

Business Software Rapid Development Platform based on SOA

Haigen Yang, Xiaoting Rui, Yixin Liu and Junyi He

*Institute of Launch Dynamics,
Nanjing University of Science and Technology, Nanjing, China
yhg_620@126.com*

Abstract

In this study, a new SOA-based business software rapid development platform (SOARDP) is presented to solve the main contradiction between the personalization of user's needs and the versatility of the software products. The platform integrates the idea of service-oriented, model-driven, Ajax, BI and so on, to improve the speed of business software development. The PICC's (People's Insurance Company of China) performance appraisal system was developed from the platform. The application shows that the platform can greatly improve the efficiency and quality of business software development.

Keywords: *SOA, Rapid Development platform, ESB, ETL, Ajax*

1. Introduction

With the rapid development of the software industry today, the main contradiction is between the personalization of user's needs and the versatility of the software products. The management models, organization structures and the business processes need to be changed with the rapid development of enterprise. The business software also must be quickly changed, in order to adapt to enterprise management. On the other hand, the software company should reduce the software development costs to improve its competitiveness. The software industry has been engaged in ongoing effort to solve the basic contradiction between the personalization of user's needs and the versatility of the software products. The solutions summed up in two directions: one is technological innovation and another is software project management, software industry has emerged a large number of technological innovations, in all technical innovations, SOA is the most meaningful and most vital.

SOA (Service-Oriented Architecture) is a set of principles and methodologies for designing and developing software in the form of services, which aims to maximize reuse neutral services to increase IT adaptability and efficiency in the applications. SOA achieves a complete separation of business and technology, which on demand through the loosely coupled of network, coarse-grained application components and distributed deployment. Channabasavaiah examined some of the problems that lead to consideration of an SOA, studied the nature of the service, constructed an application framework based on SOA and given some of the future computing environments that to develop an SOA application even more imperative [1]. Salter and Jennings used NetBeans IDE 6 to build an SOA-Based composite application [2]. In practical applications, the SOA by reusing existing business logic and data model to minimize

the costs, risk, resources and time of the software development, and also can be ensured to improve effective of IT construction and benefits of investment of the enterprise. The SOA mainly use the following several techniques.

- a) XML (Extensible Markup Language): XML for the Web Service to provide a unified data format, including messaging, service description and a description of the workflow.
- b) SOAP (Simple Object Access Protocol): lightweight protocol for exchange of XML-encoded information, to pass messages between each Web Service on component transport layer.
- c) WSDL (Web Service Definition Language): With XML to describe a network service or endpoint used to define the Web Service and call for the interaction between old and new applications to provide a contract.
- d) UDDI (Universal Description Discovery and Integration): provides the framework for the description on the Web and found that business services is the registry for web services to achieve the standards and norms

The birth of a new development model-EMA (enterprise model automation) had a revolutionary impact on business infrastructure software platform for managing software development model. Doyle and Georghiou introduced and discussed a framework to enable successful change in an EMA project [3]. EMA is based on the level of business and management, as the basic means to build the development platform and business software system. In order to solve the urgent problems and find a way to quickly build business infrastructure software platform is placed in front of the software companies. The platform needs to provide a basic framework for developing complex applications; provide a fast, convenient, easy-to-use development tools for maintenance and management the basic components; to reduce the complexity of the development of management software, shorter the development cycles, improve R&D efficiency. Chen [4] described the inherent difficulties of predicting the performance of N-tier enterprise applications built using component technologies.

Many software companies put a lot of money and manpower to research and develop related products, but lack of an actual available platform for business infrastructure software to build and develop applications quickly. By using of service-oriented architecture, component technology, Web Service technology, workflow technology, BI, to develop a rapid development platform is very meaningful.

The software platform can be divided into three levels: operating system platform, software infrastructure platform, business infrastructure software platform, Fig.1 shows the hierarchy diagram. The operating system platform is for application software to interact with the hardware platform; software infrastructure platform is to provide infrastructure for complex applications on the operating system platform; business infrastructure software platform for the realization of the separation process of software development technology and business.

