

## Improvement of the Structure of Information Integrated Service Platform for Oilfield Exploration and Development

Yanghao Wu

Science Faculty, University of Alberta, Edmonton, Canada

---

*Abstract: The SOA-based information integrated service platform is applied in the exploration and development of SL Oilfield in eastern China. With the continuous deepening of exploration and development as well as scientific research and production, the original service platform requires to be updated and expanded. The original service platform adopts mature Web Service. Based on the improvement of the original Web Service platform, the REST service platform framework is also additionally developed in this study. The REST service platform takes the oilfield exploration and development business as the core, develops diversified data services in the form of service components, which are managed uniformly through the service component container. It outputs unified data services in REST form and publishes data formats such as JSON and XML. Therefore, the data acquisition and data processing functions of the original services are improved, and new data service forms are studied for oilfield exploration and development, establishing a more comprehensive SOA-based information integrated service platform.*

*Keywords: Oilfield exploration and development; information integrated service platform; Web Service platform; REST service platform.*

---

### 1. INTRODUCTION

The SOA-based information integrated service platform is applied in the exploration and development of SL Oilfield in eastern China. With the continuous deepening of exploration and development as well as scientific research and production, it is necessary to focus on the business object model of oilfield exploration and development to improve the data acquisition and data processing of the original services. Targeting on the exploration and development, new data service forms should be studied and a completer and more comprehensive SOA-based information integrated service platform should be established. Service-Oriented Architecture (SOA) is a component model that links different functional units of an application (called services) through well-defined interfaces and contracts between these services [1]. In fact, SOA is not a specific technology, but a framework that represents a new idea and a sign of “service-oriented new IT era” [2]. Based on object-oriented technology, Web Service encapsulates data and programming elements so that different Web-based applications can access. With the promotion and application of Web Service technology, the idea of SOA has been continuously demonstrated by multiple information systems with significant benefits, and it gradually became popular. In recent years, the idea of REST has been gradually accepted along

with SOA. Websites are also starting to open APIs to developers, inciting the upsurge of REST-style Web Services. REST refers to Representational State Transfer, which is a relatively new and popular service form. Roy Fielding defines REST as a structure style that takes “minimizing latency and network interactions while maximizing the independence and expansibility of components” as the target. REST is also called RESTful Web Service, it is a new style of Web Service [3].

The information integrated service platform deals more than business logic, but also data access, business logic organization and other aspects. REST and Web Service have their own advantages. REST is very suitable for resource-based service interfaces, and is especially suitable for scenarios that require high efficiency but low security. However, the maturity of Web Service can bring convenience to the interface design that has to be provided in diversified development languages and has higher security requirements [4]. The original information integrated service platform of SL Oilfield adopts the mature Web Service platform. Based on the improvement of the original Web Service platform, the REST service platform framework is additionally developed in this study as the main focus, as well as building a brand new data service system that supplement the original information integrated service platform of SL Oilfield.

## **2. IMPROVEMENT OF THE ORIGINAL WEB SERVICE DATA SERVICE**

The Web Service data service of the exploration and development information integrated service platform for SL Oilfield has been improved, which mainly includes the following points.

### **2.1 Independence of Management Functions**

In the data service layer, the processing logics such as identity authentication and transactions belong to the general system at all levels and are separated from the data service platform. Through Aspect Oriented Programming (AOP) technology, these logics are uniformly described in independent modules (management function modules), and the relevant logic of the data service platform is realized by the interface that realizing corresponding logic of this layer.

### **2.2 Improvement of Data Access Methods**

In terms of data access mode, iBatis.NET is used to access the underlying data. iBatis.NET provides access interfaces for mainstream databases and sets the data source through configuration files.

### **2.3 Construction of Data Service Layer**

The SOA framework is adopted to build the data service layer above the database access layer. Its function is to return the information of business component requested by the user to the calling module of the application layer to complete the user’s access to the data. Considering the cross-platform and multi-client operation, it adopts Web Service and provides data interfaces to the outside through the SOAP protocol.

### **2.4 Improvement of File Access Methods**

Considering the particularity of file access, FTP service, which is very mature, is used in file access. At the same time, for reducing the load of accessing the service platform, FTP is used as a supplement to the Web Service.

