

REFERENCES

- [1] G. C. D. G. Fabio Bellifemine, *Developing Multi-Agent systems with JADE*, West Sussex, England: Jhon Wiley & Sons Ltd, 2007.
- [2] J. C. D.-w. G. J.-h. Z. Yi-Nan Guo, "A Novel Multi-agent Based Complex Process Control System and Its Application," *Lecture Notes in Control and Information Sciences*, Volume 344, pp. 319-330, 2006.
- [3] R. Tahboub, D. Lazarescu, V. lazarescu, "Software Engineering and Database Web Management in Daleelcom Design", *Proceedings of the International Conference on Information Technology: Coding and Computing(ITCC'05),IEEE,0-7695-2315-3/05*, 2005.
- [4] R. Tahboub, D. Lazarescu, "Secure Information Store Web Access in Time Sheet Management System", *Scientific Bulletin, University Politehnica of Bucharest, S. C, V. 66, N.2-4*, 2004.
- [5] Tahboub.R,Lazarescu.V, "Novel Approach for Remote Energy Meter Reading Using Mobile Agents" *Information Technology: New Generations*, 2006. ITNG 2006. Third International Conference on Date 10-12 April 2006
- [6] Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, and D. Rus, *Mobile agents for distributed information retrieval*. In Mathias Klusch, editor, *Intelligent Information Agents*, chapter 15, pp. 355-395, Springer-Verlag, 1999.
- [7] C. Bohoris, A. Liotta, G. Pavlou, "Mobile Agent Based Performance Management for the Virtual Home Environment", *Journal of Network and System Management*, pp. 133-149, June 2003.
- [8] Hongjun Li, "Intelligent Distributed Fault and Performance Management for Communication Networks". CSHCN PhD 2002-2.
- [9] Horvat D. , Cvetkovic D. , Milutinovic V. , Kocovic P. and Kovacevic V. 'Mobile Agents and Java Mobile Agents Toolkits'. *System Sciences*, 2000. *Proceedings of the 33rd Annual Hawaii International Conference on 4-7 Jan. 2000*
- [10] Rafał Leszczyzna Version 'Evaluation of Agent Platforms' Version 2.0 EUR 23508EN - 2008
- [11] Nguyen G., Dang T.T., Hluchy L., Laclavik M., Balogh Z.and Budinska I., 'Agent Platform Evaluation and Comparison', *Published by Institute of informatics, Slovak Academy of Sciences, Pellucid 5FP IST -2001-34519*, June 2002.
- [12] Vu Anh Pham,Karmouch, A. *Mobile software agents: an overview*

Communications Magazine, IEEE (Volume:36 , Issue: 7)

- [13] <http://www.recursionsw.com/products/voyager/voyager-platforms.html>
[online]
- [14] *Mudumbai, Srilekha S, Johnston, William, Essiari, Abdelilah 'Anchor Toolkit - a secure mobile agent system',06-16-2008*
- [15] *Hyacinth S. Nwana, Divine T. Ndumu, Lyndon C. Lee & Jaron C. Collis 'ZEUS: A Toolkit for Building Distributed Multi-Agent Systems'*
- [16] *F. Bellifemine, G. Caire, T. Trucco and G. Rimassa, JADE PROGRAMMER'S GUIDE, Free Software Foundation,, April-2010.*



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Appendix A : COMMUNICATION DRIVER SOFTWARE CODE

```
package metermonitor.meterreadingagent.utilities;
```

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.*;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
```

```
/**
```

```
*
```

```
* @author Dulan Ranawaka
```

```
*/
```

```
public class ReadingMeterRegister {
```

```
    Socket smtpSocket = null;
```

```
    DataOutputStream os = null;
```

```
    DataInputStream is = null;
```

```
    byte b1[] = new byte[]{3,0,8,0,0,0,0,0,0,0};
```

```
    byte b2[] = new byte[]{3,0,5,47,63,33,13,10};
```

```
    byte b3[] = new byte[]{3,0,6,6,48,53,48,13,10};
```

```
    String responseLine;
```

```
    public String Read(String hostname){
```

```
        try {
```

```
            smtpSocket = new Socket(hostname, 2010);
```

```
            os = new DataOutputStream(smtpSocket.getOutputStream());
```

```
            is = new DataInputStream(smtpSocket.getInputStream());
```

```
        } catch (UnknownHostException e) {
```

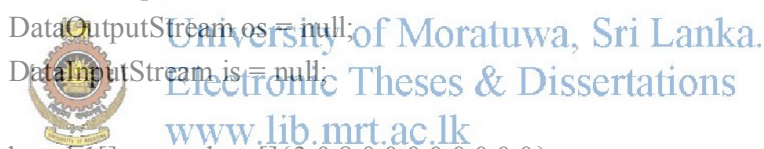
```
            System.err.println("Don't know about host: hostname");
```

```
        } catch (IOException e) {
```

```
            System.err.println("Couldn't get I/O for the connection to: hostname");
```

```
        }
```

```
    }
```



```

if (smtpSocket != null && os != null && is != null) {
    try {
        os.write(b1);
        os.write(b2);
        responseLine = is.readLine();
        os.write(b3);
        while ((responseLine = is.readLine()) != null) {
            if( responseLine.startsWith("72.7.0")){
                break;
            }
            if (responseLine.indexOf("Ok") != -1) {
                break;
            }
        }
        os.close();
        is.close();
        smtpSocket.close();
    } catch (UnknownHostException e) {
        System.err.println("Trying to connect to unknown host: " + e);
    } catch (IOException e) {
        System.err.println("IOException: " + e);
    }
}

if (responseLine != null){
    responseLine=responseLine.split("\\(")[1].split(".")[0];
}
else{
    responseLine= Integer.toString(0);
}
return responseLine;
}
}


```



Appendix B : MONITOR AGENT SOFTWARE CODE

```
package metermonitor.monitoragent.agent;

import jade.core.AID;
import jade.core.Agent;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAException;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import metermonitor.common.utilities.CommonConstants;
import metermonitor.monitoragent.behaviours.MeterAssigningBehaviour;
import metermonitor.monitoragent.behaviours.MeterPickupBehaviour;
import metermonitor.monitoragent.behaviours.MeterReadingPickupBehaviour;
import metermonitor.monitoragent.behaviours.NotifyMeterAssignmentBehaviour;
import metermonitor.monitoragent.behaviours.ReadingAgentPickupBehaviour;

/**
 *  University of Moratuwa, Sri Lanka.
 * Electronic Theses & Dissertations
 * www.lib.mrt.ac.lk
 * @author Dulan Ranawaka
 */
public class MonitorAgent extends Agent {

    List<AID> _readingAgents = new ArrayList<>();
    List<String> _meters = new ArrayList<>();
    Hashtable<String, String> _meterNumbers = new Hashtable<>();
    Hashtable<String, String> _meterIps = new Hashtable<>();
    Hashtable<AID, List<String>> _meterAssignment = new Hashtable<>();
    MonitorAgentForm _agentUiForm = null;

    @Override
    protected void setup() {
        _agentUiForm = new MonitorAgentForm(this);
        _agentUiForm.setVisible(true);
        _agentUiForm.setAgentName(getAID().getName());
    }
}
```

```

showMessage("Registering as a monitor...");

ServiceDescription serviceDescription = new ServiceDescription();
serviceDescription.setType(CommonConstants.MonitorServiceType);
serviceDescription.setName(CommonConstants.MonitorServiceName);

