

References

- [1] Duchastel, P. C. (1990). Cognitive design for instructional design. "Instructional Science", 19(6), pp.437-444. (EJ 423 422)
- [2] Fabio, B., Giovanni, C., dominic, G, 'Developing Multi-Agent Systems with JADE', Telecom Italia,Italy, Whitestein Technologies AG, Switzerland.
- [3] Fan Zhu,Aple,W.P.F, Cathy, L.R. (2008), ENGOnto: Integrated multiple English Learning Ontology for personalized Education, International Conferences on Computer Science and Software Engineering
- [4] Feber, J. (1999), Multi Agent Systems, Addison-Wesley, Reading, MA
- [5] Floyed, A.W., David, W.J., Pat, O. (2009), Is a Curriculum Management System in Your Future?,[www.allofe.com/gen/corp_generated_bin/documents/basic_module/ cms-in-your-future.pdf](http://www.allofe.com/gen/corp_generated_bin/documents/basic_module/cms-in-your-future.pdf)
- [6] Kasowitz, Abby, 'Tools for Automating Instructional Design'.Journal of ERIC Digest. Available at <http://www.ericdigests.org/1999-1/tools.html>
- [7] Locatis, C. & Park, O. (1992), Some uneasy inquiries into ID expert systems. ETR&D, 40(3), pp.87-94
- [8] Magid Nikraz1a, Giovanni Caireb, and Parisa A. Bahria, A Methodology for the Analysis and Design of Multi-Agent Systems using JADE, School of Engineering Science and Parker Center, Murdoch University, Australia
- [9] Marco, R.,Joseph, S.(2007), Curriculum Management and Review: an Ontology Based Solution, In proceedings of World Conference on E-learning in Coporate,Govermant,Helthcare and Higher Education, pp.6476-6482

- [10] Mark Britton et Al(2008), A Curriculum Review and mapping process Supported by an Electronic Database System, American Journal of Pharmaceutical Education, 72(5), pp.99
- [11] Milheim, W. (1997), Instructional design issues for electronic performance support systems. British Journal of Educational Technology, 28(2), pp.103-110
- [12] Musen, M.A, (1992), Dimensions of knowledge sharing and reuse, Computers and biomedical research 25: pp 435-467
- [13] Pavi,P., Juha,P. (2003), CUBER: A Personalized Curriculum Builder, In proceeding of the 3rd IEEE International Conference on Advanced learning Technologies
- [14] Prideaux BMJ. (2003), ABC of learning and teaching in medicine: Curriculum Design, pp 326, pp.268-270
- [15] Prof Karunananda A.S., (2009), Course Materials of Artificial Intelligence in bachelor of Information Technology.
- [16] Russell, D. M. & Pirolli, P. (1992), Computer assisted instructional design for computer-based instruction, Final Report, Working Papers, Berkeley, CA: National Center for Research in Vocational Education. (ED 354 872)
- [17] Vijayakumar Rajarajan (2006), Automating Curriculum Management Tasks, A Manuscript Style Capstone Project, College of Science and Health, India.
- [18] Wenhuan Lu et Al (2009), Ontology Aware Course Management for Curriculum Evolution Process in Higher Education, http://www.jaist.ac.jp/joint-workshop/verite/Wenhuan%20Lu_coe.pdf.
- [19] Xiang Fu, Boris,P., Cathy, L.R,Computer Aided Curriculum Planning and Sceduling, available at <http://acronyms.thefreedictionaty.com/>

- [20] [http://en.wikipedia.org/wiki/Ontology_\(Information_science\)](http://en.wikipedia.org/wiki/Ontology_(Information_science))
- [21] [http:// jade.tilab.com](http://jade.tilab.com)
- [22] The Protégé-2000 users Guide documents,
http://protege.stanford.edu/doc/users_guid/
- [23] The Protégé Ontology Editor and Knowledge Acquisition System,
<http://protge.stanford.edu>
- [24] The Multi Agent Systems, <http://www.cs.cmu.edu/softwareagents/multi.html>



