References

[1] G. Zhang, B. Eddy Patuwo, and M. Y Hu, "Forecasting with artificial neural networks:: The state of the art," *Int. J. Forecast.*, vol. 14, no. 1, pp. 35–62, 1998.

[2] "Vidushi Sharma, Sachin Rai and Anurag Dev, 2012, A Comprehensive Study of Artificial Neural Networks, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 10, October 2012." .

[3] M. Dianati, I. Song, and M. Treiber, "An introduction to genetic algorithms and evolution strategies," Technical report, University of Waterloo, Ontario, N2L 3G1, Canada, 2002.

[4] E. J. Horvitz, J. S. Breese, and M. Henrion, "Decision theory in expert systems and artificial intelligence," *Int. J. Approx. Reason.*, vol. 2, no. 3, pp. 247–302, 1988.

[5] M. Ying and Z. Jianyong, "The Study of Multi-agent-based Logistics System of Special Items," *2nd Int. Conf. Comput. Appl. Syst. Model.*, 2012.

[6] "Engene Santos Jr, Feng Zhang, 2003, Intra Organizational Logistics Management Through Multi Agent System." .

[7] J. Jorge E. Hernández, Raúl Poler, Mula and D. de La Fuente, "A multiagent based-model for the collaborative planning process in decentralized supply chain networks," in *XIII Congreso de Ingeniería de Organización*, 2009, pp. 1012–1020.

[8] W. Unin, "A Study of Multi-Agent Based Supply Chain Modeling and Management," *iBusiness*, vol. 02, no. 04, pp. 333–341, 2010.

[9] S. F. Smith and N. M. Sadeh, "A Multiagent Approachi," *Decis. Sci.*, vol. 29, no. 3, 1998.

[10] "Jayashankar M. Swaminathan, Stephen F. Smith and Norman M. Sadeh, Modeling Supply Chain Dynamics: A Multiagent Approach." . - 1998 - A Multiagent Approachi.pdf." .

[11] S. Kumar and U. Kumar, "Java Agent Development Framework," *Int. J. Res.*, vol. 1, no. 9, pp. 1022–1025, 2014.

[12] "FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS, FIPA 97 Specification, Part 2, Agent Communication Language 'fipa_OC00018.pdf.'" .

[13] V. Kumar and S. Srinivasan, "A review of supply chain management using multi-agent system," *Int. J. Comput. Sci. Issues*, vol. 7, no. 5, pp. 198–205, 2010.

[14] "Agent based supply chain management automation," in *Proceedings of the Eighth International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII)*, 2000, vol. 1006.

[15] M. Ojha, "Optimizing supply chain management using gravitational search algorithm and multi agent system," in *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011*, 2012, pp. 481–491.

[16] X. Li and S. K. Lau, "A multi-agent approach towards collaborative supply chain management," 2005.

[17] "Thierry Moyaux, Brahim Chaib-draa, and Sophie D'Amours, 'Supply Chain Management and Multiagent Systems: An Overview', Springer Abstract, Volume 28, 2006, pp 1-27." .

[18] J. C. Fransoo and M. J. Wouters, "Measuring the bullwhip effect in the supply chain," *Supply Chain Manag. Int. J.*, vol. 5, no. 2, pp. 78–89, 2000.

[19] N. Julka, R. Srinivasan, and I. Karimi, "Agent-based supply chain management—1: framework," *Comput. Chem. Eng.*, vol. 26, no. 12, pp. 1755–1769, 2002.

[20] M. d'Inverno, D. Kinny, M. Luck, and M. Wooldridge, "A formal specification of dMARS," in *Intelligent Agents IV Agent Theories, Architectures, and Languages*, Springer, 1998, pp. 155–176.

[21] D. Pereira, E. Oliveira, N. Moreira, and L. Sarmento, "Towards an architecture for emotional BDI agents," in *EPIA*, 2005, vol. 5, pp. 40–47.

[22] X. Zhu, "Agent Based Modeling for Supply Chain Management: Examining the Impact of Information Sharing," Kent State University, 2008.

[23] R. Carvalho and L. Custódio, "A multiagent systems approach for managing supply-chain problems: new tools and results," *Intel. Artif.*, vol. 9, no. 25, pp. 79–88, 2005.

[24] C. I. Popirlan, A. Stefănescu, and L. Stefănescu, "Multi-Agent Approach for Data Analysis in a Knowledge-based System for Contact Centers," *World Acad. Sci. Eng. Technol.*, vol. 59, pp. 1126–1131, 2011.

[25] A. K. Chopra and M. P. Singh, "Elements of a business-level architecture for multiagent systems," in *Programming Multi-Agent Systems*, Springer, 2010, pp. 15–30.

[26] S. Srinivasan, S. K. Singh, and V. Kumar, "Multi-agent System based Service Oriented Architecture for Supply Chain Management System," *Int. J. Comput. Appl.*, vol. 27, no. 5, pp. 12–16, 2011.