DFAgentDescription df = new DFAgentDescription();
df.setName(getAID());
df.addServices(serviceDescription);

try {
    DFService.register(this, df);
} catch (FIPAException ex) {
    Logger.getLogger(MonitorAgent.class.getName()).log(Level.SEVERE, null,
ex);
}

showMessage("Starting behaviours...");

addBehaviour(new ReadingAgentPickupBehaviour());
addBehaviour(new NotifyMeterAssignmentBehaviour(this, 10000,
_meterAssignment));
addBehaviour(new MeterReadingPickupBehaviour());
addBehaviour(new MeterPickupBehaviour());
}

public void addReadingAgent(AID readingAgent) {
    showMessage("Adding agent: " + readingAgent.getName());

    for (AID ra : _readingAgents) {
        if (ra.getName().equals(readingAgent.getName())) return;
    }

    _readingAgents.add(readingAgent);

    addBehaviour(new MeterAssigningBehaviour(_readingAgents, _meters));

    List<String> agentNames = new ArrayList<>();
    _readingAgents.stream().forEach((ra) -> {
        agentNames.add(ra.getName());
    });
}

```

```

        _agentUiForm.updateReadingAgents(agentNames.toArray(new
String[agentNames.size()]));
    }

    public void removeReadingAgent(AID readingAgent) {
        showMessage("Adding agent: " + readingAgent.getName());

        AID agentToRemove = null;

        for (AID ra : _readingAgents) {
            if(ra.getName().equals(readingAgent.getName())) {
                agentToRemove = ra;
                break;
            }
        }

        if (agentToRemove == null) return;
        _readingAgents.remove(agentToRemove);

        addBehaviour(new MeterAssigningBehaviour(_readingAgents, _meters));

        List<String> agentNames = new ArrayList<>();
        _readingAgents.stream().forEach((ra) -> {
            agentNames.add(ra.getName());
        });

        _agentUiForm.updateReadingAgents(agentNames.toArray(new
String[agentNames.size()]));
    }

    public boolean addMeter(String meter) {
        if (!_meterNumbers.containsKey(meter)) return false;
        if (_meters.stream().anyMatch(m -> m.equals(meter))) return false;

        _meters.add(_meterNumbers.get(meter));

        addBehaviour(new MeterAssigningBehaviour(_readingAgents, _meters));

        return true;
    }
}

```

```

public void updateMeterAssignment(Hashtable<AID, List<String>>
meterAssignment) {
    _meterAssignment.clear();
    _meterAssignment.putAll(meterAssignment);
}

public void addMeterInfo(String meterNumber, String meterIp) {
    if (_meterNumbers.containsKey(meterNumber) ||
_meterIps.containsKey(meterIp)) return;
    _meterNumbers.put(meterNumber, meterIp);
    _meterIps.put(meterIp, meterNumber);
}

public void updateMeterReading(String meter, int reading) {
    _agentUiForm.updateMeterReading(_meterIps.get(meter), reading);
}

public void showMessage(String message) {
    _agentUiForm.addMessage(message);
}
}

```



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk


```

package metermonitor.monitoragent.agent;

import jade.core.AID;
import jade.lang.acl.ACLMessage;
import jade.util.Logger;
import java.awt.BorderLayout;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;
import javax.swing.DefaultListModel;
import javax.swing.ListModel;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.data.general.DefaultPieDataset;

/**
 *  University of Moratuwa, Sri Lanka.
 * Electronic Theses & Dissertations
 * www.lib.mrt.ac.lk
 */
 * @author Dulan Ranawaka
 */
public class MonitorAgentForm extends javax.swing.JFrame {

    private final MonitorAgent _agent;
    private final Hashtable<String, Integer> _meterReadings = new Hashtable<>();

    /**
     * Creates new form MonitorAgentForm
     * @param agent
     */
    public MonitorAgentForm(MonitorAgent agent) {
        _agent = agent;

        initComponents();
    }

```

```

Timer chartUpdateTimer = new Timer();
chartUpdateTimer.schedule(new TimerTask() {
    @Override
    public void run() {
        updateChart();
    }
}, 3000, 3000);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    btnAddMeter = new javax.swing.JButton();
    txtMeter = new javax.swing.JTextField();
    jScrollPane2 = new javax.swing.JScrollPane();
    lstMessages = new javax.swing.JList();
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jScrollPane3 = new javax.swing.JScrollPane();
    lstReadingAgents = new javax.swing.JList();
    jScrollPane1 = new javax.swing.JScrollPane();
    lstMeters = new javax.swing.JList();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    btnAddMeter.setText("Add");
    btnAddMeter.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnAddMeterActionPerformed(evt);
        }
    });
}
}

```

```

jScrollPane2.setViewportViewView(lstMessages);

jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
.addGap(0, 0, Short.MAX_VALUE)
);
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
.addGap(0, 376, Short.MAX_VALUE)
);
jLabel1.setText("Reading Agents");
jScrollPane3.setViewportViewView(lstReadingAgents);

jScrollPane1.setViewportViewView(lstMeters);

jLabel2.setText("Meters");

jLabel3.setText("Messages");

jLabel4.setText("Meter Readings");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()

```



```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(layout.createSequentialGroup())
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(layout.createSequentialGroup())
```

```
        .addContainerGap()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(jScrollPane3,  
    javax.swing.GroupLayout.PREFERRED_SIZE, 318,  
    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel1))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```
    .addGroup(layout.createSequentialGroup()  
    .addComponent(jLabel2,  
    .addGap(226, 226, 226))  
    .addComponent(jScrollPane1,  
    javax.swing.GroupLayout.DEFAULT_SIZE, 299, Short.MAX_VALUE)))
```

```
        .addGroup(layout.createSequentialGroup())
```

```
            .addGap(342, 342, 342)
```

```
            .addComponent(txtMeter,
```

```
            javax.swing.GroupLayout.PREFERRED_SIZE, 213,  
            javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(btnAddMeter,
```

```
    javax.swing.GroupLayout.DEFAULT_SIZE,  
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
```

```
        .addGap(12, 12, 12)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(jScrollPane2,
```

```
    javax.swing.GroupLayout.PREFERRED_SIZE, 295,  
    javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

        .addComponent(jLabel3)))
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        .addContainerGap())
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel4)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
        .addComponent(jLabel1)
        .addComponent(jLabel2)
        .addComponent(jLabel3))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
        .addComponent(jScrollPane3,
javax.swing.GroupLayout.DEFAULT_SIZE, 229, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addComponent(jScrollPane1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
        .addComponent(txtMeter,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnAddMeter)))

```



```

        .addComponent(jScrollPane2))
        .addGap(14, 14, 14)
        .addComponent(jLabel4)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void btnAddMeterActionPerformed(java.awt.event.ActionEvent evt) {
    if (!_agent.addMeter(txtMeter.getText())) return;

    ListModel meterListModel = lstMeters.getModel();
    if (meterListModel == null || !(meterListModel instanceof DefaultListModel))
    {
        meterListModel = new DefaultListModel<>();
        lstMeters.setModel(meterListModel);
    }

    ((DefaultListModel)meterListModel).addElement(txtMeter.getText());
}

// Variables declaration - do not modify
private javax.swing.JButton btnAddMeter;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JList lstMessages;
private javax.swing.JList lstMeters;
private javax.swing.JList lstReadingAgents;

```



