

## A Development Methodology for a Remote Inspection System with JAVA and Socket

Yoo Rark Choi, Jae Cheol Lee, Jae Hee Kim

Korea Atomic Energy Research Institute, ISA Div., P.O.Box 105, Yuseong Daejeon, yrchoi@kaeri.re.kr

### 1. Introduction

We have developed RISYS (Reactor Inspection System) which inspects reactor vessel welds by an underwater mobile robot [1]. The system consists of a main control computer and an inspection robot which is controlled by the main control computer. Since the environments of the inspection tasks in a nuclear plant, like in other industrial fields, is very poor, serious accidents often happen. Therefore the necessity for remote inspection and control system has increased more and more. We have carried out the research for a remote inspection model for RISYS [2], and have adopted the world wide web, java, and socket technologies for it.

Client interface to access the main control computer that controls the inspection equipments is essential for the development of a remote inspection system. It has been developed with a traditional programming language, for example, Visual C++, Visual Basic and X-Window. However, it is too expensive to vend and maintain the version of a interface program because of the different computer O/S. Nevertheless web and java technologies come to the fore to solve the problems but the java interpreting typed language could incur a performance problem in operating the remote inspection system.

We suggest a methodology for developing a remote inspection system with java, a traditional programming language, and a socket programming that solves the java performance problem in this paper.

### 2. Java and Socket Programming for a Remote Inspection System

The web provides a powerful networking facility on the internet environment. A graphic user interface programmed with java on a web browser could communicate with a remote web server without any extra network programming. We can easily access a remote inspection system with java. But this way causes some problems. Java is useful at the client interface in a remote inspection system. When it joins the C language that composes main control the program of the remote inspection, it works well.

#### 2.1 Develop a client interface with the web and java

The web browser directly communicates to the web server. The request for consuming web server resources must be processed in the web server. Java provides several methods for it such as applet, JSP(Java Server Page) and JNI(Java Native Interface).

Applet is downloaded from the web server to the client web browser and it is executed on the client web browser.

JSP is a server side script language and provides a dynamic interface technique[3]. ASP, PHP, and CGI act just like JSP.

A traditional programming language code could be used in java as a java shared library. This technique is provided in JNI [4].

JSP and JNI programming is a method of a direct communication between the client web browser and the web server. Since JNI programming is a fragmented type, it is difficult to communicate with other JNI programs. When an equipment has a CUP itself and the tools are controlled by the CPU, fragment typed programming is inappropriate for a control program, because a complex control relationship could occur between the control program and the equipment.

Direct communication is not only difficult in developing complex control program of an inspection system but also slow at executing the performance of the control program.

We suggest another method to reduce the problems. It is a direct communication between the client web browser and the control program. The control program must be executed on a server side main control computer and it is a part of the control program.

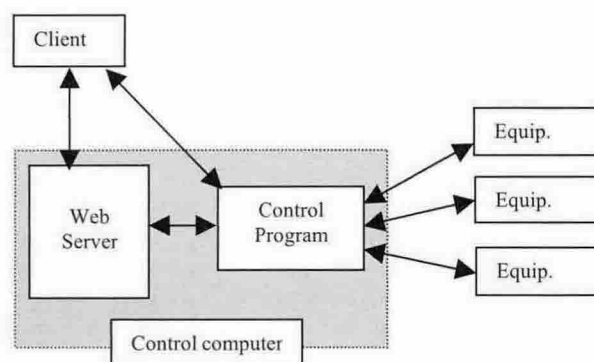


Figure 1. Direct communication between a client and a control program

We must provide other web browser for the display of the message transferred from the web server. In the applet technique, no more web browsers are necessary. Applet provides a flexible programming environment. We propose an applet based client interface and a TCP/IP based communication between the control program of inspection system server and the client interface. On an applet based client interface, JSP could also be a valuable component to develop a remote inspection system.

