Agent-based method of improving the efficiency of the e-learning

Natalia Axak [1], Andrii Tatarnykov [1], Maksym Kushnaryov[1]

[1] *Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine*

Abstract

The paper discusses the problems of modern distance learning, such as individualization of the learning process. The need to improve the processes of supporting remote learning to improve the quality of education is substantiated. The aim of the study is to improve the quality of education and the organization of the educational process in distance learning.

A set of interrelated tasks is proposed, the solution of which is to organize effective interaction between the teacher and the learning management system (LMS) using an agent-based approach. The main result that defines the novelty of the work is the formalization and integration of the following processes: (i) generating real-time suggestions to the teacher to control the performance of tasks during exams or electronic testing; (ii) monitoring students' learning during the semester with the possibility of changing the learning trajectory; (iii) monitoring parents' presence in online classes; (iv) generating recommendations for management and other stakeholders to improve online learning

The modeling of the prototype of the proposed system confirms the effectiveness of its use as a means of studying the organization of the educational process. It shows how agents can help collect and analyze data on how effectively electronic resources are used for learning by students and teachers.

Keywords

e-learning, learning management system, behavior monitoring, temporal logic, multi-agent logics

1. An introduction to the use of intelligent agents for decision-making in the educational process

In the educational process, decision-making is an important step, as it affects the quality of learning and student success. However, there are some problems that can arise when making decisions in the educational process: the absence of necessary data or insufficient processing can make the decision-making process insufficiently informed and reduce its effectiveness; limited resources, such as budget, staff, and time, can force decisions that are not the best for the quality of education and student success; insufficient qualifications of education staff can lead to misunderstanding and analysis of data, which can lead to incorrect decisions; the diversity of students in a classroom can make decision-making more complex, as students' needs and interests may vary; insufficient motivation of students can reduce the effectiveness of learning and lead to lower quality solutions; social issues, such as poverty, violence, and discrimination, can affect the quality of education and student success and complicate decision-making.

These problems can reduce the efficiency of decision-making in the educational process.

The main advantages of using agents for decision making are their ability to: process automation - agents can help automate complex decision-making processes, which can increase productivity and reduce time spent on decisions; data collection and analysis - agents can help collect and analyze large amounts of data, which allows you to identify trends and patterns that can be used to make better decisions; intelligent decisions - agents can be trained to make decisions based on previous experiences and training using machine learning and other technologies, allowing them to make better decisions based on the analysis of many factors; individualized approach - agents can help to develop customized solutions for participants in the queuing process, which allows for a

[1] *ICST-2024: Information Control Systems & Technologies, September , 23 – 25, 2024, Odesa, Ukraine*

nataliia.axak@nure.ua (N. Axak); andrii.tatarnykov@nure.ua (A. Tatarnykov); maksym.kushnarov@nure.ua (M. Kushnaryov)

0000-0001-8372-8432 (N. Axak); 0000-0002-1632-8188 (A. Tatarnykov); 0000-0002-3772-3195 (M. Kushnaryov)

more personalized approach and meet the needs of different groups.

Thus, the use of agents in the educational process is relevant and can help improve the quality of decision-making.

2. Related Works

Recently, research related to agent-based decision-making in education has taken a big step forward. The book [1] focuses on Intelligent Tutoring Systems, which are one of the most common uses of agents in education for decision-making.

Studies [2, 3, 4, 5] conducted a systematic review of the literature on the use of artificial intelligence in the educational process, in particular for decision-making.

Study [6] proposes agent-based virtual and intelligent recommendations that require information about users' profiles and preferences to recommend the right content. They applied natural language processing (NLP) techniques and semantic analysis approaches to recommend course selection for e-learners and teachers.

The content-based recommendation method provides content suggestions related to students' requests and preferences. The use of social media from an educational perspective makes it possible to provide a user-friendly interface for recommending the highest level of interaction in terms of collaboration between users and contacts [7,8,9].

The article [10] identifies the features of human and artificial intelligence decision-making along five key contingency factors: specificity of the decision space, ability to interpret the decision-making process and results, size of the alternative set, speed of decision-making, and repeatability. Based on a comparison of human and artificial intelligence decision-making along these dimensions, the article creates a new framework that describes how both decision-making methods can be combined to optimally improve the quality of organizational decision-making.

The authors of [11] propose future research directions in a triple perspective: key methodologies for Large Scale Decision Making (LSDM), AI, and data fusion for LSDM.

In [12], the authors describe a meta-reasoning policy that can be implemented by a team of agents to make effective control decisions at the meta-level based on the availability of communication in the environment. The authors synthesize the meta-reasoning policy as a solution to the reactive synthesis problem involving the level of communication in the environment and the choice of the agent's algorithm.

The authors of [13] argue that in a multi-agent environment, it is appropriate to ask what behavior the system will exhibit under the assumption that agents act rationally, following their preferences. They promote a paradigm of rational verification for multi-agent systems, as an analog of classical verification. The authors tried to automatically determine whether the given properties of a system, expressed in the form of temporal logic formulas, will be preserved in this system under the assumption that the system components (agent) behave rationally by choosing (for example) strategies that form a game-theoretic equilibrium.

The article [14] aims to provide a comprehensive view of the relationship between agents and multi-agent systems (MAS) on the one hand, and logic-based technologies on the other, by making them the subject of a systematic literature review. The resulting technologies are discussed and evaluated from two different perspectives: MAS and logic-based. The paper lists the most common logic-based technologies (47 in total) for MAS, but only a relatively small number of them conform to major technology standards.

