Design and Implementation of Agent Interaction Mechanisms for Emotionally Intelligent Tutoring Systems

Sintija Petrovica[1], Alla Anohina-Naumeca[2], [1-2]*Riga Technical University*

***Abstract* – Modern intelligent systems are built from intelligent agents, but usually agent interaction mechanisms are described at the general level omitting design and implementation details. The paper presents aspects of practical implementation of agent interaction mechanisms in JADE platform on the basis of the set of agents for modelling student's emotions in intelligent tutoring systems. The concepts of an intelligent and affective tutoring system, agent, and agent interaction are explained, as well.**

***Keywords* – Affective tutoring system, agent interaction, intelligent tutoring system, multi-agent system**

I. Introduction

Nowadays the new generation of students, who do not know the world without information and communication technology (ICT), has appeared. They use technology in their private, social, and professional life. Therefore, it is only natural that ICT is penetrating more intensively into the educational process. As a result, the well-known traditional learning environment, which implies simultaneous existence and live contact between the teacher and the student in some space and time, is replaced by lifeless machines. That is why research of the last several decades is directed towards humanization of educational technologies not only from the point of view of their operation at the level of the human-teacher, but also from the point of view of their visual appearance. One type of such systems is emotionally intelligent tutoring systems (EITS) that are capable not only to implement all constituent parts of the learning process but also to adapt them to an emotional state of the student and to show emotions (by means of an animated pedagogical agent) in reaction to student's actions in the environment. Taking into account that the modern approach in the field of artificial intelligence is related to an agent paradigm, there is a trend to build EITS, as other intelligent systems, from autonomous purposeful entities called agents. Regardless of the fact that there is a quite mature theory about communication and interaction of agents, and a lot of systems are built as multi-agent systems, in the majority of information sources the agent interaction is shown at the general level without specification of design and description of practical implementations. This research is devoted to aspects of practical implementation of agent interaction mechanisms in JADE platform. As a basis for the research, a set of agents developed in 2008 for the modelling of student's emotions in intelligent tutoring systems (ITS) is used.

The structure of the paper is as follows. Section II explains the concept of an intelligent tutoring system and an affective tutoring system. Section III gives general information about agents, their interaction and usage for development of intelligent tutoring systems. The next section describes the previously developed set of agents for the modelling of student's emotions. Section V provides the detailed design of agent interaction mechanisms. Section VI describes the implementation aspects of the developed interaction mechanisms. Section VII is devoted to experimental testing of the mechanisms. Conclusions are given at the end of the paper.

II. Intelligent and Affective Tutoring Systems

Intelligent tutoring systems are adaptive computer systems, which are grounded on the theory of learning and cognition; they emulate a human-teacher and provide benefits of one-on-one tutoring. They are called intelligent systems because their architecture and operation is typically based on the principles and methods of artificial intelligence. Such systems are of especial significance in our technological time because they come closer to implementation of a more natural learning process by adapting a learning environment (content, feedback, navigation, etc.) to the characteristics of a particular student. Adaptation is possible because of a student diagnosis module that collects and processes information about the student (his/her learning progress, problem solving behaviour, psychological characteristics, etc.) and of a student model that stores this information. Additionally to the mentioned two components, the student diagnosis module and the student model, the architecture of an intelligent tutoring system includes:

- a pedagogical module responsible for implementation of the teaching process and a pedagogical model storing teaching model and strategies;
- an expert module able to generate and solve problems in the problem domain and an expert model storing knowledge what must be taught to the student;
- a communication module managing interaction among the system and the student through different devices.

Over the past few decades, research in neuroscience and psychology [1] – [3] has shown that emotions are fundamental to learning because they have an effect on perception, attention, motivation to learn, as well as acquisition, creation, and retrieval of knowledge. As a result, the field of affective (or emotionally intelligent) tutoring systems has started to

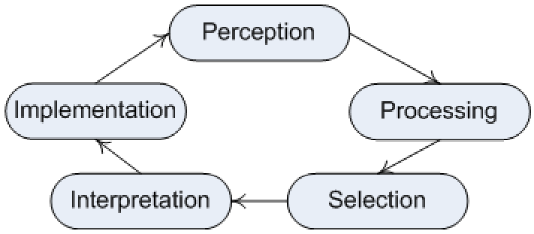

Fig. 1. The affective loop of an intelligent tutoring system

evolve by integrating into the traditional intelligent tutoring system ability to recognize student's emotions and to respond to them in an appropriate way. The modelling of student's emotions provides two main advantages in tutoring systems [4]:

- it allows adaptation of the learning process and environment not only to cognitive characteristics of the student, but also to his/her affective state;
- it allows inducing of necessary emotions to the student in order to promote more effective learning.

The system becomes more flexible, adaptive, and natural in terms of similarity to the traditional learning environment.

According to [5], an affective computational system must have a few of the following capacities: recognize, express, or possess emotions. An emotionally intelligent tutoring system [6], in turn, is defined as an intelligent tutoring system, which includes functional capabilities able to:

1. know student's emotions, and
2. induce emotions to the student in order to improve his/her performance.

As a result, such a system needs to achieve the following conditions: know the current emotional state of the student, determine the impact of an action on the student's emotional state, and select the most advantageous emotional state of the student to enhance his/her performance.

The notion of the emotionally intelligent tutoring system can be extended by an affective loop (Figure 1), which involves the tutor (the system) and the student [4]. First of all, the system perceives the current emotional state of the student through sensors from one or several sources, for example, face and/or speech. After that, features relevant to the determination of the student's emotion, for example, face features (shape and displacement of eyes, eyebrows, etc.) and/or speech features (pitch and duration of the utterance, etc.), are extracted from the information received from the sensors. Further, the features are classified with intent to obtain one of the pre-defined emotional categories. The identified emotional category is interpreted in terms of its influence on the learning process and further student's activities in the learning environment. At the next step, the