To reduce the network communication cost and use the java environment effectively, the web server and control program must be on the same computer.

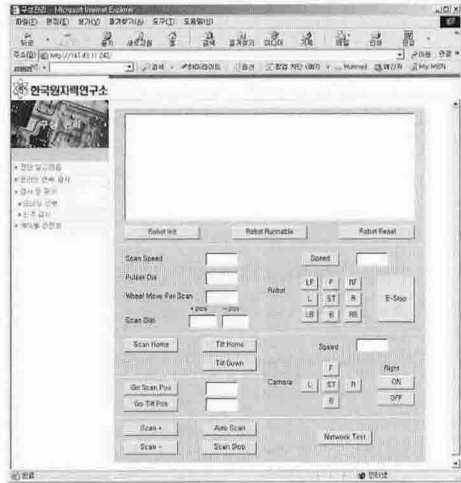


Figure 2. Applet based client interface

We use apache 1.3.19 for the web server, JDK 1.3.0 for the java programming, and Jakarta Tomcat 3.2.1 for the JSP environment to develop the system.

2.2 Structure of the TCP/IP program

The control program controls the inspection equipments and communicates with the client web browser. The control program of the RISYS was developed on Visual C++ 6.0. We refer to a TCP/IP programming method on Visual C++.

The remote inspection system that we have developed consists of some inspection equipments, a control program, and a client interface. The TCP/IP communication module is in the control program and must be connected to the inspection equipments and client web browser.

One network communication program could connect to many clients. There are two programming methods to process a 1:N connection. First method is creating sockets as many as the existing client socket number with different ports. Second one is creating just one server socket and creating a child socket with the same port whenever a client requires a connection. The second method may use thread, but thread does not guarantee a time sequence in Visual C++. A time sequence is a very important factor for an inspection system.

Because the number of client sockets is already defined, the first method is effective. We adopt the first

method and also use the server-child socket structure of the second method.

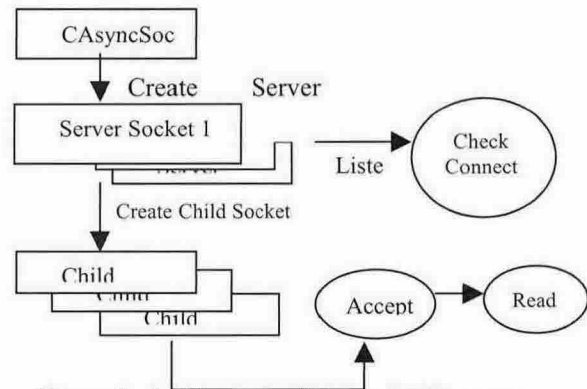


Figure 3. Architecture of the socket Program

A server socket is created with the same number of client connections. Server sockets go into a Listen() mode and create a child socket for each server socket. When a client connection event occurs, the server socket that charges the client connection makes a connection between the child socket and the client. If a child socket is already connected, a new connection is ignored in the server socket. The connected child socket goes into a Receive() mode and reads the data transferred from the client.

3. Conclusion

Web and java techniques are advantageous for a remote interface. But java has some problems in the control part. A traditional programming language could cover the problems. Thus we adopt web, java and a traditional program language in the RISYS, and create a remote inspection system.

Communication between the client web browser and the main control program of a remote inspection system is faster than a communication between the client and the web server in execution performance.

REFERENCES

[1] J. H. Kim, J. C. Lee, Y. R. Choi, Development of an automatic reactor inspection system, KAERI/RR-2239/01, 2002.  
 [2] J. H. Kim, J. C. Lee, Y. R. Choi, Development of a remote inspection system for NSSS components, KAERI/TR-2711/2004, 2004.  
 [3] H. S. Yang, S. Y. Pyo, Perfect JSP, Hanbitbook, 2000.  
 [4] "http://java.sun.com/j2se/1.3/docs/guide/jni/", Sun Microsystems, Inc.