

The Research on Power SCADA Based on J2EE Framework and IEC 61970

Zhao Qiang

North China Electric Power University, Beijing, China
zhaoqiang01@hotmail.com

Chen Danyan

North China Electric Power University, Beijing, China
daisytest@126.com

Abstract—In this paper, the problems exposed by the third generation of SCADA (Supervisory Control And Data Acquisition) based on the POSIX (Portable Operating System Interface of Unix) standards for UNIX systems are analyzed. Then it makes a brief description of the advantages of the new generation power SCADA system based on the J2EE (Java 2 Platform, Enterprise Edition) framework by using open architecture and object-oriented technologies. The integration bus characterized by standardization and component-based is one the key technologies in implementing this new SCADA system which is based on IEC (International Electrical Commission) 61970. By replacing tightly-coupled CORBA (Common Object Request Broker Architecture) to realize CIS (Component Interface Specification) interfaces, Web services components provide more standard and open interfaces for the information sharing and interoperability among application systems of power enterprises. CIM (Common Information Model) is used to describe power system resources and establish models for the database and SVG (Scalable Vector Graphics) elements. As the graphic expression method of the system, SVG with a combination of CIM achieves integrative graphic model library. The use of advanced design patterns also contributes the scalability to some extent and makes the system much more powerful in dealing with constant changes.

Index Terms—SCADA, J2EE framework, IEC 61970, Web services, SVG, EJB

I. INTRODUCTION

SCADA (supervisory control and data acquisition) system is the real-time data source of the power system automation, and it supplies a large number of real-time data to the EMS system and other power information systems [1]. As an indispensable power production technology, computer technology based, SCADA plays a great role in the safe and stable operation of the distribution network.

The SCADA system has gone through three generations since first being put into use in power network. Now it is at the critical period of the fourth generation [2]. By adopting the standards such as POSIX (portable operating system interface of Unix), Motif and C/S (client/server) mode, the third generation SCADA has realized open to some extent at the system software layer. However, there

are still some problems that appear during the practical application. The problems involve the difficulty of interoperation and sharing resources resulting from the lack of a unified data model grid and Graphics standards, the poor scalability and openness and so on. By using component technologies, Internet/Intranet technologies, and having the IEC (International Electrical Commission) 61970 as the constructing standard, the latest fourth generation of SCADA system continues to expand its integration with other systems [3]. Building a Web SCADA with above technologies will help to reduce reduplicate investment and achieve seamless integration among various systems.

II. THE DESIGN OF POWER SCADA BASED ON J2EE PLATFORM

A. Functional Requirements for the Platform from SCADA System

Informationization of the power enterprise begins very early. Many software systems were developed with different technologies and development platforms at different times. And the hardware and software conditions are awfully complex. So in this case, when developing SCADA system, we must consider the following points. Firstly, the system needs to share information with other subsystems of power system (such as geographic information systems, etc.) by a flexible and standard way. Secondly, as power automation function specifications at all levels of power network have made a clear requirement for real-time of the SCADA system, so the SCADA system must meet the real-time requirements [4]. Thirdly, it should be easy to be extended, upgraded and maintained.

J2EE (Java 2 Platform, Enterprise Edition), which is built on the Java 2 platform, is the platform enterprise application solution. It is an open, standards-based platform [5]. For enterprise applications, J2EE provides a good support in scalability, flexibility, easy maintenance, real-time, etc. Its superiority is mainly reflected in the following aspects:

- Supporting heterogeneous environment: The applications based on J2EE are platform-

independent and are not dependent on any particular operating system.

- **Expansibility:** The design based on EJB components technology can simplify the maintenance of application. Since the components can be updated and replaced independently, new functions can be added transparently by updating specific components of the application.
- **Supporting real-time:** RTSJ (Real-time specification for Java) is proposed to solve some of the limitations of Java program language. These limitations prevent its wide usage in the real-time environment. RTSJ make some improvements to Java specification from several aspects, including scheduling, memory management, threading, clock and so on.
- **Supporting XML (eXtensible Markup Language) information sharing:** XML is considered as the cornerstone of the next generation network applications. J2EE platform supports XML comprehensively, and the platform can take full advantage of platform independence of XML and J2EE platform to share information among different systems efficiently.