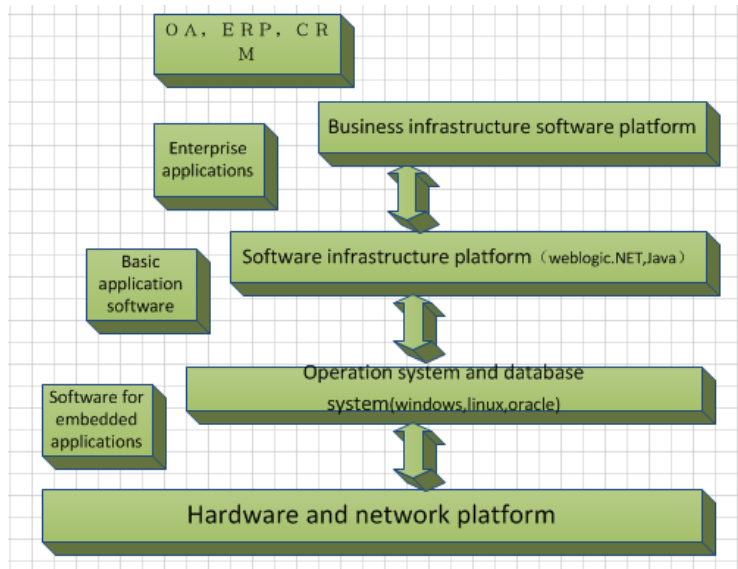


Figure 1. Levels of Software Platform

We present a platform by using service-oriented structure (SOA) to provide a series of services in the form of atomic service. The platform uses component technology to achieve software loose coupling and with highly flexible structure. The platform also uses service bus architecture, the modules or applications in the platform via the service bus to get a variety of resources. The external applications can also be easily inserted to the platform through the bus. The bus here we called it: ESB (Enterprise Service Bus). And any stand-alone applications can be connected to the ESB as a service to minimize the coupling of the system. Schmidt [5] presented a summary of the key concepts of the ESB and defined the integration model for ESB. The service bus provides a standard Web Service interface, highly standardized and high scalability.

In this paper, we try to develop a platform by using SOA to improve the competitiveness of the software products. The platform we developed can fast to build enterprise applications and achieve online zero-coding development to build enterprise applications.

2. Platform Architecture and Technology Implement

A Business infrastructure software development platform should have the following characteristics:

1. Cross-platform capabilities. The platform should support several of operating systems, such as Windows, Linux, Unix and others; support a variety of databases, including Oracle, SQL server, Mysql, Informix and so on; support a variety of middleware web servers, such as Tomcat, Weblogic, JBoss and others.
2. Highly configurable. The platform needs to reduce the amount of coding, users can through configuration to achieve a vast majority of business logic.
3. Hot-pluggable running core. Some modules by a little development or by simple configuration can be quickly integrated into the platform which needs a hot-pluggable running core.

4. High efficiency. Taking into account the carrying capacity of the system, the platform needs to improve system efficiency and carrying capacity.
5. Well user experience. By using Ajax and other rich client technologies to improve the user experience.

We designed a rapid development platform based on SOA called SOARDP, which consists of display layer, control service layer, application service layer, basic component service layer, system service layer, application development tools and ESB service bus, shown in Figure 2.