Temporal logics have been widely used in model checking as a formalism for reasoning about the execution of computer systems. They are powerful enough to define most of the properties that can be verified by reactive systems, while also providing very efficient verification algorithms [15]. Temporal logic and model checking have had a major impact on computer science and have been applied in many industrial cases. Several attempts have been made to extend temporal logic to multi-agent systems where multiple components interact, for example, Computation-Tree Logic (CTL) can only express the existence (or not) of an execution of a global system with certain properties, where the goal is to quantify the possible behavior of individual components interacting in the system. In 1997, CTL was extended to Alternating-time Temporal Logic (ATL) with the introduction of strategy quantifiers. In ATL, strategy quantifiers express the existence (or not) of a behavior of one of the agents (or a coalition), so that any final execution in the global system satisfies this property. Study [16] is related to multi-agent logic and its application in computer science. The authors work with multi-agent logic based on relational models. They determine that time availability relations can have gaps or places of forgotten time.

The authors of [17] study the problem of learning to satisfy temporal logic specifications with a group of agents in an unknown environment that can exhibit probabilistic behavior. From a learning perspective, these specifications provide a rich formal language with which to capture tasks or goals, while from a logic and automated verification perspective, the implementation of learning capabilities allows for practical applications in large, stochastic, unknown environments.

The temporal logic of actions (TLA) – is a logic for specifying and reasoning about parallel systems. The systems and their properties are represented in the same logic, so the statement that a system conforms to its specification and the statement that one system implements another are expressed by logical consequence. TLA is very simple; its syntax and full formal semantics are summarized in about a page. Report [18] introduces TLA and describes how it is used to define and verify parallel algorithms.

Education systems include a variety of components such as learning management, progress tracking systems, e-textbooks, etc. The challenge is to interact and integrate intelligent agents with these systems to ensure that they work together effectively. For a system to be successful, agents need to be able to quickly adapt to new requirements and change their behavior accordingly.

Given these problems, further research and development of intelligent agents in the educational process is aimed at solving these issues, ensuring accessibility, flexibility, ethical use and involvement of all participants in the educational process.

The aim of the work is to improve the quality of distance learning decision-making based on the analysis of video information about student behavior.

To achieve this goal, the following tasks are solved: (i) to develop the model of an agent-oriented decision support system for distance learning; its components and their interaction; (ii) identify user needs based on the collected data and current trends in education; (iii) formalize the process of generating recommendations for teachers to support decision-making.

3. The model of an agent-oriented decision support system for distance learning

This paper proposes a system that integrates various components: a video surveillance subsystem that detects certain actions or circumstances that may indicate possible violations during e-testing [19]; a multi-agent system whose agents interact with the video surveillance system in real time [20], as well as analyze the collected data and provide recommendations; temporal logic to determine the logic of the system's response to various events and states in real time; a decision-making system that automatically responds to detected violations in real time; mechanisms for automatically responding to detected violations, such as sending notifications to administrators or teachers.

To implement such a system we propose a model of an agent-based decision support system (AoDSS) for distance learning, which is a conceptual description of the architecture and principles of the system. This model uses the agent-based paradigm to optimize and improve the learning process in a remote format. In such a system, agents are key components that contribute to the organization and optimization of the learning environment.

The proposed AoDSS model is represented as a set of:

$$AoDSS = \langle MAS, S_F, S_B, S_S \rangle, \tag{1}$$

where MAS – multi-agent system, S_F – the client side subsystem (Frontend) is presented in various formats, which allows users to interact with the system AoDSS from various devices, S_B – subsystem of the server part (Backend) Provides data processing and the ability to customize client applications, S_S – a subsystem for synchronizing client and server parts. The multi-level architecture of an agent-based decision support system for distance learning is shown in Fig. 1.

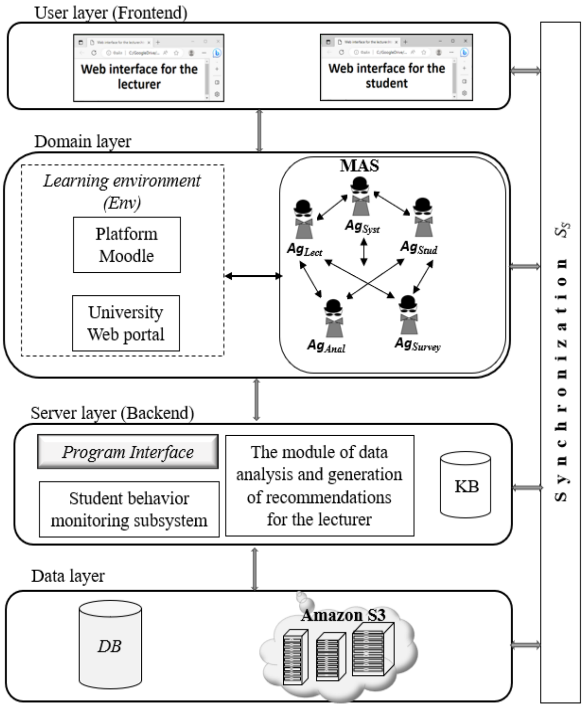


Figure 1: Architecture of an agent-based decision support system for distance learning.

The combination of the sequence diagram and the diagram of the system activities for the decision-making process during distance learning to provide recommendations to the teacher and student is shown in Fig. 2.

Actions performed by AoDSS participants are marked with rectangles. Consideration of the diagram begins with the registration of participants in the educational process on the server using the teacher and student interfaces contained in the client subsystem.

Solid arrows show messages that are sent in any case, and dashed arrows show messages exchanged by agents whose sending depends on a condition.

Types of messages used between agents: cfp - used for announcing the task; propose - used for making a proposal; inform - used to indicate completion of the task.