[27] Haitham Al-zu'bi, (2010), Applying Electronic Supply Chain Management Using Multi-Agent System: A Managerial Perspective,International Arab Journal of e-Technology, Vol, 1, No. 3,.

[28] Y. Fu, R. Piplani, R. De Souza, and J. Wu, "Multi-agent enabled modeling and simulation towards collaborative inventory management in supply chains," in *Simulation Conference, 2000. Proceedings. Winter*, 2000, vol. 2, pp. 1763–1771.

[29] H. A. Rady, "Multi-Agent System for Negotiation in a Collaborative Supply Chain Management.," *Int. J. Video Image Process. Netw. Secur.*, vol. 11, no. 5, 2011.

[30] Y. Chen, Y. Peng, T. Finin, Y. Labrou, S. Cost, B. Chu, R. Sun, and R. Willhelm, "A negotiation-based multi-agent system for supply chain management," *Work. Notes Agents*, vol. 99, 1999.

[31] B. Hettige, A. S. Karunananda, and G. Rzevski, (2012), "Multi-agent System Technology for Morphological Analysis",Sri Lnka Association of Artificial Intelligence, (SLAAI).

[32] K. Psarris and World Scientific and Engineering Academy and Society, Eds., *Proceedings of the 6th WSEAS International Conference on Telecommunications and Informatics (TELE-INFO '07), 6th WSEAS International Conference on Signal Processing (SIP '07): Dallas, Texas, USA, March 22 - 24, 2007*. Athens: WSEAS Press, 2007.

[33] F. Wu, S. Zilberstein, and X. Chen, "Online planning for multi-agent systems with bounded communication," *Artif. Intell.*, vol. 175, no. 2, pp. 487–511, Feb. 2011.

[34] M. de Weerdt, A. ter Mors, and C. Witteveen, "Multi-agent Planning," *Dept Softw. Technol. Delft Univ. Technol.*, 2003.

[35] I. J. Chen and A. Paulraj, "Understanding supply chain management: critical research and a theoretical framework," *Int. J. Prod. Res.*, vol. 42, no. 1, pp. 131–163, Jan. 2004.

[36] G. P. Kurien and M. N. Qureshi, "Study of performance measurement practices in supply chain management," *Int. J. Bus. Manag. Soc. Sci.*, vol. 2, no. 4, pp. 19–34, 2011.

[37] R. P. Kampstra, J. Ashayeri, and J. L. Gattorna, "Realities of supply chain collaboration," *Int. J. Logist. Manag.*, vol. 17, no. 3, pp. 312–330, 2006.

[38] "George Rzevski, 08 New Directions in Searching for Artificial Intelligence.pdf." .

[39] N. Fabbe-Costes and M. Jahre, "Supply chain integration and performance: a review of the evidence," *Int. J. Logist. Manag.*, vol. 19, no. 2, pp. 130–154, Aug. 2008.

[40] R. H. Ballou, "Business logistics: importance and some research opportunities," *Gest. Produção*, vol. 4, no. 2, pp. 117–129, 1997.

[41] T. Blecker and Hamburg International Conference of Logistics, Eds., *Innovative methods in logistics and supply chain management: current issues and emerging practices*, 1. ed. Berlin: epubli, 2014.

[42] J. Storey, C. Emberson, J. Godsell, and A. Harrison, "Supply chain management: theory, practice and future challenges," *Int. J. Oper. Prod. Manag.*, vol. 26, no. 7, pp. 754–774, Jul. 2006.

[43] "Eugénio Oliveira, Klaus Fischer and Olga Stepankova, 1999, Multi-agent Systems: Which Research for which Applications." .

[44] F. L. Bellifemine, *Developing multi-agent systems with jade*. Hoboken, NJ: John Wiley, 2007.

[45] O. M. Shehory, *Architectural properties of multi-agent systems*. Carnegie Mellon University, The Robotics Institute, 1998.

[46] "Amit K. Chopra and Munindar P. Singh, 'Agent Communication', MIT Press, 2011, www.csc.ncsu.edu/faculty/mpsingh/papers/mas/Agent-Communication-chapter.pdf." .

[47] "Anand S. Rao and Michael P. Georgef, 'BDI Agents: From Theory to Practisce', Aprial 1995." .

[48] B. Anandampilai, "Content-Based Multicasting Using JADE," *Int. J. Soft Comput.*, vol. 2, no. 3, pp. 422–425, 2007.

[49] "NICHOLAS R. JENNINGS, MICHAEL WOOLDRIDGE, KATIA SYCARA, A Roadmap of Agent Research and Development, 1998 'roadmap.pdf.'" .

[50] K. P. Sycara, "Multiagent systems," *AI Mag.*, vol. 19, no. 2, p. 79, 1998.