B. The Architecture of SCADA System Based on J2EE Platform

Developing SCADA system with advanced J2EE technologies can resolve a series of problems existing in the traditional SCADA. J2EE platform is the ideal platform to develop the fourth generation of SCADA system. According to the J2EE architecture, the SCADA system is designed to be four-tier, distributed truss, as illustrated in Fig. 1.

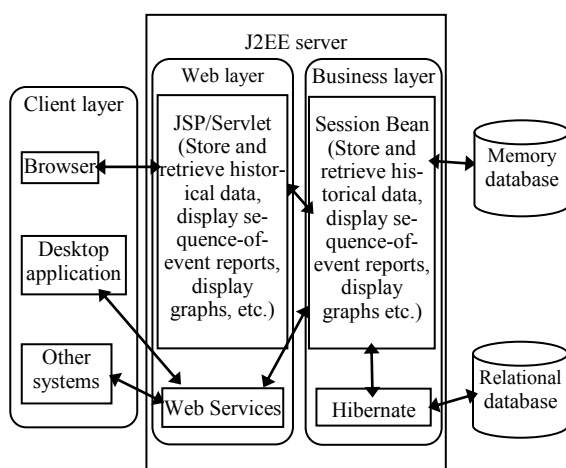


Figure 1. SCADA system based on J2EE platform.

In view of strong interaction, more secure access mode and lower network traffic of C/S mode, and distribution, rich graphical display of B/S (Browser/Server) mode, this system is based on B/S and C/S mixed mode: on the one hand, the client browser supplies the users with rich graphic display and user interaction functions, including telesignalling and telemetry statements, sequence-of-

event report, primary electric wiring diagram, etc., on the other hand, independent Java applications, which run directly on the client operating system(or Java virtual machine), can make full use of client resources and computing ability, and the result is that it will offer richer functions for the master side, e.g., analyzing the real-time or historical data, monitoring the devices, analyzing the real-time or historical data, etc.

At present, many monitoring systems take advantage of ActiveX controls or Java Applet, or raster graphics generated by server for the client downloading. In despite of good display effect and rich interaction, ActiveX controls or Java Applet make the client security level lower and the direct result is that it may lead to security risks such as the illegal invasion. Moreover, ActiveX controls have poor portability and are difficult to be upgraded and maintained. Also publishing raster graphics is not realistic for the SCADA system which requires high real-time. In light of the merits, such as XML-based Vector Graphics based and strong interaction, of SVG (Scalable Vector Graphics) technology, monitor screens like a wiring diagram are displayed by SVG. Comparing with above measures, Web SVG graphics can gain more openness by using the third-party plug-ins to display SVG graphics. What's more, the XML features of SVG enable the client to make further analysis and treatment after receiving SVG graphics.

JSP (Java Server Pages) and Java Servlet in the Web layer respond to the requests from the client. Having received those requests, Servlet calls the EJB (Enterprise JavaBeans) components in the business layer, and then returns the processing results to the client. In addition, for the purpose of improve integration capability with other systems instead of only providing services for itself, after analyzing the requirements of other systems that share information with this system, according to IEC 61970, some business methods are further wrapped and published as Web services to be called.

The detailed business logic of the function modules of SCADA systems is wrapped by EJB component. EJB, providing a unified interface to the external systems, is the core of the whole system design. Its component characteristic enables the system to meet the diverse requirements of power enterprises easily. Former SCADA systems use entity Bean to encapsulate business entities while the session Bean will call the corresponding entity Bean to read and write data and keep synchronization of objects and data. In order to reduce complexity and make the system light as much as possible, in this system, we make an improvement to the traditional calling mechanism.

Generally, data are stored in relational database, such as SQL Server or Oracle. The mount of data dealt by SCADA systems is huge and the relationships among the various data are always very complex, therefore, it is necessary to introduce a convenient and efficient ORM (Object-Relational Mapping) solution. This system adopts Hibernate, a very excellent persistence layer, to establish the dynamic mapping with the relational database. It separates persistent classes that created by Java

and the underlying database [6]. The session Bean will call the Hibernate to encapsulate business logic. It provides lightweight object encapsulation for JDBC (Java Database Connectivity). It not only allows the mapping of mapped tables with the domain objects, but also provides data query and recovery mechanisms. The above mechanisms are transparent to users. Hibernate can replace entity Bean in the J2EE applications to complete the important task of data persistence [7].

