Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 10, No. 3, 2023

Design and Implementation of Data Service Middleware in Earthquake Precursor Instrument Platform

Can Zhou[1, a], Xiyin Liang[1, *], Ke Li [2, b] and Lijie Wang [2, c]

[1] Engineering Research Center of Gansu Province for Intelligent Information Technology and Application, China
[2] College of Physics and Electronic Engineering, Northwest Normal University, China

[*] **Corresponding author**: Xiyin Liang (Email: zhoucan0301@foxmail.com), [a] lvsq9707@163.com, [b] 2548626332@qq.com, [c]1085052680@qq.com

Abstract: With the gradual development of the earthquake signal early warning system, more and more earthquake precursor signal observation instruments and equipment have been added to the earthquake signal early warning system, which has also caused various compatibility issues. This article designs a data service middleware solution based on B/S architecture for the interaction between earthquake precursor signal observation equipment and remote access terminals on the earthquake precursor instrument platform. Combining embedded technology, it achieves compatible and universal deployment of heterogeneous terminals, and achieves the ideal function of data service middleware. At the same time, in the data service middleware system, real-time data access, presentation, and log upload and download of earthquake precursor instruments are implemented, as well as remote editing of basic information such as instrument IP addresses and station positions, as well as instrument status monitoring and other functions.

Keywords: Earthquake precursors; Data service middleware system; Data monitoring.

1. Introduction

Earthquakes are a highly destructive natural phenomenon that poses a huge threat to the safety of life and property. However, the occurrence of earthquakes is a relatively complex natural phenomenon, accompanied by a variety of earthquake precursor anomalies [1]. In order to predict and observe earthquakes, various instruments and equipment for observing earthquake precursor signals are needed. These factors have led to the increasing complexity of the usage specifications and design architecture of the entire precursor signal observation network, leading to the normalization of operation in a clustered manner on heterogeneous platforms, and the cost of development and maintenance is also increasing day by day [2]. Therefore, in order to address platformization differences and simplify service application development, this article designs a data service middleware solution based on the B/S architecture for the interaction between earthquake precursor signal observation equipment and remote access terminals. Combining embedded technology, it achieves compatible and universal deployment of heterogeneous terminals, effectively filling the various shortcomings of the original instrument platform system [3].

2. System architecture design

Based on the overall demand for earthquake precursor signal observation equipment, network communication model design analysis, and various characteristics that can communicate on different heterogeneous platforms in the future development trend, the embedded middleware system should meet the following requirements in this regard:

(1) In response to the demand for multiple data formats on the application side, the middleware system needs to provide multiple types of data application services.

(2) For multiple types of communication protocol requests, multiple network communication interface models need to be reserved.

(3) For different heterogeneous platforms, it is necessary to have a unified data format and implement basic functions.

(4) The middleware installed on the local hardware platform needs to have data file storage, local data processing, and decision-making capabilities.

Because middleware is a software used to connect the bottom operating system layer and the upper application program, it is an independent system service software that can coordinate network resources and communication [4]. Middleware is usually integrated on development or runtime platforms. That is to say, as a "messenger" of platform and communication, it is only applied in distributed systems and is called a middleware program [5]. Middleware can black box complex underlying systems, unify specific development environments, and greatly reduce workload for different types of heterogeneous platforms, requiring only a single development. Middleware makes the development of application layer software simpler, shortens the development cycle of supporting software systems, and reduces the difficulty and cost of system operation, maintenance, modification, and operation management. Separate complex problems from upper level application development, and provide standardized and unified data information services for subsequent development of multiple types of applications and interactive systems.

Embedded middleware is a type of system software commonly used between embedded system software and application interaction end. It utilizes a series of lightweight systems suitable for embedded platforms, such as the cropped Linux operating system, to provide basic usage functions and provide a relevant environment for the development and operation of upper level application software, enabling embedded development software to run on different underlying operating systems on different platforms [6]. The key points of embedded middleware are as follows:

(1) It is a system software that runs between embedded

operating systems and embedded application software, serving as a link between the services between the two.

(2) It is deployed on embedded devices to provide support for information exchange between applications and databases, including information transmission, security maintenance, and improving work energy efficiency;

(3) It can provide a unified development environment platform for embedded applications, allow the use of API interfaces, and directly utilize middleware to develop programs. The developed programs can run directly on middleware.