[51] B. Marzougui and K. Barkaoui, "Interaction Protocols in Multi-Agent Systems based on Agent Petri Nets Model," *Interaction*, vol. 4, no. 7, 2013.

Appendix A

Essentials of Multi Agent technology

A.1   Introduction

This appendix consists of the features in multi-agent systems and also describes the difference between complex system and others.

CLASSES/ Features	RANDOM SYSTEMS	COMPLEX SYSTEMS	STABLE SYSTEMS	ALGORITHMIC SYSTEMS
Predictability	Total uncertainty	Considerable uncertainty	No uncertainty	No uncertainty
Behavior	Random	Emergent	Planned	Deterministic
Norms of behavior	Total freedom of behavior	Some external guidance is essential	Governed by laws and regulations	Follows instructions
Degree of organization	None	Self-organization	Organized	Rigidly structured
Degree of control	None	Self-control by self-organization	Centralized control	No need for control
Irreversible changes	Random changes	Co-evolves with environment	Small temporary deviations possible	None
Operating point	None	Operates far from equilibrium	Operates at an equilibrium	Operates according to the specification

Table   A.1: A Multi Agent System Features

Source: George Rzevski, a New Direction of Research into Artificial Intelligence

Design of MAS for SCM

B.1 Introduction

As said in design chapter, this appendix consist of various design diagrams to align with implementation stage.
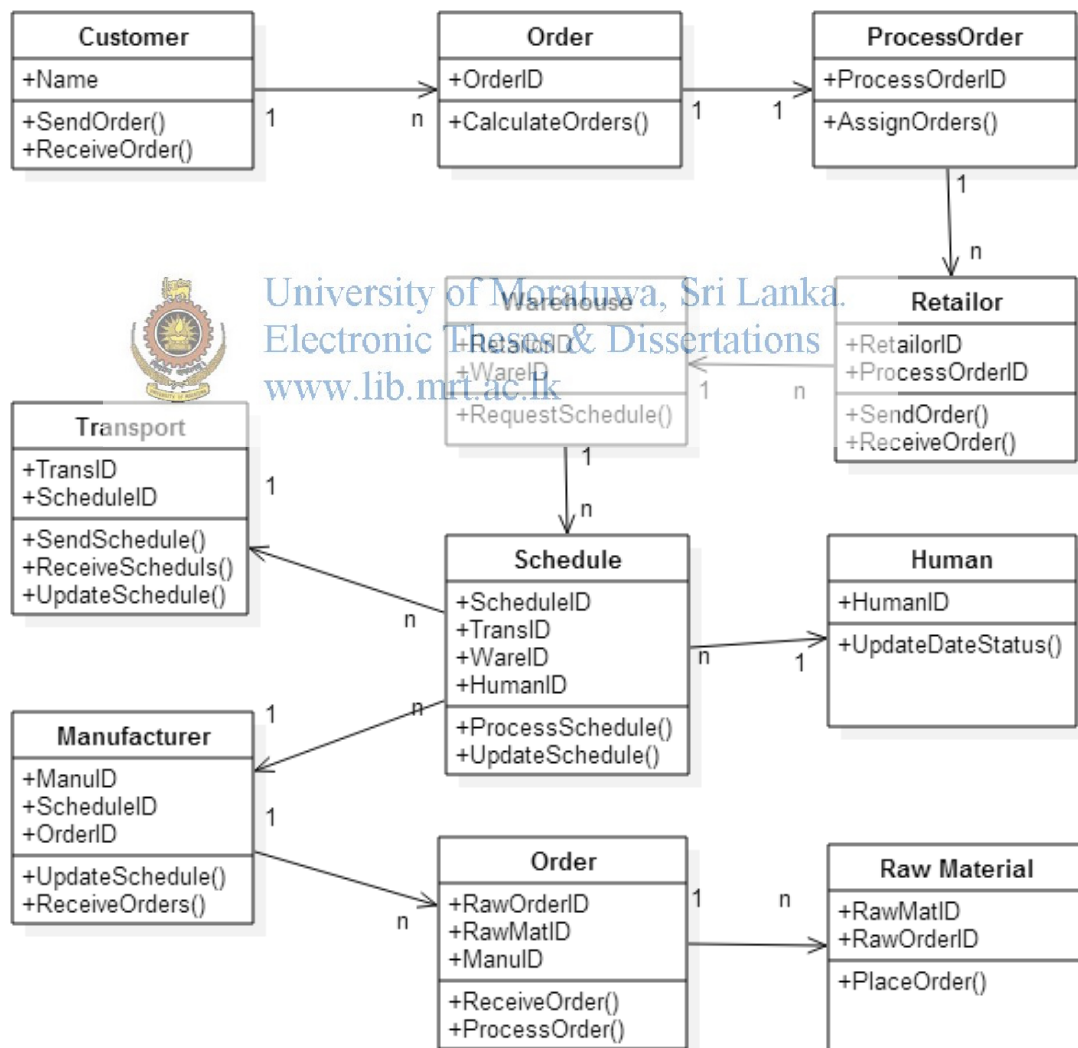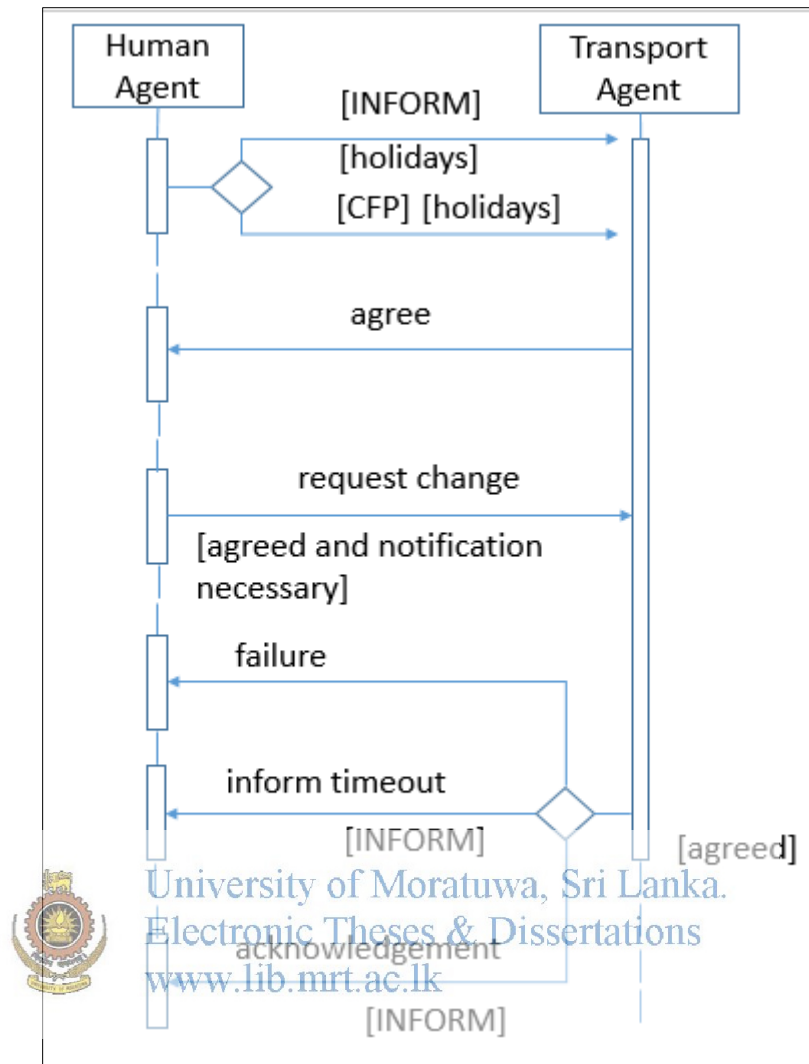

