

ภาคผนวก

ภาคผนวก ก

โปรแกรมภาษาจาวาที่ใช้ทำเซิร์ฟเลต และ ไฟล์ HTML

1. EsMainPanel.java

```
//EsMainPanel.java
import java.io.*;
import java.util.*;
import javax.servlet.*;
import jess.*;

// displays a query
public class EsMainPanel {
    String query;          // query caption
    int items;             // nr of query items
    String factComponents[]; // image file names, labels, and assertions
    final String mainUrl = "http://www.phitsa.com/"; //Phitsanulok Radar URL
    EsReteControl reteCon;

    public EsMainPanel(String query, int items, String[] factComponents) {
        this.query = query;
        this.items = items;
        this.factComponents = factComponents;
    }

    public EsMainPanel(String query, int items, String[] factComponents,
        EsReteControl reteCon) {
        this.query = query;
        this.items = items;
        this.factComponents = factComponents;
        sendPanel(reteCon);
    }

    public void sendPanel(EsReteControl reteCon) {
        int rows, cols;
        Hashtable assertions = new Hashtable();
        ServletOutputStream toClient = reteCon.getClient();

        int codeSum = 0; // generate a signature to identify this query
        for (int i=0; i<query.length(); i++)
            codeSum += query.charAt(i);
        String querySignature = Integer.toString(codeSum);
        reteCon.setQuerySignature(querySignature);

        try { // generate HTML for the query page
            toClient.println("<html><head><title>Main Page</title>" +
                "<meta http-equiv=\"Content-Type\" content=\"text/html; " +
                "charset=windows-874\">");

            if (reteCon.getSeqError())
                toClient.println("<FONT FACE=\"Arial,Helvetica\">" +
                    "<B>Starting Over:</B>" +
                    "<BLINK>Please do not use BACK or FORWARD " +
```

```

        "when responding to queries.</BLINK>");
        rows = items / 4 + 1;
        cols = (items <= 2 || items == 4) ? 2 : 3;

        toClient.println("<table width=\"75%\" border=\"0\"
align=\"center\">");
        toClient.println("<tr>");
        toClient.println("<td align=\"center\" height=\"70\" " +
"bgcolor=\"#336699\">");
        toClient.println("<div align=\"left\"><font face=\"Arial, " +
"Times New Roman, Times, serif\"> " +
"<b><font size=\"5\" color=\"#FFFFFF\"> " +
"Main Page</font></b></font></div>");
        toClient.println("</td>");
        toClient.println("</tr>");
        toClient.println("<tr>");
        toClient.println("<td height=\"259\" valign=\"top\" " +
"bgcolor=\"#99ccff\">");
        toClient.println("<p align=\"center\"><font face=\"Arial, " +
"Times New Roman, Times, serif\"><h3 align=\"center\"> " +
query + "</h3></font></p>");
        toClient.println("<FORM name=\"startup\" method=POST " + "action=\""
+ reteCon.getResponse().encodeURL("/servlet/EsServlet")
+ "\">");
        toClient.println("<div align=\"center\">");

        int next = 0;
        for (int r=1; r<=rows; r++) {
            for (int c=1; c <= cols && next < items; c++) {
                toClient.println("<INPUT TYPE=\"submit\" NAME=\"action\"
VALUE=\"" +
                    factComponents[next*2+0] + "\"><br>"); // add a button
                assertions.put(factComponents[next*2+0],
factComponents[next*2+1]);
                // link button label and assertion for later use
                next++;
            }
            toClient.println( "<TR> " +
"<TD BGCOLOR=\"#CCCCCC\"><CENTER> " +
"<INPUT TYPE=\"submit\" NAME=\"action\" " +
"VALUE=\"restart\"></CENTER></TD></TR>");

            // link the Restart button with (restart)
            assertions.put("restart", "(restart)");
            toClient.println("</TABLE>"); // end table
            toClient.println("<INPUT type=hidden name=\"querysig\" value=\"" +
querySignature + "\">"); // embed the query signature in the
form
            toClient.println("</FORM><FONT FACE=\"Arial,Helvetica\">");

            toClient.println("<CENTER><P><A HREF=\"/expert/\"> " +
"Return to the ATCR-33S Expert System startup page</A></CENTER>");
            toClient.println("<P><CENTER><A HREF=\"" + mainUrl +
"\>Return to Phitsanulok Radar home page</A></CENTER>");
            toClient.println("</FONT></BODY></HTML>");
        } catch (IOException e) {}
        reteCon.setAssertions(assertions); // hand assertions to reteCon
    }
}