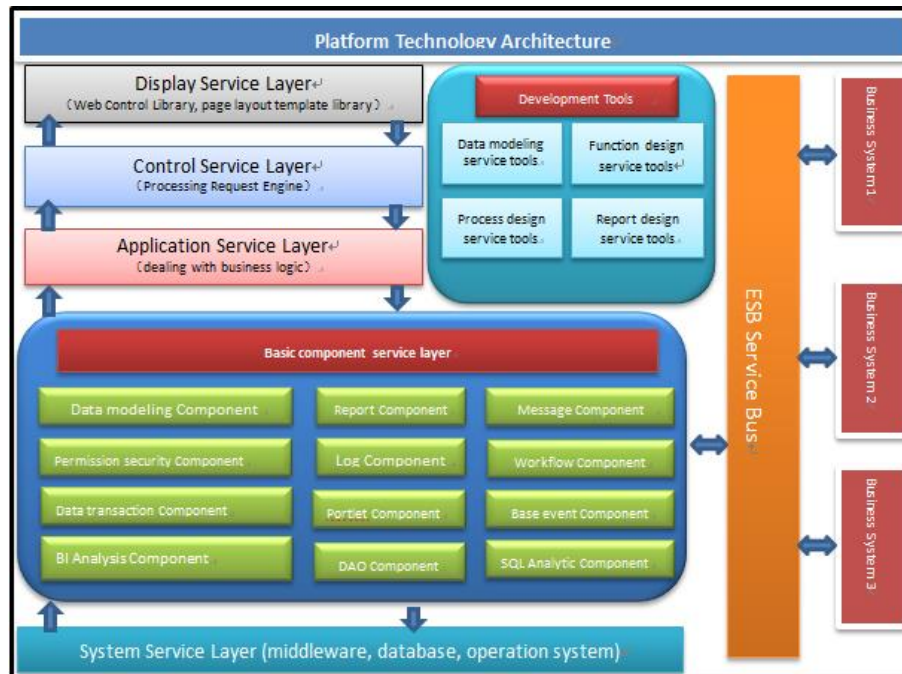


Figure 2. Platform Technology Architecture

- Display service layer: by using cross-browser UI components and rich client technology to reduce development effort of user interfaces, which provides various types of page layout, and offers a variety of html controls.
- Control services layer: the layer provides a unified request processing engine, which is responsible for processing all front-end requests. According to the request parameters, the layer calls the business logic processes object or component and then feedback the data to the display service layer.
- Application service layer: the layer provides a unified standard specification for building business objects. All business objects are based on a unified base class, which provides a wealth of parameters to simplify the business objects.
- Basic component service layer: the layer provides a large amount of business components to support the platform running and ensures that the system is stable, efficient, and safer. The components are designed based on model-driven.
- Application development tools: the tools provide development methods for designing and development applications. The simple and easy to use development tools can

greatly improve the efficiency of software development. The tools including data modeling tool, functional design tool, business logic processing design tool, reporting design tool.

- ESB service bus: the bus is based on a standard SOAP protocol as a data transfer protocol for the service requested, which includes service core; service registry; addressing center; service integration center, service security center; authentication and authorization center.

2.1. Heterogeneous Database Access Service

When enterprise, government and other users construct IT systems, database systems may involve a variety of types, so business infrastructure platform should deal with the difference heterogeneous databases. Chenetal [6] presented a development of an integrated environment acting as a software agent for discovering correlative attributes of data objects from multiple heterogeneous resources. Heterogeneous database heterogeneity includes:

1. Heterogeneous computer architecture and operating system, each database may run on different computers: mainframe computers, minicomputers, workstations;
2. DMBS heterogeneous; heterogeneous of database systems (such as Oracle, SQL Server, DB2, etc.). In different databases, the sql grammar is different.

This article is mainly to solve the DMBS heterogeneous, which presents a heterogeneous database access service to solve the problems of heterogeneous of databases. The applications access data through the access service to shield the heterogeneous between different databases, heterogeneous database access service architecture is shown in Figure 3 below.

