The Intelligent Management System for Operation and Maintenance of Shanxi Seismic Station Network Based on JAVA

Yan LIANG and Yun-feng GAO
Shanxi Earthquake Agency, Taiyuan Shanxi 030021, China

Keywords: JAVA, JOPENS, Operation, Maintenance, Noise.

Abstract. This paper adopted JAVA and MySQL to design the intelligent management system for operation and maintenance of Shanxi Seismic Station Network. The system mainly realized station and equipment information management, status monitoring, automatic update the station logs and instrument parameters of JOPENS, real-time monitoring and calculation of station noise and chart output. This system can effectively improve the operation and maintenance efficiency of the station and make the daily work streamline.

Introduction

At present, Shanxi Seismic Network contains 57 seismic stations and 57 strong motion stations. Maintenance personnel will face a huge workload, including daily check station fault and reports the stations log, noise calibration. In addition, most of the work stays in manual finishing stage, the management is less informative. Some parameters cannot be synchronized in the JOPENS system in time when one equipment were replaced in the station. In addition, when the replacement record is incomplete or not timely, the waveform availability rate is reduced, which affects the accuracy of the earthquake output. Some seismologists have done a lot of development and research on operation maintenance adopted B/S or C/S mode to realize information management of stations and equipment, or status monitoring of instrument operation and communication links [1, 2].

System Development

Requirements Analysis

In Shanxi Seismic Network, the monitoring management of station and equipment operation and the calculation of noise are realized separately at present. The timeliness of parameter update is very poor in this pattern. It is inconvenience to maintenance and data analysis. The intelligent management system for operation and maintenance of the Shanxi Seismic Network will integrate the daily work related to operation and maintenance of the network, including station information management, equipment information management, operation status monitoring, automatic update of station logs and instrument parameters in JOPENS system, real-time monitoring and calculation of station noise, and export statistics chart. The system can standardize operation and maintenance, improve work efficiency and ensure data quality. The UML system use case diagram is shown in Fig. 1.
Summary Design

System Processing Flow

This paper analyzes the system processing flow based on the daily work flow of the station maintenance of Shanxi Seismic Network. See Fig. 2.

The system automatically connects to the database of JOPENS, monitors communication connection and real-time data stream of the station, judgments station status and informs the
maintenance staff of station fault information when it starts up. After troubleshooting, the maintenance staff shall report the maintenance records of the station and equipment. If the equipment is replaced, the station and equipment parameters shall be automatically synchronized to JOPENS database, and the log records of station in JOPENS database shall be updated at the same time. In addition, the power spectrum probability density and root-mean-square amplitude of station noise are calculated automatically and periodically. According to the above data, the user can make statistics of the station and equipment maintenance record, the station evaluation monthly report, the equipment replacement roadmap, the station noise monthly report and chart.

**System Structure**

The system adopts B/S structure (Browser/Server mode), which makes the core part of the system function realization focus on the Server and simplifies the development, maintenance and use of the system. The system is built in Eclipse development environment, with Java as the background development language, MySQL as the background database management system, and Geoserver as the map server, combined with the Amap API for map display.

The system uses a classic three-tier architecture framework for model building and interactive processing control. This system establish entity by com.entity. The com.dao contains dao and the implementation methods of the dao. The com.servlet handles the browser's request and response through request and response. The com.util achieves connection, query, modification, etc. with the database. The com.filter implements the function of filter. All pages are displayed in the JSP page.

**Function Module**

According to the requirements of the intelligent management system of operation and maintenance of Shanxi Seismic Network, it can be divided into six parts: information management, operation monitoring, log maintenance, parameter synchronization, noise calculation and chart output. The specific functional module is shown in Fig. 3.

![System function module diagram.](image)

**Database Design**

The system uses MySQL 5.6 to create a database named snms. The database entities used in the system contains user entity, data entity, seismometer entity, data collector entity, communication unit entity, station log entity, equipment log entity, link status entity and data flow entity. According to these entity objects, can obtain the basic model of the data table structure, form a complete data structure. The database includes 9 data tables, wherein, the table structures of the station information table and the station log information table are shown in Table 1 and Table 2.
Table 1. Structure of sta.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>NULL</th>
<th>Primary Key</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>INT(11)</td>
<td>NO</td>
<td>YES</td>
<td></td>
<td>Automatic increase</td>
</tr>
<tr>
<td>name</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Station name</td>
</tr>
<tr>
<td>code</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Station code</td>
</tr>
<tr>
<td>longitude</td>
<td>DOUBLE</td>
<td>NO</td>
<td></td>
<td></td>
<td>Longitude</td>
</tr>
<tr>
<td>latitude</td>
<td>DOUBLE</td>
<td>NO</td>
<td></td>
<td></td>
<td>Latitude</td>
</tr>
<tr>
<td>elevation</td>
<td>DOUBLE</td>
<td>NO</td>
<td></td>
<td></td>
<td>Elevation</td>
</tr>
<tr>
<td>stalevel</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Station level</td>
</tr>
<tr>
<td>statype</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Station type</td>
</tr>
<tr>
<td>seigeomodel</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Seismometer model</td>
</tr>
<tr>
<td>seigeoserial</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Seismometer serial</td>
</tr>
<tr>
<td>datacollmodel</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Data collector model</td>
</tr>
<tr>
<td>datacollserial</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Data collector serial</td>
</tr>
<tr>
<td>comunitmodel</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Communication unit model</td>
</tr>
<tr>
<td>comunitserial</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Communication unit serial</td>
</tr>
<tr>
<td>ipadd</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>IP address</td>
</tr>
<tr>
<td>port</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Port</td>
</tr>
<tr>
<td>location</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Detailed location</td>
</tr>
<tr>
<td>starttime</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td>NULL</td>
<td></td>
<td>Running start time</td>
</tr>
</tbody>
</table>