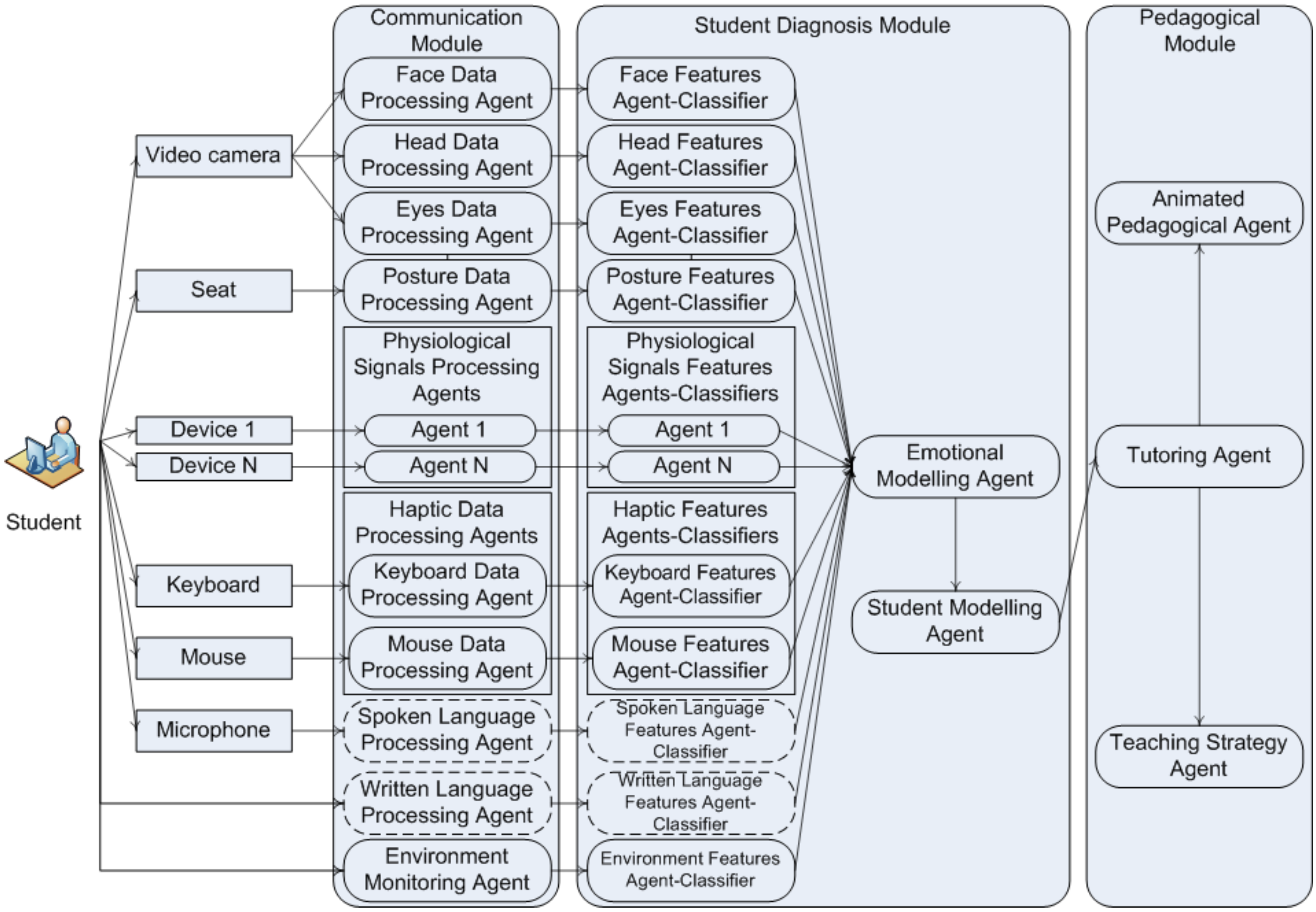

Fig.2. The set of agents for the modelling of student's emotions in intelligent tutoring systems

TABLE 1

INPUT CATEGORIES FOR THE DETERMINATION OF A STUDENT'S EMOTIONAL STATE

Category	Sub-category (if any)	Some examples of features
Educational environment	Student's actions within the environment and their characteristics	Difficulty of the performed action, student reaction time, navigation in the application, absence of an action, action unrelated to the educational application, chosen options, help request or refuse
	Events within the environment	End of the test, types of made mistakes, correct/incorrect answer/solution, usefulness of help
	Characteristics	Level of difficulty of a task
Face and head	Face	Shape and displacement of eyes, eyebrows, mouth, mouth corners, lips
	Eyes	Pupil size, blink rate and duration, saccadic data, gaze duration and fixations, eyelid's degree of openness
	Head	Likelihood of nod, likelihood of shake
Spoken language	Content	Speaks using exclamations, speaks words from a specific list of words
	Characteristics	Pitch, duration and intensity of the utterance, silence, phonetic features
Posture	-	Net pressure, prior and post change, reference change, net coverage
Physiological signals	-	Electromyography, finger temperature, finger pulse amplitude, heart rate, blood pressure, respiration activation
Haptic cues	Usage of the keyboard	Typing speed, usage of BkSp or Delete keys, hitting of unrelated keys
	Usage of the mouse	Mouse pressure
Written language	Content	Speech act
	Characteristics	Length of an answer, verbosity

system selects a pedagogical action (or actions), which can be directed towards one of two aspects: to change the emotional state of the student, for example, frustration or boredom, if its influence on the learning process is considered negative, or to maintain the emotional state of the student, such as enthusiasm or confidence, if it has a positive effect on the learning process. Moreover, an emotion, which the system (more precisely an animated pedagogical agent resided in the system) will express, must be selected taking into account the current emotional state of the student, the chosen pedagogical action, and the goal of system's influence on the student. After that, the animated pedagogical agent expresses its emotion and implements the pedagogical action. The loop repeats.

Regardless of the fact that research in the field of affective tutoring systems is at its active phase, at the moment only few successful implementations are known, for example, Easy with Eve [7], Edu-Affe-Mikey [8], AutoTutor [9].

III. AGENTS AND THEIR USAGE IN THE DEVELOPMENT OF INTELLIGENT TUTORING SYSTEMS

The development of agent-based systems is the modern approach in the field of artificial intelligence. The general and widely accepted definition of an agent was provided by Russell and Norvig: "an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators" [10]. Typically, agents are understood as entities that possess the following capabilities: autonomy, adaptive behaviour, reactivity, abilities to plan, predict, reason, learn, and operate in complex dynamic environments. The common practice is to combine multiple such entities in a single system in such a way forming a multi-agent system. There agents communicate using some language and try to achieve goals that a single agent would not be able to achieve in isolation.

