

Power Terminal Communication Access Network Monitoring System Scheme Based on Design Patterns

Shengchao Yan*, Desheng Wu, Jiang Zhu

State Grid Electric Power Research Institute, Nanjing 210003, China.

*Corresponding author e-mail: yanshengchao@sgepri.sgcc.com.cn

Abstract. In order to realize patterns design for terminal communication monitoring system, this paper introduces manager-workers, tasks-workers design patterns, based on common design patterns such as factory method, chain of responsibility, facade. Using these patterns, the communication monitoring system which combines module-groups like networking communication, business data processing and the peripheral support has been designed successfully. Using these patterns makes this system have great flexibility and scalability and improves the degree of systematic pattern design structure.

1. Introduction

Power terminal communication network monitoring system, as an important pillar of power communication business network system, takes the function of monitoring the system operating status and predicting the performance operating trend. In order to make the system more reliable and design the system structure clearly, it is necessary to study many design patterns to improve the structuration and flexibility of system.

Design pattern [1], which is used repeatedly and is well-known and is cataloged, is a summary of code design experience. Generally speaking, using design pattern is to reuse code and make code easier to be understood and ensure the reliability of code. Therefore, design patterns are widely used to design monitoring system, such as network monitoring, video surveillance, mine safety monitoring, turbine monitoring and environmental monitoring [2-6], and good results are achieved. With power's own characteristics, technology used in power communication network monitoring system is still more conservative, and generally the design pattern of data flow DFD is adopted. Therefore, when a new generation of power communication network monitoring system is designed, it is particularly important which kind of design pattern is used to improve the modularization of system [7].

Based on a combination of common design patterns including factory method, the Chain of Responsibility and Proxy, manager-workers and task-workers are proposed in this article. And then, power communication network monitoring system, which includes modules of network communication and processing business data and supporting peripheral systems, is realized. Manager-workers and task-workers make the system have flexibility and scalability, and improve the degree of systematic pattern design structure.



2. Module of design patterns

2.1. Basic Design Patterns

Common design patterns include the following categories, and these are used in a large number of basic designs as a basis, such as Factory method pattern which gets classes we need by calling different methods, rather than to instantiate concrete instance, and wraps what instance creates; Facade pattern which provides a set of interfaces in subsystem with a unified high-level interface to makes the subsystem easier to use; Chain of Responsibility By decoupling between the sender and receiver for a request, more than one object has a chance to handle this request. And Proxy provides a proxy to control access to the object for the other objects. Based on these patterns, we introduce manager-workers, tasks-workers design patterns to improve the design.

2.2. Manager-Workers Pattern

The extensible manager-workers pattern takes thread pattern. Manager provides a common access interface (generally Singleton pattern) for information, and as the execution object of task, workers are the main pattern of communication interface. This pattern provides an asynchronous pattern, that is, the indirect access to workers. Since this pattern has more threads (thread pool) which provide a greater throughput rate for network communication, this pattern has the ability to face the strong data pressure. Application example of network communication interface module takes the manager-workers pattern. By providing start () and stop () method, this pattern provides Facade pattern method which starts / stops network communication interface modules, as shown in Figure 1.