Appendix A

Design Diagrams

Operations of agents in this project have been designed using AUML collaboration diagrams showing partners of interaction with the operation. The dotted lines represent life-lines of agents. The arrow shows the interactions and the packages show the applied agent interaction protocol. Figure A1 shows the operation of module agent, Figure A2 shows the operation of credit agent, Figure A3 shows the operation of level agent, Figure A4 shows the operation of design agent and Figure A5 shows the operation of prerequisites agent

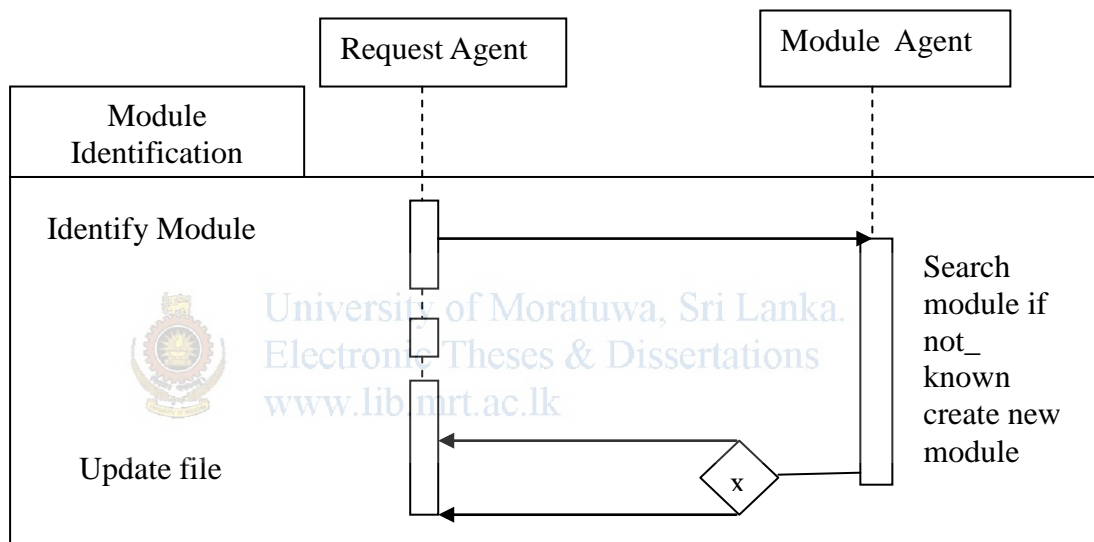


Figure A.1: Operation on Module Agent

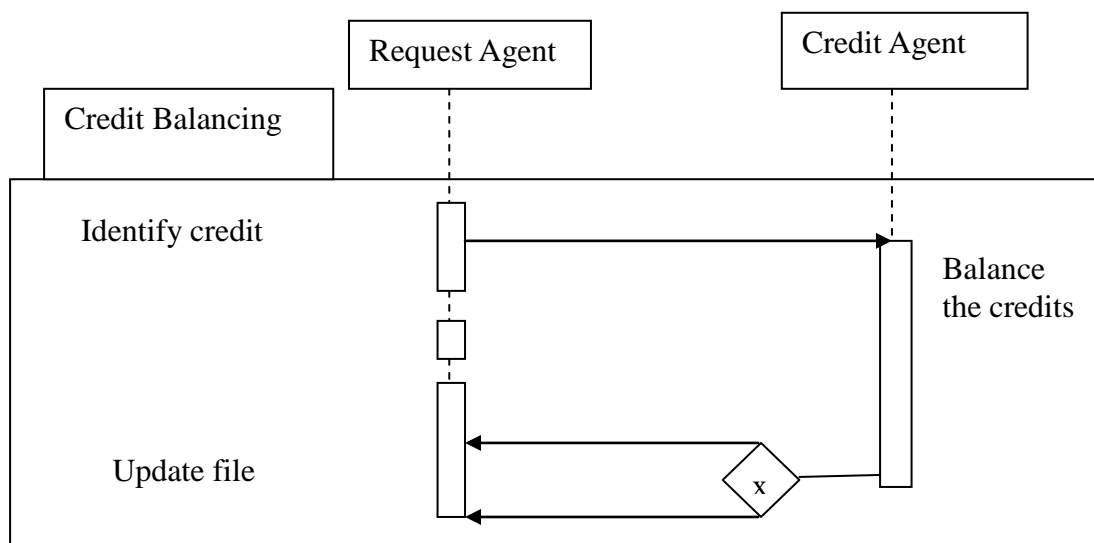


Figure A.2: Operation on Credit Agent