Figure B.1: Class Diagram for SCM
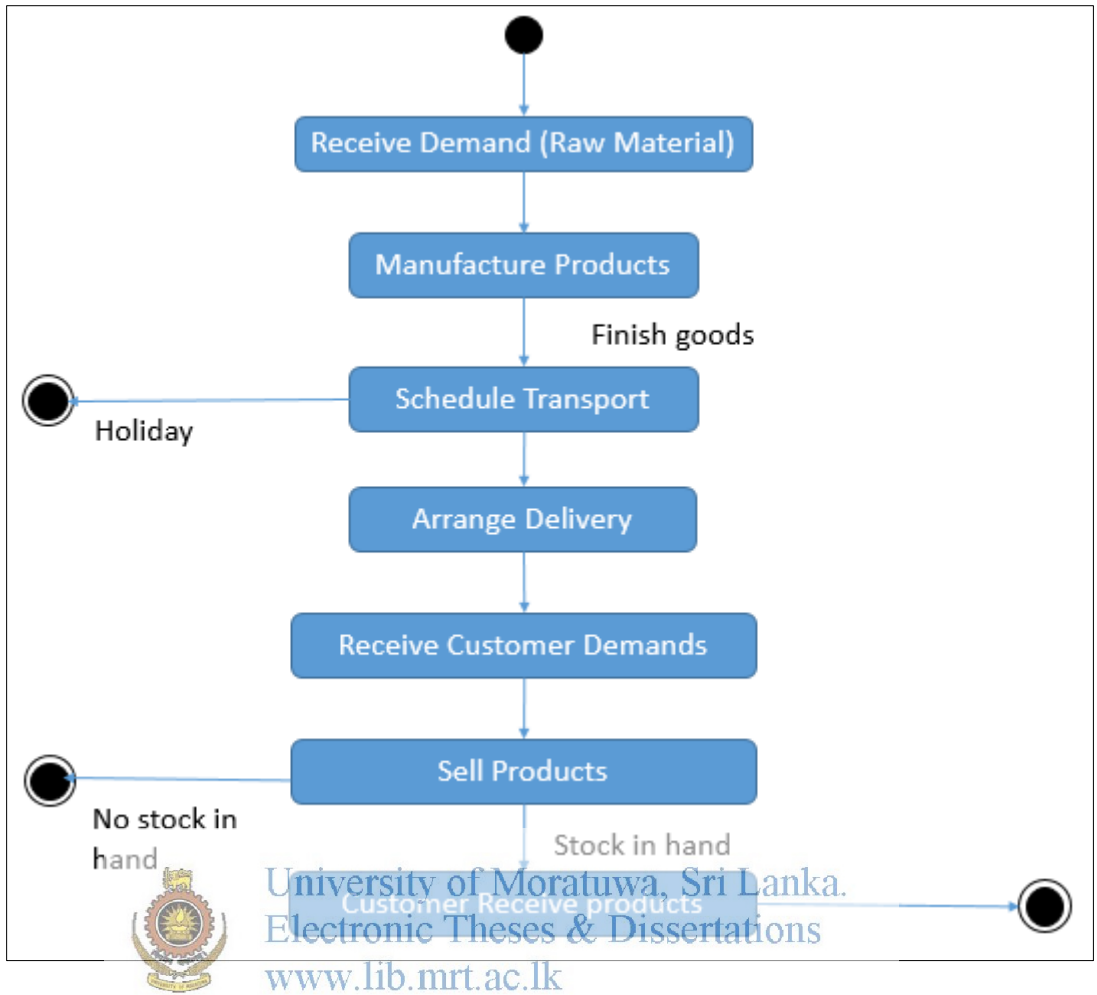
Figure B.2: Sequence Diagram for Human Agent Interaction