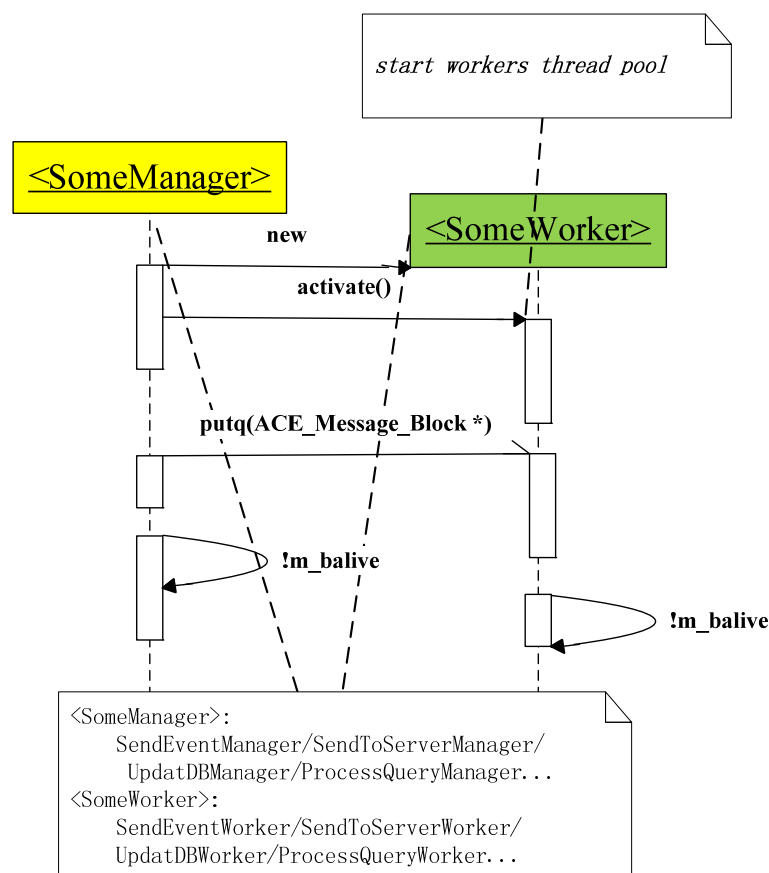


Figure 1. Manager -workers pattern sequence

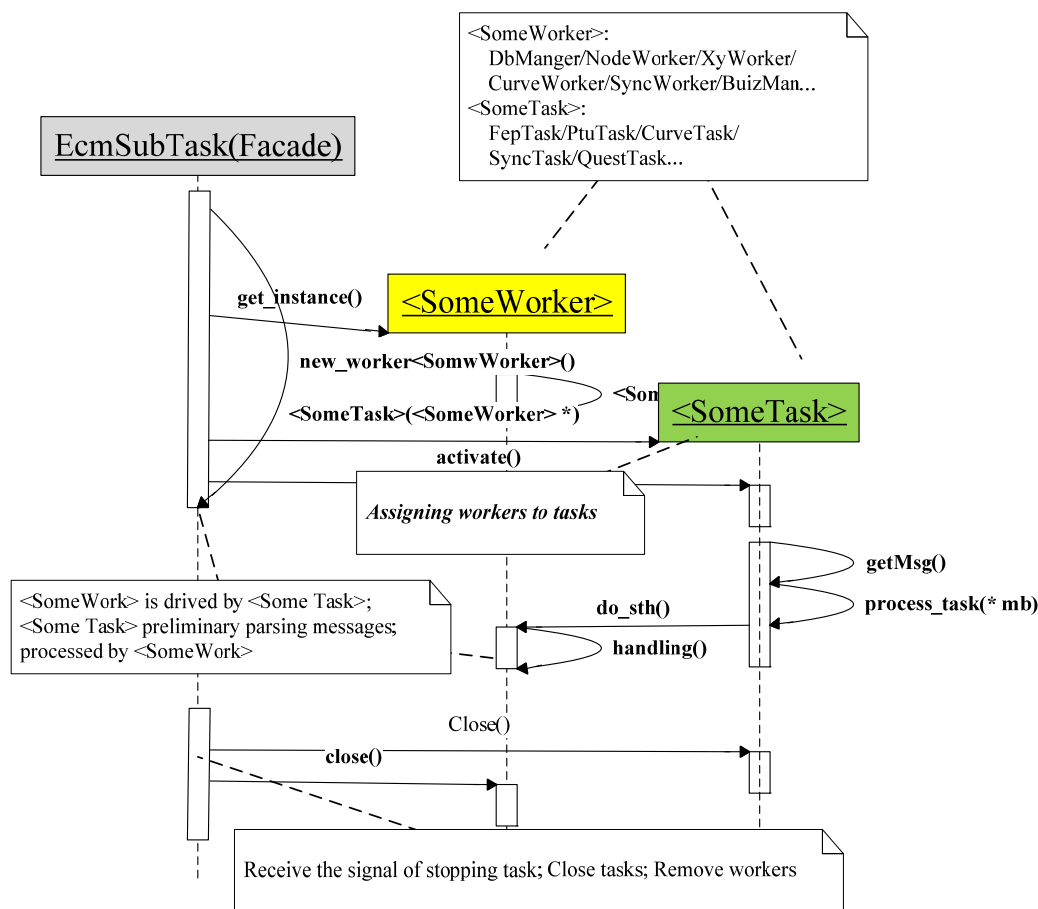


Figure 2. Task-workers pattern sequence

2.3. Task-Workers Pattern

The extensible task-workers pattern likes manager-workers pattern. Task takes thread pattern, and as the object actually processed by task, workers are the main pattern of processing business data module. This pattern has the function of adding or removing tasks, so the system has great flexibility and scalability, and then this pattern provides the basis for dynamic control module.

Application example of processing business data module takes the task-workers pattern. By providing start () and stop () method, this pattern provides Facade pattern method which starts / stops processing business data modules. In this method the application of the task-workers pattern and Facade pattern based on method template starts in the perspective of Larger-Granularity or customizes each module needed to enable, as shown in Figure 2.

Pattern is not independent of each other, and the combination of patterns is often needed to use. On the basis of common design patterns and the two design patterns proposed by the paper, the system is designed and constructed by flexibly using the best results of each pattern, as described below.

3. Design of System Structure

The system designed in this article includes three parts: network communication interface, processing business data, and periphery support environment. As shown in Figure 3, the left is the client, and the right is the Front-end Processor, data and data sources such as the backup server, and the middle is the core processing module of the system.