It is considered that interaction and communication are defining characteristics of multi-agent systems. Usually, the exchange of information among agents is known as communication [11]. In turn, interaction in multi-agent systems is the ongoing exchange of data involving more than two agents. During interactions, agents are in contact with each other directly, through another agent, or through the environment [12]. One of the ways to initiate and maintain interaction among agents is their participation in communication that enables agents to base their decisions on more complete knowledge of overall situation [13].

Effective interaction and interoperation among agents requires three fundamental and distinct components [14]: a common language; a common understanding of the knowledge exchanged; the ability to exchange whatever is included in the first two components.

In 1995, the Foundation for Intelligent Physical Agents (FIPA) began its work on developing standards that were focused on agent communication language (ACL) for agent systems [15].

The integration of multiple agents produces an easy maintainable, extensible, and robust modular architecture of a system because of the following advantages [16], [17]: agents' ability to solve complex problems in which data, expertise, or control is distributed; computational efficiency related to computational parallelism; reliability due to availability of redundant agents with the same capabilities or dynamic changing of the set of cooperating agents; extensibility because agents can be easy added, changed, or removed, etc.

Taking into account agent properties, multi-agent architecture is especially appropriate for development of intelligent tutoring systems due to the following reasons:

- the system must plan the learning process and dialogue with the student;
- intelligent tutoring systems perform multiple and very different tasks, inter alia monitoring of and reacting to student's behaviour in the learning environment, student's knowledge assessment, choosing of learning material and tasks, provision of feedback and help, adaptation of teaching strategies, etc.
- functioning conditions of the system are changing with each student's action in the learning environment and the system must demonstrate reactive behaviour;
- the system collects information about the student and must consider his/her cognitive, psychological, and affective characteristics in order to adapt the learning process;
- the system architecture is composed of several constituent parts. Each of them has a set of quite independent functions. However, all parts must interact in order to achieve the common goal.

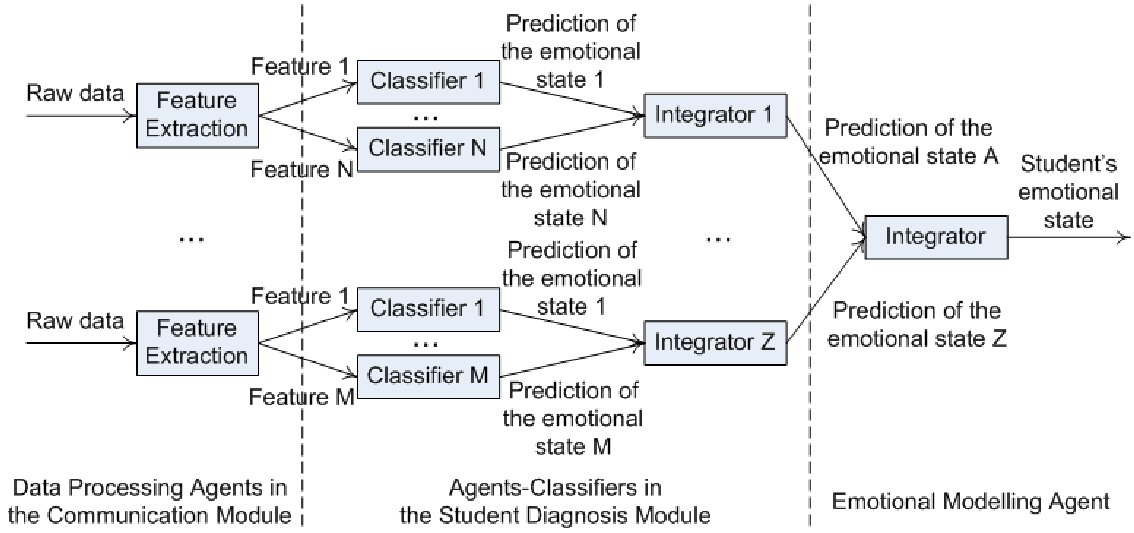

Fig.3. The decision making process concerning the student's emotional state

IV. Agents for Detection of Student's Emotions

In 2008, a set of agents for the modelling of student's emotions in intelligent tutoring systems was developed by the researchers of the Department of System Theory and Design of Riga Technical University (Figure 2). The set includes a number of agents [4] for the two main modules of the traditional architecture of an intelligent tutoring system: the communication module and the student diagnosis module.

The main task of the agents in the communication module is to receive raw data from sensors such as video camera, posture seat, microphone, etc. and to extract from them features which are used for the determination of the student's emotional state. According to Table 1, features can be very different and multiple. Moreover, they can be received from a single sensor (features of face, head, eyes, posture, and spoken language) or a set of sensors. For example, such features of the face as a shape and displacement of eyes, eyebrows and mouth are extracted from video of only one camera. In turn, in order to receive data about various physiological signals different devices are used for different signals. For this reason multiple Physiological Signal Processing Agents are included in the set. Each of them manages a particular device which simultaneously can be used for capturing of several signals. Two agents, the Spoken Language Processing Agent and the Written Language Processing Agent, are displayed by a dotted line because they are necessary and can be implemented only in the case if the system provides interaction in the form of natural language dialogue. Other agents can be implemented in any intelligent tutoring system. The Written Language Processing Agent and the Environment Monitoring Agent do not have external devices (sensors) attached to them. The first of them captures dialogue in a special field of the system's user interface. The second agent manages interface tools such as buttons, menus, input fields, panels, etc.

The agents of the communication module after the extraction of necessary features send them to the agents-classifiers of the student diagnosis module. The latter carry out the following functions:

- receipt of the extracted features;
- classification of the features;
- integration of results of particular classifiers in order to get more exact prediction of the student's emotional state;
- storing of the results both of the classification and the integration;
- transferring of the predicted emotional state to the Emotional Modelling Agent.

The Emotional Modelling Agent located in the student diagnosis module makes a final decision about the student's emotional state by receiving the predictions of the student's emotion from the agents-classifiers and by integrating them.