Figure B.3: State Transition Diagram

Implementation of the MAS Solution for SCM

C.1   Introduction

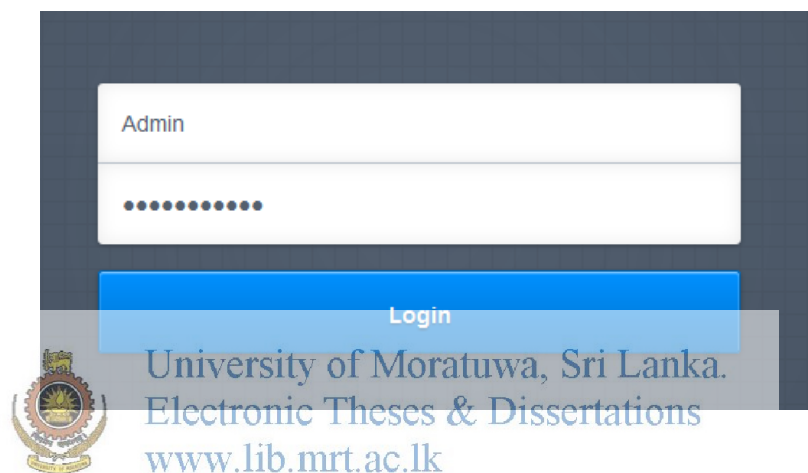During this appendix have included important code segments, figures and data have used in implementation stage.