```

private javax.swing.JTextField txtMeter;
// End of variables declaration

public void setAgentName(String agentName) {
    this.setTitle(agentName);
}

public void updateReadingAgents(String[] readingAgents) {
    DefaultListModel<String> readingAgentListModel = new
DefaultListModel<>();

    for (String readingAgent : readingAgents) {
        readingAgentListModel.addElement(readingAgent);
    }

    lstReadingAgents.setModel(readingAgentListModel);
    updateChart();
}

public void updateMeterReading(String meter, int reading) {
    if ( _meterReadings.containsKey(meter) ) {
        _meterReadings.replace(meter, reading);
    } else {
        _meterReadings.put(meter, reading);
    }
}

private void updateChart() {
    DefaultCategoryDataset objDataset = new DefaultCategoryDataset();

    Iterator<Map.Entry<String, Integer>> it = _meterReadings.entrySet().iterator();

    while (it.hasNext()) {
        Map.Entry<String, Integer> entry = it.next();

        if (objDataset.getColumnKeys().contains(entry.getKey())) {
            objDataset.setValue(entry.getValue(), "X", entry.getKey());
        } else {
            objDataset.addValue(entry.getValue(), "X", entry.getKey());
        }
    }
}

```

```

    }

    JFreeChart objChart = ChartFactory.createBarChart("Meter Readings", "Meter
Serial Number", "Meter Readings (kVA)", objDataset, PlotOrientation.VERTICAL,
false, true, false);

    ChartPanel CP = new ChartPanel(objChart);
    jPanel1.setLayout(new java.awt.BorderLayout());
    jPanel1.removeAll();
    jPanel1.add(CP, BorderLayout.SOUTH);
    jPanel1.validate();
}

public void addMessage(String message) {
    ListModel meterIdListModel = lstMessages.getModel();
    if (meterIdListModel == null || !(meterIdListModel instanceof
DefaultListModel))
    {
        meterIdListModel = new DefaultListModel<>();
        lstMessages.setModel(meterIdListModel);
    }
    if(((DefaultListModel)meterIdListModel).size()>10)
    {
        ((DefaultListModel)meterIdListModel).clear();
        ((DefaultListModel)meterIdListModel).removeAllElements();
    }
    ((DefaultListModel)meterIdListModel).addElement(message);
}
}
}

```




```

package metermonitor.monitoragent.behaviours;

import jade.core.AID;
import jade.core.behaviours.OneShotBehaviour;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.List;
import metermonitor.monitoragent.agent.MonitorAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class MeterAssigningBehaviour extends OneShotBehaviour {

    private final List<AID> _readingAgents;
    private final List<String> _meters;

    public MeterAssigningBehaviour(List<AID> readingAgents, List<String> meters)
    {
        this._readingAgents = readingAgents;
        this._meters = meters;
    }

    @Override
    public void action() {
        if(myAgent instanceof MonitorAgent) {
            ((MonitorAgent)myAgent).showMessage("Calculating meter assignment...");
        }

        if (_readingAgents.size() == 0 || _meters.size() == 0) return;

        Hashtable<AID, List<String>> meterAssignment = new Hashtable<>();

        _readingAgents.stream().forEach((readingAgent) -> {
            meterAssignment.put(readingAgent, new ArrayList<>());
        });

        for (int i = 0; i < _meters.size(); i++) {
            meterAssignment.get(_readingAgents.get(i %
                _readingAgents.size())).add(_meters.get(i));
        }
    }
}

```



```

    }
    if(myAgent instanceof MonitorAgent) {
        ((MonitorAgent)myAgent).updateMeterAssignment(meterAssignment);
    }
}
}
package metermonitor.monitoragent.behaviours;

import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import metermonitor.monitoragent.agent.MonitorAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class MeterPickupBehaviour extends CyclicBehaviour {

    @Override
    public void action() {
        MessageTemplate messageTemplate =
        MessageTemplate.MatchPerformative(ACLMessage.INFORM_REF);
        ACLMessage msg = myAgent.receive(messageTemplate);

        if (msg != null) {
            if(myAgent instanceof MonitorAgent) {
                ((MonitorAgent)myAgent).showMessage("Meter received from agent: " +
                msg.getSender().getName());

                String[] messageContent = msg.getContent().split("@");
                if(messageContent.length == 0) return;

                ((MonitorAgent)myAgent).addMeterInfo(messageContent[0],
                messageContent[1]);
            }
        }
        else {
            block();
        }
    }
}
}

```

```

package metermonitor.monitoragent.behaviours;

import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import metermonitor.monitoragent.agent.MonitorAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class MeterReadingPickupBehaviour extends CyclicBehaviour {

    @Override
    public void action() {
        MessageTemplate messageTemplate =
        MessageTemplate.MatchPerformative(ACLMessage.CFP);
        ACLMessage msg = myAgent.receive(messageTemplate);

        if (msg != null) {
            if (myAgent instanceof MonitorAgent)
                ((MonitorAgent)myAgent).showMessage("Meter reading received from
agent: " + msg.getSender().getName());

            String[] messageContent = msg.getContent().split("\\$");
            if(messageContent.length == 0) return;

            ((MonitorAgent)myAgent).updateMeterReading(messageContent[0],
Integer.parseInt(messageContent[1]));
        }
        else {
            block();
        }
    }
}

```

```

package metermonitor.monitoragent.behaviours;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;
import java.util.Hashtable;
import java.util.List;
import metermonitor.monitoragent.agent.MonitorAgent;
/**
 *
 * @author Dulan Ranawaka
 */
public class NotifyMeterAssignmentBehaviour extends TickerBehaviour {

    private final Hashtable<AID, List<String>> _meterAssignment;

    public NotifyMeterAssignmentBehaviour(Agent a, long period, Hashtable<AID,
List<String>> meterAssignment) {
        super(a, period);
        this._meterAssignment = meterAssignment;
    }
    @Override
    protected void onTick() {
        if(myAgent instanceof MonitorAgent) {
            ((MonitorAgent)myAgent).showMessage("Notifying meter assignment...");
        }
        _meterAssignment.forEach((a, m) -> {
            AID readingAgent = a;
            if (m.size() > 0) {
                ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
                msg.addReceiver(readingAgent);

                String messageContent = "";
                for (String meter : m) {
                    messageContent = messageContent + meter + "$";
                }
                messageContent = messageContent.substring(0, messageContent.length() -
1);

                msg.setContent(messageContent);
                myAgent.send(msg);
            }
        });
    }
}

```

```

    }
  });
}
}
package metermonitor.monitoragent.behaviours;

import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import metermonitor.common.utilities.CommonConstants;
import metermonitor.monitoragent.agent.MonitorAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class ReadingAgentPickupBehaviour extends CyclicBehaviour {

    @Override
    public void action() {
        MessageTemplate messageTemplate =
        MessageTemplate.MatchPerformative(ACLMessage.INFORM);
        ACLMessage msg = myAgent.receive(messageTemplate);

        if (msg != null) {
            if(myAgent instanceof MonitorAgent) {
                ((MonitorAgent)myAgent).showMessage("Message received from agent: "
+ msg.getSender().getName());

                if(msg.getContent().equals(CommonConstants.StartupMessage))
                    ((MonitorAgent)myAgent).addReadingAgent(msg.getSender());
                else if(msg.getContent().equals(CommonConstants.ShutdownMessage))
                    ((MonitorAgent)myAgent).removeReadingAgent(msg.getSender());
            }
        }
        else {
            block();
        }
    }
}
}

```

Appendix C : ACCESS AGENT SOFTWARE CODE