### 3. SUPPLEMENTARY DEVELOPMENT OF REST DATA SERVICE

In addition to improving the original Web Service platform, this paper focuses on supplementing the REST service platform framework. Taking the oilfield exploration and development business object model as the core, this paper develops various data services in the form of service components, which are under unified management through the service component container and output unified data services in REST form and publish data formats such as JSON and XML. JSON (JavaScript Object Notation) adopts a completely language-independent text format, but also inherits conventions of the C language family (including C, C++, C#, Java, JavaScript, Perl, Python, etc.). These features make JSON an ideal data-interchange language, convenient for both human reading and compiling, as well as machine parsing and generating.

#### 3.1 MVC Mode Structure

Model-View-Controller (MVC) mode has always been a very important architectural mode in computer science (Figure 1). It is very powerful when dealing with separation of concerns in applications. It is well used in the Web-based application system framework.

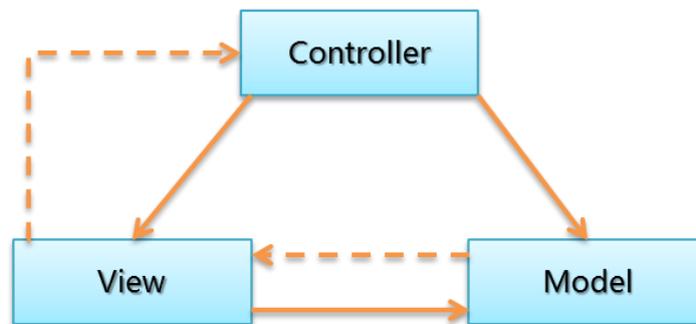


Fig. 1 Phase analysis of arrangement

This mode is frequently used in network programming. First of all, the model is a class that represents a domain. These domain objects often encapsulate the data stored in the database and the code used to manipulate these data and implement the business logic of specific domains. Secondly, views are pages that generated dynamically, which are data streams returned by different terminals. For example, the HTML returned by the browser or the JSON returned by the Web terminal. Finally, the controller is a special class that manages the relationship between views and models. It communicates with the model and decides which view to present. It is a service provided based on the user's Action, that completes the concatenation of the model and the view.

The architectural styles of MVC and REST are basically tailor-made for data service platform. Through analysis on the application of the existing digitalized oilfields, it is found that the current application is mainly based on the presentation of the model and the service for the model. With the model itself, it is exactly the architectural system with MVC style.

M: The data model directly operated by the user.

V: Presentation of the data model.

C: Services that provide operations on models and services provided for the users' operational logic. Considering that users may use different terminals, V may use different technologies and methods to present data models. However, the operations for the same business logic are the same. Therefore, the naturally layered framework of the digital oilfield is partially used to provide the data model and services for that model, while the other part is to present the data model on different terminals. In this

environment, the basic interaction logic is client-server request. The URI not only unambiguously identifies a reference in the global address space, but also tells us how to interpret that request. REST provides the best interpretation for the extension of oilfields based on the MVC framework.

Through the URI, the data model required by different terminals can be provided. For example, data presentation forms based on XML or JSON can be provided for desktop and web terminals, while JSON or images are provided directly for mobile terminals. As V, URI provides different data model presentation ways for terminals.

The data model and the services that provide operations to the data model are positioned as M is totally in line with the definition style of the MVC framework. Here, the services that provide operations for the data model are extracted separately and denoted by S, which is for better separating the structure. The separation of S (Services) can be developed as a separate API. On the one hand, it facilitates the reuse of components; on the other hand, it provides a better way to distribute workloads for developers.

C provides response services specifically for the users' logical operations. Using the S separated from M to provide users with M, and then it is directly published in the form of URI with RESTful style.

### 3.2 REST Data Service Framework

The REST data service framework (Fig. 2) adopts the technology of ASP.NET MVC2 to realize the MVC+RESTful style. Through ASP.NET MVC2 project, it is published as a website. It can fully utilize the features of the MVC framework to separate the concerns of M and C. The Unity container provides the system features of IOC and supports the expansibility and manageability of the system.