The real-time database is not only the foundation and core of the SCADA system supporting platform, but the basis of building other subsystems as well. The real-time, openness and safety performance of the system are determined to a large extent by the real-time database [6]. The real-time database is created and managed by way of file and memory mapping, which can ensure real-time data's unlost by refreshing the disk file synchronously through the memory of the real-time database. The memory's application, release, access, concurrency and storage are most achieved by the operating system and the memory space of the file mapping is only restricted by physical space.

The relational database and the real-time database are both adopted in this system. The former stores historical data while the latter stores telemetry, telesignalling and electrical measurement coming from plant side RTU (Remote Terminal Unit). When the electrical equipments are described in accordance with CIM (Common Information Model), the system can read objects from and write objects into the CIM library by the object-based API (Application Programming Interface). The functions, such as power network analysis and so on, access the database through relation-based API, whose main task is to read data from and write data into two-dimensional tables.

III. THE IMPLEMENTATION OF POWER SCADA

A. Information Expression and Multi-system Integration Based on IEC 61970

In order to reduce the costs of adding a new power system function and to protect previous investments, SCADA system should be more standard and open as far as possible. For such a demand, the biggest difficulty is the lack of a common information model [8]. Fortunately, IEC released a standard design specification for the application developers using real-time information, that is, IEC 61970. IEC 61970 is the basic standard for the integration and interoperation of dispatching automation systems. Following the IEC 61970 means that we can solve interoperability issues of grid data model.

This power SCADA system follows IEC 61970 to accomplish the information exchange and interoperation among power systems. The integration platform is illustrated in Fig. 2. It mainly encompasses CIM database, import and export tool for CIM/XML, CIS (Component Interface Specification) interface implementation and Web services integration bus. Other interactive systems could access SCADA system through Web services that wrap the CIS interfaces.

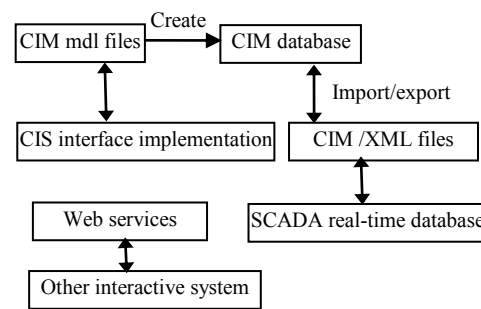


Figure 2. Information expression and multi-system integration.

CIM model is an abstract model used to describe all the main objects of power enterprise. In the field of electricity, CIM is used in the development and integration of electrical engineering, planning, management, operation and business applications. The standard method it uses to describe power system resources is to provide object classes, attributes and their relationships. Setting up database model according to the CIM model ensures the SCADA system open. Master station system store a variety of data collected in the CIM database. We should create object models of system before building a database according to the CIM model. The choice of model classes depends on the scope of application and properties. In this system, the main packages we choose to implement include the core package, topology package, measuring package and other packages related with CIS service interfaces.

The structure of CIM database is built automatically in light of mdl files by conversion tool. As the relational data model of relational databases is different from the object model of CIM, so it is necessary to make a conversion. When converting between CIM model and relational database, that is to map with CIM model and relational model, there are at least one fact we should consider carefully, that is the contradiction between the object-oriented models and the relational database system. As the relationship of CIM model is complex, so mapping into a relational database table completely will inevitably generate a lot of relation tables, and it will increase the number of database tables. If only mapping the CIM objects and regardless the mapping of the relationship, the database will not be able to confirm to the CIM model, and the result is to lost the meaning of the standard.