Figure 3 displays the overall process of decision making. The structure of each group of agents in terms of input, output, and processing is shown. The decision making process corresponds to the decision on level fusion of data. The process shown in Figure 3 is used as a basis for design and implementation of agent interaction mechanisms by making only minor changes in this process. The initial part of the process, where the raw data is received and feature extraction is done, is replaced with a text file which already contains these extracted features (Section VII). This is done because real tests require different devices–sensors from which the Data Processing Agents would receive raw data.

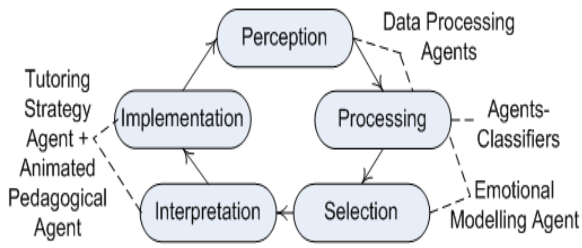

Fig.4. Involvement of the agents into the affective loop

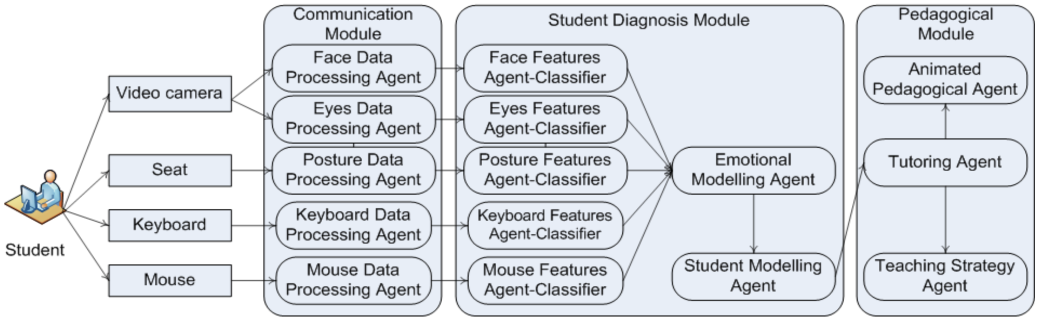

Fig. 5. The set of agents used for design of interaction mechanisms

The determined student's emotional state alongside with the information from other agents of the student diagnosis module is sent to the Student Modelling Agent located in the pedagogical module (Figure 2). The Student Modelling Agent transfers the received information to the Tutoring Agent, which activates different agents in the pedagogical module. In relation to the student's emotional state, the Teaching Strategy Agent chooses the suitable teaching strategy, but the Animated Pedagogical Agent, in turn, selects and expresses an emotional state. The common goal of both mentioned agents is to maintain or to change the emotional state of the student.

The involvement of the agents in the previously described affective loop (Section II) is shown in Figure 4.

V. DESIGN OF AGENT INTERACTION MECHANISMS

Currently, the most part of agent development methodologies are based on Unified Modelling Language (UML) developed by Object Management Group (OMG). As the main reason for UML usage is mentioned that starting over with a new modelling language for agents would be neither useful nor productive. Instead, multi-agent systems would benefit from an improvement of existing trusted methods. Agent UML or AUML provides such a solution. AUML first appeared in 1999 as an attempt to bring together research on agent-based software methodologies and emerging standards for object-oriented software development [18], [19].

This research is based on the adapted version of the previously developed set of agents for the modelling of student's emotions in ITS (Figure 2). The set includes only those agents that working together would be able to identify a student's emotional state quite precisely and could be implemented in any intelligent tutoring system (Figure 5).

AUML sequence diagrams are used for a detailed design of interaction mechanisms among agents, because in multi-agent systems these diagrams represent the exchange of messages using protocols. Protocols show sequences of allowed messages among agents [18]. FIPA ACL is used for message representation in sequence diagrams. During research, interaction mechanisms among agents were designed for the modelling of student's emotions based on his/her face data, eyes data, posture data, mouse, and keyboard usage features. In this paper the determination of the student's emotional state based on student's facial features such as shape of eyes and eyebrows, as well as shape of mouth and displacement of lip corners is shown and described in detail (Figure 6), because all other mentioned mechanisms can be developed in a similar manner.

The designed sequence diagram shows that the Emotional Modelling Agent sends a request to the Facial Feature Agent-Integrator to send a prediction of an emotional state after integrating all predictions of the emotional state from agents-classifiers. Data about a student, whose emotion will be determined, are sent also in this request. Then the Facial Feature Agent-Integrator sends a request to the Agent-Classifier by Eyes and Eyebrows to send its determined student's emotional state by shape of eyes and eyebrows. Next, the Agent-Classifier by Eyes and Eyebrows sends a request to the Face Data Processing Agent to send eye and eyebrow features. The Face Data Processing Agent sends the required data or failure message if failed to extract them from video received from a sensor – video camera. Then the Agent-Classifier by Eyes and Eyebrows on the basis of the received eye and eyebrow features determines prediction of an emotional state. If classification of these features is successful, the agent sends the result to the Facial Feature Agent-Integrator. Otherwise, the failure message about unsuccessful classification is sent. After that the Facial Feature Agent-Integrator sends a request to the Agent-Classifier by Mouth and Lip Corners to send its determined student's emotional state by shape of mouth and displacement of lip corners. Then the Agent-Classifier by Mouth and Lip Corners sends a request to the Face Data Processing Agent to send mouth and lip features. The Face Data Processing Agent sends the required mouth and lip features or failure message, if failed to extract them. Then the Agent-Classifier by Mouth and Lip Corners on the basis of the received mouth and lip features determines prediction of an emotional state. If classification of these features is successful, then the agent sends a prediction to the Facial Features Agent-Integrator. Otherwise, the failure message about unsuccessful classification is sent. Next, the Facial Feature Agent-Integrator combines the received predictions of emotional states, thus, getting the student's final prediction of an emotional state from facial features and sends it to the Emotional Modelling Agent. If it was not possible to combine the received predictions of emotional states, then the Facial Feature Agent-Integrator sends a failure message to the Emotional Modelling Agent.

