A Web Service Software Design Pattern Targeting Power System Data Analytics and its Application on State Estimation Data

Zhantao Fan[1,a], Yuan Xu[2,b*], Yiping Chen[1,c], Ligang Zhao[2d], Yan Guo[1,e], Hongyue Zhen[2,f], Wenjun Mao[1,g], Tinghui Zhou[2,h], Xiaoshan Wu[2,i], Guanbiao Huang[2,j], Yingqi Tang[2,k], Changxiang Wang[2,l]

[a]fanzt@csg.cn, [b*]xuyuan@csg.cn, [c]chenyp4@csg.cn, [d]zhaolg@csg.cn, [e]guoyan@csg.cn, [f]zhenhy@csg.cn, [g]maowenjun@im.ehv.csg, [h]zhouth@csg.cn, [i,]wuxs@csg.cn, [j]huanggb@csg.cn, [k]tangyq@csg.cn, [l]wangchangxiang@csg.cn

[1]China Southern Power Grid, Guangzhou 510663, Guangdong Province, China
[2]State Key Laboratory of HVDC, Electric Power Research Institute, CSG, Guangzhou 510663, Guangdong Province, China

Abstract More and more data-driven analyzing methods are proposed to thoroughly reveal risks of the high proportion new energy power system in academia. However, the development of digital infrastructure of power grid enterprises can't keep pace, due to the security requirements, and, more directly, the data silos between different data services. A web service software design pattern targeting power system data analytics is introduced in this paper to improve data interaction efficiency, and the pattern is validated by an actual application on state estimation data.

Keywords: Software Design Pattern, Power System Data Analytics, State Estimation

1. Introduction

With the rapid development of new energy power generation technology and integrated energy systems construction, the increasing uncertainty of both power supply and energy consumer is bringing severe challenges to the safe and stable operation of the power grid[1]. Thanks to the innovative breakthroughs in computing ability and efficient algorithms, researchers propose more and more novel data-driven analyzing methods in power forecasting[2], stability assessment[3], system operation[4], etc. However, the accessibility and transformation efficiency of power grid data is limiting the application of those methods. On the one hand, it is very difficult to establish a common data set or benchmark with real operation data, due to the strict confidentiality requirements. On the other hand, in power gird enterprise, data-related applications are often highly customized, resulting in a lack of interaction between different data services. Therefore, a well designed web service software pattern is not only necessary but also serves as an effective framework to develop a cohesive application and a key factor in enhancing the power system analyzing level.

Previous studies and implementations have contributed new insights to the design of power grid software. Many researchers have focused on enhancing multi-source data governance and convergence, recognizing its importance in this domain[5]. Additionally, microarchitecture and

ICIDDT 2023, November 03-05, Zhenjiang, People's Republic of China

DOI 10.4108/eai.3-11-2023.2342227

microservice approaches based on cluster techniques have gained significant prominence, assuming increasingly crucial roles in power grid software design[6]. Moreover, there is a growing demand for improved response speed and real-time performance in these programs[7].

This paper propose a web service software design pattern targeting power system data analytics. Firstly, typical requirements of power system data service, including confidentiality, uniformity, etc, are thoroughly summarized and accurately described. Secondly, the principle of the proposed design pattern is introduced, focusing on its modularity, flexibility and adaptability to evolving further requirements. Finally, through actual application on state estimation data, the proposed pattern is validated.

2. Requirements Description

2.1 Data Confidentiality

Data confidentiality requirements involve ensuring the privacy and confidentiality of sensitive information within a system. Unexpected data leaking of power grid data, such as grid structure, substation coordinates, electrical parameters of power equips, etc, raises risks of potential malicious attack in both cyber and physical sources[8]. According to China Power Monitoring System Safety Protection Provisions, all operation service system within power generation enterprises and power grid enterprises shall be divided into production control areas and management information areas. Usually, there are IV~V areas in practice and severe data transfer restrictions between different areas.

2.2 Data Uniformity

Data uniformity generally requires accurate, reliable and synchronized across components, which is essential in distributed systems and micro-service based architectures. Specifically, in the context of this paper, the focus is maintaining consistency between various grid operation and analyzing systems. These systems include grid state estimation, electromechanical and electromagnetic transient simulation, wide area measurement, among others. Although these systems serve the same physical power grid, there are substantial disparities in their approach to description and modeling. Consequently, the data utilization efficiency of software will become extremely low if data uniformity is not prioritized during the design stage.

2.3 Data Presentation

There are various modes of interaction on power grid analyzing data. E.g. Statistic data on power generation and consumption is typically collected by measurement terminals, stored in databases, and accessed as required; It is essential to present the procedure data from the analyzing module, such as the iterations of power flow calculation and transient simulation, to the analysis and monitoring crew in real-time, either in a row-by-row or point-by-point manner; The most common method for transferring and applying power gird models and other electrical equipment simulation models is through files, which can be in standard or non-standard formats. All these interaction modes needs different data transforming techniques. Hence, a thoroughly designing on data presentation modes takes high priority during the development of power system web services.