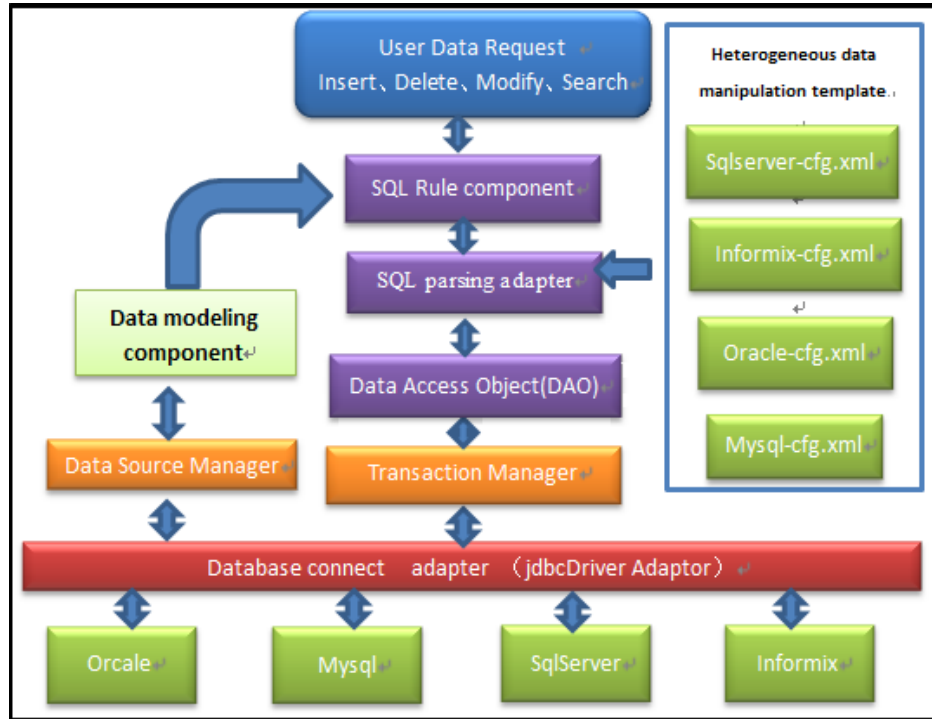


Figure 3. Heterogeneous Database Access Service

If the user needs to request data from the database, the SQL rule component generates the type of data request model from data modeling component, the SQL parses adapter loads SQL operation template statements depending on the different types of databases. According to different data sources, Data Access Object (DAO) connects to different databases. The transaction manager is provided between the Data Access Object and database connect adapter, which is used to handle data commit, rollback and other operations. Database connect adapter via jdbc Driver to access different databases. This can shield the different database types and provide a seamless, heterogeneous database access service. We designed a XML based sql template file to distinguish different database sql statements. The following codes show manipulation template of Oracle database.

```
<elements>
<element name='create_table'>
<![CDATA[create table <param:table_name/> (<param:table_fields/>);]]>
</element>
<element name='alter_table_name'>
<![CDATA[rename table <param:old_table_name/> to <param:table_name/>;]]>
</element>
...
</elements>
```

2.2. Hot-pluggable System Running Core

The hot-pluggable system running core is the core module of the platform. This module should have a good technical architecture that can facilitate to insert a variety of business components, integrate interfaces and access services for a variety of function modules. The client does not need to invoke the business component object directly, but through the system running core to invoke business components.

Many J2EE [7] platforms generally use Spring [8] to manage the related business components, by using basic JavaBean instead of the EJB to solve the complexity of enterprise application development. Through the Inversion of Control (IOC) and Aspect Oriented Programming (AOP) [9], Spring not only supports XML metadata configuration also supports annotations, which defines object classes in XML, changes a class only needs to modify the defined XML. This achieves hot-pluggable of the object, which is somewhat similar to the USB interface in hardware.

After research Spring framework and in order to more easily to provide services, we redesigned a hot-pluggable running core for the platform. The hierarchy of the running core is shown in Figure 4. First, the ServiceController class loads and handles various business components; second, the system initializes StartupController object responsibly to load the business objects from server.xml; third, SystemFactory object creates these business objects. In the last, the ServiceController calls the “execute (RequestContext requestContext)” method, according to the context of information to implement the corresponding components. SystemFactory equivalent to the Spring’s BeanFactory used to construct different classes. The ServiceController equivalent to the Spring’s Core Object used to establish and maintain these components.