```

2. EsSelectPanel.java

```
//EsSelectFault.java
import java.io.*;
import java.util.*;
import javax.servlet.*;
import jess.*;

// displays a query
public class EsSelectPanel {
    String query;          // query caption
    int items;             // nr of query items
    String factComponents[]; // labels, and assertions
    final String mainUrl = "http://www.phitsa.com/"; //Phitsanulok Radar URL
    EsReteControl reteCon;

    public EsSelectPanel(String query, int items, String[] factComponents) {
        this.query = query;
        this.items = items;
        this.factComponents = factComponents;
    }

    public EsSelectPanel(String query, int items, String[] factComponents,
        EsReteControl reteCon) {
        this.query = query;
        this.items = items;
        this.factComponents = factComponents;
        sendPanel(reteCon);
    }

    public void sendPanel(EsReteControl reteCon) {
        int rows, cols;
        Hashtable assertions = new Hashtable();

        ServletOutputStream toClient = reteCon.getClient();

        int codeSum = 0; // generate a signature to identify this query
        for (int i=0; i<query.length(); i++)
            codeSum += query.charAt(i);
        String querySignature = Integer.toString(codeSum);
        reteCon.setQuerySignature(querySignature);

        try { // generate HTML for the query page
            toClient.println("<html><head><title>Question Page</title>" +
                "<meta http-equiv=\"Content-Type\" content=\"text/html;\" +
                \"charset=windows-874\">");

            if (reteCon.getSeqError())
                toClient.println("<FONT FACE=\"Arial,Helvetica\">" +
                    " <B>Starting Over:</B>" +
                    " <BLINK>Please do not use BACK or FORWARD " +
                    "when responding to queries.</BLINK>");
            rows = items / 4 + 1;
            cols = (items <= 2 || items == 4) ? 2 : 3;

            toClient.println("<table width=\"75%\" border=\"0\"
align=\"center\">");
            toClient.println("<tr>");
            toClient.println("<td align=\"center\" height=\"70\" " +
                "bgcolor=\"#336699\">");
            toClient.println("<div align=\"left\"><font face=\"Arial,\" +
                \"Times New Roman, Times, serif\">" +
                "<b><font size=\"5\" color=\"#FFFFFF\">" +
                "Question page</font></b></font></div>");
            toClient.println("</td></tr>");
```

```

toClient.println("<tr><td height=\"259\" valign=\"top\" \" +
    \"bgcolor=\"#99ccff\">");
toClient.println("<p align=\"center\"><font face=\"Arial, \" +
    \"Times New Roman, Times, serif\"><h3 align=\"center\">\" +
    query + "</h3></font></p>");
toClient.println("<FORM name=\"startup\" method=POST \" + \"action=\"\" +
    + reteCon.getResponse().encodeURL(\"/servlet/EsServlet\")
    + \">");
toClient.println("<div align=\"center\">");
int next = 0;
toClient.println("<select name=\"action\">");
    for (int r=1; r<=rows; r++) {
        for (int c=1; c <= cols && next < items; c++) {
            toClient.println("<option VALUE=\"\" +
                factComponents[next*2+0] + \">\" +
                factComponents[next*2+0] + "</option>"); // add a selection
list
assertions.put(factComponents[next*2+0],factComponents[next*2+1]);
    // link button label and assertion for later use
    next++;
        }
    }
toClient.println("</select>");
toClient.println("<INPUT TYPE=\"submit\" NAME=\"action\" \" +
    \"VALUE=\"submit\">"); // add a button
toClient.println("</TABLE>"); // end table
toClient.println("<INPUT type=hidden name=\"querysig\" value=\"\" +
    querySignature + \">"); // embed the query signature in the
form
toClient.println("</FORM><FONT FACE=\"Arial,Helvetica\">");
toClient.println("<CENTER><P><A HREF=\"\"/expert/\">\" +
    \"Return to the ATCR-33S Expert System startup
page</A></CENTER>");
toClient.println("<P><CENTER><A HREF=\"\" + mainUrl + \">\" +
    \"Return to Phitsanulok Radar home page</A></CENTER>");
toClient.println("</FONT></CENTER></BODY></HTML>");
} catch (IOException e) {}
reteCon.setAssertions(assertions); // hand assertions to reteCon
}
}