According to relationship between the relational data model of RDBMS (Relational Database Management System) and CIM object model, system follows following converting rules:

- Map a class to a table.
- Map the attributes of the class to the fields of the table.
- All properties of the top-level object will be owned by its all children. CIM database follows such a feature. In order to achieve inheritance mapping, the system map both parent class and subclasses into tables. And then add an attribute that is the primary key, the name of the primary key is 'PK_<base class name>', The attributes set

of the tables which have the same primary key in the inheritance trees are all attributes of a specific equipment. Another attribute added is the type, which is used to describe what the specific object that in the base table actually is.

- For zero/one-to-one/many relationship, it needs to add a foreign key to one/many table. If this zero/one relationship is an inheritance tree, then this foreign key should be named 'FK_<base class name>'.
- For many-to-many relationship, it needs to create an associated table, whose attributes are the key attributes of the two tables, and the primary key of the associated table is primary key combination of the two tables.

The data source of CIM database is SCADA real-time database. And the CIM/XML import/export tool will import XML files into CIM database if the whole grid model changes. Because the time spent on importing the entire power system model is very long, it is undesirable to re-import the entire system because of a minor change to the system, especially such a change is very frequent and normal. Therefore, the system uses the difference model to describe the differences between two RDF (Resource Description Framework) models to improve efficiency.

IEC 61970 recommends SVG as the standard graphics format for power system graphics interaction. Graphics control is realized by capturing messages provided by SVG. The system creates graphics models on the basis of CIM models, and reads and stores the graphics data with SVG and Ajax technology. Such an implementation approach achieves the integration of graphics, model and database, and solves the problem of refreshing the real-time electrical model information dynamically. CIM-based SVG graphics can refresh partially when the real-time data arrive and query equipment parameters.

CIM-based SVG graphics models can express not only the geometric properties of graphic elements, but also the resources properties of the graphics. Thus, elements in this system are generally described in the form of the following Fig. 3.

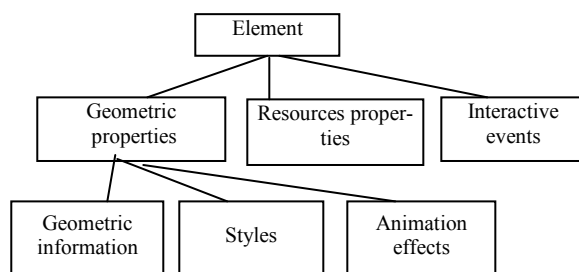


Figure3. SVG description of power elements.

In the SVG files, it can use custom element labels to describe application properties of graphics element. The properties name and value can be specified arbitrarily. For example, we can use element <Properties> to describe the application attributes of graphics elements. And attribute name and attribute value of <Properties>

describes application attribute name and attribute value of a graphic element. For the application properties of more complex equipment, we can associate the application properties, which are stored in the database, of the device with the graphic element by the corresponding ID of graphic element. Compared with the way of storing the properties directly in SVG files, this method can separate the graphic files and model files of application properties, and the simplified SVG file has a higher degree of reusability. The latter method getting data information stored in external database is much more convenient, especially for real-time data.

The SVG graphics have the ID number of each equipment, and the number is equal to the equipment ID in the power grid model. The association between SVG graphics and CIM models is kept by this method. The ID number of the equipment in the SVG file can be got by parsing RDF/ID of the equipment in the CIM file, and then gain the graphic description of the equipment. The integration of graphics and database based on CIM models provide a set of advanced graphics display and operation tools. It updates corresponding equipment properties into appropriate tables while generating electrical components on the monitor screen, and it also can check to ensure that the data stored into the database are right and set up the relationships automatically between equipment on the graphics and the data in the database.

CIS provides the standard interfaces to access the data for application software. The CIS interface implementation implements GDA (Generic Data Access) interface and HSDA (High Speed Data Access) interface. The former provides insertion, update, deletion, and filtering query functions while the latter is used to transfer massive data efficiently. The data transferred by HSDA includes real-time data, computing data, equipment parameters and commands passed to the SCADA system, and these data often carry a time stamp and quality code. Third-party users can develop new applications or transfer data flexibly by calling the standard GDA and HSDA interface on the Web service integration bus. The typical applications are that the client gets grid model data by calling GDA methods or get measured values by calling HSDA methods.