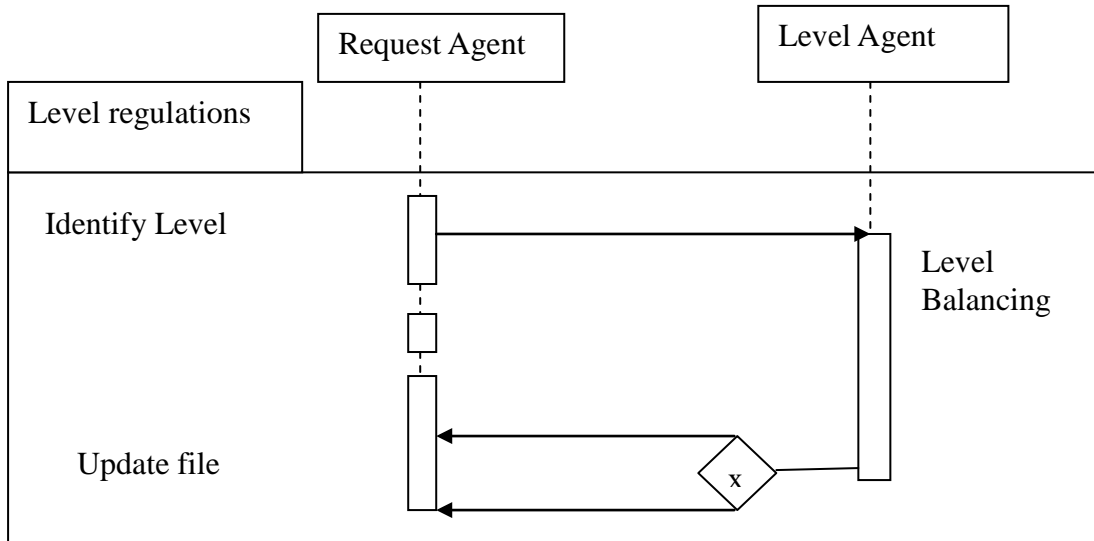


Figure A.3: Operation on Level Agent

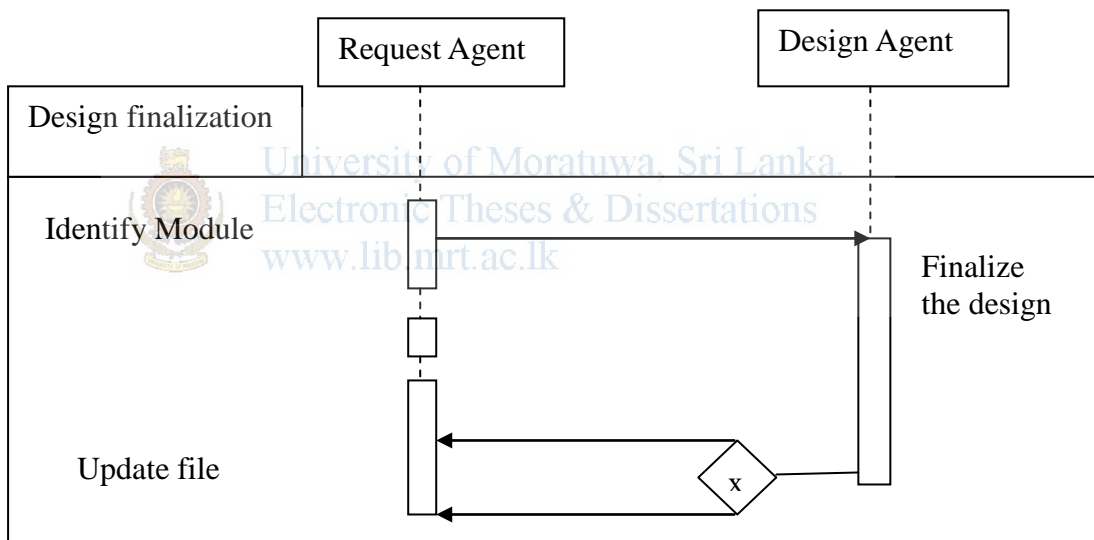


Figure A.4: Operation on Design Agent

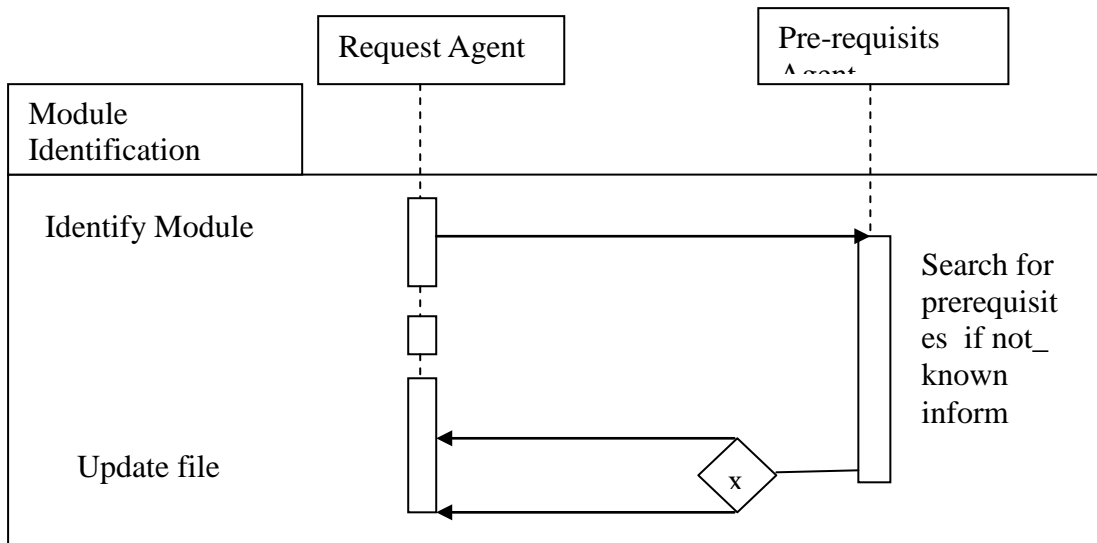


Figure A.5: Operation on Prerequisites Agent



Curriculum Ontology

This appendix' will show some classes created from ontology.