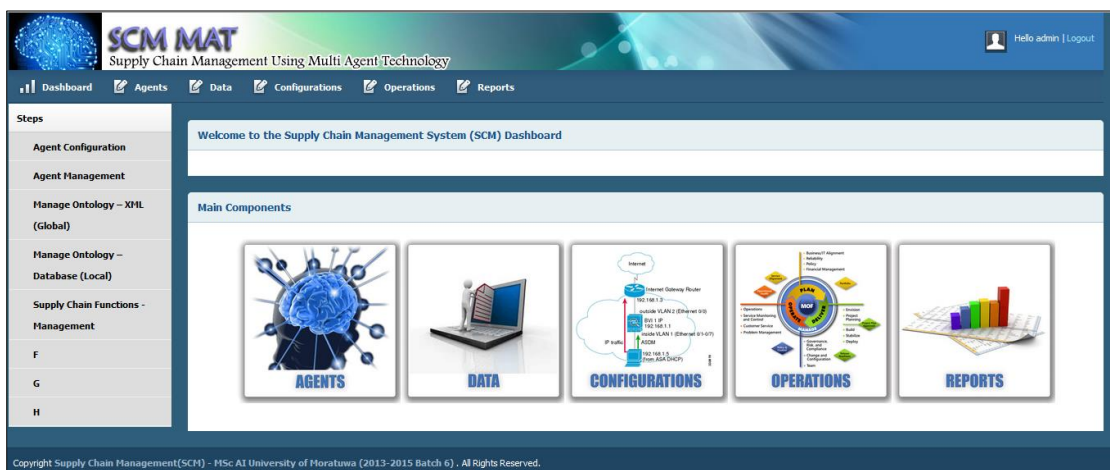Figure C.1: Login Screen of MASSCM

Figure C.2: Dashboard of MASSCM

Raw Material Agents Required Quantity Schedule - Update

Agent Name (AID)	R1	Ex: D1,D2
Quantity	3250	
	Update	

Figure C.3: Ontology Update form

			schedule_id	manu_aid	tr_aid	confirm_date	confirm_status
☐	✏	✗	1	M5	T2	2015-04-01	1
☐	✏	✗	2	M5	T2	2015-04-01	1
☐	✏	✗	3	M5	T2	2015-04-01	1
☐	✏	✗	4	M5	T2	2015-04-01	1
☐	✏	✗	5	M5	T2	2015-04-01	1
☐	✏	✗	6	M5	T2	2015-04-01	1
☐	✏	✗	7	M5	T2	2015-04-01	1
☐	✏	✗	8	M5	T2	2015-04-01	1
☐	✏	✗	9	M5	T2	2015-04-01	1
☐	✏	✗	10	M5	T2	2015-04-01	1

Figure C.4: Transport Schedule Data

Source: Link Natural Products (PVT) LTD

			tr_autu_id	tr_aid	tr_sdate	tr_edate	tr_time	tr_qty	read_status	manu_aid	tr_date_holiday
☐	✎	✗	1	T1	2015-04-01	2015-04-01	09:00:00	2000	1	M3	1
☐	✎	✗	2	T2	2015-04-01	2015-04-02	12:00:00	300	1	M5	0
☐	✎	✗	3	T3	2015-04-02	2015-04-02	08:30:00	4000	1	M2	0
☐	✎	✗	4	T1	2015-04-02	2015-04-02	06:30:00	200	1	M3	0
☐	✎	✗	5	T3	2015-04-01	2015-04-01	14:30:00	250	1	M2	0
☐	✎	✗	6	T2	2015-04-02	2015-04-02	23:00:00	500	1	M5	0
☐	✎	✗	7	T3	2015-04-01	2015-04-01	12:00:00	700	1	M2	0
☐	✎	✗	8	T1	2015-04-07	2015-04-08	06:30:00	500	1	M3	0
☐	✎	✗	9	T3	2015-04-06	2015-04-07	12:00:00	300	1	M2	0
☐	✎	✗	10	T2	2015-04-02	2015-04-02	06:30:00	2000	1	M5	0

Figure C.5: Transport Schedule Data (Confirmed by Manufacturers)

			tr_manu_autoid	tr_manu_aid	tr_manu_date	tr_manu_confirm
☐	✎	✗	1	M1	2015-04-01	0
☐	✎	✗	2	M2	2015-04-06	0
☐	✎	✗	3	M3	2015-04-02	0
☐	✎	✗	4	M4	2015-04-02	0
☐	✎	✗	5	M5	2015-04-01	0

Figure C.6: Transport Schedule of Manufacturers Agents

```
public class DataAccess {

    Statement stmt = null;

        // Select SQL query.

        public ResultSet SelectSql(String query) {

                ResultSet rs = null;
                Statement stmt = null;

                try {

                        DbCon useDbCon = new DbCon(); // Creating instance for a DbCon class.
                        Connection newCon = useDbCon.conn; // Accessing variable which used in DbCon class.

                        stmt = newCon.createStatement();
                        rs = stmt.executeQuery(query);

                }
                        catch (SQLException ex){
                        ex.printStackTrace();
                }
```

Figure C.7: Code for MYSQL Data Base Connection

```java
private ACLMessage msg;

public ACLMessage getMessage() {
    return msg;
}

public myReceiver(Agent a, int millis, MessageTemplate mt) {
    super(a);
    timeOut = millis;
    template = mt;
}

public void onStart() {
    wakeupTime = (timeOut < 0 ? Long.MAX_VALUE
            : System.currentTimeMillis() + timeOut);
}

public boolean done() {
    return finished;
}

public void action() {
    if (template == null) {
        msg = myAgent.receive();
    } else {
        msg = myAgent.receive(template);
    }
}
```

Figure C.8: Message Space Agent Code

```java
// T - Transport

myAgent.getContainerController().createNewAgent("T1","pk1.TransportAgent",null).start();
myAgent.getContainerController().createNewAgent("T2","pk1.TransportAgent",null).start();
myAgent.getContainerController().createNewAgent("T3","pk1.TransportAgent",null).start();

// W - Warehouse

myAgent.getContainerController().createNewAgent("W1","pk1.WarehouseAgent",null).start();
myAgent.getContainerController().createNewAgent("W2","pk1.WarehouseAgent",null).start();
myAgent.getContainerController().createNewAgent("W3","pk1.WarehouseAgent",null).start();

// H - Human Agent

myAgent.getContainerController().createNewAgent("H1","pk1.HumanAgent",null).start();
```