The following steps show how to get the whole grid model:

- 1) Get "ResourceID" for each class by calling method "get_uris". ResourceID is used not only in CMI/XML data model files as the foundation of resources reference and correlation between resources, but in the data access interfaces, such as GDA, as the foundation of remark and resources usage between different software .
- 2) Get all record values in the tables by calling method "get_filtered_extnt_values". It is less efficient to transport all data in one method call when the system has to gain large amounts of data. So in this system, we adopt the integrator in some methods to allow the client to get an arbitrary amount of resources in one time. If the integrator

has no resource when the client finishes, it will be recycled automatically.

Fig. 4 is a sequence paradigm illustrates the above calling process.

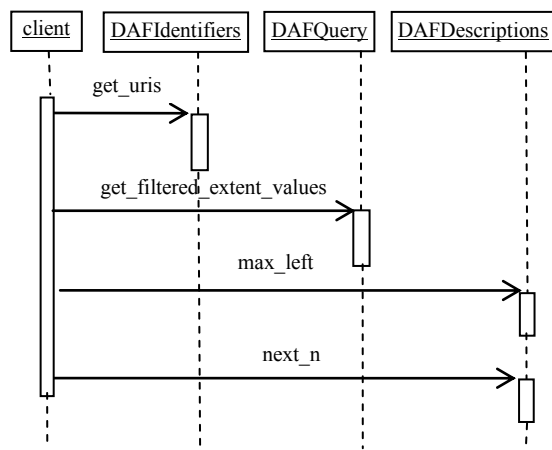


Figure4. A sequence paradigm.of getting the whole gird model

The following steps show how to get the whole gird model:

- 1) Create a session with the server by calling method "create_data_access_session".
- 2) Get an access point by calling method "node_home", and then find the root node through the access point.
- 3) Call method "find_by_parent" circularly to get all the resources at all levels. After getting all the resources, call the method "find_by_parent" of all leaf nodes to get all the measured values.
- 4) Create a group after getting the whole tree and put subscribed measured values. If all these steps have been done, what we need to do is only to wait the change of data.

The traditional integration solutions for heterogeneous system are private and complex, that is to say, different components can't exchange data directly [9, 10]. In this system, we adopt new component model Web services, which are introduced in the chapter "Component Model" of IEC 61970's appendix A. When compared with CORBA, Web services provide a different, Internet standard based, any-to-any system integration model, and it is more suitable for the integration of old and new system. Web services based, loosely coupled enterprise application integration solutions are of great significance in promoting power enterprises to achieve wider application integration. The steps show how to realize Web service CIS are as follows:

The CIS provided by IEC 61970 is identified by IDL (Interface Description Language), which is only for CORBA (Common Object Request Broker Architecture) [11]. If want to develop Web services, we have to translate IDL CIS into WSDL (Web Service Definition Language) CIS according to translation rules. OMG (Object Management Group) has defined CORBA to WSDL/SOAP (Simple Object Access Protocol) Inter-

working specification and WSDL/SOAP to CORBA Interworking specification for the mutual translation of IDL and WSDL. The following example illustrates how to translate IDL defined ResourceIDService interface to WSDL defined. It concerns three aspects: message definition, port type definition and operation definition.

The ResourceIDService interface defined by IDL is as follows:

```

Interface ResourceIDService{
    ResourceIDSequence get_resource_ids(in URISequen
ce uris) raises(LookupError);
}

```

The corresponding ResourceIDService interface defined by WSDL is as follows:

Message definition:

```

<message name=" ResourceIDService.get_resource_id
s">

```

```

  <part name="uris" type="DAFIdentifiers.URISequen
ce"/>

```

```

</message>

```

```

<message name=" ResourceIDService.get_resource_id
sResponse">

```

```

  <part name="_return" type="DAFIdentifiers.Resource
eIDSequence"/>

```

```

</message>

```

Port type definition and operation definition:

```

<portType name=" ResourceIDService">

```

```

  <operation name=" get_resource_ids">

```

```

    <input message="tns: ResourceIDService.get_resou
rce_ids"/>

```

```

    <output message="tns: ResourceIDService.get_reso
urce_idsResponse"/>

```

```

    <fault name="CORBA.SystemException" message="
"corba:CORBA.SystemExceptionMessage"/>

```

```

    <fault name=" DAFIdentifiers.LookupError" messa
ge=" tns:_exception. DAFIdentifiers.LookupError"/>

```

```

  </operation>

```

```

</portType>

```

Build CIS Web services server by the JAX-RPC (Java API for XML-Based RPC) server-side model. Here some tools, such as Axis, can be used to mapping WSDL CIS into Java classes.