This article designs a new type of embedded data service middleware suitable for earthquake precursor signal observation instruments based on the functional characteristics of the required middleware and the requirements of application hardware platforms. This new middleware can combine the characteristics of earthquake precursor signal instruments and software, increase the network operation ability of earthquake systems, and achieve multiple functions of the application end. The designed new data service middleware architecture model is shown in Figure 1.

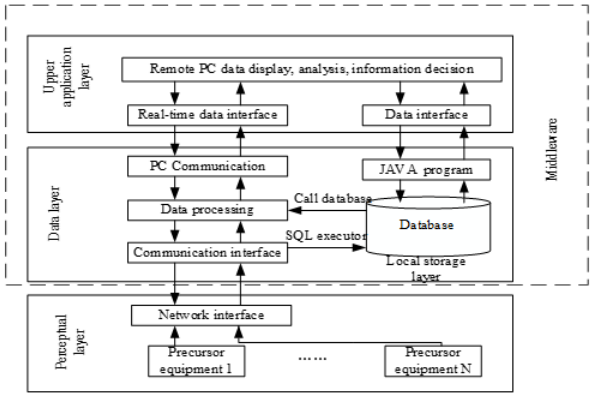

Figure 1. Middleware Structure Model Diagram

This article constructs a new data service middleware model, which is a key component of the earthquake precursor signal observation instrument network. It is mainly composed of three parts: the upper application layer, the data layer, and the bottom perception layer. Each layer structure reserves multiple interfaces, including real-time communication interfaces, precursor network coordinator interfaces, and historical data interfaces. These reserved interface modules are interconnected through the middleware system. The local area network plays a wireless link function in the entire middleware system, enabling the upper application layer to receive, receive, and respond to the data layer on the PC end; The RS232 interface of the device gateway in the middleware, which is the communication interface between the coordinator and the precursor network coordinator interface of the seismic precursor signal instrument, can be connected to achieve the middleware's function of obtaining precursor signal data from the underlying perception layer through a serial port; The collected data of relevant precursor signals comply with the relevant data protocol section of the "Network Communication of Special Equipment for Earthquake Precursor Network" protocol; The embedded web server forwards data requests sent by the application layer, issues instructions to the data layer, and receives responses.

3. System functional design

Because the main tasks of middleware systems are instrument platform monitoring and data processing, they have different functional requirements for different structural layers. For the relevant processing services of the upper application layer, it is beneficial for users to read and download precursor data, modify the network information of some sensors in the seismic precursor signal observation instrument network, manage user related requirements, and report whether the equipment is operating normally [7]. Has the following many characteristics and application functions:

(1) PC side web browser direct control: breaking away from the traditional C/S framework, directly using a universal web browser as an interactive interface application software, achieving control of all types of precursor sensing devices without the need to download third-party applications, greatly reducing usage costs.

(2) Precursor device monitoring: Currently, the platform precursor data of middleware systems consists of temperature precursor devices, pressure precursor devices, and rainfall precursor devices. Users can check the precursor devices at any time through the interactive page in their web browser to see if they are in normal working status and promptly handle abnormal device situations. The specific system functional division designed by the middleware is shown in Figure 2.

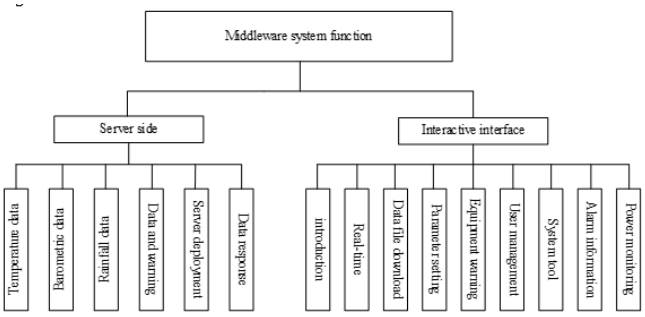

Figure 2. Middleware system functional business diagram

4. System test and function implementation

Before conducting middleware web side testing, it is necessary to port the system package compiled in Idea to the hardware platform. As the relevant migration runtime environment has been installed and configured in the previous text, after migration, simply enter the command Java jar+ system package to deploy and run the relevant services. Because the middleware system designed in this article adopts the Spring Boot framework, a series of related deployment hardware platform information will appear when loading the middleware system package, indicating that the entire middleware system package is already running and the related functions are being deployed.