Table 2. Structure of logsta.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Datatype</th>
<th>NULL</th>
<th>Primary Key</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>INT(11)</td>
<td>NO</td>
<td>YES</td>
<td></td>
<td>Automatic increase</td>
</tr>
<tr>
<td>staname</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Station name</td>
</tr>
<tr>
<td>stacode</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Station code</td>
</tr>
<tr>
<td>operator</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Operator</td>
</tr>
<tr>
<td>start_time</td>
<td>DATETIME</td>
<td>NO</td>
<td></td>
<td></td>
<td>Fault start time</td>
</tr>
<tr>
<td>end_time</td>
<td>DATETIME</td>
<td>NO</td>
<td></td>
<td></td>
<td>Fault end time</td>
</tr>
<tr>
<td>occur_type</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Fault type</td>
</tr>
<tr>
<td>treat_type</td>
<td>VARCHAR(45)</td>
<td>NO</td>
<td></td>
<td></td>
<td>Treatment type</td>
</tr>
<tr>
<td>lon_con</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td></td>
<td></td>
<td>Log content</td>
</tr>
<tr>
<td>remark</td>
<td>VARCHAR(45)</td>
<td>YES</td>
<td></td>
<td></td>
<td>Remarks</td>
</tr>
<tr>
<td>creat_time</td>
<td>DATETIME</td>
<td>NO</td>
<td></td>
<td></td>
<td>Create time</td>
</tr>
</tbody>
</table>

Detailed Design

For the function modules of the system and 9 data tables of the database to be implemented, this paper define entity classes such as User, Sta, Seismo, Dataco, Dcm, Logsta, Logequ, Linkstatus, Recordlogsta and Recordlogequ in com.entity, establish the method classes of Dao and DaoImpl corresponding to each entity in com.dao to implement various algorithms and handle the database, realize the method call and parameter passing of each entity corresponding to the Dao by com.servlet, establish the connection of database and the operation class named DBconn in com.util, create a filter class EncodingFilter in com.filter to implement the filter interface and become a servlet filter. The detailed design of some classes and interfaces is as follows:

Database Connection and Operation Class

Methods of connection, querying, modifying and closing data are established in DBconn class. Part of the connection code is as follows:

```java
Class.forName("com.mysql.jdbc.Driver");
conn = DriverManager.getConnection(url,username,password);
```

Filter Class

The system solves the Chinese gibberish through the configuration of EncodingFilter class. Part of the source code is as follows:
Display of Equipment Replacement Roadmap

The route of device replacement is obtained by calling servlets to read the device replacement records and corresponding locations. The roadmap display is realized by using the Amap API. Part of the source code is as follows [3]:

```javascript
var map = new AMap.Map('map');
var polyline = new AMap.Polyline({
  path: path,
  showDir: true,
  strokeColor: '#3366bb',
  strokeWeight: 10
});
map.add([polyline])
map.setFitView()
```

Monitoring Station Communication Link

The system executes the ping command through the Java call console to determine whether the communication link of the station is normal. Part of the source code is as follows [4]:

```java
String pingCommand = "ping " + ipAddress + " -n " + pingTimes + " -w " + timeOut;
System.out.println(pingCommand);
Process p = r.exec(pingCommand);
if (p == null) {
  return false;
}
```

Station Maintenance Record Statistics

The user selects the time period and the station to query, retrieves the corresponding records from the database and displays them to the browser. At the same time, the records can be exported. Part of the source code is as follows [5]:

```java
List<Logsta> record = lsd.search2(t1,t2,staname);
XSSFWorkbook workBook = null;
workBook = new XSSFWorkbook();
XSSFSheet sheet = workBook.createSheet();
workBook.setSheetName(0,"info");
for(int i=0;i<dataList.length;i++){
  XSSFRow row = sheet.createRow(i);
  String[] oneRowData = dataList[i];
  for(int j=0;j<oneRowData.length;j++)
  {
    row.createCell(j).setCellValue(oneRowData[j]);
  }
}
File file = new File("E:\export\"+fileName+".xls");
FileOutputStream outStream = new FileOutputStream(file);
workBook.write(outStream);
```
System Interface Display

Station Map
The station map shows the geographic location map and remote sensing image map respectively through Amap API. As shown in Fig. 4.

![Remote sensing image of stations.](image)

Figure 4. Remote sensing image of stations.

Equipment Replacement Roadmap
The system displays the roadmap of one device which the user selected to the browser page. As shown in Fig. 5.

![Equipment replacement roadmap.](image)

Figure 5. Equipment replacement roadmap.

Station Maintenance Record Statistics
The user selects the start time, the end time and the station to looks up the maintenance record. And the user can choose to export the record. As shown in Fig. 6.
Conclusion

The system has completed some functions, such as station operation monitoring, station and equipment maintenance log processing, station and equipment distribution map, etc., which can simplify the daily work process of maintenance staff and improve efficiency. In the next work, the remaining functional modules of the system should be developed to achieve reading the station data stream from JOPENS database, calculate the station noise and monitor station data quality.

Acknowledgement

This research was financially supported by The Spark Program of Earthquake Technology of CEA (XH18010Y).

References