The method using Web services to establish integration bus improves the loosely coupled nature and reusability of the SCADA system and achieves information sharing in CIM definition scope. The third party can visit the SCADA's run information and grid model in a standard manner and can expanse their own data types easily.

Adoption of IEC 61970 makes the SCADA system be truly open and standardized, and it brings common international standards for the integration platform of the enterprise automation system. Each EMS application can have their own information description inside. As long as adhering to common information model at the application (or component) interface semantic level, different applications produced by different manufacturers or of different systems can access the public data by the same way (e.g. using XML). It is a real and effective exchange of information.

B. Visiting EJB Component with Design Patterns

For the purpose of reducing the creation and destruction of the Java object, singleton pattern restricts the creation of a class instance and provides a global accessing point to access the instance. A class which is designed by this pattern has only one instance. The instances created by factory pattern usually implement the same interface. All the instances created by the factory method pattern usually implement a same interface, where all methods owned by the instances are identified, and these methods always have different implementation in different implementing classes. The caller doesn't need to care about the specific implementation of methods, and thus it reduces the cost of heterogeneous systems.

The present system combines factory method pattern and singleton pattern when creating EJB delegate factory class. The factory class returns Remote interface according to the JNDI (Java Naming and Directory Interface) given. As the system only needs one instance of factory class and doesn't need to create repeatedly, so the factory class is designed in terms of singleton pattern. Furthermore, HashMap is used to cache EJB remote interfaces. All the means used is to reduce the cost of the system and improve the response speed. The steps for the EJB factory to get EJB remote interface by JNDI are as follows:

```
public class RemoteFactory {
    private static RemoteFactory factory;
    //cache EJBRemote with HashMap
    private Map remoteInterfaces;
    private Context context;
    .....
    public EJBObject lookup(String jndi, Class homeClass)
    throws NamingException, CreateException,
    RemoteException {
        EJBObject remote = (EJBObject)remoteInterfaces.get(h
        omeClass);
        if (remote == null) {
            Object obj = context.lookup(jndi);
            CommHome home = (CommHome)
            PortableRemoteObject.narrow(obj, homeClass);
            remote=home.create();
            remoteInterfaces.put(homeClass, home.create());
        }
        return remote;
    }
    .....
}
```

- 1) Create an EJB factory class: Create an EJB factory class named RemoteFactory, and declare an instance of HashMap for caching EJB remote interface and a private and static instance of RemoteFactory which will be returned by a synchronous method.
- 2) Declare the constructor of RemoteFactory to be private.
- 3) Create a lookup method: Create a lookup method whose return type is EJBObject. In this method, lookup the object of EJBObject in the above HashMap cache pool firstly. If it does exist, then return it. Otherwise, lookup all the home interfac-

es' parent interface by JNDI in the context, and then a new object of EJBObject which will be put into the existing HashMap later will be created by the interface.

The CommHome interface which inherits EJBHome interface has the create method to return the object EJBObject. All the Home interfaces of EJB components inherit the CommHome interface. When the clients need to call EJB components after creating a factory class, the clients only need to call the lookup method of the EJB factory, and pass the JNDI and the Home interface of EJB components. This common method avoids a large number of repeat lookups and calls for the method "create" from the client.