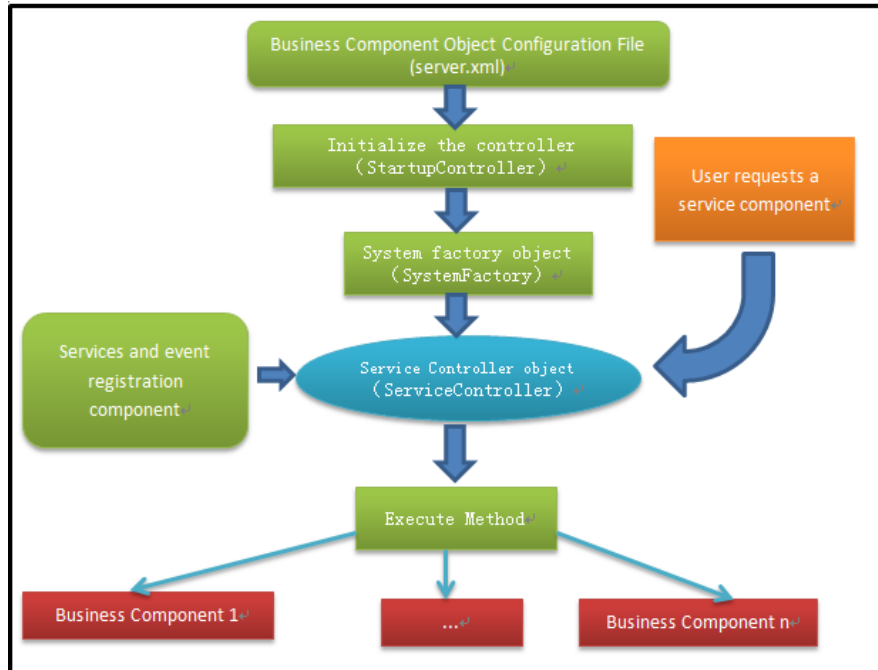


Figure 4. Systems Running Core

The ServiceController is the most important class, which responsible for loading classes dynamically. In common Java application, the class loads automatically by the JVM, but has limitations in the development of the high flexibility programs, because the JVM depends on the class name to load a class, but not based on user input parameters. When the program is running, the application can not mount a new class, needs re-modify, compile and deploy a new module. By using dynamic class loading technology can solve these problems. In order to realize Java class loading dynamic can use this code:

```
Class clzz = null;
try { // Create a component class object
clzz = Class.forName(sClassName);
} catch (ClassNotFoundException e)
{
return false;
}
try { // Create an instance of the Component Object
clzzObj = clzz.newInstance();
} catch (IllegalArgumentException e)
{
return false;
}
```

```
Method method = null;//Get component method object  
Class[] clzzParams = getParamClass(params);  
try {  
    method = clzz.getMethod(sMethodName, clzzParams);  
}  
catch (SecurityException e) {return false;}
```

2.3. Extjs Template-based Web Presentation Technology

JSP technology provides a flexible, feature-rich Web presentation layer technology. However, the logical code in JSP, the business code and data are mixed, cannot separate the presentation layer from the logic layer, and so code reuse and maintainability of the system are becoming very difficult. Many software companies provide many template technologies, such as XSLT, Velocity, FreeMarker, try to isolate the presentation layer from logical layer. In order to avoid the shortcomings of the JSP, it seems to return back to the early web only supports Servlet technology.

In Web2.0 era, the rise of Ajax technique has a significant breakthrough in user experience and data transmission, which provides a new solution for presentation layer. The core of Ajax is the XMLHttpRequest object to complete the asynchronously data exchange and process, Dom object used to handle the XML data returned from the server side. The software industry provides a variety of solutions for implementing of Ajax (DWR, AJAX, Tags, Echo2, etc.).

Another important purpose of the development platform is to allow a staff who less familiar with the programming to quickly build applications based on business needs. A visual online designer via drag and drop can design business application is needed.

This platform is based on the Extjs framework, developed a very flexible and easy to use module. The module provides a large number of rich UI components, with its powerful features and good presentation, to ensure that the specification and performance of the software interface. All the pages of the platform are static web pages, which can greatly reduce the burden of the server; provide a reliable guarantee for the platform to support enterprise applications, Figures 5-6 are the designers of the web presentation layer interface, including a grid and form designer.