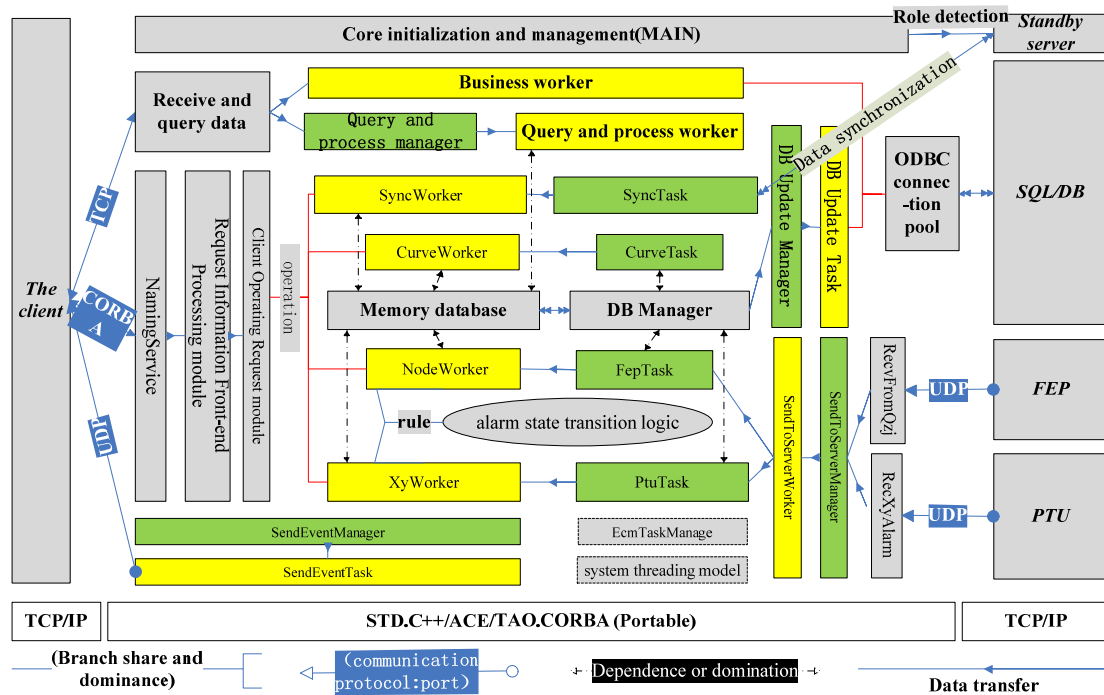


Figure 3. System structure

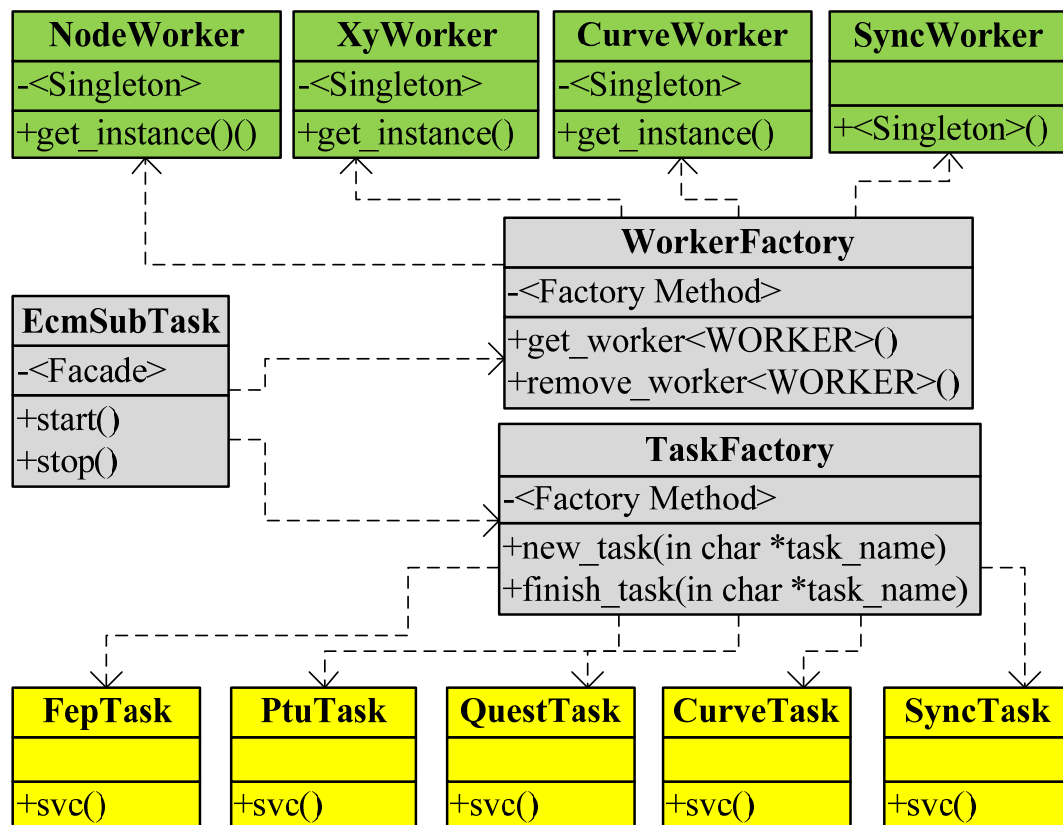


Figure 4. Business data processing

4. System application of design pattern

4.1. Processing Business Data

This module is responsible for processing data obtained, and accessing relational database and memory database, and sending the event message to the client. This module includes the following content: FepTask(FEP data task), PtuTask(PTU data task), NodeWorker(processing node worker), XyWorker(processing protocol worker), CurveTask(recording curve task), CurveWorker(recording curve worker), SyncTask(data synchronization task), SyncWorker(data synchronization worker), RDB(query of relationship database), and the core initialization, etc. as shown in Figure 4.

(1) From open-source portable package, MDB(memory database) saves the necessary data from SQL relational databases, which is the data basis of monitoring and processing data. DbManger(database management module) is used to initialize and manage in-memory database, and to proxy local requests for SQL relational database. By using ODBC to isolate the actual content of heterogeneous databases, the code does not need to be rewritten on the basis of different databases.