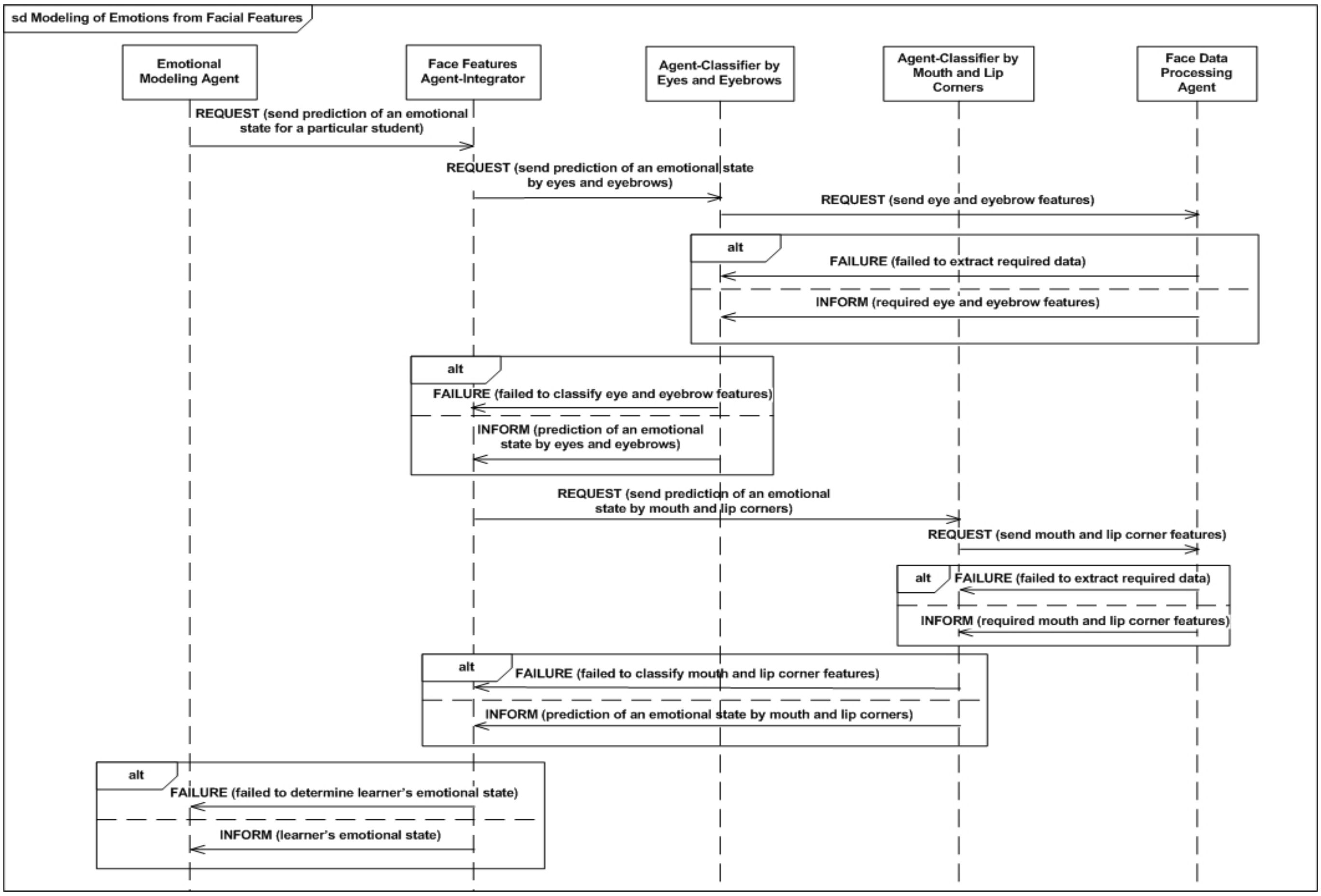

Fig. 6. Modelling of emotions from facial features

VI. Implementation of Agent Interaction Mechanisms

Due to the fact, that for practical implementation of the previously developed agent interaction mechanisms agent communication and FIPA standards (FIPA ACL communication language is used) play an important role, JADE agent platform is chosen for implementation. Currently, JADE is one of the most used FIPA-compliant agent platform [20]. JADE is fully developed in Java, as well as agent development is done using Java. Moreover, JADE provides graphical user interface for management of the agent platform, and it has a set of graphical tools, which can help in management and debugging phase. In this case, testing of the implemented agent interaction mechanisms will be carried out using JADE debugging tool called *Sniffer Agent*, which allows tracking, monitoring, and debugging of conversations between agents. In order to develop agents in Java, software development platform Eclipse IDE (Integrated Development Environment) is used together with JADE agent platform. Therefore, it will be possible to develop agents, run them, and follow their actions using JADE.

As previously mentioned, practical implementation of agent interaction mechanisms for modelling of emotions from facial features is done, because on the basis of facial features it is possible to quite precisely determine the most part of the emotions that may arise in the learning process (delight, confusion, boredom, frustration, flow, and neutral). Thus, agents could determine appropriate emotions for each received facial feature (shape of eyes, eyebrows, mouth and displacement of lip corners), a table is created showing all possible facial feature expressions and emotions characterized by these features (Table 2).

Practically implementing the previously developed table in Java, lists of emotion names are defined, for example, for shape of eyes:

- `static List<String> closed=Arrays.asList("Bored");`
- `static List <String> tightened = Arrays.asList("Delighted", "Confused", "Bored", "Frustrated", "Flow");`

Depending on the received facial feature expressions, appropriate lists of emotions are selected using method and if-then rules shown below.

```
byUpperFeatures (eyes, eyebrows)
  if (eyes.equals(„Closed”) )
    {eyeEmotions = closed;}
  if (eyes.equals(„Tightened”))
    {eyeEmotions = tightened;}
    ...
  if (eyebrows.equals(„Rised”))
    {eyebrowEmotions = rised;}
  if (eyebrows.equals(„No changes”))
    {eyebrowEmotions = noChanges;}
```

TABLE 2

FACIAL FEATURE EXPRESSIONS AND APPROPRIATE EMOTIONS

Facial features	Emotions
Shape of eyes:	
Closed	Bored
Tightened	Delighted, Confused, Bored, Frustrated, Flow
Opened	Bored, Flow, Neutral
Shape of eyebrows:	
Lowered	Confused, Flow
No changes	Delighted, Bored, Flow, Neutral
Raised	Delighted, Bored, Frustrated, Flow
Shape of mouth:	
Pressed	Frustrated
Closed	Delighted, Confused, Bored, Flow, Neutral
Opened	Delighted, Bored
Stretched	Bored
Displacement of lip corners:	
Pulled down	Confused
No changes	Bored, Flow, Neutral
Pulled straight	Bored, Frustrated, Flow
Pulled up	Delighted