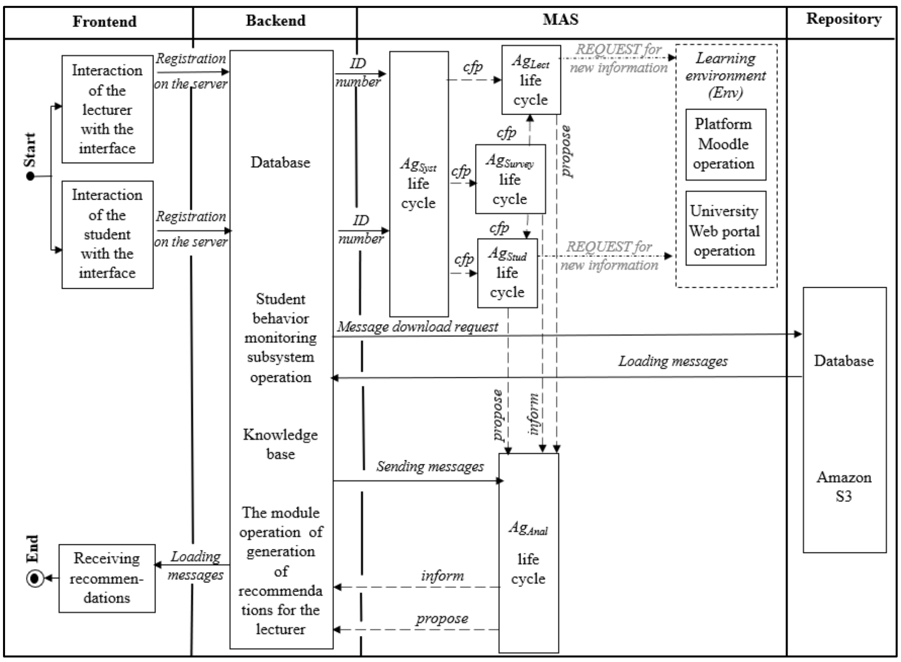


Figure 2: The combination of the sequence diagram and the activity diagram reflects dynamic aspects of AoDSS behavior.

3.1. Multi-agent system

In formula (2), a multi-agent system is represented as a set of three components:

$$MAS = \{Ag, Act, Env\}, \quad (2)$$

which consists of a set of program agents $Ag = \{Ag_{Stud}, Ag_{Lect}, Ag_{Anal}, Ag_{Syst}, Ag_{Survey}\}$, that operate in the environment *Env*, where AgS_{tud} – student agent monitors the activity and academic progress of students, collects data on completed assignments, tests, and other information for further analysis; Ag_{Lect} – the teaching agent monitors courses, assignments, and student feedback, generates recommendations for improving materials and teaching methods to enhance teaching and decision-making in the educational environment; Ag_{Anal} – the data analytics agent is responsible for processing and analyzing data collected from students and teachers, identifying patterns, trends, and important connections to support decision-making; Ag_{Syst} – the system agent manages the interaction between other agents, distributes tasks, ensures the continuity of the system and coordination of processes; Ag_{Survey} – the survey agent sends questionnaires to users, collects responses and stores them in the database; $Act = \{A_{Stud}, A_{Lect}, A_{Anal}, A_{Syst}, A_{Survey}\}$ – set of actions of agents ($A_i = a^i_1, a^i_{2, \ldots,} a^i_n$; i=*Stud, Lect, Anal, Syst, Survey*); $Env = \{E_{\mathrm{Mn}}, E_{\mathrm{Md}}, E_{\mathrm{Un}}\}$ – a set of states of the environment of Student behavior monitoring subsystem (E_{Mn}), the environment of LMS Moodle (E_{Md}) and the environment of University Web portal (E_{Un}), ($E_j = \{e^j_1, e^j_2, \ldots, e^j_m\}, j = \overline{(Mn, Md, Un)}$).

Agents are described by functions $Ag_i\colon f_{\mathrm{i}}\left(ended\ with\ e^i_u\right) \to A_i,\ (u = \overline{1, m})$. The set of all agent's runs in an environment is the behavior of the agents $F(Ag, Env)$.

Agent functionality Ag_{Stud} includes:

- Data collection and analysis

$$Y^a = \begin{bmatrix} y^a_{11} & \cdots & y^a_{1m} \\ \cdots & \cdots & \cdots \\ y^a_{n1} & \cdots & y^a_{nm} \end{bmatrix},$$

(*n*=1,2,…,*N*; *N* – is the number of students enrolled in the *m*-th discipline of the curriculum ($m \in M$), *M* – number of disciplines in an academic semester) on student attendance and activity during classes

- Analysis of the collected data

$$Y^k = \begin{bmatrix} y^k_{11} & \cdots & y^k_{1m} \\ \cdots & \cdots & \cdots \\ y^k_{n1} & \cdots & y^k_{nm} \end{bmatrix}$$

to determine the student's level of knowledge (the number of points at the current moment of the discipline m) and to identify weaknesses in their understanding of the material.

- Training a model based on new data obtained from student data

$$Y^{Stud} = \{Y^a, Y^k\}.$$

Agent functionality Ag_{Lect} includes:

- Collecting and aggregating various data, such as student grades, course activity

$$Y^{Lect} = \left\{\min_{y^a_{im}} Y^{Stud}, \max_{y^a_{im}} Y^{Stud}, \min_{y^k_{im}} Y^{Stud}, \max_{y^k_{im}} Y^{Stud}, \ldots\right\}.$$

- Use of analytical methods and algorithms to identify useful connections, patterns, and trends in the accumulated data $\mathrm{Y^{Lect}}, \mathrm{Y^{Stud}}$.
- Based on data analysis $\mathrm{Y^{Lect}}, \mathrm{Y^{Stud}}$ the agent creates individual recommendations on teaching methods, selection of materials, creation of additional explanations and examples to improve students' understanding of the material of the course organization [21].
- Tracking student attendance and impact on the effectiveness of the learning process, as well as providing appropriate feedback.