1. CurriculumOntology class

```
package ontology;

import jade.content.onto.*;
import jade.content.schema.*;

public class CurriculumOntology extends Ontology implements
CurriculumVocabulary {

// -----> The name identifying this ontology
public static final String ONTOLOGY_NAME = "Curriculum-Ontology";

// -----> The singleton instance of this ontology
private static Ontology instance = new CurriculumOntology();

// -----> Method to access the singleton ontology object
public static Ontology getInstance() { return instance; }

// Private constructor
private CurriculumOntology() {
super(ONTOLOGY_NAME, BasicOntology.getInstance());

try {

// ----- Add Concepts

        ConceptSchema cs = new ConceptSchema(MODULE);
        add(cs, Module.class);

        cs.add(MODULE_CURRICULUM , (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
        cs.add(MODULE_LEVEL, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
        cs.add(MODULE_SUBCODE, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
        cs.add(MODULE_TITLE, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
        cs.add(MODULE_LECTURE, (PrimitiveSchema)
getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);
        cs.add(MODULE_LAB, (PrimitiveSchema)
getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);
        cs.add(MODULE_CREDIT, (PrimitiveSchema)
getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);

        ConceptSchema cs1 = new ConceptSchema(OUTLINE);
        add(cs1, Outline.class);
        cs1.add(OUTLINE_CODE , (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
        cs1.add(OUTLINE_OUTLINE, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
```

```

    ConceptSchema cs2 = new ConceptSchema(NEWOUTLINE);
    add(cs2,NewOutline.class);
    cs2.add(NEWOUTLINE_PRE , (PrimitiveSchema)
getSchema(BasicOntology.STRING),0, ObjectSchema.UNLIMITED);
    cs2.add(NEWOUTLINE_OUTLINE, (PrimitiveSchema)
getSchema(BasicOntology.STRING),0, ObjectSchema.UNLIMITED);
    cs2.add(NEWOUTLINE_EQU, (PrimitiveSchema)
getSchema(BasicOntology.STRING),0, ObjectSchema.UNLIMITED);

// ----- Add AgentActions

// Agent Action - CreditBalance
AgentActionSchema as;

add( as = new AgentActionSchema(CREDIT_BALANCE),
CreditBalance.class);
as.add(CREDIT_BALANCE_CURRICULUM, (PrimitiveSchema) getSchema
(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(CREDIT_BALANCE_LEVEL, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(CREDIT_BALANCE_SUBCODE, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(CREDIT_BALANCE_Lecture, (PrimitiveSchema)
getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);
as.add(CREDIT_BALANCE_LAB, (PrimitiveSchema)
getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);
as.add(CREDIT_BALANCE_CREDIT, (PrimitiveSchema)
getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);
as.add(CREDIT_BALANCE_TYPE, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(CREDIT_BALANCE_GPA, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);

add( as = new AgentActionSchema(LEVEL_BALANCE), LevelBalance.class);
as.add(LEVEL_BALANCE_CURRICULUM, (PrimitiveSchema) getSchema
(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(LEVEL_BALANCE_LEVEL, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(LEVEL_BALANCE_SUBCODE, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(LEVEL_BALANCE_TYPE, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(LEVEL_BALANCE_GPA, (PrimitiveSchema)
getSchema(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(LEVEL_BALANCE_TCREDIT, (PrimitiveSchema)
getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);

add( as = new AgentActionSchema(NEWCREDIT), NewCredit.class);
as.add(NEWCREDIT_CREDIT, (PrimitiveSchema) getSchema
(BasicOntology.FLOAT), ObjectSchema.MANDATORY);

add( as = new AgentActionSchema(PREREQ), PreReq.class);
as.add(PREREQ_SUBCODE, (PrimitiveSchema) getSchema
(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(PREREQ_OUTLINE, (PrimitiveSchema) getSchema
(BasicOntology.STRING),0, ObjectSchema.UNLIMITED);

add( as = new AgentActionSchema(DELOPERATION), DelOperation.class);

```



```

as.add(DELOPERATION_OUTLINE, (PrimitiveSchema) getSchema
(BasicOntology.STRING), ObjectSchema.MANDATORY);

add( as = new AgentActionSchema(DESOPERATION), DesOperation.class);

as.add(DESOPERATION_LEARNOB, (PrimitiveSchema) getSchema
(BasicOntology.STRING), ObjectSchema.MANDATORY);
as.add(DESOPERATION_CURRICULUM, (PrimitiveSchema) getSchema
(BasicOntology.STRING), ObjectSchema.MANDATORY);
}

catch (OntologyException oe) {
oe.printStackTrace();
}
}
} // BankOntology

```

2. Curriculum Vocabulary Class

```

package ontology;

public interface CurriculumVocabulary
{

public static final String CREDIT_AGENT="Credit Agent";
public static final String LEVEL_AGENT="Level Agent";
public static final String DESIGN_AGENT="Design Agent";
public static final String SUBJECT_AGENT="Subjet Agent";
public static final String MESSAGE_AGENT="Message Agent";
public static final String PREREQ_AGENT="PreReq Agent";
public static final String REQ_AGENT="Req Agent";
//public static final String REQUEST_AGENT="Request Agent";

        public static final String CREDIT_BALANCE = "creditBalance";
        public static final String CREDIT_BALANCE_CURRICULUM =
"curriculum";
        public static final String CREDIT_BALANCE_LEVEL = "level";
        public static final String CREDIT_BALANCE_SUBCODE =
"subcode";
        public static final String CREDIT_BALANCE_LECTURE =
"lecture";
        public static final String CREDIT_BALANCE_LAB = "lab";
        public static final String CREDIT_BALANCE_CREDIT = "credit";
        public static final String CREDIT_BALANCE_TYPE = "type";
        public static final String CREDIT_BALANCE_GPA = "GPA";

        public static final String LEVEL_BALANCE = "LevelBalance";
        public static final String LEVEL_BALANCE_CURRICULUM =
"curriculum";
        public static final String LEVEL_BALANCE_SUBCODE = "subcode";
        public static final String LEVEL_BALANCE_LEVEL = "level";
        public static final String LEVEL_BALANCE_TYPE = "type";
        public static final String LEVEL_BALANCE_GPA = "GPA";
        public static final String LEVEL_BALANCE_TCREDIT = "Tcredit";

        public static final String MODULE = "Module";