After the middleware system runs on the hardware platform, the PC connects to the hardware platform and uses the IP settings of the hardware platform and the open port of the embedded server in the middleware system to link to the middleware system. At the beginning of deployment, relevant program deployment information will be generated on the hardware platform, as shown in the system deployment information interface in Figure 3.

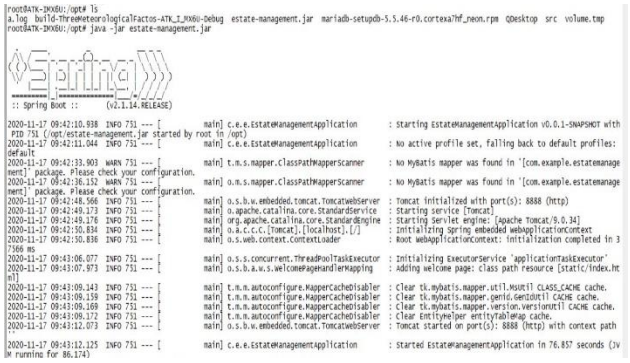

Figure 3. System Deployment Interface

After the deployment is successful and the user or administrator identity is verified, you can enter the PC interactive system. You can see the instrument name, station code, device ID, and device IP in the upper left corner, and multiple secondary jump pages below, including instrument introduction, real-time measurement data, data file download, parameter settings, device warning reminders, user management, system tools, alarm information, power monitoring, and other functional interfaces.

Figure 4. Real time measurement data interface

Click on real-time measurement data on the left side to jump to the corresponding interface, as shown in Figure 4, where the latest measured precursor signal data on the hardware platform can be read in real-time. You can also download the historical precursor data of TXT text to the PC for storage through the data file download page. Parameter settings can be used to set some basic instrument information, such as station code, device ID, open port, etc. There are also some basic IP settings, time correction, IP operation records, as well as some equipment operation alarm records and power monitoring.

5. Conclusion

This article introduces the design and implementation of a new type of embedded data service middleware for earthquake precursor signal observation instruments. Based on the hardware equipment of the earthquake precursor device platform, an embedded data service middleware model was constructed, and the entire middleware system software design was implemented, including embedded software design and environmental installation. Among them, the entire middleware system runs on the local hardware platform, and the browser on the PC connects to the local router. You can view precursor device information and real-time data by opening the browser page, download and store historical data files locally, and then analyze relevant data. The middleware system designed in this article can achieve remote control of data services related to earthquake precursor observation instruments, perform data queries, and modify basic information of instruments and stations, meeting the functional requirements of the middleware system for precursor equipment.

Acknowledgements

The work content of this article has received strong assistance and support from institutions and individuals such as the Gansu Provincial Seismological Bureau and the Gansu Intelligent Information Technology and Application Engineering Research Center of Northwest Normal University.

References

[1] Ma N. Direct economic loss assessment model of earthquake based on machine learning algorithm [D]. Institute of Engineering Mechanics, China Earthquake Administration, 2020.

[2] Li Yishi. Questions and Answers on basic knowledge of earthquake prevention and disaster Reduction [J]. Disaster Prevention Expo, 2013(06):30-35.

[3] Adam Wolfgang,Foitzik Volker,Siering Till Christian. CORBA im Steuerungsbau [J]. Zeitschrift für wirtschaftlichen Fabrikbetrieb, 2022,93(9).

[4] Bao Liqun, Li Fangfang. Indoor Air Quality remote monitoring System based on Embedded Web [J]. Automation & Instrument,2019, (05):10-14.

[5] Ren Tian. Design and Implementation of CAT Monitoring System Server [D]. Nanjing University, 2018..

[6] WANG Chao. Research and Application of Protocol Conversion Gateway Based on Embedded Middleware [D]. Northern University for Nationalities,2017.

[7] Zhou Zhenggui. Node Injection Control Logic Design for Intelligent Home System Based on Wireless Sensor Network[J]. Journal of Sensors,2022,2022.