```

3. EsQueryPanel.java

```

//EsQueryText.java
import java.io.*;
import java.util.*;
import javax.servlet.*;
import jess.*;

// displays a query
public class EsQueryPanel {
    String query; // query caption
    int items; // nr of query items
    String factComponents[]; // labels, and assertions
    final String mainUrl = "http://www.phitsa.com/"; //Phitsanulok Radar URL
    EsReteControl reteCon;

    public EsQueryPanel(String query, int items, String[] factComponents) {
        this.query = query;
        this.items = items;
        this.factComponents = factComponents;
    }
}

```

```

public EsQueryPanel(String query, int items, String[] factComponents,
    EsReteControl reteCon) {
    this.query = query;
    this.items = items;
    this.factComponents = factComponents;
    sendPanel(reteCon);
}

public void sendPanel(EsReteControl reteCon) {
    int rows, cols;
    Hashtable assertions = new Hashtable();

    ServletOutputStream toClient = reteCon.getClient();

    int codeSum = 0; // generate a signature to identify this query
    for (int i=0; i<query.length(); i++)
        codeSum += query.charAt(i);
    String querySignature = Integer.toString(codeSum);
    reteCon.setQuerySignature(querySignature);

    try { // generate HTML for the query page
        toClient.println("<html><head><title>Question Page</title>" +
            "<meta http-equiv=\"Content-Type\" content=\"text/html; " +
            "charset=windows-874\">");

        if (reteCon.getSeqError())
            toClient.println("<FONT FACE=\"Arial,Helvetica\">" +
                " <B>Starting Over:</B>" +
                " <BLINK>Please do not use BACK or FORWARD " +
                "when responding to queries.</BLINK>");

        rows = items / 4 + 1;
        cols = (items <= 2 || items == 4) ? 2 : 3;

        toClient.println("<table width=\"75%\" border=\"0\"
align=\"center\">");
        toClient.println("<tr>");
        toClient.println("<td align=\"center\" height=\"70\" " +
            "bgcolor=\"#336699\">");
        toClient.println("<div align=\"left\"><font face=\"Arial, " +
            "Times New Roman, Times, serif\">" +
            "<b><font size=\"5\" color=\"#FFFFFF\">" +
            "Question page</font></b></font></div>");
        toClient.println("</td></tr>");
        toClient.println("<tr><td height=\"259\" valign=\"top\" " +
            "bgcolor=\"#99ccff\">");
        toClient.println("<p align=\"center\"><font face=\"Arial, " +
            "Times New Roman, Times, serif\"><h3 align=\"center\">" +
            query + "</h3></font></p>");
        toClient.println("<FORM name=\"startup\" method=POST " + "action=\"" +
            + reteCon.getResponse().encodeURL("/servlet/EsServlet")
            + "\">");
        toClient.println("<div align=\"center\">");
        int next = 0;
        for (int r=1; r<=rows; r++) {
            for (int c=1; c <= cols && next < items; c++) {
                toClient.println("<INPUT TYPE=\"submit\" NAME=\"action\" " +
                    "VALUE=\"" + factComponents[next*2+0] + "\">"); // add a
button
                assertions.put(factComponents[next*2+0],
factComponents[next*2+1]);
                // link button label and assertion for later use
                next++;
            }
        }
    }
}