C. The Integration of EJB and Hibernate

As the object-relational mapping solution of Java, Hibernate mainly maintains the mapping relationship of objects created by object-oriented design work and tables of RDBMS and the persistence operations. It provides a convenient framework for the mapping from object-oriented domain model to the traditional relational database. Hibernate is integrated with EJB in this system, and it takes charge of access the database. After Hibernate being deployed to the Weblogic application server, the session Bean can call Hibernate to deal with businesses. The steps of the integration are as follows:

- 1) Deploy Hibernate: Deploy Hibernate to the Weblogic application server. After the deployment, then Weblogic will generate a JNDI for Hibernate.
- 2) Create a StartUp class: In order to ensure the pre-creating of Hibernate's SessionFactory, the system needs to create a StartUp class which implements the Weblogic T3StartupDef interface. This class will begin to run in the background when the application server starts. The core method is as follows:

```
public String startup(String name, Hashtable args)
throws Exception {
    Configuration con = new Configuration()
    .configure("/hibernate.cfg.xml");
    SessionFactory sf = con.buildSessionFactory();
}
3) Operate the database through Hibernate: Session Bean can get the Hibernate SessionFactory through the above-mentioned Hibernate JNDI. After getting the instance, session Bean can operate the database through Hibernate.
Context ctx = new InitialContext();
SessionFactory sf = (SessionFactory)
ctx.lookup("hibernate/session_factory");
```

D. The Real-time Java Technology

As a data acquisition and supervisory control system, an important feature of SCADA system is real-time. Because of Java's inherent uncertainty such as thread management, garbage collection, class loading and so on [12], the standard Java applications that run on general JVM (Java Virtual Machine) could only meet soft real-time demand in a few hundred milliseconds. To overcome these limitations, Java real-time specification was raised,

and it enables Java to be widely applied in real-time systems.

SCADA systems use multi-threading technology when acquire real-time data. When the fault accidents take place, SCADA system should give reaction timely, however, the conventional Java thread uses the time slice rotation scheduling mechanism. The mechanism doesn't provide any priority protections to threads. Therefore, if the system can't react in the first time when accidents happen, the consequences will be very serious or even disastrous.

The two types of RTSJ real-time thread are the general `RealtimeThread` class and `NoHeapRealtimeThread` class and other mechanisms provided by RTSJ can meet the needs of real-time data collection of SCADA systems. General `RealtimeThread` is used to the task which can tolerant the delay brought about by the GC (Garbage Collection) while as the high-priority `NoHeapRealtimeThread`'s being rarely interrupted or seized by the garbage collection, so it is often used in the demanding task which can't tolerate delay. Fig. 5 shows the priority of the different threads.

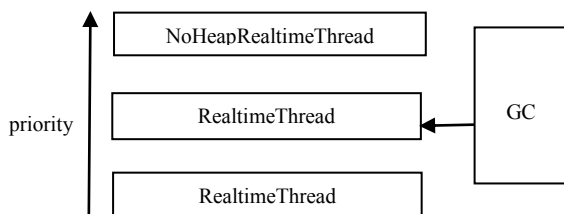


Figure5. Priority of thread and GC mechanism.

By using fixed priority scheduling strictly, real-time thread scheduling of RTSJ is perceptible. The class `HighResolutionTime` and `Clock` provide high resolution clocks for real-time systems and can be accurate to nano-second [13]. The delay is between about 10 microseconds and 20 microseconds for the dual-processor workstation using Java RTS.

To provide real-time functions, the SCADA system requires the underlying operating system to support real-time, for example, Solaris 10 has a real-time scheduler. Also the program needs to import the `javax.realtime` package before we use any functions provided by RTSJ implementation.

Follow following steps can create and start a real-time thread.

- 1) Initialize parameters: Initialize related parameters of the constructor of real-time thread. A typical constructor is like this: `RealtimeThread(SchedulingParameters scheduling, ReleaseParameters release, MemoryParameters memory, MemoryArea area, ProcessingGroupParameters group, java.lang.Runnable logic)`. The functions for the above parameters are listed below: `SchedulingParameters` is used to designate priority, `ReleaseParameters` to designate real-time tasks to be periodic, aperiodic or sporadic, `MemoryParameters` to restrict the usage of memory allocator,

`MemoryArea` to designate the type of memory pool(heap, immortal memory or scope memory) for the real-time thread, `ProcessingGroupParameters` to control the activities of the aperiodic thread, and logic to cite a instance of a class that implements the `Runnable` interface.

- 2) Instantiating the real-time thread: Instantiating a real-time thread object by using the constructor mentioned above.
- 3) Start the thread: Call the "start" method of the realtime thread.