```
package metermonitor.meterreadingagent.agent;

import jade.core.AID;
import jade.core.Agent;
import jade.lang.acl.ACLMessage;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import metermonitor.common.utilities.CommonConstants;
import metermonitor.meterreadingagent.behaviours.AssignmentPickupBehaviour;
import metermonitor.meterreadingagent.behaviours.NotifyMeterReadingBehaviour;
import metermonitor.meterreadingagent.behaviours.NotifyStartupBehaviour;

/**
 *
 * @author Dulan Ranawaka
 */
public class MeterReadingAgent extends Agent {

    Hashtable<AID, List<String>> _meterReadingRequests = new Hashtable<>();
    Hashtable<String, List<AID>> _monitors = new Hashtable<>();
    MeterReadingAgentForm _agentUiForm = null;

    @Override
    protected void setup() {
        _agentUiForm = new MeterReadingAgentForm(this);
        _agentUiForm.setVisible(true);
        _agentUiForm.setAgentName(getAID().getName());

        showMessage("Starting behaviours...");

        addBehaviour(new NotifyStartupBehaviour(this, 15000));
        addBehaviour(new AssignmentPickupBehaviour(_meterReadingRequests));
        addBehaviour(new NotifyMeterReadingBehaviour(this, 5000, _monitors));
    }

    @Override
    protected void takeDown() {
        showMessage("Notifying shutdown to monitors...");

        if (_meterReadingRequests.isEmpty()) return;

        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
```

```

    Iterator<Map.Entry<AID, List<String>>> it =
_meterReadingRequests.entrySet().iterator();

    while (it.hasNext()) {
        Map.Entry<AID, List<String>> entry = it.next();

        msg.addReceiver(entry.getKey());
    }

    msg.setContent(CommonConstants.ShutdownMessage);
    send(msg);
}

public void updateMeterReadingRequests(AID monitor, List<String> meters) {
    showMessage("Updating meter reading requests...");

    Iterator<Map.Entry<AID, List<String>>> it =
_meterReadingRequests.entrySet().iterator();

    while (it.hasNext()) {
        Map.Entry<AID, List<String>> entry = it.next();

        if (entry.getKey().getName().equals(monitor.getName())) {
            it.remove();
        }

        _meterReadingRequests.put(monitor, meters);

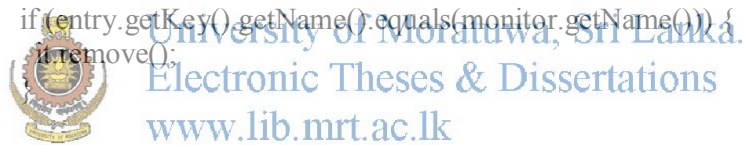
        _meterReadingRequests.forEach((m, mt) -> {
            _agentUiForm.updateMeters(mt);
        });

        for (String meter : meters) {
            if (_monitors.containsKey(meter)) {
                List<AID> existingMonitors = _monitors.get(meter);

                if (!existingMonitors.stream().anyMatch(m ->
m.getName().equals(monitor)))
                    existingMonitors.add(monitor);
            }
            else {
                List<AID> monitors = new ArrayList<>();
                monitors.add(monitor);

                _monitors.put(meter, monitors);
            }
        }
    }
}

```



```
    }  
  }  
}  
  
public void shutdown() {  
    _agentUiForm.setVisible(false);  
    doDelete();  
}  
  
public void showMessage(String message) {  
    _agentUiForm.addMessage(message);  
}  
}
```



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk


```

package metermonitor.meterreadingagent.agent;

import java.util.List;
import javax.swing.DefaultListModel;
import javax.swing.ListModel;

/**
 *
 * @author Dulan Ranawaka
 */
public class MeterReadingAgentForm extends javax.swing.JFrame {

    private final MeterReadingAgent _readingAgent;

    /**
     * Creates new form MeterReadingAgentForm
     * @param readingAgent
     */
    public MeterReadingAgentForm(MeterReadingAgent readingAgent) {
        this._readingAgent = readingAgent;

        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane3 = new javax.swing.JScrollPane();
        jList2 = new javax.swing.JList();
        jScrollPane1 = new javax.swing.JScrollPane();
        lstMeters = new javax.swing.JList();
        btnShutdown = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        lstMessages = new javax.swing.JList();
        jLabel2 = new javax.swing.JLabel();

        jList2.setModel(new javax.swing.AbstractListModel() {
            String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };
            public int getSize() { return strings.length; }
            public Object getElementAt(int i) { return strings[i]; }
        });

```

```

});
jScrollPane3.setViewportView(jList2);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jScrollPane1.setViewportView(lstMeters);

btnShutdown.setText("Shutdown");
btnShutdown.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnShutdownActionPerformed(evt);
    }
});

jLabel1.setText("Meters");

jScrollPane2.setViewportView(lstMessages);

jLabel2.setText("Messages");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                        layout.createSequentialGroup()
                            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                                layout.createSequentialGroup()
                                    .addGap(0, 0, Short.MAX_VALUE)
                                    .addComponent(btnShutdown))
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                                    layout.createSequentialGroup()
                                        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                                            layout.createSequentialGroup()
                                                .addComponent(jScrollPane1,
                                                    javax.swing.GroupLayout.DEFAULT_SIZE, 275, Short.MAX_VALUE)
                                                .addGroup(layout.createSequentialGroup()
                                                    .addComponent(jLabel1)
                                                    .addGap(0, 0, Short.MAX_VALUE)))
                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel2)
    .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 294,
javax.swing.GroupLayout.PREFERRED_SIZE))))
    .addContainerGap()
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
    .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel1)
    .addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jScrollPane2)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 203, Short.MAX_VALUE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 66,
Short.MAX_VALUE)
    .addComponent(btnShutdown)
    .addContainerGap()
);

pack();
} // </editor-fold>

private void btnShutdownActionPerformed(java.awt.event.ActionEvent evt) {
    _readingAgent.shutdown();
}

// Variables declaration - do not modify
private javax.swing.JButton btnShutdown;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JList jList2;
private javax.swing.JScrollPane jScrollPane1;

```

```

private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JList lstMessages;
private javax.swing.JList lstMeters;
// End of variables declaration

public void setAgentName(String agentName) {
    this.setTitle(agentName);
}

public void updateMeters(List<String> meters) {
    DefaultListModel<String> meterListModel = new DefaultListModel<>();

    meters.stream().filter((meter) ->
(!meterListModel.contains(meter))).forEach((meter) -> {
        meterListModel.addElement(meter);
    });

    lstMeters.setModel(meterListModel);
}

public void addMessage(String message) {
    ListModel meterIdListModel = lstMessages.getModel();
    if (meterIdListModel == null || !(meterIdListModel instanceof
DefaultListModel))
    {
        meterIdListModel = new DefaultListModel<>();
        lstMessages.setModel(meterIdListModel);
    }
    if(((DefaultListModel)meterIdListModel).size()>10)
    {
        ((DefaultListModel)meterIdListModel).clear();
        ((DefaultListModel)meterIdListModel).removeAllElements();
    }
    ((DefaultListModel)meterIdListModel).addElement(message);
}
}

```

```

package metermonitor.meterreadingagent.behaviours;

import jade.core.AID;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import java.util.Arrays;
import java.util.Hashtable;
import java.util.List;
import metermonitor.meterreadingagent.agent.MeterReadingAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class AssignmentPickupBehaviour extends CyclicBehaviour {

    private final Hashtable<AID, List<String>> _meterReadingRequests;

    public AssignmentPickupBehaviour(Hashtable<AID, List<String>>
meterReadingRequests) {
        this._meterReadingRequests = meterReadingRequests;
    }