```

```

toClient.println( "<TR>" +
    "<TD BGCOLOR=\"#CCCCC\"><CENTER>" +
    "<INPUT TYPE=\"submit\" NAME=\"action\" VALUE=\"restart\">" +
    "</CENTER></TD></TR>" );
// link the Restart button with (restart)
assertions.put("restart", "(restart)");
toClient.println("</TABLE>"); // end table
toClient.println("<INPUT type=hidden name=\"querysig\" value=\"" +
    querySignature + "\">"); // embed the query signature in the
form
toClient.println("</FORM><FONT FACE=\"Arial,Helvetica\">");
toClient.println("<CENTER><P><A HREF=\"/expert/\">" +
    "Return to the ATCR-33S Expert System startup
page</A></CENTER>");
toClient.println("<P><CENTER><A HREF=\"" + mainUrl + "\">" +
    "Return to Phitsanulok Radar home page</A></CENTER>");
toClient.println("</FONT></CENTER></BODY></HTML>");
} catch (IOException e) {}
reteCon.setAssertions(assertions); // hand assertions to reteCon
}
}

```

4. EsAnsPanel.java

```

// EsAnsPanel.java
import java.io.*;
import java.util.*;
import javax.servlet.*;
import jess.*;

// sends a conclusion back to the user
public class EsAnsPanel {
    String query; // query caption

    public EsAnsPanel(String query, EsReteControl reteCon) {
        this.query = query;
        sendPanel(reteCon);
    }

    public void sendPanel(EsReteControl reteCon) {
        int rows, cols;
        Hashtable assertions = new Hashtable();

        ServletOutputStream toClient = reteCon.getClient();

        int codeSum = 0; // generate a signature to identify this query
        for (int i=0; i<query.length(); i++)
            codeSum += query.charAt(i);
        String querySignature = Integer.toString(codeSum);
        reteCon.setQuerySignature(querySignature);

        try { // generate HTML for the query page
            toClient.println("<html><head><title>Corrective action
page</title>" +
                "<meta http-equiv=\"Content-Type\" content=\"text/html; " +
                "charset=windows-874\">");
            toClient.println("</HEAD><BODY BGCOLOR=\"#B1F3BD\">");

            if (reteCon.getSeqError())
                toClient.println("<FONT FACE=\"Arial,Helvetica\">" +
                    "<B>Starting Over:</B>" +
                    "<BLINK>Please do not use BACK or FORWARD " +
                    "when responding to queries.</BLINK>");
        }
    }
}

```

```

        toClient.println("<FORM name=\"startup\" method=POST " +
"action=\"\"
        + reteCon.getResponse().encodeURL("/servlet/EsServlet")
        + "\">");
        toClient.println("<table width=\"75%\" border=\"0\"
align=\"center\">");
        toClient.println("<tr>");
        toClient.println("<td align=\"center\" height=\"70\" " +
        "bgcolor=\"#336699\">");
        toClient.println("<div align=\"left\"><font face=\"Arial, " +
        "Times New Roman, Times, serif\">" +
        "<b><font size=\"5\" color=\"#FFFFFF\">" +
        "Corrective action page</font></b></font></div>");
        toClient.println("</td></tr>");
        toClient.println("<tr><td height=\"259\" valign=\"top\" " +
        "bgcolor=\"#99ccff\">");
        toClient.println("<p align=\"center\"><font face=\"Arial, " +
        "Times New Roman, Times, serif\"><h3 align=\"center\">" +
        query + "</h3></font></p>");
        toClient.println("<FORM name=\"startup\" method=POST " + "action=\"\"
        + reteCon.getResponse().encodeURL("/servlet/EsServlet")
        + "\">");
        toClient.println( "<TR>" +
        "<TD BGCOLOR=\"#CCCCCC\"><CENTER>" +
        "<INPUT TYPE=\"submit\" NAME=\"action\" VALUE=\"\" +
        "restart\"></CENTER></TD></TR>" );

        // link the Restart button with (restart)
        assertions.put("restart", "(restart)");
        toClient.println("</TABLE>"); // end table
        toClient.println("<INPUT type=hidden name=\"querysig\" value=\"\" +
        querySignature + "\">"); // embed the query signature in the
form
        toClient.println("</FORM><FONT FACE=\"Arial,Helvetica\">");

        toClient.println("<CENTER><P><A HREF=\"/expert/\">" +
        "Return to the ATCR-33S Expert System startup
page</A></CENTER>");
        toClient.println("</FONT></BODY></HTML>");

        } catch (IOException e) {}
        reteCon.setAssertions(assertions); // hand assertions to reteCon
    }
}