Agent functionality Ag_{Survey} includes:

- Sending out questionnaires: $A_{Survey}(send) = f_{send}(Users, Survey) \to SurveyResponses$, where Users - a set of users (students and teachers), Survey - a questionnaire, $f_{send}(Users, Survey)$ – the function of sending questionnaires to users.
- Collecting responses: $A_{Survey}(collect) = f_{collect}(SurveyResponses) \to CollectedData$, where $SurveyResponses = \{x_1, x_2, \ldots\}$ – answers to the questionnaire, *CollectedData* - collected data, $f_{collect}(SurveyResponses)$ – function of collecting answers to questionnaires.

- Saving answers in the database: $A_{Survey}(store)=f_{store}(CollectedData,Database)$, where *Database* - database to store the collected data, $f_{store}(CollectedData,Database)$ – the function of saving the collected data in the database.

Agent functionality A_{Anal} includes:

- Processing of collected data: $A_{Anal}(process)=f_{process}(CollectedData) \rightarrow ProcessedData$, where *ProcessedData* – processed data, $f_{process}(CollectedData)$ – function of processing the collected data.
- Data analysis: $A_{Anal}(analyze)=f_{analyze}(ProcessedData) \rightarrow Insights$, where *Insights* - Identified patterns and trends, $f_{analyze}(ProcessedData)$ – function of analyzing the processed data.
- Generating recommendations: $A_{Anal}(recommend)=f_{recommend}(Insights) \rightarrow Recommendations$, where *Recommendations* – recommendations for decision-making, $f_{recommend}(Insights)$ – function of generating recommendations based on data analysis.

Agents Ag_{Survey} and Ag_{Anal} have clearly defined functional responsibilities and interact to collect, process, and analyze data that helps to make decisions in the distance learning system.

3.2. Modeling agent decisions

At the abstract level, agents have four properties - knowledge, goals, conditions, and actions - where knowledge is provided through a simulator of the roles of scenario developers, lecturers, and students participating in distance learning. We developed a simulator to model the scenario. Simulator constructs are used to emulate the actions of lecturers and students. The simulator provides the user with a level of abstraction that exactly matches the domain problem description and eliminates unnecessary overhead of customizing frameworks and writing technical code for a specific application. In this interpreter, we use 10 key concepts necessary for modeling a decision-making scenario in distance learning, which are presented in the ontology in Fig. 3.

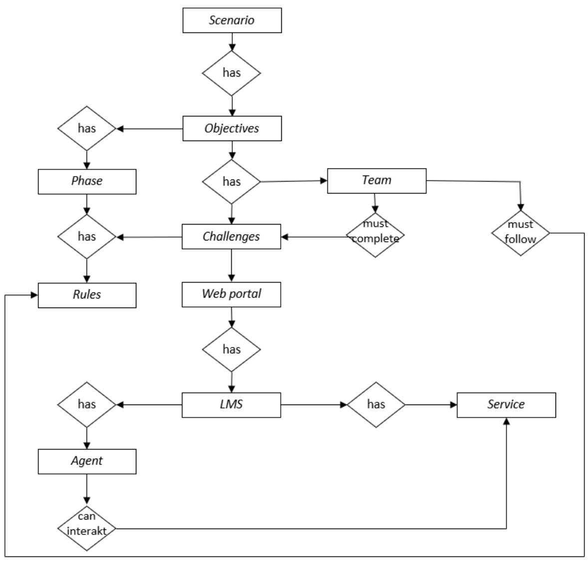


Figure 3: Ontology of concepts for the scenario of providing recommendations for improving the quality of education.

1 *Scenario*. The following types of scenarios have been implemented: generating real-time suggestions for the teacher to control the completion of tasks during exams or e-testing; monitoring the mastery of educational materials by students during the semester with the possibility of changing the learning path; parents' control over the presence of their children in online classes; and analyzing the satisfaction of using the online platform.

The main properties of the scenario are defined: Name is used to define the name of the event within which the scenario will be executed. Start date: a date value that indicates the start date of the scenario in the format dd.mm.yyyy. End Date: A date value that indicates the end date of the scenario in the format dd.mm.yyyy. Start time: a time value that indicates the start time of the scenario in 24-hour format: h:min. End time: a time value that indicates the end time of the scenario in 24-hour format: hh:min. Goals: A list of all goals of the scenario. Agents: A list of all agents active during the scenario. Rules: A list of all the rules to be followed in the scenario. Teams: A list of all the teams involved in the scenario.

2. Web portal. Concepts *Web portal* is used to determine the site of the educational institution that is present in the scenario. It has the following properties: Public IP address. User accounts with the following properties: Username, Password, User ID, Groups in which the user is present, Home indicates the user's home folder.

3. LMS. Concepts. *LMS* is used to define the online learning platform that is present in the scenario. It has the following properties: Public IP address. User accounts with the following properties: Username, Password, User ID, Groups in which the user is present, Home indicates the user's home folder.

4. *Service*. Services are used to make the scenario more realistic.

5. *Challenge*. The term *Challenge* is used to represent an exercise or task that needs to be completed. It has the following properties: Prerequisites - a list of other challenges that need to be completed before you can access the current task.