(2) Taking Chains of Responsibility, QuestTask responds to client operation requests sent through CORBA service channel. After being preliminary parsed, the requests are processed by the target workers module. Taking task-workers pattern, there are 4 modules in this module group, described as follows in turn.

(3) FepTask is described as follows. By pre-analysis and preliminary processing, this thread is distinguished into tele-signaling, telemetry, tele-signaling displacement and site packets according to the type of packet header. Querying nodes table in MDB, if the corresponding record exists, the structure block resolved will be processed by NodeWorker, and the process is synchronization. NodeWork responds to the node processing requests of other modules(such as FepTask, PtuTask, client operation task), and then corresponding processing is executed based on alarm state transition logic. Specific methods are as follows:

- a. Checking whether it is locked, it will skip if it is locked and continue if it isn't.
- b. If the switch of node alarm state occurs, the alarm will be recorded or cleared in SQL relational databases and MDB, and then the alarm event will be sent to the client by alarm event manager.
- c. Update the latest state information of each node. As shown in Figure 5.

(4) PtuTask is similar to FepTask. By pre-analysis and preliminary processing, this thread is processed by corresponding workers, and the process is synchronization. XyWorker is similar to NodeWorker.

(5) CurveTask is responsible for timing, and CurveWorker is driven to record curve data by CurveTask at appropriate intervals. CurveWorker is responsible for checking node information recorded by the curve, and based on the rule of recording, the curve data is added into the relational database.

(6) Synv Task is responsible for timing, and SyncWorker is driven to check and synchronize data at appropriate intervals. SyncWorker is responsible for checking the node status of FEP, and the alarm status will be changed when timeout.

The application of the above design pattern makes the system automatically increase or remove the task according to the number of messages being processed, which makes the system have great flexibility and scalability.