```

5. EsReteControl.java

```

//EsReteControl.java
import java.util.*;
import java.io.*;
import javax.servlet.http.*;
import javax.servlet.*;
import jess.*;

public class EsReteControl {
    private Rete reteEngine;
    private Jess jessParse;
    private ServletOutputStream toClient;
    private HttpServletResponse res;
    private boolean firstTime = true, assertion = false;
    private boolean reset=false, seqError=false;
    private String fact, querySignature;
    private Hashtable assertions;

```

```

public EsReteControl(HttpServletResponse res) {
    this.res = res;
    reteEngine = new Rete();
    reteEngine.store("ReteControl", this);
}

public void run() {
    String KBFileName = null;
    if (firstTime) {
        firstTime = false;

        FileReader rulesFile;
        try { // open the CLIPS file
            KBFileName = "c:\\tomcat41\\webapps\\ROOT\\psres.clp";
            //KBFileName = "/home/virtual/site328/fst/var/www/html/psres.clp";
            rulesFile = new FileReader(KBFileName);

        } catch (FileNotFoundException fnfe) {
            rulesFile = null;
            //System.out.println("EsReteControl: cannot open: " +
KBFileName);
        }

        if (rulesFile != null) { // parse the CLIPS file
            jessParse = new Jess(rulesFile, reteEngine);
            try {
                jessParse.parse(false);
            } catch (JessException je) {
                //System.out.println("EsReteControl: error in jessParse.parse: " +
je);
            }
            try {
                rulesFile.close();
            } catch (Throwable exc) {}
        }
    }

    if (assertion) {
        assertion = false;
        try {
            reteEngine.executeCommand( "(assert " + fact + ")" );
            reteEngine.run();
        } catch (JessException je) {
            // System.out.println("EsReteControl: error in executeCommand / run: " +
je);
        }
    }

    if (reset) {
        reset = false;
        try {
            reteEngine.reset();
            reteEngine.run();
        } catch (JessException je) {
            //System.out.println("EsReteControl: error in reset / run: " +
je);
        }
    }
}

public void resetRete(ServletOutputStream sos) {
    seqError = false;
    toClient = sos;
    reset = true;
}

```

```

    }

    public void assertFact(String n_querySignature,
        String n_label, ServletOutputStream sos) {
        toClient = sos;          // save toClient for response
        if ( !(querySignature.equals(n_querySignature)) ) {
            seqError = true; // start over when a sequence error is detected
            reset = true;
        }
        else {
            seqError = false;
            // find the assertion to match the returned label
            fact = (String) assertions.get(n_label);
            if (fact.equals("(restart)"))
                reset = true;
            else
                assertion = true;
        }
    }
}

public boolean getSeqError() { return seqError;}
public ServletOutputStream getClient() { return toClient; }
public HttpServletResponse getResponse() { return res; }

public void setAssertions(Hashtable n_a) { assertions = n_a;}
public void setQuerySignature(String qs) { querySignature = qs; }
public void setSession(HttpServletResponse res) {
    this.res = res;
}
}