    @Override
    public void action() {
        MessageTemplate messageTemplate =
MessageTemplate.MatchPerformative(ACLMessage.REQUEST);
        ACLMessage msg = myAgent.receive(messageTemplate);

        if (msg != null) {
            if(myAgent instanceof MeterReadingAgent) {
                ((MeterReadingAgent)myAgent).showMessage("Reading request received
from agent: " + msg.getSender().getName());

                String[] meters = msg.getContent().split("\\$");
                if(meters.length == 0) return;

                ((MeterReadingAgent)myAgent).updateMeterReadingRequests(msg.getSender(),
Arrays.asList(meters));
            }
            else {
                block();
            }
        }
    }
}

```

```

}
package metermonitor.meterreadingagent.behaviours;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Random;
import metermonitor.meterreadingagent.agent.MeterReadingAgent;
import metermonitor.meterreadingagent.utilities.ReadingMeterRegister;
/**
 *
 * @author Dulan Ranawaka
 */
public class NotifyMeterReadingBehaviour extends TickerBehaviour {

    private final Hashtable<String, List<AID>> _monitors;

    public NotifyMeterReadingBehaviour(Agent a, long period, Hashtable<String,
List<AID>> monitors) {
        super(a, period);
        this._monitors = monitors;
    }
    @Override
    protected void onTick() {
        if(myAgent instanceof MeterReadingAgent && !_monitors.isEmpty()) {
            ((MeterReadingAgent)myAgent).showMessage("Informing meter
readings..");
        }

        Iterator<Map.Entry<String, List<AID>>>> it = _monitors.entrySet().iterator();
        while (it.hasNext()) {
            Map.Entry<String, List<AID>> entry = it.next();
            ACLMessage msg = new ACLMessage(ACLMessage.CFP);

            for (AID monitor : entry.getValue()) {
                msg.addReceiver(monitor);
            }
            ReadingMeterRegister Meter_reder =new ReadingMeterRegister();

            msg.setContent(entry.getKey() + "$" + Meter_reder.Read(entry.getKey()));
            myAgent.send(msg);
        }
    }
}

```

```

    }
}
package metermonitor.meterreadingagent.behaviours;

import jade.core.AID;
import jade.core.behaviours.OneShotBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAException;
import jade.lang.acl.ACLMessage;
import java.util.logging.Level;
import java.util.logging.Logger;
import metermonitor.common.utilities.CommonConstants;
import metermonitor.meterreadingagent.agent.MeterReadingAgent;
import metermonitor.meterreadingagent.utilities.ReadingAgentConstants;

/**
 *
 * @author Dulan Ranawaka
 */
public class NotifyShutdownBehaviour extends OneShotBehaviour {
    private AID[] _monitorAgents;

    @Override
    public void action() {
        ServiceDescription serviceDescription = new ServiceDescription();
        serviceDescription.setType(CommonConstants.MonitorServiceType);

        DFAgentDescription df = new DFAgentDescription();
        df.addServices(serviceDescription);

        try {
            DFAgentDescription[] monitors = DFService.search(myAgent, df);
            _monitorAgents = new AID[monitors.length];
            for (int i = 0; i < monitors.length; i++) {
                _monitorAgents[i] = monitors[i].getName();
            }
        } catch (FIPAException ex) {
            Logger.getLogger(MeterReadingAgent.class.getName()).log(Level.SEVERE,
null, ex);
        }
        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
        for (AID monitorAgent : _monitorAgents) {
            msg.addReceiver(monitorAgent);
        }
        msg.setContent(CommonConstants.ShutdownMessage);
    }
}

```

```

        myAgent.send(msg);
    }
}
package metermonitor.meterreadingagent.behaviours;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.TickerBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAException;
import jade.lang.acl.ACLMessage;
import java.util.logging.Level;
import java.util.logging.Logger;
import metermonitor.common.utilities.CommonConstants;
import metermonitor.meterreadingagent.agent.MeterReadingAgent;
import metermonitor.monitoragent.agent.MonitorAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class NotifyStartupBehaviour extends TickerBehaviour {
    private AID[] _monitorAgents;

    public NotifyStartupBehaviour(Agent a, long period) {
        super(a, period);
    }

    @Override
    protected void onTick() {
        if(myAgent instanceof MeterReadingAgent) {
            ((MeterReadingAgent)myAgent).showMessage("Searching for monitors..");
        }

        ServiceDescription serviceDescription = new ServiceDescription();
        serviceDescription.setType(CommonConstants.MonitorServiceType);

        DFAgentDescription df = new DFAgentDescription();
        df.addServices(serviceDescription);

        try {
            DFAgentDescription[] monitors = DFService.search(myAgent, df);
            _monitorAgents = new AID[monitors.length];

```



```

        for (int i = 0; i < monitors.length; i++) {
            _monitorAgents[i] = monitors[i].getName();
        }
    } catch (FIPAException ex) {
        Logger.getLogger(MeterReadingAgent.class.getName()).log(Level.SEVERE,
null, ex);
    }

    if(myAgent instanceof MeterReadingAgent) {
        ((MeterReadingAgent)myAgent).showMessage("Informing startup to
monitors..");
    }

    ACLMessage msg = new ACLMessage(ACLMessage.INFORM);

    for (AID monitorAgent : _monitorAgents) {
        msg.addReceiver(monitorAgent);
    }

    msg.setContent(CommonConstants.StartupMessage);
    myAgent.send(msg);
}
}

```



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Appendix D : DATABASE AGENT SOFTWARE CODE

```
package metermonitor.dataentryagent.agent;

import jade.core.AID;
import jade.core.Agent;
import java.util.AbstractList;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.List;
import metermonitor.dataentryagent.behaviours.MonitorDiscoveryBehaviour;
import metermonitor.dataentryagent.behaviours.NotifyMetersBehaviour;

/**
 *
 * @author Dulan Ranawaka
 */
public class DataEntryAgent extends Agent {

    private List<AID> _monitorAgents = new ArrayList<AID>();
    private Hashtable<String, String> _meters = new Hashtable<>();
    private DataEntryAgentForm _agentUiForm = null;
    @Override
    protected void setup() {
        _agentUiForm = new DataEntryAgentForm(this);
        _agentUiForm.setVisible(true);
        _agentUiForm.setAgentName(getAID().getName());

        showMessage("Starting behaviours...");

        addBehaviour(new MonitorDiscoveryBehaviour(this, 15000, _monitorAgents));
    }

    public boolean addMeter(String meterNumber, String meterIp) {
        if (_meters.containsKey(meterNumber) || _meters.containsValue(meterIp))
            return false;

        _meters.put(meterNumber, meterIp);
        addBehaviour(new NotifyMetersBehaviour(_monitorAgents, meterNumber,
            meterIp));
        return true;
    }

    public void showMessage(String message) {
        _agentUiForm.addMessage(message);
    }
}
```

```

}
package metermonitor.dataentryagent.agent;

import javax.swing.DefaultListModel;
import javax.swing.ListModel;

/**
 *
 * @author Dulan Ranawaka
 */
public class DataEntryAgentForm extends javax.swing.JFrame {

    private final DataEntryAgent _dataEntryAgent;

    /**
     * Creates new form DataEntryAgentForm
     * @param dataEntryAgent
     */
    public DataEntryAgentForm(DataEntryAgent dataEntryAgent) {
        this._dataEntryAgent = dataEntryAgent;

        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        txtMeterNumber = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        txtMeterIp = new javax.swing.JTextField();
        btnAdd = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        lstMessages = new javax.swing.JList();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        lstMeters = new javax.swing.JList();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```



University of Moratuwa, Sri Lanka.