6. *Team*. The concept of a *team* is used to define the role of participants in the learning experience. Teams consist of lecturers or students who try to overcome challenges presented in a scenario. It has the following properties: Type is used to define the type of team, i.e. lecturers, students, or parents. Members contains the contact information of each team member.

7. *Agent*. Agents are used to automatically perform specific tasks in training. It has the following properties: Type is used to define the type of agent, such as student agent, teaching agent, etc.

8. *Phase*. The scenario can be broken down into several phases, for example, monitoring behavior, controlling attendance, providing recommendations, etc. It has the following properties: Type is used to define the type of phase, such as start, middle, and end. Scenario goals for a particular phase. Scenario rules for a particular phase. Agent – a list of agents that depend on the phase.

9. *Objectives*. A description of the scenario objectives that must be met to successfully complete the scenario. A scenario can have one or more objectives depending on the complexity of the scenario.

10. *Rules*. Rules contain information for commands in a scenario, for example, "Use of mobile devices during the exam is prohibited." It has the following properties: Type is used to specify the type of rule, such as allowed or denied. Text contains the rules of the scenario in text format

Temporal logic is important for analyzing and managing temporal aspects in systems where various events and processes occur over time. Therefore, we will create a formal specification of agents using temporal logic for decision making in distance learning. This approach allows us to create accurate and systematic models of agents and their behavior in a virtual learning environment. The formalization of agents and their behavior allows you to ensure the quality of learning and optimize the process, which is especially important in distance learning.

Formulas in temporal logic express relationships between events, states, or properties of objects at different points in time. To create formulas, we use:

- Logical operators. Temporal logic uses logical operators such as AND, OR, NOT, IMPLIES, EQUIV to build complex expressions from basic temporal or logical statements.
- Temporal operators. Temporal logic has its own temporal operators, such as "Next" (N), "Until" (U), "Eventually" (F), "Always" (G), which are used to express relationships between events at different points in time, operator "Release" (R) indicates that the event it wraps will be true until the second event occurs.
- Parameterization. Formulas can contain parameters that depend on a specific context or system conditions. This allows you to generalize formulas and ensure their use in different scenarios.
- Quantifiers. Temporal logic can use quantifiers such as "For All" (∀) and "Exists" (∃) to express general or existing relationships between events.
- Diamond. Operator (◇) is used to express the possibility of an event in the future, i.e. "sometime in the future"; the operator (◆), which expresses the obligation or necessity of an event in the future, i.e. "later in the future it will be true".
- Event and state identifiers. Formulas can contain identifiers of events, states, or object properties that are used to describe temporal aspects and their relationships with system states.
- Actions and events. Formulas can include a description of actions, events, or observations that occur in the system and their impact on the states and properties of the system over time.

• Relationships to other logical and mathematical systems. Formulas in temporal logic can use constructs and concepts from other logical and mathematical systems to further express temporal properties.

One approach is to use a formal language such as Linear Temporal Logic (LTL) to define temporal logic rules. LTL expressions include time operators that allow you to express conditions and state changes over time.

A rule for detecting misbehavior over time, i.e., if there is a moment in time when misbehavior is detected, then the agent should respond:

LTL expression: $\Diamond$(Unlawful_Behavior).

A rule to check for misbehavior at a certain frequency, i.e., if misbehavior is detected during each "Period" of steps, then the agent should react:

LTL expression: $\Diamond$(Unlawful_Behavior) U
(every_n_steps(Period)).

The rule for the appeal period, i.e.: "There is a point in time when the breach notification was sent, and this event is true until the appeal response is received."

LTL expression: $\Diamond$ (Sent_Notification_of_Violation) R $\Diamond$ (Received_Appeal_Response).

A rule to prohibit the use of aids, i.e. if a student has used aids, the agent must block the test:

LTL expression: $\Diamond$ (Used_Auxiliary_Tools) $\longrightarrow$ $\Diamond$(Block_Test).

The rule for applying warnings, i.e. if a violation is detected, the agent must send a warning:

LTL expression: $\Diamond$(Violation_Detected) $\longrightarrow$ $\Diamond$(Send_Warning).

For the formal specification of agent-based decision making in distance learning, we use the formal description language TLA+ (Temporal Logic of Actions). TLA+ allows modeling and formalizing systems with temporal aspects, and helps to express and verify the properties of specific systems, including distributed systems and algorithms.

This specification language allows you to formally define systems, taking into account their logic and dynamics. This helps to avoid misunderstandings and allows you to accurately define the expected behavior of the system.

The formal specification of an agent-oriented decision support system for distance learning is given by the following formulas:

```
   ◇(Unlawful_Behavior)
   ◇(Unlawful_Behavior) U (every_n_steps(Period))
   ◇(Sent_Notification_of_Violation) R ◇(Received_Appeal_Response)
   ◇(Used_Auxiliary_Tools) → ◇(Block_Test)
   ◇(Violation_Detected) → ◇(Send_Warning)
   G (time = 0 → (testQuestions = <<q1, q2, q3>> ∧ studentAnswers = <<>> ∧
teacherRecommendations = <<>> ∧ violations = <<>> ∧ appeals = <<>>))
   G (time < MaxTime → (∃ student ∈ DOMAIN(studentAnswers): X
(studentAnswers[student][time'] = CHOOSE answer ∈ testQuestions)))
   G (time = MaxTime → X GenerateTeacherRecommendations)
   G (GenerateTeacherRecommendations → (CheckForViolations ∧ CheckForAppeals ∧
CheckForWarning))
   G (CheckForViolations → (violations' = ...))
   G (CheckForAppeals → (appeals' = ...))
   G (CheckForWarning → (IF violations' > WarningThreshold THEN
teacherRecommendations' = AppendWarning(teacherRecommendations) ELSE
teacherRecommendations' = teacherRecommendations ENDIF))
   G (AppendWarning(recommendations) → (teacherRecommendations' =
Append(recommendations, "Warning")))
   G (AppealDecision → (∃ appeal ∈ appeals: (IsValidAppeal(appeal) ∧
ProcessAppeal(appeal))))
   G (WarningDecision → (IF violations > WarningThreshold THEN teacherRecommendations'
= AppendWarning(teacherRecommendations) ELSE teacherRecommendations' =
teacherRecommendations ENDIF))
```