Figure C.9: Agent Initialization Code

```
///Access XML content
try {

    //File ontoXmlFile = new File("C://Users//Manoj Lap//Documents//NetBeansProjects//MAS_SCM
    File ontoXmlFile = new File("C://Users//Manoj Lap//Documents//NetBeansProjects//MAS_SCM//
    JAXBContext jaxbContext = JAXBContext.newInstance(pk2.ontology.AgentConfiguration.class);

    Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();
    mySelf = (AgentConfiguration) jaxbUnmarshaller.unmarshal(ontoXmlFile);
```

Figure C.10: Common Domain Ontology Access (XML Access)

```
for (int i = 0; i < agents.length; i++) {

    msg.addReceiver(/*new AID("M" + i, AID.ISLOCALNAME)*/agents[i].getName());
    par.addSubBehaviour(new myReceiver(this, 2000, template) {
        public void handle(ACLMessage msg) {
                                          // Ontology should goes here............

            if (msg != null) {
                int offer = Integer.parseInt(msg.getContent());

                System.out.println("Got quote Rs:" + offer
                        + " from " + msg.getSender().getLocalName());
                if (offer <= bestPrice) {
                    bestPrice = offer;
                    bestOffer = msg;

                }
            }
    });
}
```

Figure C.11: Sending Multicast Messages to Manufacturer Agents

```
seq.addSubBehaviour(new myReceiver(this, 2000,
        MessageTemplate.and(
                MessageTemplate.MatchConversationId(msg.getConversationId()),
                MessageTemplate.or(
                        MessageTemplate.MatchPerformative(ACLMessage.AGREE),
                        MessageTemplate.MatchPerformative(ACLMessage.REFUSE)))) {

            private static final long serialVersionUID = 1L;
```

Figure C.12: Agent Performatives Code

```
seq.addSubBehaviour(new DelayBehaviour(this, rnd.nextInt(3000)) {
    public void handleElapsedTimeout() {
        if (bestOffer == null) {
            System.out.println("Got no quotes");
        } else {
                                // Inser query.............

            String query = "INSERT INTO agents_manufacture"
                    + "(mf_best_price, mf_aid, mf_conid) VALUES "
                    + "('" + bestPrice + "','" + bestOffer.getSender().getLocalName() +
            try {
                da.InsertSql(query);
            } catch (SQLException ex) {
                ex.printStackTrace();
            }

            System.out.println("\nBest Price Rs:" + bestPrice
                    + " from " + bestOffer.getSender().getLocalName());
            ACLMessage reply = bestOffer.createReply();

                                // Ontology should goes here....................
            if (bestPrice <= 50) {
                reply.setPerformative(ACLMessage.REQUEST);
                reply.setContent("" + rnd.nextInt(80));
                System.out.print("ORDER at Rs:" + reply.getContent() + " ");
                send(reply);
            }
        }
    }
```

Figure C.13: Ontology Update and Send Reply to Manufacturers

```
addBehaviour(new MessageReceivingBehaviour());
// This code need to modify......................................................
MessageTemplate mt1 =
        MessageTemplate.and(
            MessageTemplate.MatchPerformative( ACLMessage.INFORM ),
            MessageTemplate.MatchSender( new AID( "T1",
                                            AID.ISLOCALNAME))) ;

MessageTemplate mt2 =
        MessageTemplate.and(
            MessageTemplate.MatchPerformative( ACLMessage.INFORM ),
            MessageTemplate.MatchSender( new AID( "T2",
                                            AID.ISLOCALNAME))) ;
MessageTemplate mt3 =
        MessageTemplate.and(
            MessageTemplate.MatchPerformative( ACLMessage.INFORM ),
            MessageTemplate.MatchSender( new AID( "T3",
                                            AID.ISLOCALNAME))) ;
```

Figure C.14: Message Template for to Receive Messages from Transport Agent

```
protected void onTick() {

    ACLMessage receiveFromT1 = receive(mtT1);
    if(receiveFromT1 != null){
        try {
            String maid1 = "";
            String traid1= "";
            Date agDate1 = null;
            try {
                //System.out.println(receiveFromT1.getContent()+" ||| "+receiveFromT1.getSend
                String sql1 = "SELECT DISTINCT manu_aid,tr_aid,confirm_date FROM agents_deliv
                rs = da.SelectSql(sql1);
                while(rs.next()){
                    maid1 = rs.getString("manu_aid");
                    traid1 = rs.getString("tr_aid");
                    agDate1 = rs.getDate("confirm_date");
                }   } catch (SQLException ex) {
                    ex.printStackTrace();
                }

            // print
            System.out.println("Transport agent "+traid1+" has agreed with manufacturer agent

            // Arrange ware date and trans date are equals

            String conSql1 = "SELECT ware_aid FROM agents_delivery_ware WHERE ware_aid='"+myA
            rs = da.SelectSql(conSql1);
            boolean val1 = rs.next();
            if(val1 == false){
            System.out.println("No confirmation among manufacturer, transport and warehouse a
```