2.4 Cross-platform

Cross-platform requirements encompass more than just the ability of applications to provide a unified, standardized, and seamless presentation across different clients, such as desktop terminals, mobile devices, and various browsers. They also involve the demand for services to adapt to different operating systems and chip architectures on the server side. Cross-platform compatibility reduces the need for platform-specific development and allows wider reach and accessibility for users. Achieving this requires careful consideration of responsive design, adaptability to different screen sizes, and compatibility with various browsers and operating systems.

2.5 Other Requirements

Other general web service software design requirements are briefly introduced in Table 1.

Table 1. Other General Web Service Requirements

Requirement	Content
Scalability	Scaling resources according to the actual load level automatically.
Reliability	Ensuring the system operates normally even in the event of component failures or unexpected glitches.
Performance	Optimizing algorithms and data structures.
Maintainability	Focusing on modularity, code conventions, and documentation.
Monitorability	Integrating mechanisms for logging, performance metric collection, error alerts, and so on.
Testability	Using appropriate testing frameworks and tools that are easy to simulate and isolate external dependencies.
Community Support	Choosing popular open-source frameworks and libraries to receive better support and collaboration.

3. Design Pattern Principle

3.1 Data Masking and Synthetic Grid

Due to the high confidentiality requirements of power grid data, the data acquisition process is highly restricted, severely limiting the efficiency of data utilization. Data masking techniques are commonly used to transfer grid data securely while also preventing the leakage of sensitive information. Statistics and cryptography are two main data masking methods. Statistical methods such as sampling, shuffling, randomization, and substitution focus on hiding sensitive information within the full data set while maintaining statistical characteristics. In comparison, cryptography uses encryption and tokenization methods to ensure that the full data set can be provided without being decoded. Designer can choose between the two approaches after balancing the cons and pros.

Synthetic grid[9] or synthetic network is a cutting-edge technique used to construct a simulated power system that imitates the characteristics of the original grid while concealing all sensitive information. Leveraging graph theory and statistics, synthetic methods are capable of generating a detailed model of the fabricated grid. This technology offers a balanced approach to data confidentiality and the preservation of the original power system's characteristics. In scenarios where overall performance characteristics of the grid are the focus, and strict data accuracy is not required, synthetic network technology is an effective solution. For instance, it can be used to study the frequency response characteristics of the grid. In addition, the local synthetic method can conceal partial areas of the original grid data while presenting a complete grid structure. Designers can apply this technique based on the purpose of data service.

3.2 Data Conversion Module

Incompatibility and lack of uniformity are significant factors contributing to the prevalence of data silos. Incompatibility arises from the tendency of different systems to focus solely on fulfilling their basic functions without considering the interconnectedness of data with other systems. On the other hand, lack of uniformity stems from the absence of a universally unique identifier for the same power equipment data across the numerous operation monitoring and analysis systems employed by power grid companies. Consequently, it becomes imperative to incorporate a comprehensive and independent data conversion module that can deliver a resilient, scalable, and maintainable data service. When designing such a module, the following principles are strongly recommended:

(1) Global Identifier Assignment

Considerable efforts have been made to establish universally unique identifiers for each power system equipment. However, the implementation of relevant standards has been limited to specific fields. Therefore, until grid companies establish a centralized identifier registration authority miraculously, it would be beneficial to develop a global identifier assignment mechanism within the developing service. This mechanism can be treated as a unique identifier in all data interactions.

(2) Separated Naming Module

Data mapping is an essential and vital process in the complex multi-system data interaction of power grid companies, as it helps to improve data application efficiency. However, the manual mapping process can be time-consuming and require a significant amount of work. To streamline this process, a separate naming module can be implemented. This module treats names or identifiers as individual instances that are referenced by data instances, rather than setting the name as a fixed attribute. By doing so, the data mapping process becomes equivalent to adding aliases of data instances to the naming entities. Introducing a separate naming module can greatly assist in automating the data mapping process, reducing the workload on personnel, and improving overall efficiency.

(3) General Data Format

In real-time monitoring applications of the power system, it is common to utilize relatively basic data formats typically used in the transport and network layers. However, working with these formats often requires specific parsing programs, leading to a potential issue of technological monopolies. To enhance data utilization and collaborative efficiency, it is recommended to adopt a standardized data transmission specification for non-real-time services. In this context, employing a widely accepted and supported general data format, such as XML, JSON, CSV, etc, can greatly benefit the power system. These formats have proven its accuracy, reliability, and efficiency in various industries and enjoys extensive community support.