Thus, the generation of agent behavior formulas is based on the analysis of the Moodle Platform, University Web portal and Student behavior monitoring subsystem, the identification of important events and their relationships over time, and the selection of appropriate temporal operators to express the desired temporal properties of the system.

3.3. Recommendation subsystem

To solve the urgent problems of each of the participants in the educational process, a subsystem for assessing student and teacher satisfaction has been developed. This approach allows for effective management of internal learning processes in an educational institution. The implementation of the subsystem for assessing student satisfaction using the online platform consists of the following stages.

Stage 1: Set the goal of increasing student and faculty satisfaction to a certain level. To measure satisfaction, we define key performance indicators (KPIs) (Table 1).

Table 1
Criteria for evaluating the effectiveness of e-learning

Factors	Features	Description
Student	x_1	Convenience of the interface
	x_2	Availability of materials
	x_3	Quality of support
	x_4	Overall impression
Teacher	x_7	The quality of teaching
	x_8	Support of teachers
	x_9	Interactivity
	x_{10}	Professional development
Educational institution	x_{11}	Infrastructure
	x_{12}	Support and management
	x_{13}	Funding
	x_{14}	Academic results
Discipline	x_{15}	Content relevance
	x_{16}	Teaching methods
	x_{17}	Practicality
	x_{18}	Adaptability
General evaluation criteria	x_{19}	Usability (ease of use)
	x_{20}	Efficiency
	x_{21}	Cost-effectiveness
	x_{22}	Security

We develop questionnaires with questions that cover all aspects of using the platform. This questionnaire includes both closed and open-ended questions, which allows us to obtain both quantitative and qualitative data on student satisfaction with the online platform.

Stage 2. The online platform Google Forms was chosen for the surveys.

Questionnaire processing consists of collecting responses, processing data, analyzing results, and preparing a report.

Questionnaire processing algorithm:

Step 1. Placement of the questionnaire: Publish the questionnaire on an online platform or send it out by email. Set a deadline for submitting responses.

Step 2. Data collection: Saving all answers in a single database or spreadsheet.

Step3. Data preparation: Exporting the collected responses to a format convenient for processing (CSV or Excel). Checking the data for completeness and correctness.

Step 4. Categorization of answers: Dividing data into categories according to the questionnaire questions.

Step 5. Calculation of quantitative indicators: For closed-ended questions (scoring on a scale from 1 to 5), calculate mean, median, mode, and other statistics. Build graphs and charts to visualize the results.

1. Calculation of quantitative indicators: For closed-ended questions (scoring on a scale from 1 to 5), calculate mean, median, mode, and other statistics. Build graphs and charts to visualize the results.
2. Analysis of open responses: Analyzing open-ended responses to identify key themes and trends. Build word clouds or other visual representations to help interpret the results.

3. Interpretation of quantitative data: Identify key trends and problem areas based on the calculated statistical indicators. Comparison of results between different groups (e.g., among students by age, course, etc.).

4. Interpretation of qualitative data: Identification of the main topics and problems mentioned by participants in open-ended responses. Assessment of the general mood and tone of the answers.

5. Preparation of the report: Prepare a report that includes key findings from the data analysis. Include graphs, charts and other visuals to illustrate the results.

6. Recommendations: Developing recommendations based on the results to improve the online platform. Identification of priority areas for implementation of changes.

7. Communication of results: Presentation of the report to the stakeholders (school administration, teachers, technical support). Publishing a summary of the main results for participants.

Create the file survey_responses.csv (Table 2), which will contain the survey data, containing columns that correspond to the survey questions.

Table 2
An example of a survey_responses.csv file that will contain student survey data

Interface usability	Material accessibility	Support quality	Overall satisfaction	Liked elements	Improvement suggestions	Support comments	General feedback
4	5	3	4	Easy to use	Add more features	Good support	Overall satisfied
3	4	2	3	User-friendly	Improve speed	Average support	Needs improvement
5	5	4	5	Intuitive design	More tutorials	Excellent support	Very satisfied

This file contains data from the survey, where each row represents the response of one survey participant to different questions. Each column corresponds to a specific question or aspect that was rated by the participants on a scale from 1 to 5, where 1 is very bad and 5 is very good.

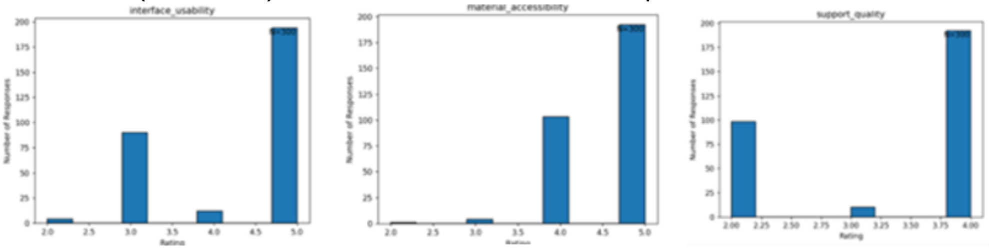
This data will be used to analyze user satisfaction and identify areas for product or service improvement (Fig. 4).