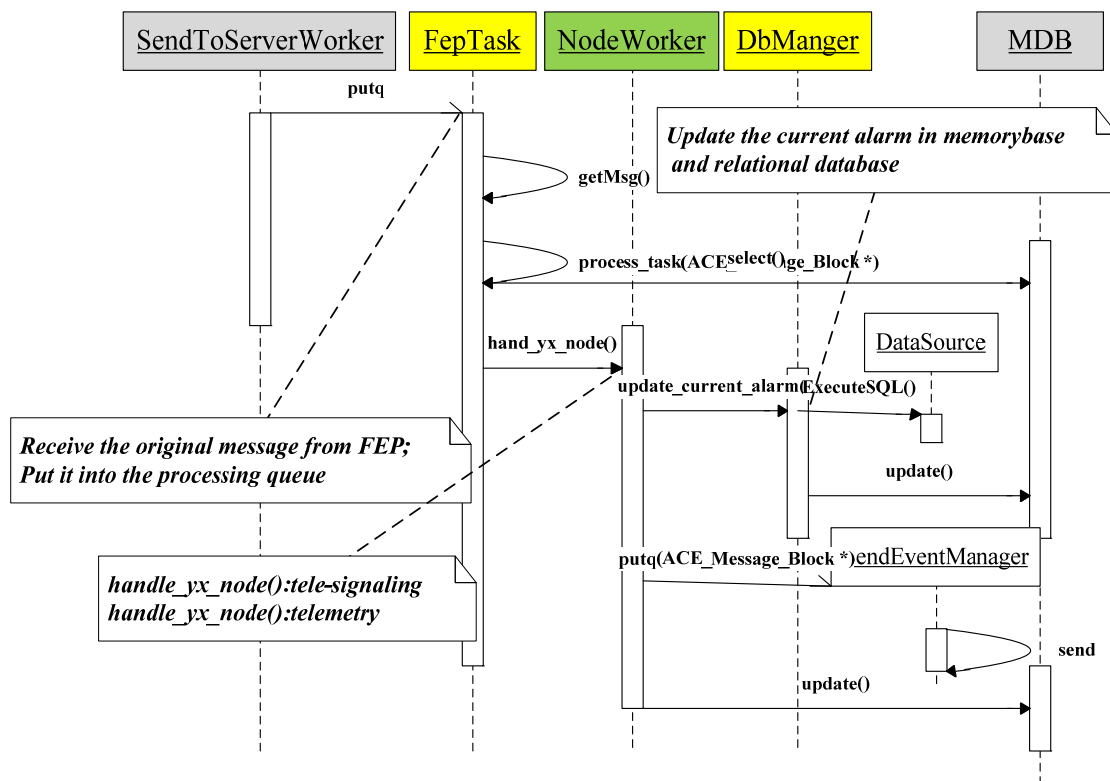


Figure 5. Pattern application sequence

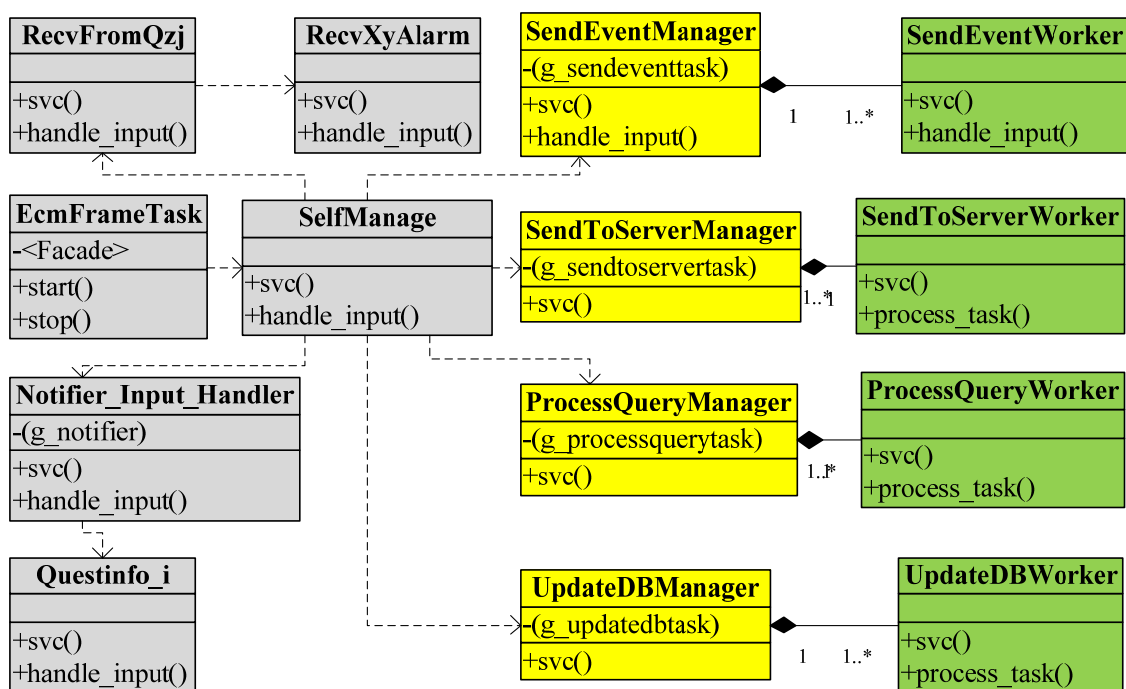


Figure 6. Communication network interface module

4.2. Communication network interface

Communication network interface module is responsible for processing data and achieving communication with the external objects, including sending and receiving. This module contains the following: RecvFromQzj, RecXyAlarm, SendToServerManager, SendEventWorker etc. Unless specify, the following modules are the thread. This module used manager-workers pattern mentioned in the previous section, and Worker modules are multi-threaded thread pool. As shown in Figure 6.

There are four modules apply manager-workers pattern, in turn described as follows:

(1) SendToServerManager receives message sent from RecvFromQzj and RecXyAlarm, and then sends to SendEventWorker which is multi-threaded thread pool. By the primary analysis of message header, the message is sent to the corresponding business task module (FEP task or FEP task) to process by convention (CNM3 is FEP message header or REP is PTU message header).