Electronic Theses & Dissertations

www.lib.mru.ac.lk


```

        .addComponent(txtMeterIp,
javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel3)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addGap(18, 18, 18)
            .addComponent(txtMeterNumber,
javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGap(0, 0, Short.MAX_VALUE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
        .addGroup(layout.createSequentialGroup()
            .addGap(260, 260, 260)
            .addComponent(btnAdd,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 302,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel4))
        .addContainerGap()
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(jLabel4))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jScrollPane2,
            javax.swing.GroupLayout.PREFERRED_SIZE, 201,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel1)
    .addComponent(txtMeterNumber,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtMeterIp,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(btnAdd))
    .addComponent(jScrollPane1))
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);

pack();
} // </editor-fold>

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    if (!_dataEntryAgent.addMeter(txtMeterNumber.getText(),
txtMeterIp.getText())) return;

ListModel meterListModel = lstMeters.getModel();
if (meterListModel == null || !(meterListModel instanceof DefaultListModel))
{
    meterListModel = new DefaultListModel<>();
    lstMeters.setModel(meterListModel);
}
}

```

```

        ((DefaultListModel)meterListModel).addElement(String.join("@",
txtMeterNumber.getText(), txtMeterIp.getText()));

    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnAdd;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JList lstMessages;
    private javax.swing.JList lstMeters;
    private javax.swing.JTextField txtMeterIp;
    private javax.swing.JTextField txtMeterNumber;
    // End of variables declaration

    public void setAgentName(String agentName) {
        this.setTitle(agentName);
    }

    public void addMessage(String message) {
        ListModel meterIdListModel = lstMessages.getModel();
        if (meterIdListModel == null || (meterIdListModel instanceof
DefaultListModel))
        {
            meterIdListModel = new DefaultListModel<>();
            lstMessages.setModel(meterIdListModel);
        }
        if(((DefaultListModel)meterIdListModel).size()>10)
        {
            ((DefaultListModel)meterIdListModel).clear();
            ((DefaultListModel)meterIdListModel).removeAllElements();
        }
        ((DefaultListModel)meterIdListModel).addElement(message);
    }
}

```

```

package metermonitor.dataentryagent.behaviours;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.TickerBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAException;
import jade.lang.acl.ACLMessage;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import metermonitor.common.utilities.CommonConstants;
import metermonitor.dataentryagent.agent.DataEntryAgent;
import metermonitor.meterreadingagent.agent.MeterReadingAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class MonitorDiscoveryBehaviour extends TickerBehaviour {
    private List<AID> _monitorAgents;

    public MonitorDiscoveryBehaviour(Agent a, long period, List<AID>
monitorAgents) {
        super(a, period);

        _monitorAgents = monitorAgents;
    }

    @Override
    protected void onTick() {
        if(myAgent instanceof DataEntryAgent) {
            ((DataEntryAgent)myAgent).showMessage("Searching for monitors..");
        }

        ServiceDescription serviceDescription = new ServiceDescription();
        serviceDescription.setType(CommonConstants.MonitorServiceType);

        DFAgentDescription df = new DFAgentDescription();
        df.addServices(serviceDescription);

        try {
            DFAgentDescription[] monitors = DFService.search(myAgent, df);

            _monitorAgents.clear();

```



```
    for (int i = 0; i < monitors.length; i++) {
        _monitorAgents.add(monitors[i].getName());
    }
} catch (FIPAException ex) {
    Logger.getLogger(MeterReadingAgent.class.getName()).log(Level.SEVERE,
null, ex);
}
}
}
```



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```

package metermonitor.dataentryagent.behaviours;

import jade.core.AID;
import jade.core.behaviours.OneShotBehaviour;
import jade.lang.acl.ACLMessage;
import java.util.List;
import metermonitor.dataentryagent.agent.DataEntryAgent;

/**
 *
 * @author Dulan Ranawaka
 */
public class NotifyMetersBehaviour extends OneShotBehaviour {

    private List<AID> _monitorAgents;
    private String _meterNumber;
    private String _meterIp;

    public NotifyMetersBehaviour(List<AID> monitorAgents, String meterNumber,
String meterIp) {
        _monitorAgents = monitorAgents;
        _meterNumber = meterNumber;
        _meterIp = meterIp;
    }
    @Override
    public void action() {
        if(myAgent instanceof DataEntryAgent) {
            ((DataEntryAgent)myAgent).showMessage("Notifying meters...");
        }

        for (AID monitorAgent : _monitorAgents) {
            ACLMessage msg = new ACLMessage(ACLMessage.INFORM_REF);
            msg.addReceiver(monitorAgent);

            String messageContent = String.join("@", _meterNumber, _meterIp);
            msg.setContent(messageContent);
            myAgent.send(msg);
        }
    }
}

```



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Appendix E : DELAY TIME MEASUREMENTS

```
Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.35 Delay Time: 6307ms Msg byte:4417
Reply from Meter IP:10.217.5.53 Delay Time: 6330ms Msg byte:4087
Reply from Meter IP:10.217.5.13 Delay Time: 6319ms Msg byte:4089
Reply from Meter IP:10.217.5.52 Delay Time: 6328ms Msg byte:4198
Reply from Meter IP:10.217.5.61 Delay Time: 6303ms Msg byte:4130
Reply from Meter IP:10.217.5.67 Delay Time: 6306ms Msg byte:4415
Reply from Meter IP:10.217.5.57 Delay Time: 6353ms Msg byte:4446
Reply from Meter IP:10.217.5.23 Delay Time: 6330ms Msg byte:3983
Reply from Meter IP:10.217.5.59 Delay Time: 6326ms Msg byte:3891
Reply from Meter IP:10.217.5.2 Delay Time: 6335ms Msg byte:4415
Reply from Meter IP:10.217.5.62 Delay Time: 6332ms Msg byte:3921
Reply from Meter IP:10.217.5.73 Delay Time: 6327ms Msg byte:4212
Reply from Meter IP:10.217.5.34 Delay Time: 6311ms Msg byte:3942
Reply from Meter IP:10.217.5.13 Delay Time: 6331ms Msg byte:4139
Reply from Meter IP:10.217.5.41 Delay Time: 6340ms Msg byte:3891
Reply from Meter IP:10.217.5.29 Delay Time: 6315ms Msg byte:4235
Reply from Meter IP:10.217.5.96 Delay Time: 6324ms Msg byte:4371
Reply from Meter IP:10.217.5.4 Delay Time: 6349ms Msg byte:4053
Reply from Meter IP:10.217.5.54 Delay Time: 6334ms Msg byte:3903
Reply from Meter IP:10.217.5.61 Delay Time: 6333ms Msg byte:4401
Reply from Meter IP:10.217.5.5 Delay Time: 6348ms Msg byte:4263
Reply from Meter IP:10.217.5.36 Delay Time: 6309ms Msg byte:4183
Reply from Meter IP:10.217.5.58 Delay Time: 6357ms Msg byte:4291
Reply from Meter IP:10.217.5.93 Delay Time: 6304ms Msg byte:4006
Reply from Meter IP:10.217.5.35 Delay Time: 6326ms Msg byte:4003
```



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations

www.lib.mrt.ac.lk