These graphs help to understand how users perceive a product or service and identify possible areas for improvement.

The result (Fig. 5.a) shows a statistical description of the survey data of students and teachers by several parameters. Each row of the table represents different statistical metrics for each parameter (each column) of the data.

Figure 5.b shows the Correlation Matrix to show the relationship between the different survey questions. The cells show the correlation coefficients between pairs of questions.

The color scale shows the strength of the correlation, where warm colors (red) indicate a positive correlation and cold colors (blue) indicate a negative correlation. High positive values (close to 1) indicate that when one question has high scores, the other question tends to have high scores as well. High negative values (close to -1) indicate an inverse relationship.



a) b) c)

Figure 4: a) The interface_usability graph shows the distribution of user ratings on the interface usability scale. b) The material_accessibility graph shows the distribution of material accessibility scores. c) The support_quality graph shows the distribution of support quality scores.

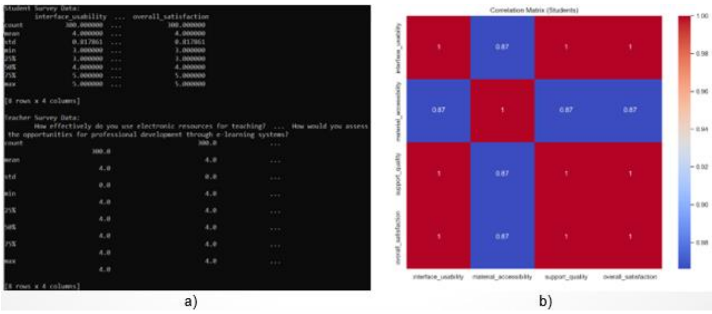


Figure 5: a) Statistical description of the survey data of students and teachers. b) A correlation matrix that shows the degree of linear dependence between the questions of the student survey.

Each element of the matrix is a correlation coefficient between two questions. For example, if the value in the student correlation matrix is 0.87 in the position (1, 2), it means that the answers to the first question have a strong positive linear relationship with the answers to the second question.

The correlation matrix helps to identify which aspects of the survey can be interrelated, which can be useful for further analysis and decision-making.

Part of the report contains recommendations for further improvement of the product or service for students: (i) improve user interface; (ii) enhance accessibility of educational materials; (iii) optimize technical support.

Based on the data analysis, the following recommendations were made for teachers: (i) Improve support for technical and methodological issues – If the average value for this criterion is less than 4, technical and methodological support for teachers should be improved. (ii) Enhance opportunities for interactive interaction with students - If the average value for this criterion is less than 4, you need to increase opportunities for interactive interaction with students. (iii) Provide more options for professional development – If the average value for this criterion is less than 4, more opportunities for professional development of teachers should be provided.

Based on the analysis of the results of the survey of teachers on their satisfaction with the use of the online platform, several main areas for improvement can be identified:

1. Increase the efficiency of using electronic resources for teaching. According to the survey, teachers evaluate the effectiveness of the use of electronic resources for teaching with an average of 4 out of 5 points. This shows overall satisfaction, but also indicates that there is room for improvement. Recommendations: Simplify navigation and access to materials. Add the ability to create interactive learning materials (e.g., interactive presentations, video tutorials with interactive elements). Introduce new test formats and automatic tools for assessing students' knowledge.

2. Support for teachers in the use of e-learning systems. Technical and methodological support received an average score of 4 out of 5, which indicates a sufficient level of satisfaction, but also points to the need for improvement. Recommendations: Provide round-the-clock technical support through various channels (chat, phone, email). Regularly conduct trainings on how to use the platform, including new features and best practices. Provide an opportunity to order individual consultations to solve specific problems.

3. Expanding opportunities for interactive interaction with students. Opportunities for interactive interaction with students are rated 4 out of 5, which shows the need for additional features to improve interaction. Recommendations: Introduce additional tools for video conferencing, interactive forums and chats. Expand feedback opportunities, for example, by adding the ability to conduct surveys and questionnaires in real time. Introduce gamification elements to increase student motivation.

4. Assessment of opportunities for professional development of teachers. Opportunities for professional development through the use of e-learning systems received an average score of 4 out of 5, which indicates the need to expand such opportunities. Recommendations: Introduce professional development programs that include courses, webinars, and the possibility of obtaining certificates. Create a platform for the exchange of experience and best practices among teachers. Introduce a system of incentives for participation in professional development programs.

The results of the survey show the overall satisfaction of teachers with various aspects of e-learning, as the average scores are at the level of 4 or higher.

4. Conclusion

The article proposes an agent-based method for improving the efficiency of e-learning, which consists of the following stages:

Stage 1: Analysis and preparation. The main goals of the system are identified: improving learning efficiency, automating processes, increasing student and teacher satisfaction. The requirements for the system from users: teachers, students, and the administration of the educational institution are identified. The existing distance learning systems and their limitations were analyzed and modern technologies and methods that can be used in agent-based systems were investigated.

Stage 2. System design. The main components of the system were identified: a video surveillance subsystem, a multi-agent system, and a decision-making system. The architecture of interaction between the components was designed. The types of agents (AgStud, AgLect, AgAnal, AgSyst, AgSurvey) and their roles in the system were defined, and the functionality of each agent was determined: data collection, analysis, and recommendation generation. The algorithms of agents' work are described. The process of generating recommendations for teachers and automating responses to events is formalized.