(2) SendEventManager receives the request of alarm event, and then sends this alarm to PushAlarmEvent which sends to the client. SendEventWorker, which is multi-threaded thread pool, receives the alarm message from SendEventManager, and then sends this message to the client with UDP protocol.

(3) ProcessQueryManager receives the request of querying the memory libraries, and then sends this request to ProcesQueryWorker, which is multi-threaded thread pool. After receiving the request, ProcesQueryWorker queries the memory libraries and then return the result to the client.

(4) UpdateDBManager receives the request of updating SQL relational database from processing business data module, and then sends this request to UpdateDBWorker, which is multi-threaded thread pool. According to this request, UpdateDBWorker updates SQL relational database.

By applying the above design patterns, sending and receiving a strong network data can be achieved, and near real-time asynchronous communication function can be also achieved.

4.3. Peripheral Supporting Environment

As the application basis for pattern, peripheral supporting module provides pattern application with environments, as follows:

(1) The function that the system threading model increases and removes tasks, makes the system have great flexibility and scalability, and provides dynamic controller module with a basic module(task-workers) which is an extension of ACE threading module and supports the buffer pattern of the receive queues.

(2) By registering in this EcmTaskManage module, the thread supports the calling of other objects by the way of name to reduce the coupling.

(3) Alarm conversion logic rules.

(4) By NamingService based on CORBA, the client sends the command to “Request Information Front-end Processing module” of the server. And then this command is sent to “Client Operating Request module” to process client request, and the server behavior is changed.

(5) Establishing ODBC connection pooling accelerates the speed that server connects to SQL relational database, and improves the efficiency of database accessing.

5. Conclusion

Based on the above model, manager-workers and task-workers design patterns are applied in system. , we managed to design the communication monitoring system by combining module-groups like networking communication, business data processing and the peripheral support, which has great flexibility and scalability and improves the degree of systematic pattern design structure. Combining its own characteristics of power communication network, the next step is needed to study the specific design patterns to achieve power communication reliability analysis system [8-10].

References

- [1] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software, ISBN 0-201-63361-2.
- [2] HAO Li: Design and Implementation the Network Monitoring Framework Based on Java Design Patterns [D], DALIAN JIAOTONG UNIVERSITY, 2010.
- [3] GUO Tian-tian, YIN Hong-liang, GAO Ting and LV Mao-chao: Application and Research of Software Design Patterns in Mine Safety Monitoring System Developing [J], COAL MINE MACHINERY, 2011, 32(2).
- [4] ZHANG Jian: The Research and Implementation of Remote Monitoring System of the Steamer Group Based on J2EE Design Patterns [D], North China Electric Power University, 2009(10).
- [5] LIU Bing-jun and TAN Guo-bin: The Application of Design Pattern in Designing and Developing Network Video Monitoring System [J], CHINA SCIENCE AND TECHNOLOGY INFORMATION, 2010(11).
- [6] LI Li and CHEN Jun-jie: Application of MVC Design Patterns on Displaying Interface of Environment Inspect and Control System [J], SHIP ELECTRONIC ENGINEERING, 2010, 30(7).
- [7] TANG Yun-shan, WANG Ping, ZHANG Hui-yong, CAI Xin-zhong and WU De-sheng: New generation power communication monitor and management system [J], TELECOMMUNICATIONS FOR ELECTRIC POWER SYSTEM, 2010, 31 (1).
- [8] Shengchao Yan, Lisha Chen, Juan Qi. Fault Analysis of Communication Network Based on Combined Rules of Fault-Tree and Code Book. Applied Mechanics and Materials. 2014, Vols. 556-562P: 5072-5075.
- [9] Shengchao Yan, Juan Qi. Research on Service Impact Analysis for Power Communication Network Based on N-1 Principle. Advanced Materials Research, 2014 Vols. 846-847: 396-399.
- [10] Shengchao Yan, ,Shunli Lv ,Juan Qi. Study and application on power communication operation mode checking and analysis method. 5thICCCNT, IEEE- 33044: ICCCNT 2014.