```
Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.40 Delay Time: 4732ms Msg byte:3915
Reply from Meter IP:10.217.5.55 Delay Time: 4711ms Msg byte:4051
Reply from Meter IP:10.217.5.53 Delay Time: 4705ms Msg byte:4427
Reply from Meter IP:10.217.5.16 Delay Time: 4712ms Msg byte:4364
Reply from Meter IP:10.217.5.35 Delay Time: 4701ms Msg byte:4301
Reply from Meter IP:10.217.5.96 Delay Time: 4709ms Msg byte:4442
Reply from Meter IP:10.217.5.84 Delay Time: 4713ms Msg byte:4246
Reply from Meter IP:10.217.5.85 Delay Time: 4722ms Msg byte:4209
Reply from Meter IP:10.217.5.24 Delay Time: 4723ms Msg byte:3922
Reply from Meter IP:10.217.5.75 Delay Time: 4732ms Msg byte:4062
Reply from Meter IP:10.217.5.65 Delay Time: 4738ms Msg byte:4245
Reply from Meter IP:10.217.5.15 Delay Time: 4701ms Msg byte:4357
Reply from Meter IP:10.217.5.83 Delay Time: 4712ms Msg byte:4117
Reply from Meter IP:10.217.5.64 Delay Time: 4734ms Msg byte:4289
Reply from Meter IP:10.217.5.74 Delay Time: 4723ms Msg byte:4406
Reply from Meter IP:10.217.5.36 Delay Time: 4703ms Msg byte:4001
Reply from Meter IP:10.217.5.1 Delay Time: 4704ms Msg byte:4184
Reply from Meter IP:10.217.5.25 Delay Time: 4720ms Msg byte:4249
Reply from Meter IP:10.217.5.20 Delay Time: 4701ms Msg byte:4284
Reply from Meter IP:10.217.5.44 Delay Time: 4700ms Msg byte:4256
Reply from Meter IP:10.217.5.37 Delay Time: 4718ms Msg byte:3944
Reply from Meter IP:10.217.5.53 Delay Time: 4708ms Msg byte:4261
```

Delay Time Measurements when Two Meter Access Agent Running

```

Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.85 Delay Time: 3412ms Msg byte:3904
Reply from Meter IP:10.217.5.31 Delay Time: 3445ms Msg byte:4107
Reply from Meter IP:10.217.5.5 Delay Time: 3432ms Msg byte:3887
Reply from Meter IP:10.217.5.18 Delay Time: 3496ms Msg byte:3999
Reply from Meter IP:10.217.5.72 Delay Time: 3414ms Msg byte:4274
Reply from Meter IP:10.217.5.91 Delay Time: 3462ms Msg byte:3928
Reply from Meter IP:10.217.5.92 Delay Time: 3498ms Msg byte:4021
Reply from Meter IP:10.217.5.61 Delay Time: 3548ms Msg byte:4375
Reply from Meter IP:10.217.5.60 Delay Time: 3475ms Msg byte:3923
Reply from Meter IP:10.217.5.94 Delay Time: 3525ms Msg byte:4337
Reply from Meter IP:10.217.5.91 Delay Time: 3528ms Msg byte:4297
Reply from Meter IP:10.217.5.17 Delay Time: 3533ms Msg byte:4248
Reply from Meter IP:10.217.5.26 Delay Time: 3450ms Msg byte:4366
Reply from Meter IP:10.217.5.18 Delay Time: 3412ms Msg byte:4404
Reply from Meter IP:10.217.5.66 Delay Time: 3439ms Msg byte:4229
Reply from Meter IP:10.217.5.73 Delay Time: 3508ms Msg byte:3884
Reply from Meter IP:10.217.5.52 Delay Time: 3556ms Msg byte:4070
Reply from Meter IP:10.217.5.28 Delay Time: 3542ms Msg byte:4189
Reply from Meter IP:10.217.5.1 Delay Time: 3427ms Msg byte:3987
Reply from Meter IP:10.217.5.17 Delay Time: 3533ms Msg byte:4420
Reply from Meter IP:10.217.5.81 Delay Time: 3463ms Msg byte:4182
Reply from Meter IP:10.217.5.21 Delay Time: 3469ms Msg byte:4279
Reply from Meter IP:10.217.5.47 Delay Time: 3400ms Msg byte:3915

```

Delay Time Measurements when Three Meter Access Agent Running

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```

Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.77 Delay Time: 2572ms Msg byte:3919
Reply from Meter IP:10.217.5.55 Delay Time: 2416ms Msg byte:4253
Reply from Meter IP:10.217.5.28 Delay Time: 2531ms Msg byte:4080
Reply from Meter IP:10.217.5.7 Delay Time: 2591ms Msg byte:3971
Reply from Meter IP:10.217.5.36 Delay Time: 2491ms Msg byte:4047
Reply from Meter IP:10.217.5.12 Delay Time: 2580ms Msg byte:4351
Reply from Meter IP:10.217.5.39 Delay Time: 2436ms Msg byte:4255
Reply from Meter IP:10.217.5.20 Delay Time: 2435ms Msg byte:4360
Reply from Meter IP:10.217.5.14 Delay Time: 2590ms Msg byte:4436
Reply from Meter IP:10.217.5.59 Delay Time: 2447ms Msg byte:4362
Reply from Meter IP:10.217.5.94 Delay Time: 2552ms Msg byte:4149
Reply from Meter IP:10.217.5.78 Delay Time: 2524ms Msg byte:4405
Reply from Meter IP:10.217.5.40 Delay Time: 2502ms Msg byte:4339
Reply from Meter IP:10.217.5.60 Delay Time: 2501ms Msg byte:4000
Reply from Meter IP:10.217.5.15 Delay Time: 2487ms Msg byte:4297
Reply from Meter IP:10.217.5.70 Delay Time: 2506ms Msg byte:4104
Reply from Meter IP:10.217.5.9 Delay Time: 2544ms Msg byte:4080
Reply from Meter IP:10.217.5.75 Delay Time: 2488ms Msg byte:4321
Reply from Meter IP:10.217.5.45 Delay Time: 2553ms Msg byte:4097
Reply from Meter IP:10.217.5.67 Delay Time: 2464ms Msg byte:4163
Reply from Meter IP:10.217.5.41 Delay Time: 2460ms Msg byte:4009
Reply from Meter IP:10.217.5.92 Delay Time: 2499ms Msg byte:4341
Reply from Meter IP:10.217.5.85 Delay Time: 2501ms Msg byte:4363

```

Delay Time Measurements when Four Meter Access Agent Running

```
Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.95 Delay Time: 2108ms Msg byte:4241
Reply from Meter IP:10.217.5.50 Delay Time: 2142ms Msg byte:4092
Reply from Meter IP:10.217.5.98 Delay Time: 2129ms Msg byte:4160
Reply from Meter IP:10.217.5.30 Delay Time: 2103ms Msg byte:4121
Reply from Meter IP:10.217.5.42 Delay Time: 2114ms Msg byte:4255
Reply from Meter IP:10.217.5.58 Delay Time: 2153ms Msg byte:4120
Reply from Meter IP:10.217.5.21 Delay Time: 2124ms Msg byte:4180
Reply from Meter IP:10.217.5.77 Delay Time: 2121ms Msg byte:4204
Reply from Meter IP:10.217.5.63 Delay Time: 2142ms Msg byte:4245
Reply from Meter IP:10.217.5.88 Delay Time: 2145ms Msg byte:4148
Reply from Meter IP:10.217.5.57 Delay Time: 2122ms Msg byte:4225
Reply from Meter IP:10.217.5.73 Delay Time: 2155ms Msg byte:4291
Reply from Meter IP:10.217.5.94 Delay Time: 2143ms Msg byte:4254
Reply from Meter IP:10.217.5.84 Delay Time: 2110ms Msg byte:3943
Reply from Meter IP:10.217.5.0 Delay Time: 2116ms Msg byte:4251
Reply from Meter IP:10.217.5.26 Delay Time: 2115ms Msg byte:4290
Reply from Meter IP:10.217.5.73 Delay Time: 2130ms Msg byte:4216
Reply from Meter IP:10.217.5.5 Delay Time: 2151ms Msg byte:3910
Reply from Meter IP:10.217.5.93 Delay Time: 2140ms Msg byte:4282
Reply from Meter IP:10.217.5.34 Delay Time: 2153ms Msg byte:4120
Reply from Meter IP:10.217.5.88 Delay Time: 2151ms Msg byte:4005
Reply from Meter IP:10.217.5.68 Delay Time: 2113ms Msg byte:4164
Reply from Meter IP:10.217.5.83 Delay Time: 2136ms Msg byte:4162
```