```

6. EsServlet.java

```

// EsServlet.java
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class EsServlet extends HttpServlet {
    HttpSession session;
    EsReteControl reteCon;

    public void init(ServletConfig Conf) throws ServletException {
        super.init(Conf);
    }

    public void doPost(HttpServletRequest req, HttpServletResponse res)
        throws IOException {
        res.setContentType("text/html"); // declare response type to client
        // find out what the client wants
        String action = req.getParameter("action");
        if (action.equals("start"))
            startNewSession(req, res);
        else if (action.equals("finish")) {
            continueSession(req, res, action);
        }
        else
            continueSession(req, res, action);
    }

    private void startNewSession(HttpServletRequest req, HttpServletResponse
res)
        throws IOException {

```

```

        session = req.getSession(true);           // get the session
        reteCon = new EsReteControl(res);          // make a new ReteCon

        try { // bind reteCon to session
            session.setAttribute("ReteCon", reteCon);
        } catch (IllegalStateException e) {}
        reteCon.resetRete(res.getOutputStream()); // initialize it
        reteCon.run();
    }

    private void continueSession(HttpServletRequest req,
        HttpServletResponse res, String action)
        throws IOException {
        session = req.getSession(false); // get the session

        if (session == null) { // if session has been dropped, restart
            System.out.println((new Date()).toString()+" null sesssion:
"+action);
            startNewSession(req, res);
            return;
        }

        try { // to get a reteCon from the session and assert the new fact
            // get the bound ReteCon
            reteCon = (EsReteControl) session.getAttribute("ReteCon");
            reteCon.setSession(res); // bind the new response object to the
reteCon
            reteCon.assertFact(req.getParameter("quersig"),
                action, res.getOutputStream());
            reteCon.run(); // change to start if ReteCon is threaded
        } catch (IllegalStateException e) {
            // old session has expired, so start a new one
            startNewSession(req, res);
        }
    }
}

```

7. stares.html

```

<!DOCTYPE html PUBLIC "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>

    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

    <title>ATCR-33S Diagnosis and Troubleshooting Expert System</title>
</head>
<body>
<center>
    <table width="75%" border="0">
        <tr>
            <td height="44" bgcolor="#336699">
                <div align="center"><b><font size="+1" color="#ffffff">Primary
Surveillance
                Radar Expert System</font></b> </div>
            </td>
        </tr>
        <tr>
            <td bgcolor="99ccff" height="286">
                <p>The ATCR-33S Diagnosis and Troubleshooting Expert System is an
<i>expert
                system</i> designed to help all staffs of Surveillance System
Engineering

```

```

        Department,<a href="http://www.aerothai.or.th"> Aeronautical Radio
of
        Thailand Co.,Ltd.</a> identify problem encountered during their
work.
    </p>
    <p>After you start the expert system, it will guide you through a
series
        of questions about the problem and then recommend you a solution.
</p>
    <center>
        <form name="startup" method="POST" action="/servlet/EsServlet">
            <input type="hidden" name="action" value="start">
            <input type="submit" value="Start Expert System" name="submit">
        </form>
    </center>
        <div align="right"><a href="assist.html"> assistant page </a>
</div>
    </td>
</tr>
<tr>
    <td bgcolor="#99ccff">
        <p>Contacts and further information: </p>
        <ul>
            <li> For comments or questions about Primary radar , contact <a
href="http://www.phitsa.com/contact/">
                Phitsanulok radar staff</a> , Surveillance Sytem Engineering
Department
                , Aeronautical Radio of Thailand .</li>
            <li> For technical questions or comments about design and
programming
                of this web site, contact <a href="mailto:g4440159@cm.edu">
developer</a>,
                Information Technology and Management, Graduate Study, Chiangmai
University
                , Thailand. </li>
        </ul>
    </td>
</tr>
</table>
</center>
<center>
    <applet code="EsApplet.class" height="25" width="300"
bgcolor="#ffffff"></applet>
</center>
<hr width="50%">
<br>
<center>
    <font face="Arial,Helvetica"><font size="-2">@Copyright 2003 bar-
foo.com</font></font>
</center>

</body>
</html>

```