连交通大学，2009

[25] Bram Smeets,Uri Boness and Roald Bankras，Beginning Google Web Toolkit From Novice to Professional，Apress，2008

[26] Robert Hanson and Adam Tacy，GWT in Action Easy Ajax with the Google Web Toolkit，Manning，2007

[27] Dan Sanderson， Programming Google App Engine，O'REILLY Google Press，2010

[28] 张锦辉．企业数据报表统一管理平台的研发与应用 [D] ．上海:华东师范大学，2008

[29] 董龙飞，唐文忠．灵活查询技术研究和实现 [N] ，控制与决策， 2003

[30] 徐秀君．浅谈 web 的应用开发框架的整合设计 [N] ，IT 及高新技术，2010

[31] 吉融．GWT 入门与经典实例解析 [M] ．北京:机械工业出版社，2008

[32] 掌明，王经桌．基于 Erwin 模型的 MIS 系统研究 [N] ．控制管理，2006 (10) 111-113

[33] 杜巍．企业级通用报表引擎开发 [D] ．上海:复旦大学，2006

Acknowledgements

I would like to express my gratitude to my Professor Daniel Krob in CNAM, my professor Zhang Ling in SCUT, my tutors Mr. FANET and Mr. AMBROSE in AREVA, and all those who helped me during the internship and the writing of this thesis. The Professors and tutors gave me instructive advice and useful suggestions on my thesis.

I also want to express my gratitude to AREVA, it provides a good environment for me to do the internship project, and express my gratitude to all the colleagues. Mr. BARFETY is a very good French Professor. I spent a very happy time with him and my colleagues. I learnt much from them.

At last, my thanks would go to my beloved family for their loving considerations and great confidence in me all through these years. I also owe my sincere gratitude to my friends and my fellow classmates who gave me their help and time in listening to me and helping me work out my problems during the difficult course of the thesis.