The emotion's determination process (feature → emotion) in general can be described by the following steps:

- The Data Processing Agent after the Agents–Classifiers' request to send necessary features carries out feature reading from the text file.
- The Data Processing Agent sends required features to the Agents–Classifiers.
- The Agents–Classifiers after receiving features pass them to the previously mentioned method that using if-then rules selects appropriate emotion lists, for example, the emotion lists that are appropriate to the shape of eyes and to the shape of eyebrows.
- The method after selecting the emotion lists carries out their comparison using the set theory operation called intersection. It allows finding the set of emotions that are in both compared lists. For this purpose the already defined method *retainall()* is used in Java.
- All Agents–Classifiers send common emotions to the Agent–Integrator.
- The Agent–Integrator after receiving all emotions from the Agents–Classifiers carries out comparison and determines the common emotion. If there is more than one emotion, then the agent selects only one with the highest probability.
- The Agent–Integrator sends the resulting emotion to the Emotional Modelling Agent.

Agents also must have common understanding of the knowledge in order to be able to communicate with each other and successfully exchange knowledge. Therefore, before the implementation of agents and their interaction mechanisms the ontology is created that represents existing things and their dependencies in the domain. In this case, in order that agents could understand identified emotions and exchange obtained results with each other, "emotion-ontology" is developed.

First of all, vocabulary of terms is created that will be used in creating concepts and agent actions. Some of created terms related to a student, such as name, surname, and student card number are shown below.

```
public static final String STUDENT = "STUDENT";
public static final String STUDENT_NAME = "name";
public static final String STUDENT_SURNAME =
"surname";
public static final String STUDENT_CARD_NO="cardNo";
```

After defining all the terms, appropriate Java classes are created. The previously defined terms are used as class names or class attributes in these classes. When all terms are defined and appropriate Java classes are created, it is possible to create emotion-ontology and link terms to appropriate classes, specifying whether it is a concept or an agent action. It is also necessary to specify the structure for every created concept and agent action, for each of them defining the existing slots, their type, as well as specifying for each slot whether it is mandatory or optional. A slot can have also cardinality, which means that the slot can contain 1 or more elements of defined type. The creation of "emotion-ontology" and example with concept "Student", as well as its defined slots are shown below.

```
public static final String NAME=„emotion-ontology";
ConceptSchema cs =
(ConceptSchema)getSchema(STUDENT);
cs.add(STUDENT_NAME,
(PrimitiveSchema)getSchema(BasicOntology.STRING),
ObjectSchema.MANDATORY);
cs.add(STUDENT_SURNAME,
(PrimitiveSchema)getSchema(BasicOntology.STRING),
ObjectSchema.MANDATORY);
```

After creating the method for the determination of emotions based on facial features and emotion-ontology, development of agents can be done. Together five agents must be developed, including all agents specified in interaction mechanism (Figure 6) – Emotional Modelling Agent, Facial Feature Agent-Integrator, Agent-Classifier by Eyes and Eyebrows, Agent-Classifier by Mouth and Lip Corners – and Face Data Processing Agent. Development of each agent is carried out in a similar manner. Every agent:

- registers ontology and content language (*Semantic Language*) used for messages;

```
getContentManager().registerLanguage(codec);
getContentManager().registerOntology(ontology);
```

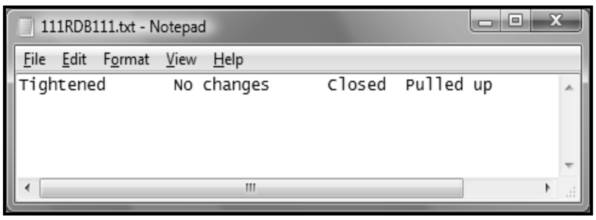

Fig. 7. The text file containing student's facial features

- registers its offered service and searches for agents, whose offered services are necessary for him;

```
sd.setType( "Emotional-modeling" );
try { DFService.register(this, dfd ); }
ServiceDescription sd1 = new ServiceDescription();
sd1.setType("Facial-feature-integration");
try { DFAgentDescription[] result =
DFService.search(this, template);}
```

- starts initial behaviour that consists of several steps, for example, two steps;

```
my2StepBehaviour mybehaviour = new
my2StepBehaviour(this);
addBehaviour(mybehaviour);
public void action() {
switch (state){
case FIRST: {op1();  state = SECOND; break;}
case SECOND:{op2(); state = FIRST; finished = true;
break;} } }
```

- after executing some action (message sending and receiving, determination of an emotion, etc.) informs about it on the screen, for example, after sending student's data;

```
student.setName ("Juris");
student.setSurname("Kalnins");
student.setCardNo("111RDB111");
ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
msg.setContentObject(student);
send(msg);
System.out.println("Agent "+getLocalName()+": Sent
student's information:\n"+student.toString()+"\n");
```

- after executing all steps of the behaviour using *reset()* method restores behaviour at initial state, thus the result is multi-step cyclic behaviour;

```
public void reset() {
super.reset();
finished = false;
state = FIRST; }
```

After all agents have completed their behaviour, the Emotional Modelling Agent waits for 30 seconds, thus imitating the situation in which possibly the student's current emotional state changes in the learning process. These 30 seconds will be used during the testing of the implemented agent interaction mechanisms in order to change facial feature expressions that will be saved in a text file or to follow agent sent messages using JADE.

VII. Experimental Testing of the Implemented Agent Interaction Mechanisms

Before running agents, it is necessary to define agent local names, that can be freely chosen by a developer, and specify an agent type or one of the previously implemented agent classes.

For testing the implemented agent interaction mechanisms, a text file is used in which student's facial features are saved. In this case, the following facial features are chosen: eyes – tightened, eyebrows – no changes, mouth – closed, and lip corners – pulled up, in order to match one particular emotion – delighted (Figure 7).