Delay Time Measurements when Five Meter Access Agent Running



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```
Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.36 Delay Time: 1006ms Msg byte:3987
Reply from Meter IP:10.217.5.14 Delay Time: 958ms Msg byte:4395
Reply from Meter IP:10.217.5.75 Delay Time: 993ms Msg byte:4117
Reply from Meter IP:10.217.5.35 Delay Time: 952ms Msg byte:4298
Reply from Meter IP:10.217.5.32 Delay Time: 1019ms Msg byte:4008
Reply from Meter IP:10.217.5.33 Delay Time: 1035ms Msg byte:3932
Reply from Meter IP:10.217.5.65 Delay Time: 952ms Msg byte:4106
Reply from Meter IP:10.217.5.86 Delay Time: 951ms Msg byte:3951
Reply from Meter IP:10.217.5.86 Delay Time: 961ms Msg byte:4303
Reply from Meter IP:10.217.5.20 Delay Time: 1030ms Msg byte:4373
Reply from Meter IP:10.217.5.90 Delay Time: 1040ms Msg byte:4131
Reply from Meter IP:10.217.5.7 Delay Time: 977ms Msg byte:4246
Reply from Meter IP:10.217.5.23 Delay Time: 968ms Msg byte:3970
Reply from Meter IP:10.217.5.92 Delay Time: 1041ms Msg byte:4212
Reply from Meter IP:10.217.5.53 Delay Time: 1030ms Msg byte:4180
Reply from Meter IP:10.217.5.87 Delay Time: 974ms Msg byte:3908
Reply from Meter IP:10.217.5.21 Delay Time: 997ms Msg byte:4376
Reply from Meter IP:10.217.5.70 Delay Time: 1037ms Msg byte:4204
Reply from Meter IP:10.217.5.5 Delay Time: 1018ms Msg byte:4357
Reply from Meter IP:10.217.5.47 Delay Time: 1024ms Msg byte:3935
Reply from Meter IP:10.217.5.33 Delay Time: 1004ms Msg byte:4346
Reply from Meter IP:10.217.5.19 Delay Time: 959ms Msg byte:4299
```

Delay Time Measurements when Ten Meter Access Agent Running

```
Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.82 Delay Time: 763ms Msg byte:4178
Reply from Meter IP:10.217.5.83 Delay Time: 795ms Msg byte:4132
Reply from Meter IP:10.217.5.37 Delay Time: 794ms Msg byte:4339
Reply from Meter IP:10.217.5.89 Delay Time: 755ms Msg byte:4300
Reply from Meter IP:10.217.5.79 Delay Time: 781ms Msg byte:4026
Reply from Meter IP:10.217.5.17 Delay Time: 792ms Msg byte:4140
Reply from Meter IP:10.217.5.93 Delay Time: 812ms Msg byte:3976
Reply from Meter IP:10.217.5.2 Delay Time: 784ms Msg byte:4008
Reply from Meter IP:10.217.5.15 Delay Time: 823ms Msg byte:4253
Reply from Meter IP:10.217.5.85 Delay Time: 820ms Msg byte:4392
Reply from Meter IP:10.217.5.47 Delay Time: 784ms Msg byte:4402
Reply from Meter IP:10.217.5.75 Delay Time: 751ms Msg byte:4370
Reply from Meter IP:10.217.5.69 Delay Time: 771ms Msg byte:4078
Reply from Meter IP:10.217.5.14 Delay Time: 770ms Msg byte:4259
Reply from Meter IP:10.217.5.92 Delay Time: 757ms Msg byte:4198
Reply from Meter IP:10.217.5.38 Delay Time: 769ms Msg byte:3976
Reply from Meter IP:10.217.5.52 Delay Time: 825ms Msg byte:4055
Reply from Meter IP:10.217.5.22 Delay Time: 792ms Msg byte:3960
Reply from Meter IP:10.217.5.53 Delay Time: 798ms Msg byte:3988
Reply from Meter IP:10.217.5.91 Delay Time: 759ms Msg byte:4221
Reply from Meter IP:10.217.5.21 Delay Time: 801ms Msg byte:4156
Reply from Meter IP:10.217.5.30 Delay Time: 825ms Msg byte:4276
```

Delay Time Measurements when Fifteen Meter Access Agent Running

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mru.ac.lk

```
Monitor@10.217.5.252:1099/JADE
Reply from Meter IP:10.217.5.28 Delay Time: 644ms Msg byte:4087
Reply from Meter IP:10.217.5.86 Delay Time: 609ms Msg byte:4403
Reply from Meter IP:10.217.5.30 Delay Time: 634ms Msg byte:4230
Reply from Meter IP:10.217.5.44 Delay Time: 632ms Msg byte:4066
Reply from Meter IP:10.217.5.28 Delay Time: 642ms Msg byte:4008
Reply from Meter IP:10.217.5.74 Delay Time: 671ms Msg byte:3933
Reply from Meter IP:10.217.5.18 Delay Time: 636ms Msg byte:4318
Reply from Meter IP:10.217.5.16 Delay Time: 671ms Msg byte:4100
Reply from Meter IP:10.217.5.56 Delay Time: 627ms Msg byte:4293
Reply from Meter IP:10.217.5.64 Delay Time: 621ms Msg byte:4270
Reply from Meter IP:10.217.5.54 Delay Time: 600ms Msg byte:4025
Reply from Meter IP:10.217.5.87 Delay Time: 669ms Msg byte:4421
Reply from Meter IP:10.217.5.50 Delay Time: 678ms Msg byte:3927
Reply from Meter IP:10.217.5.55 Delay Time: 604ms Msg byte:4316
Reply from Meter IP:10.217.5.97 Delay Time: 661ms Msg byte:4385
Reply from Meter IP:10.217.5.43 Delay Time: 674ms Msg byte:4204
Reply from Meter IP:10.217.5.10 Delay Time: 633ms Msg byte:3977
Reply from Meter IP:10.217.5.72 Delay Time: 662ms Msg byte:3914
Reply from Meter IP:10.217.5.96 Delay Time: 678ms Msg byte:4350
Reply from Meter IP:10.217.5.71 Delay Time: 642ms Msg byte:4439
Reply from Meter IP:10.217.5.97 Delay Time: 632ms Msg byte:4159
Reply from Meter IP:10.217.5.70 Delay Time: 645ms Msg byte:4287
```

Delay Time Measurements when Twenty Meter Access Agent Running