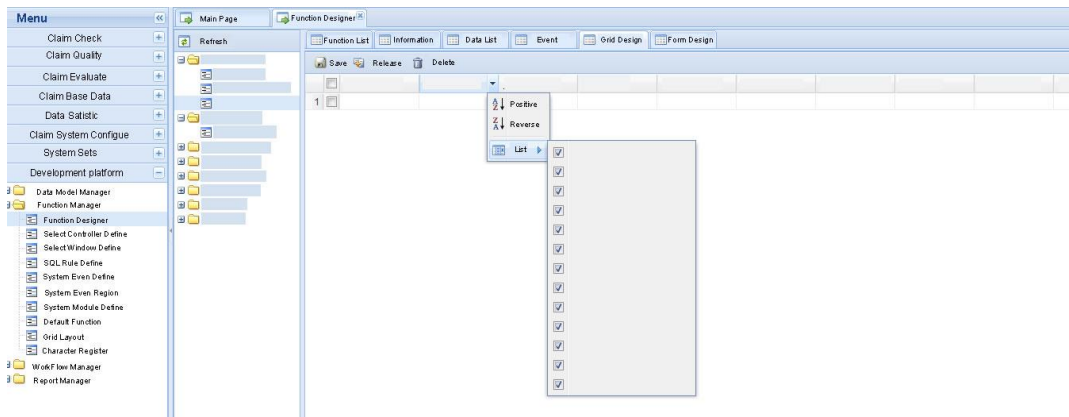


Figure 5. Grid Page Designer

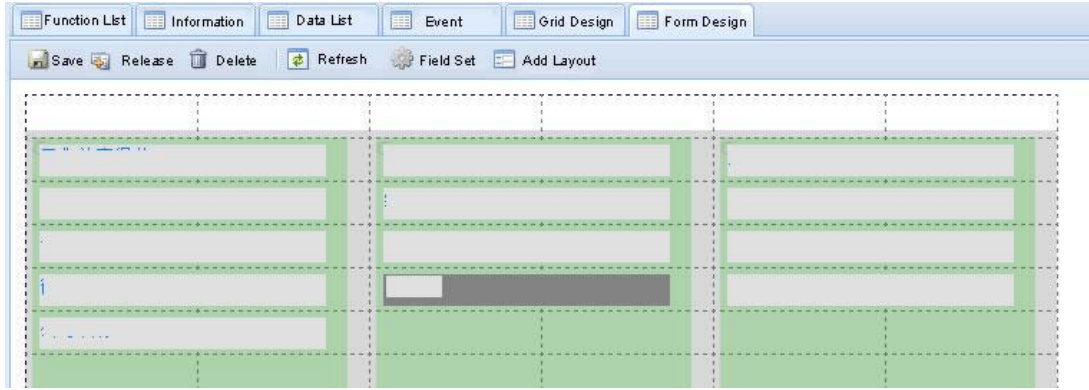


Figure 6. Form Page Designer

2.4. ETL and BI Service

Business Intelligence (BI) includes modern data warehouse technology, online analytical technology, data mining and data display technology for data analysis in order to achieve business value. BI is usually understood as existing enterprise data change into knowledge to help to make a decision; BI extracts the data from different useful enterprise data and cleanup, to ensure the correctness of the data. The data go through the process of Extraction, Transformation and Load get the valuable data, which is an ETL process. On this basis, the use of appropriate query and analysis tools (data mining tools, OLAP tools), the managers of the enterprise in the decision-making process could use the valuable knowledge.

ETL (Extract-Transform-Load) is a process in database usage and especially in data warehousing that involves:

- Extracting data from outside sources
- Transforming it to fit the operational needs (which can include quality levels)
- Loading it into the end target (database or data warehouse)

The platform designs a lightweight ETL model as a service running in a Java environment to support extract data from the relational database, text, Excel or XML files and transform the data in accordance with a set of rules and the end result is output to the relational database, text, Excel, XML or PDF files. Data extraction and store operation consist of a series of tasks, tasks execute sequentially according to rules configured by users. ETL service workflow is shown in Figure 7:

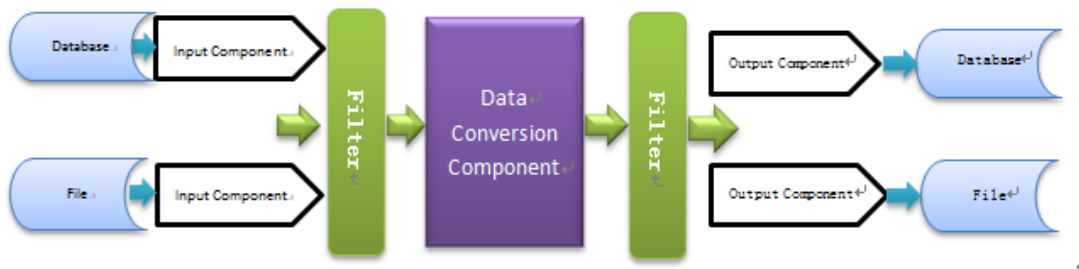


Figure 7. ETL Service Workflow

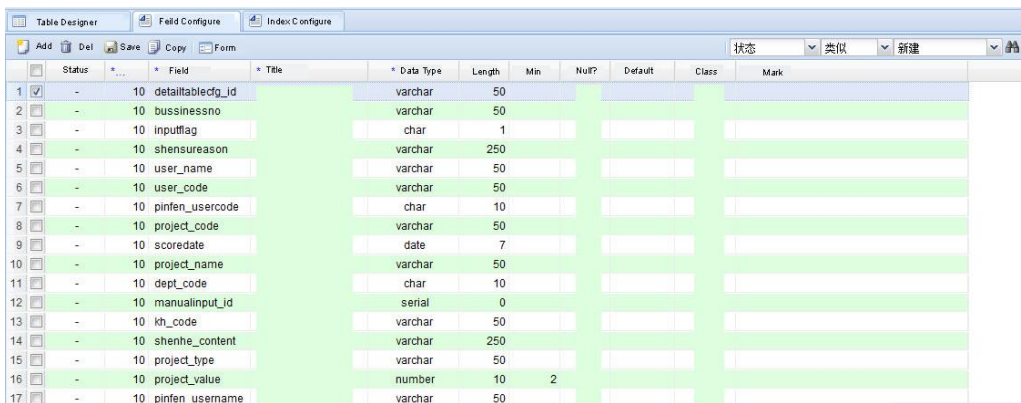
Here, Input Component gets data from the source database or file. Then, the data will be filtered by the Filter, which used to clean the data. Data Conversion Component converts a set of data by business rules and submits the results to another Filter to clean the data. The Output Component will output the clean data to the target database or file. Each input component is running in a separate thread.

- 1) Input Component: This component is used to read data from the source database or file, each input data must follow the master-slave relationship.
- 2) Output Component: This component is used to write data to the target database or file, each output data also must follow the master-slave relationship.
- 3) Filter Component: This component is used to clean the data by data validation, code conversion and some pretreatments. The filter receives a single input or output data and returns the processed data.
- 4) Data Conversion Component: This component is used to convert the data by a set of rules. The rules may be some sql statements or regular expressions.

3. The Rapid Development of Applications

This section shows the development of Jiangsu PICC performance appraisal system by using the SOARDP, as an instance, introduces how to achieve rapid development via the platform.

Data is the base of business application software, so first need to use data model tools to modeling data structure. It can be convenient to define the data structure online via the data model designer. This designer is similar to the SQL Server enterprise manager, which has the ability to create, drop, modify tables, and support the definition of table fields. The data types of table fields support integer, char, varchar, serial, float, *etc...* This module shields the heterogeneity of the database via the technology of Section 2.1, the user interface of the designer shows in Figure 8.



Status	Field	Title	Data Type	Length	Min	Null?	Default	Class	Mark
<input checked="" type="checkbox"/>	10	detailtablecdg_id	varchar	50					
<input type="checkbox"/>	10	bussinessno	varchar	50					
<input type="checkbox"/>	10	inputflag	char	1					
<input type="checkbox"/>	10	shensureason	varchar	250					
<input type="checkbox"/>	10	user_name	varchar	50					
<input type="checkbox"/>	10	user_code	varchar	50					
<input type="checkbox"/>	10	pinfen_usercode	char	10					
<input type="checkbox"/>	10	project_code	varchar	50					
<input type="checkbox"/>	10	scoredate	date	7					
<input type="checkbox"/>	10	project_name	varchar	50					
<input type="checkbox"/>	10	dept_code	char	10					
<input type="checkbox"/>	10	manualinput_id	serial	0					
<input type="checkbox"/>	10	kh_code	varchar	50					
<input type="checkbox"/>	10	shenhe_content	varchar	250					
<input type="checkbox"/>	10	project_type	varchar	50					
<input type="checkbox"/>	10	project_value	number	10		2			
<input type="checkbox"/>	10	pinfen_username	varchar	50					

Figure 8. Data Model Designer

After data modeling is completed by the data model designer, the platform provides a business function designer, to achieve some simple business logic designed online. This designer includes: functional model definition, functional field definition, functional events definition, grid and form page designer. By using these tools to develop applications can

improve development efficiency and make maintenance easy, and can achieve zero-coding to develop applications, as shown in Figures 5, 6, 9, 10.