After running the agents and using the previously created text file, defined agents' text messages are displayed in Eclipse console. For example, the Agent-Classifier by Eyes and Eyebrows after receiving upper facial features (shape of eyes and eyebrows) outputs the following message in the console:

```
EyesEyeBrowsAgentClassifier: Received student's eye
features – Eyes: Tightened, Eyebrows: No changes
```

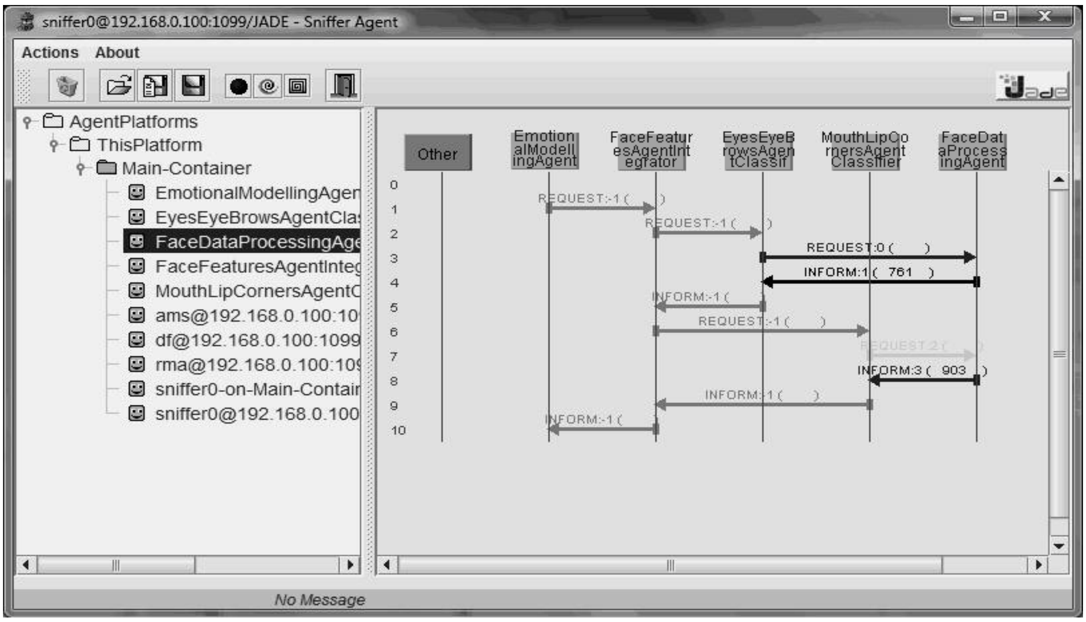

Fig. 8. Messages sent among agents and sequence of these messages using Sniffer Agent

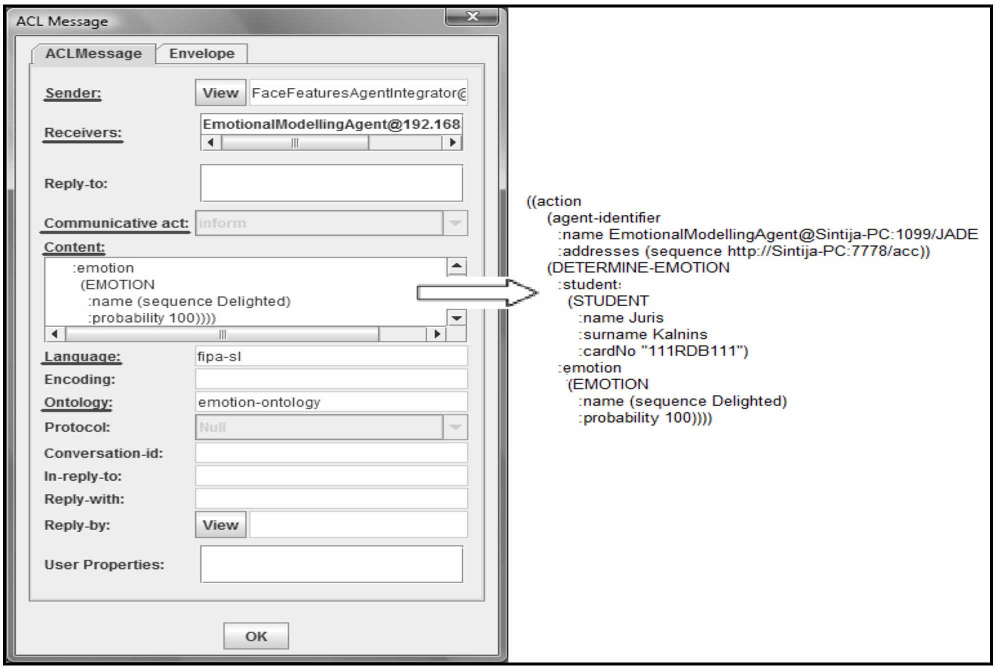

Fig. 9. Full information about the sent message

The Face Data Processing Agent after receiving a request to send lower facial features (shape of mouth and displacement of lip corners) from the Agent-Classifier by Mouth and Lip Corners outputs in the console messages about the received request and facial features that were read from a text file:

```
FaceDataProcessingAgent:
MouthLipCornersAgentClassifier sent request message
to send mouth features!
Data from file-mouth: Closed, lip corners: Pulled up
```

As already mentioned, it is possible to follow agent interaction and their sent messages using JADE. In this case, messages sent among agents and sequence of these messages can be seen using *Sniffer Agent* (Figure 8). Figure 8 clearly shows that the agent interaction mechanism is similar to the previously designed one.

Sniffer Agent also allows viewing each sent message in detail and seeing full information about it – a message sender, receiver, communicative action, message content, ontology, content language, etc. Observing the last INFORM message in Figure 9, it is possible to see all the previously mentioned information about this message (Figure 9). Message content contains student's data – a name, surname, and card number, as well as the determined emotion and its probability.

CONCLUSIONS

The concepts of intelligent and affective tutoring systems and architecture of such systems are described in this paper. Also the overview of agent technology and its usage in development of intelligent tutoring systems is examined to find out the concept and characteristics of the agents, as well as identify general interaction and communication mechanisms among them.