```

```

public static final String MODULE_CURRICULUM = "curriculum";
public static final String MODULE_LEVEL = "level";
public static final String MODULE_TITLE= "title";
public static final String MODULE_SUBCODE = "subcode";
public static final String MODULE_LECTURE = "lecture";
public static final String MODULE_LAB = "lab";
public static final String MODULE_CREDIT = "credit";

public static final String NEWCREDIT = "Newcredit";
public static final String NEWCREDIT_CREDIT = "credit";

public static final String OUTLINE = "Outline";
public static final String OUTLINE_CODE = "code";
public static final String OUTLINE_OUTLINE= "outline";

public static final String PREREQ = "PreReq";
public static final String PREREQ_SUBCODE = "subcode";
public static final String PREREQ_OUTLINE= "outline";

public static final String DELOPERATION = "Deloperation";
public static final String DELOPERATION_OUTLINE= "outline";

public static final String DESOPERATION = "DesOperation";
public static final String DESOPERATION_LEARNOB= "learnOB";
public static final String DESOPERATION_CURRICULUM=
"curriculum";

public static final String NEWOUTLINE = "NewOutline";
public static final String NEWOUTLINE_PRE = "pre";
public static final String NEWOUTLINE_OUTLINE= "outline";
public static final String NEWOUTLINE_EQU= "equ";
}

```

3. Module Class

```

package ontology;

import jade.content.*;

public class Module implements Concept {
// -----

private String curriculum;
private String level;
private String subcode;
private float lab;
private float lecture;
private float credit;

public String getCurriculum() {
return curriculum;
}

public String getLevel() {
return level;
}

public String getSubcode() {

```

```

        return subcode;
    }

    public float getLecture() {
        return lecture;
    }

    public float getLab() {
        return lab;
    }

    public float getCredit() {
        return credit;
    }

    public void setCurriculum(String curriculum) {
        this.curriculum = curriculum;
    }

    public void setLevel(String level) {
        this.level = level;
    }

    public void setSubcode(String subcode) {
        this.subcode = subcode;
    }

    public void setLecture(float lecture) {
        this.lecture = lecture;
    }
    public void setlab(float lab) {
        this.lab = lab;
    }

    public void setCredit(float credit) {
        this.credit = credit;
    }
}

```

4. Problem class

```

package ontology;

import jade.content.*;

public class Problem implements Concept{
// -----

    private int num;
    private String msg;

    public int getNum() {
        return this.num;
    }

    public String getMsg() {
        return this.msg;
    }
}

```

```

    }

    public void setNum(int num) {
        this.num = num;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }
}

```

5. Information Class

```
package ontology;
```

```
import jade.content.*;
```

```

public class Information implements AgentAction {
// -----

    //private int type;
    private String curriculum;

    //public int getType() {
    //    return type;
    //}

    public String getCurriculum() {
        return curriculum;
    }

    //public void setType(int type) {
    //    this.type = type;
    //}

    public void setCurriculum(String curriculum) {
        this.curriculum = curriculum;
    }
}

```

6. CreditBalance Class.

```
package ontology;
```

```
import jade.content.*;
```

```

public class CreditBalance implements AgentAction{

    private String curriculum;
    private String level;
    private String subcode;
    private float lab;
    private float lecture;
    private float credit;
    private String type;
    private String GPA;

    public String getCurriculum() {

```

```
        return curriculum;
    }

    public String getLevel() {
        return level;
    }

    public String getSubcode() {
        return subcode;
    }

    public float getLecture() {
        return lecture;
    }

    public float getLab() {
        return lab;
    }

    public float getCredit() {
        return credit;
    }

    public String gettype() {
        return type;
    }
}

public String getGPA() {
    return GPA;
}

public void setCurriculum(String curriculum) {
    this.curriculum = curriculum;
}

public void setLevel(String level) {
    this.level = level;
}

public void setSubcode(String subcode) {
    this.subcode = subcode;
}
```