Figure C.15: Warehouse Agent Ontology Management

```
String genCID() {
    if (cidBase == null) {
        cidBase = getLocalName() + hashCode()
                + System.currentTimeMillis() % 10000 + "_";
    }
    return cidBase + (cidCnt++);
}


ACLMessage newMsg(int perf, String content, AID dest) {
    ACLMessage msg = newMsg(perf);
    if (dest != null) {
        msg.addReceiver(dest);
    }
    msg.setContent(content);
    return msg;
}
```

Figure C.16: Warehouse Agent Utility Methods

```
@Override
protected void setup() { // Setup start
    super.setup(); //To change body of generated methods, choose Tools | Template
    addBehaviour(new TickerBehavior(this, 20000));
    addBehaviour(new delayBehaviorOnTrans (this, 30000));


    SequentialBehaviour seq = new SequentialBehaviour();
    addBehaviour(seq);

    ParallelBehaviour par = new ParallelBehaviour(ParallelBehaviour.WHEN_ALL);
    seq.addSubBehaviour(par);


    // This behavior is created to send agreed dates from transport agent and ma

    seq.addSubBehaviour(new DelayBehaviour(this, 90000){
```

Figure C.17: Transport Agent Different Behaviors

```
// Creat data base connection to obtain holidays...........

query = "SELECT DISTINCT tr_sdate FROM agents_transport_schedule_manu WHERE tr_date_holiday='1'";
String sdata = "";
rs = da.SelectSql(query);
    try {
        while(rs.next()){
            holiday = rs.getDate("tr_sdate");

            String dateUpdate = "INSERT INTO agents_human_holiday (haid,h_holi_dates) VALUES "
                    + "('"+myAgent.getLocalName()+"','"+holiday+"')";
            da.InsertSql(dateUpdate);


            sdata = sdata+holiday+", ";

        }   } catch (SQLException ex) {
        ex.printStackTrace();
    }

    msg.setContent("Hi I am human agent\n, please reschedule the delivery confirmed on "+sdata+" due

    //System.out.println(sdata);
    for(int i = 1; i<=3; i++){
    msg.addReceiver(new AID("T"+i, AID.ISLOCALNAME));
    send(msg);
    removeBehaviour(this);
}
```

Figure C.18: Human Agent Interference

```
// To w1

    ACLMessage toW1 = new ACLMessage(ACLMessage.INFORM);
    toW1.setContent("Need product X with quantity 1000");
    toW1.addReceiver(new AID("W1", AID.ISLOCALNAME));
    send(toW1);
    System.out.println(toW1+" --- "+toW1.getSender().getLocalName());
// To w2

    ACLMessage toW2 = new ACLMessage(ACLMessage.INFORM);
    toW2.setContent("Need product Y with quantity 2000");
    toW2.addReceiver(new AID("W2", AID.ISLOCALNAME));
    send(toW2);
    System.out.println(toW2+" --- "+toW2.getSender().getLocalName());
// To w3

    ACLMessage toW3 = new ACLMessage(ACLMessage.INFORM);
    toW3.setContent("Need product Z with quantity 3000");
    toW3.addReceiver(new AID("W3", AID.ISLOCALNAME));
    send(toW3);
    System.out.println(toW3+" --- "+toW3.getSender().getLocalName());
}
```

Figure C.19: Retailor and Warehouse Agents Negotiation

```
ParallelBehaviour par = new ParallelBehaviour(ParallelBehaviour.WHEN_ALL);
seq.addSubBehaviour(par);

seq.addSubBehaviour(new DelayBehaviour(this, 100050){
public void handleElapsedTimeout(){

    // To w1

    ACLMessage toRe1 = new ACLMessage(ACLMessage.INFORM);
    toRe1.setContent("Need products......!");
    toRe1.addReceiver(new AID("RE1", AID.ISLOCALNAME));
    send(toRe1);
    System.out.println(toRe1+" --- "+toRe1.getSender().getLocalName());

    // To w2

    ACLMessage toRe2 = new ACLMessage(ACLMessage.INFORM);
    toRe2.setContent("Need products......!");
    toRe2.addReceiver(new AID("RE2", AID.ISLOCALNAME));
    send(toRe2);
    System.out.println(toRe2+" --- "+toRe2.getSender().getLocalName());

}
});
```

Figure C.20: Customer and Retailer Negotiation