Figure 9. Functional Module Definition

Number	Coding	Name	Method	Page Type	Shown?	Hide?	Index	Hot Key
1	uploadexcel		uploadexcel()	.grid,chkgrid,editgrid,			10	
2	pre		preRecord()	.form,chkform,			10	
3	next		nextRecord()	.form,chkform,			20	
4	create		create()	.form,grid,subform,			105	N
5	delete		del()	.grid,subgrid,			110	D
6	save		save()	.form,subform,			120	S
7	copy_eg		editCopy()	.editgrid,grid,subgrid,sube...			125	W
8	expbt		expbt()	.grid,chkgrid,editgrid,			220	T

Figure 10. Functional Events Definition

The system not only provides the grid and form designer, but also provides a wealth of events support, including complex data operations, Excel import and export, multi-dimensional search analysis, ETL functions. These events can be configured as user operations for users to operate. The business function designer according to the configuration auto generate static Extjs files, Extjs files include some Ajax code to communicate with back-end business logics, which largely reduce the burden on the server, increase the carrying capacity of the system. The system also provides some services like: time task service, report task service, portal service, which greatly simplifies the user's development.

According to PICC performance appraisal system development can be found, an application developed by using pure coding needs one week development, but by using SOARDP only needs one day. From the user point of view, the platform not only saves development time, but also business people can also build complex systems by simple training.

4. Conclusion

In this paper, we designed a rapid development platform based on SOA, to solve the main contradiction between the personalization of user's needs and the versatility of the software products. The platform integrates the ideas of service-oriented, model-driven, Ajax, BI and so on, to improve the speed of business software development. Users can through drag and drop to develop applications online. From the PICC performance appraisal system development can be seen that the platform can greatly improve development efficiency and save a lot of resources for enterprise IT construction.

Acknowledgements

We would like to acknowledge the National Natural Science Foundation of China (Grant No. 10902051, 11102089).

References

- [1] K. Channabasavaiah, K. Holley and E. Tuggle Jr., "Migrating to a service-oriented architecture", IBM Developer Works, <http://www.ibm.com/developerworks/library/ws-migratesoa2/>, (2003).
- [2] D. Salter and F. Jennings, "Building SOA-Based Composite Applications Using NetBeans IDE 6", Packet Press, (2008).
- [3] S. Doyle and J. Georghiou, "Software review: A process change model to meet the Enterprise Marketing Automation (EMA) vision", J. Database Marketing, vol. 8, (2001), pp. 176-182.
- [4] S. Chen, Y. Liu, I. Gorton and A. Liu, "Performance prediction of component-based applications", J. Syst. Software, vol. 74, (2005), pp. 35-43.
- [5] M. T. Schmidt, B. Hutchison and P. Lambros, "The enterprise service bus: Making service-oriented architecture real", IBM Systems Journal, vol. 44, no. 4, (2005), pp. 781-797.
- [6] M. Chenetal, Q. Zhu and Z. Chen, "An integrated interactive environment for knowledge discovery from heterogeneous data resources", Inform. Software Technol., vol. 43, (2001), pp. 487-496.
- [7] C. Nan and K. K. Ma, "Java's futures: Challenge and opportunity", IT Prof., vol. 6, (2004), pp. 19-26.
- [8] R. Johnson, J. Hoeller, A. Arendsen, C. Sampaleanu, R. Harrop, *et al.*, "The spring framework-reference documentation", version 2.5.6. <http://static.springsource.org/spring/docs/2.5.x/spring-reference.pdf>, (2008).
- [8] R. Johnson, "J2EE development frameworks", Computer, vol. 38, (2005), pp. 107-110.
- [9] J. Bloch and P. Chan, "Effective Java: Programming Language Guide", Prentice-Hall Inc., Seattle, USA (2002).