Generally in the system, the common `RealtimeThread` will be used in the mission that can tolerate delays caused by the garbage collector. High-priority `NoHeapRealtimeThread` is rarely interrupted or occupied, so they will be used in the most demanding tasks which can't tolerate delays.

IV. CONCLUSION

After studying the architecture of the third-generation SCADA system which is based on UNIX platform, aiming at being in line with the IEC 61970 standard, a series improvement were made on J2EE four-tier architecture with many new software technologies that sprang up in recent years to the traditional power. The system completed the mapping between the database of conventional SCADA system and the CIM model, and CIM based SVG graphics modeling mechanism realized the integration of graphics, digit and model. The new generation of SCADA system is constructed by component technology. The general software bus realized by the Web services component model, which implement CIS interface, can provide functions like model data to all level applications. Compared with the data bus realized by CORBA, the data bus realized by Web services is much more standardized and open. Facts have proved that the new system has better flexibility and scalability, and it is more convenient to integrate with other systems. It will make a significant and effective contribution in the power automation system. The realization of this system is a useful attempt for the development of the J2EE architecture based power system software.

REFERENCES

- [1] Mao Wenjuan. "Research of Web service-based real-time data access platform for SCADA system," *Southwest Jiaotong University, ChengDu*, pp. 1-3, April 2007.
- [2] Zhao Duolin. "Web SCADA system architecture based on B/S structure," *Nanjing University of Science and Technology, Nanjing*, pp. 1-2, April 2006.
- [3] Ma Zhengwu. "Summarize of SCADA&RTU in the course of realizing the enterprise automation," *Programmable Controller & Factory Automation, Shenzhen*, pp. 18-20, 2005.
- [4] Xiong Bing, Zhang Aimin, zhang Hang. "Study of improving the real-time performance of SCADA in electric power system," *High Voltage Apparatus, Xi'an*, vol. 38(03), pp. 16-18, 2002.
- [5] Xiao Jianfeng. "J2EE platform in research and application of system framework design," *Wuhan University of Technology, Wuhan*, 2006.

- [6] Wu Aiping, Chen Jianyun, Qu Zhijian. "Design and implementation of SCADA system real-time database based on CIM," *Proceedings of the CSU-EPSA, Tianjin*, vol. 20(02), pp. 105-109, April 2008.
- [7] Gan Shuman, *Open Source Framework Technologies and Practices Based on Eclipse*. Beijing: Publishing House of Electronics Industry, 2007.
- [8] Gao Xiang, Zhang Peichao, Zhang Jianmin, *Application Technologies of Fault Information System*. Beijing: China Electric Power Press, 2007.
- [9] Wang Fei. "Analysis and implementation of integration framework for Web Services and CORBA," *Southeast University, Nanjing*, pp. 4-11, 2006.
- [10] Yu Junxiang. "The research and application of EAI based on Web service," *Zhejiang University, Hangzhou*, pp. 6-21, 2006.
- [11] Zhang Shenming, Liu Guoding. "Introduction of standard IEC 61970," *Automation of Electric Power Systems, Nanjing*, vol. 26(14), pp. 1-6, 2002.
- [12] Zhong Baohua. "The research and application of real-time Java," *Guangdong University of Technology, Guangdong*, 2004.
- [13] Peter C.Dibble, *Real-time Java platform programming*. Beijing: China Machine Press, 2003.

Zhao Qiang (Shanxi, China, 1959), Computer Engineering of Xidian University in 1983, Master Degree, Xi'an, China. Major field of study is the research on distributed computing technology and its application in power system.

He now works in North China Electric Power University, Beijing, China as an associate professor. He used to engage in the work of dispatching computer applications in Electric Power. The systems he ever developed include SCADA system of power grid, short-term load forecasting, flow calculation, reliability analysis of power system operation and so on. He used to publish more than 20 articles and several books. He now interests in power system automation.

Chen Danyan (Zhejiang, China, 1985) Computer Application Technology of North China Electric Power University, Master Degree, Beijing, China Major field of study is the application of computer technology in power system.

She now studies in dept. of Computer Science and Technology, North China Electric Power University, Beijing, China as a graduate student. She now engages in the work of developing data exchange platform of state grid. She has published two articles.