(4) Validation Mechanisms

Detailed data validation plays a crucial role in efficiently identifying program bugs. It encompasses various aspects, such as verifying data types, ranges, formats, mandatory fields, and adhering to specific business requirements. Going beyond the basics, high-level validation involves mathematical analysis, logical derivation, and algorithmic checks. In addition to a robust and intelligent validation process, the significance of thorough validation result reporting cannot be overstated, particularly when encountering validation errors. To generate comprehensive reports, it is essential not only to ensure a high level of completion in the validation mechanism but also to employ highly decoupled program modules and advanced data structures.

3.3 Transmission Protocol

Nowadays, different data presentation forms require different transmission protocols in the architecture of anterior and posterior end separation software design. The most common protocols used are HTTP (Hyper Text Transmission Protocol) and HTTPS (HTTP Secure), which can meet the majority of requirements such as querying parameters or statistical results from a data center, validating access authority, and achieving operation results from the web.

However, HTTP is not sufficient for handling real-time and pseudo real-time data due to its time-consuming handshake process and unidirectional connection characteristic. In such scenarios, WebSocket, a newly developed instant messaging technique based on HTML (Hyper Text Markup Language) V5.0, is more suitable.

Additionally, various standard and self-defined file data are extensively utilized in power grid operations and business systems. Therefore, FTP (File Transmission Protocol) is a better choice for those with higher requirements for file transfer rate, volume, and frequency.

In summary, selecting the appropriate transmission protocol is crucial in modern anterior and posterior end separation software design architecture. While HTTP and HTTPS can fulfill most requirements, WebSocket[10] and FTP are better suited for real-time data handling and file transfer respectively.

3.4 Compatibility and Consistency

To ensure compatibility and consistency across all operating systems, such as Windows, macOS, and mainstream release versions of Linux, in client and server both, it is essential to follow these design principles:

(1) Choose a cross-platform framework or language, such as Java, Python, React Native, and Electron, that allow developer to write code once and deploy it on multiple platforms.

(2) Use libraries that are compatible with multiple platforms to ensure application runs smoothly across all operating systems.

(3) Consider the various screen sizes and resolutions of different devices and design software to be responsive and adaptable to different screen sizes.

(4) Thoroughly test on multiple platforms: It's important to test software on different operating systems to ensure it works as intended and is free of bugs.

(5) Provide consistent user experience: Ensure that the user interface and experience are consistent across all platforms to provide a seamless experience for users.

(6) Deploy applications using containerization techniques, such as Docker. By encapsulating all necessary dependencies within an individual container, it creates an optimal environment that ensures the highest performance and efficiency.

4. Case Validation

4.1 Application Architecture Sketch

A web service application on processing state estimation data is taken as an example to validate the proposed design principles in this section. The architecture sketch and main workflow of the service is shown in figure 1.

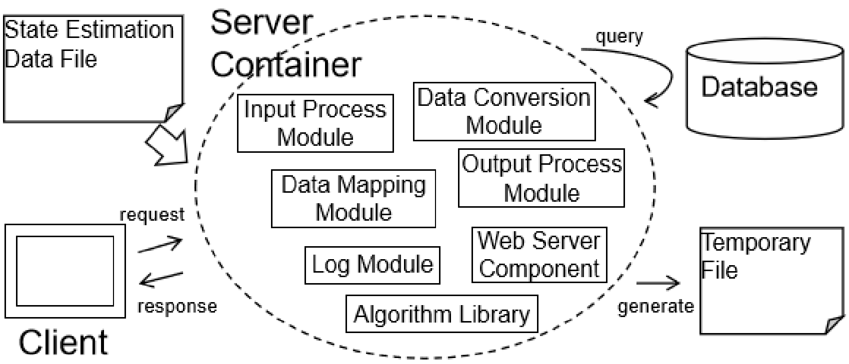

Fig 1. A State Estimation Data Service Architecture Sketch

In the architecture sketch (Fig1), the entire service application is deployed within a server container and comprises seven main components. The core modules, namely the data conversion module and data mapping module, play a vital role in transforming input data into the desired format. The input and output process module handles data I/O operations, querying from the database, and generating temporary data files. The web server component handles client requests and responses. The log module captures application states and information, while the algorithm library provides essential functions for each module.

To elaborate further, the application is developed using *Python 3.12*, a versatile and cross-platform programming language. The separate data conversion module ensures a resilient, scalable, and maintainable data transformation process. For the web server component, *Flask 1.1.14* with *Gunicorn 21.0.1* is chosen, providing necessary protocols such as HTTP, Web Socket, FTP, etc. The database relies on *MySQL 5.7*, an open-source database, and *SqlAlchemy 1.2*, a Python database toolkit, which facilitates seamless integration with the data mapping module while maintaining a connection pool for efficient data querying.