Taking into account the previously reviewed theoretical material on EITS and agent use in development of intelligent systems, the detailed interaction mechanisms among agents in an EITS are designed. The set of agents for the modelling of student's emotions in intelligent tutoring systems developed in 2008 is used as the basis for research. Design of agent interaction mechanisms in an emotionally intelligent tutoring system, considering student's facial features, is shown in this paper. In a similar way, interaction mechanisms among agents for the modelling of student's emotions based on his/her eyes data, posture data, mouse, and keyboard usage features are designed.

Experimental evaluation of the designed agent interaction mechanisms is performed by practically implementing modelling of emotions from facial features using development platform Eclipse together with JADE agent platform.

Experimental evaluation allows concluding that emotions can be quite precisely determined, if there are specific features for each of the emotions. However, there can be situations where it is necessary to make a choice from contradicting results, for example, "bored" and "flow". This is due to the fact that these emotions most precisely can be determined by the student's posture. Therefore, a student's emotion could be determined more accurately, if other designed agent interaction mechanisms were also implemented.

Design and implementation of agent interaction mechanisms described in this paper are carried out to supplement the existing theoretical research on design of agent interaction mechanisms in multi-agent systems with their practical implementation.

The results obtained can be used in the existing intelligent tutoring systems by extending their functionality with student's emotion identification or in development of new

EITS. Moreover, in future it is planned to use real devices – sensors for the extraction of necessary features.

REFERENCES

[1] C. Hinton, K. Miyamoto, B. Della-Chiesa, "Brain Research, Learning and Emotions: implications for education research, policy and practice," Eur.J.Edu., vol.43, no.1, pp.87-103, 2008.

[2] C.Lawson, "The connections between emotions and learning". [Online]. Available: http://www.cdl.org/resource-library/articles/connect_emotions.php [Accessed: Sept. 27, 2011].

[3] E.W.Taylor, „Transformative learning theory: a neurobiological perspective of the role of emotions and unconscious ways of knowing." Int. J. of Lifelong Education, vol.20, no.3, pp.218-236, May-June 2001.

[4] A. Anohina and L. Intenberga, "The Set of Agents for the Modelling of Student's Emotions in Intelligent Tutoring Systems," in *12th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2008)*, 2008, pp.73-78.

[5] R.Picard, ***Affective computing.*** Cambridge: MIT Press, 1997.

[6] M.Ochs and C.Frasson, "Optimal emotional conditions for learning with an intelligent tutoring system," in *7th International Conference on Intelligent Tutoring Systems*, 2004, pp.845-847.

[7] A. Sarrafzadeh, A. Alexander, F. Dadgostar, C. Fan, A. Bigdeli, "How do you know that I don't understand? A look at the future of intelligent tutoring systems," Comp.in Hum.Beh., vol.24, no.4, pp.1342-1363, 2008.

[8] E.Alepis, M. Virvou, K. Kabassi, "Requirements analysis and design of an affective bi-modal intelligent tutoring system: the case of keyboard and microphone," Stud. in Comp. Intel., vol.104, pp.9-24, 2008.

[9] S. D'Mello, A. Graesser, R.W. Picard, "Toward an affect-sensitive AutoTutor", Intel.Syst., vol.22, iss.4, pp.53-61, 2007.

[10] S.Russell and P.Norvig, *Artificial intelligence: a modern approach*. Prentice Hall, 2009.

[11] T. Finin et. al., KQML as an Agent Communication Language, in *J. Bradshaw, ed, Software Agents*, Cambridge, MA: MIT Press, 1995.

[12] D. Keil and D.Q. Goldin, "Indirect Interaction in Environments for Multi-agent Systems", in *Proceedings of E4MAS*, 2005, pp.68-87.

[13] R. Becker et al., "Analyzing Myopic Approaches for Multi-Agent Communication," Computational Intelligence, vol. 25, no. 1, pp.31-50, Feb. 2009.

[14] Y. Shoham and K. Leyton-Brown, *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge, UK: Cambridge University Press, 2009.

[15] M. Wooldridge, *An Introduction to Multi-Agent Systems*, 2nd ed. New York: John Wiley & Sons, 2009.

[16] "Multi-agent systems". [Online]. Available: http://www-2.cs.cmu.edu/~softagents/multi.html. [Accessed: Sept. 27, 2011].

[17] K. Sycara, "Multiagent systems," AI Magazine, vol.19, no.2, pp.79-92, 1998.

[18] B. Bauer et. al., "Agent UML: A Formalism for Specifying Multiagent Software Systems," International Journal of Software Engineering and Knowledge Engineering, vol. 11, no. 3, pp.207-230, Jun. 2001.

[19] M. Huget, "Agent UML Notation for Multiagent System Design," IEEE Internet Computing, vol. 8, no. 4, pp.63-71, Jul/Aug. 2004.

[20] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, "JADE – A Java Agent Development Framework," in *Multi-Agent Programming: Languages, Platforms and Applications*, Springer, 2005, pp. 125–148

Sintija Petrovica obtained the master degree in computer systems in 2011 from Riga Technical University, Latvia.

Since 2011 she is a Scientific Assistant at Riga Technical University, Department of System Theory and Design. Her main research fields are intelligent tutoring systems and agent technologies.

Sintija Petrovica received the Award for the Best Paper at the international conference on E-Learning and the Knowledge Society in 2011.

Address: Department of System Theory and Design, Riga Technical University, Kalku Str. 1, Riga, LV-1658, Latvia, Phone: +371 26569654, E-mail: SintijaPetrovica@gmail.com

Alla Anohina-Naumeca obtained the doctoral degree in the sub-field of system analysis, modelling and design of the field of information technology in 2007 from Riga Technical University, Latvia.

She is a Lecturer at the Department of System Theory and Design of the mentioned university. Alla Anohina-Naumeca has a ten-year experience of teaching in the field of computer science. Her main research fields are intelligent tutoring systems, computer-assisted assessment systems, and artificial intelligence.

Alla Anohina-Naumeca received Gebert Rüf Stiftung's Swiss Baltic Net Graduate Award for outstanding research projects and active participation at international conferences in 2008, as well as Werner von Siemens Excellence Award for the doctoral thesis in 2007 and the master thesis in 2002.

Address: Department of System Theory and Design, Riga Technical University, Kalku Str. 1, Riga, LV-1658, Latvia, Phone: +371 67089595, Fax: +371 67089584, E-mail: alla.anohina-naumeca@rtu.lv