7. Level Balance Class

```
package ontology;
import jade.content.*;
public class LevelBalance implements AgentAction{
    private String curriculum;
    private String level;
    private String type;
    private String GPA;
    private String Subcode;

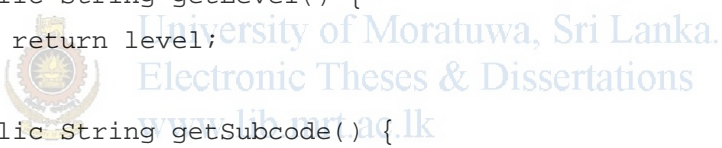
    private float Tcredit;

    public String getCurriculum() {
        return curriculum;
    }

    public String getLevel() {
        return level;
    }
    public String getSubcode() {
        return Subcode;
    }
    public String gettype() {
        return type;
    }

    public String getGPA() {
        return GPA;
    }

    public float getTCredit() {
        return Tcredit;
    }
}
```



8. Outline Class

```
package ontology;
```

```
import jade.content.Concept;
```

```
public class Outline implements Concept{
```

```
    private String subcode;
```

```
    private String[] outline;
```

```
    public String getSubcode() {
```

```
        return subcode;
```

```
    }
```

```
    public String[] getoutline() {
```

```
        return outline;
```

```
    }
```

```
    public void setSubcode(String subcode) {
```

```
        this.subcode = subcode;
```

```
    }
```

```
    public void setoutline(String[] outline) {
```

```
        this.outline = outline;
```

```
    }
```

```
    }
```



Appendix C

The Questionnaire

This section includes Questionnaire which has been given to the people who involve for the testing of the curriculum design system.

Name :-

Designation: -

Curriculum design process is more time consuming and many parties are involving for curriculum designing. Conventional computing aims at developing programs to do specific task without been able to improve the performance. Therefore conventional software for curriculum design does not provide facilities to automate the design process to meet dynamically changing requirement in a complex environment. Also modern software for curriculum design partially addresses the above issues.

Therefore this project aims to provide automated system to handle the issue in designing curriculum among the negotiation within different parties with use of Multi-Agent technology.

General Details.

1. Do you ever involve in a design/review process of curriculum?

- Yes
 No

If the answer is Yes, answer for the below questions, if not please ignore the below questions and stop answering.

2. Your experience in designing/reviewing curriculums,

- High
 Moderate
 Avarage

Technology details

3. Your rate of the knowledge of Multi Agent Technology,

- High
- Moderate
- Avarage
- low

4. Do you think that use of Multi Agent Technology for designing curriculumms is better idea than other technologies,

- Yes
- No

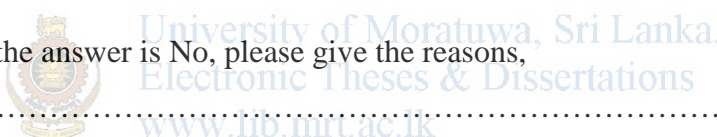
5. If the answer is No, please give the reasons,

.....

.....

.....

.....



Design

6. Is the final design of the system is user friendly?

- Yes
- No

7. Do you think that the final design of the user interface should be improved?

- Yes
- No

8. If yes, what should be improved?

.....
.....
.....
.....

Implementation

9. Do you satisfied with the rate of credit balancing

- Yes
 No

10. If Yes, Satisfactory of credit balancing is

- High
 Moderate
 Avarage
- 

11. Do you satisfied with the way of prerequisite modules checking

- Yes
 No

12. If Yes, Satisfactory of prerequisites modules checking is

- High
 Moderate
 Avarage

13. Do you satisfied with the way of equivalent module checking

- Yes
 No

14. If Yes, Satisfactory of equivalent module checking is

- High
- Moderate
- Avarage

15. The support in content (learning outcome) editing is

- High
- Moderate
- Avarage

Remarks

.....

.....

.....

.....