4.2 Details in Data Conversion Module

The mission of conversion module is to transform the estimation data in file into a power flow calculation form and the flow chart of the data conversion module is described in figure 2.

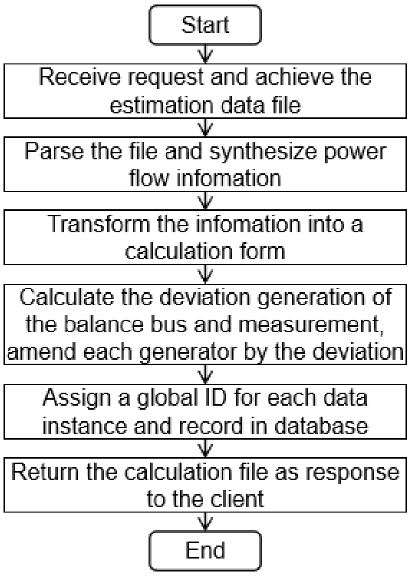

Fig 2. Main Flow Chart of the Data Conversion Module.

In Fig 2, the main data conversion process consists of six essential steps. It begins with a POST request, a HTTP request method, initiated by the client, specifying the desired target estimation data file for calculation. The input module handles and parses the file, while the conversion module synthesizes all necessary power flow information and organizes it within an internal data structure, facilitating easy transformation into a calculation format in step 3.

Subsequently, a data validation mechanism is implemented by calculating the deviation in power generation between the estimated values and the actual measurements. This mechanism also adjusts the output of each generator to offset the identified deviation. Once the validation is completed, a global identifier is assigned to each data instance and recorded in the database. Finally, the program concludes by returning the calculation file as a response to the client and awaits the next request.

The case module introduced in the current section is effectively functioning on the Power Dispatching Cloud Platform of China Southern Power Grid.

5. Conclusion

This paper introduces a web service software design pattern addressing power system data analytics, after conducting a thorough requirements analysis. By demonstrating its application in processing state estimation data, the proposed design pattern is verified and provides a template for future software design within China Southern Power Grid.

References

[1] L.P.M.I.Sampath ,Gupta A ,Ong Y , et al., "Evolutionary Multitasking Framework to Support Optimal Power Flow Under Rapid Load Variations," Southern Power System Technology, Papers 11(10), 103-114(2017).

[2] Wang Bo, Feng Shuanglei, Liu Xiaolin, et al., "Short-Term Wind Power Prediction Considering Identification and Testing of Transitional Weather Processes," Southern Power System Technology, Online Paper, http://kns.cnki.net/kcms/detail/44.1643.tk.20230523.1726.004.html.

[3] Zhang Pei, Zhu Zhujun, Liu Zhao, et al., "Online Assessment Method for Static Voltage Stability of New Power System Based on RReliefF-BP Network." Southern Power System Technology, Paper 17(03), 65-74(2023).

[4] XI Lei, DU Xiong, LI Yanying, et al., "Automatic Generation Control Algorithm Based on Ensemble Learning with the Idea of Reinforcement Learning," Southern Power System Technology, Paper 17(07),74-82(2023).

[5] XU Hongqiang, CAI Yu, WAN Xiong, et al., "Architecture and Key Technologies for Big Data Platform in Power Grid," Power System Technology, Paper 45(12),4798-4807(2021)

[6] XIAO Yi, LI Junye, FANG Biwu, et al., "Microservice Based Data Integration Platform and Architecture Design of Online Security and Stability Analysis System for A Bulk Power System," Proceedings of the CSEE, Proc 42(S1),126-135(2022).

[7] Zhao Lin, Zhou Erzhuan, Fan Long, et al., "Architecture of Online Analysis System for Power Grid Oriented to Rapid Response," Automation of Electric Power System, Paper 43(18), 95-101(2019).

[8] K. Chatterjee, V. Padmini and S. A. Khaparde, "Review of cyber attacks on power system operations," 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, India, 2017, pp. 1-6, doi: 10.1109/TENCONSpring.2017.8070085.

[9] H. Sadeghian, S. H. Elyas and Z. Wang, "A Novel Algorithm for Statistical Assignment of Transmission Capacities in Synthetic Grid Modeling," 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, OR, USA, 2018, pp. 1-5, doi: 10.1109/PESGM.2018.8585532.

[10] A. Hasibuan, M. Mustadi, I. E. Y. Syamsuddin and I. M. Anis Rosidi, "Design and implementation of modular home automation based on wireless network, REST API, and WebSocket," 2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Nusa Dua Bali, Indonesia, 2015, pp. 362-367, doi: 10.1109/ISPACS.2015.7432797.