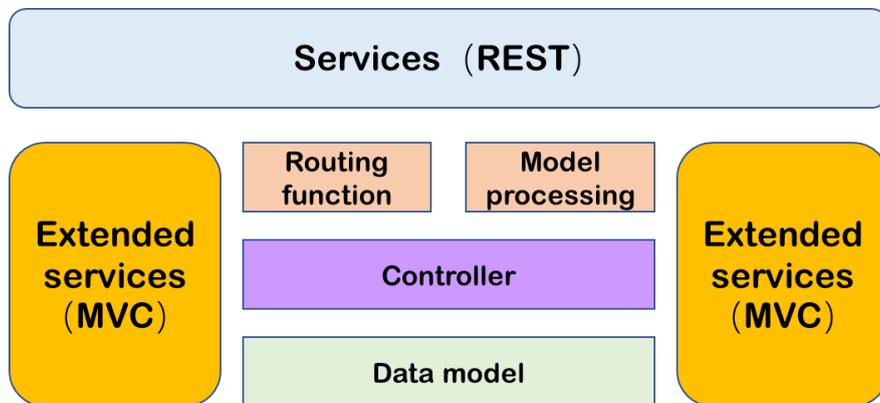


Fig. 2 Schema of REST Data Service Platform

In the process of system startup, RESTHTTP Application completely rewrites the original system behaviors by using the template mode:

```
UnityContainer = OnCreateContainer() new UnityContainer();//Use the Unity container to implantation the IOC mode.
```

```
OnInstallBootStrapper();//The constraints of the installation system, called boot constraints.
```

```
OnInstallServiceEngine(); //The public API that needs to be used when installing the controller.
```

Normally, these APIs are the S separated in the M above.

```
OnInstallServices();// Installation services. It uses the scan service catalog to automatically install the service according to the service interface, which is more concise than XML registration.
```

```
OnBootStrapperStart();//Execute the installed boot constraints to constrain the behavior of the system.
```

The REST data service platform uses the Unity container for management, and provides constraints for the system through `OnInstallBootStrapper()` at the beginning of the system booting. `RESTHTTPApplication` provides three boot constraints of `RestfulRoutesBootStrapper` (routing boot), `RestfulControllerFactoryBootStrapper` (service construction factorial boot) and `RestfulErrorHandlerBootStrapper` (error handler boot). Among them, `RestfulRoutesBootStrapper` completes the routing generation of the system, and completes the routing guidance of data services such as general query, customized query, file processing, and domain model in this part. Developers can modify this part to achieve different identification of URIs in RESTful. `RestfulControllerFactoryBootStrapper` uses a self-defined factory method to replace the default factory of the system to obtain services from the Unity container, so as to achieve the expansibility of the service. `RestfulErrorHandlerBootStrapper`, on the other hand, replaces the system's default ERROR handling with a self-defined `RestfulErrorHandler`. `OnBootStrapperStart()` completes the boot constraint at the end of the service booting, guiding the URI resource to the designated controller through the routing function. Data models are generated in the controller, and then processed according to the terminal type in a REST way. Then it will be returned to the terminal in different data formats (such as JSON or HTML).

#### **4. CONCLUSION**

All data of oilfield exploration and development are resources in the field of exploration and development business, and the most suitable way to publish resources is REST. Therefore, this paper studies the REST service form, and builds a brand new data service system based on the metadata model and business object model, which supplements the original data service platform.

Taking the oil and gas field exploration and development business object model as the core, the REST information integrated service platform framework develops various data services in the form of service components, in which realize a standard interface of the platform framework. The platform framework manages all data services uniformly through the service component container, loads and runs each data service when the platform starts, and provides unified REST-form services and data in JSON and other formats. Adopted the technology of ASP.NET MVC2, this service platform framework realizes MVC+RESTful style, and is capable of publishing ASP.NET MVC2 project as a website. It can fully utilize the features of MVC framework to separate the concerns of M and C, and the adoption of Unity container provides the system features of IOC, providing support for the expansibility and manageability of the system. SL Oilfield and other old oilfields in eastern China are building digitalized oilfields. Facing the application requirements and actual conditions of various businesses, as well as various data sources such as databases, graph files and big data formats, the service platform provides data access in various modes, including metadata-based ORM mapping object access, SQL-based data access, and file-based access. After extracting the functions of the data access layer into contracts, users can easily switch between different implementations at any time, which provides better data services for the exploration and development of oilfield.

#### **REFERENCES**

- [1] SOA[EB/OL]. <http://www.soa.org/>, 2000.
- [2] IBM DW. Service-Oriented Architecture[EB/OL]. <http://tech.ccidnet.com/>, 2005.

- [3] Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures[D]. Dissertation: the University of California, Irvine, 2000.
- [4] REST and SOAP[EB/OL]. <http://www.infoq.com/cn/articles/REST-soap-when-to-use-each>, 2010.