Stage 3. Development and implementation. Program modules for each of the agents were developed and integrated into a single system. A video surveillance subsystem was implemented to monitor student behavior during classes and algorithms for recognizing events and anomalies were set up. Functional testing of each component of the system was carried out. Integration testing was performed to ensure correct interaction between the components.

The proposed agent-based method for improving the efficiency of e-learning is a comprehensive approach that covers all stages from requirements analysis to continuous system improvement. The use of intelligent agents allows automating the processes of monitoring, analysis, and decision-making, which contributes to improving the quality of education and increasing the satisfaction of students and teachers.

5. References

[1] R. A. Sottilare, A. C. Graesser, X. Hu & A. M. Sinatra, (Eds.), Design recommendations for intelligent tutoring systems: Volume 6-team tutoring (Vol. 6), US Army Research Laboratory, 2018.

[2] N. Jalal, “Editorial of the first issue of the international journal of learning analytics and artificial intelligence for education.” International journal of learning analytics and artificial intelligence for education 1.1 (2019): 4-7.

[3] F. Ouyang, P. Jiao, “Artificial intelligence in education: The three paradigms,” Computers and Education: Artificial Intelligence, 2, (2021). doi.org/10.1016/j.caeai.2021.100020.

[4] M. Cukurova, C. Kent, R. Luckin, “Artificial intelligence and multimodal data in the service of human decision-making: A case study in debate tutoring,” British Journal of Educational Technology, 50 (6), (2019): 3032-3046. doi.org/10.1111/bjet.12829.

[5] C. Campbell, “The three paradigms of Artificial intelligence in secondary school education,” Computers and Education: Artificial Intelligence, 2 (2022). doi:10.1016/j.caeai.2021.100020

[6] Z. Shahbazi, Y. C. Byun, “Agent-based recommendation in E-learning environment using knowledge discovery and machine learning approaches,” Mathematics, 10(7) (2022). doi.org/10.3390/math10071192.

[7] A. Krouska, C. Troussas, M. Virvou, “SN-Learning: An exploratory study beyond e-learning and evaluation of its applications using EV-SNL framework,” Journal of Computer Assisted Learning, 35(2) (2019). doi.org/10.1111/jcal.12330.

[8] M.S. Rafiq, X. Jianshe, M. Arif, P. Barra, “Intelligent query optimization and course recommendation during online lectures in E-learning system,” J Ambient Intell Human Comput 12, 10375–10394 (2021). doi.org/10.1007/s12652-020-02834-x.

[9] B. Mbipom, S. Craw, S. Massie, “Improving e-learning recommendation by using background knowledge,” Expert Systems *38*(7) (2021). doi.org/10.1111/exsy.12265.

[10] Y. R. Shrestha, S. M. Ben-Menahem, G. Von Krogh, "Organizational decision-making structures in the age of artificial intelligence," California Management Review, 61(4) (2019). doi.org/10.1177/0008125619862257.

[11] R. X. Ding, I. Palomares, X. Wang, G. R. Yang, B. Liu, Y. Dong, & F. Herrera, "Large-Scale decision-making: Characterization, taxonomy, challenges and future directions from an Artificial Intelligence and applications perspective," Information fusion, 59 (2020): 84 –102. doi.org/10.1016/j.inffus.2020.01.006.

[12] E. Carrillo, S. Yeotikar, S. Nayak, M. K. M. Jaffar, S. Azarm, J. W. Herrmann, H. Xu, "Communication-aware multi-agent metareasoning for decentralized task allocation", IEEE Access, 9 (2021): 98712-98730. doi: 10.1109/ACCESS.2021.3096229.

[13] A. Abate, J. Gutierrez, L. Hammond, P. Harrenstein, M. Kwiatkowska, M. Najib, M. Wooldridge, "Rational verification: game-theoretic verification of multi-agent systems", Applied Intelligence, 51(9) (2021): 6569-6584. https://doi.org/10.1007/s10489-021-02658-y.

[14] R. Calegari, G. Ciatto, V. Mascardi, et al., "Logic-based technologies for multi-agent systems: a systematic literature review", Auton Agent Multi-Agent Syst 35(1) (2021). doi.org/10.1007/s10458-020-09478-3.

[15] N. Markey, Temporal logics for multi-agent systems, in: 42nd International symposium on mathematical foundations of computer science, 84, 2017, pp. 1-3. https://doi.org/10.4230/LIPIcs.MFCS.2017.84.

[16] V. V. Rybakov, "Temporal multi-agents' logics modelling dynamic temporal accessibility relations", Procedia Computer Science, 207 (2022): 475-482. doi.org/10.1016/j.procs.2022.09.102.

[17] L. Hammond, A. Abate, J. Gutierrez, M. Wooldridge, "Multi-agent reinforcement learning with temporal logic specifications". arXiv preprint arXiv:2102.00582, (2021). https://doi.org/10.48550/arXiv.2102.00582.

[18] L. Lamport, "The temporal logic of actions. ACM transactions on programming languages and systems" (TOPLAS), 16(3) (1994): 872-923. doi.org/10.1145/177492.17772.

[19] N. Axak, A. Tatarnykov, The behavior model of the computer user, IEEE, in: 17th International conference on computer sciences and information technologies (CSIT), 2022, pp. 458-461. doi: 10.1109/CSIT56902.2022.10000499.

[20] N. Axak, M. Kushnaryov, A. Tatarnykov, The Agent-Based Learning Platform, in: XI International scientific and practical conference "Information control systems and technologies", V. 3513, 2023, pp. 263-275.

[21] N. G. Axak, L. E. Gryzun, O. V. Shcherbakov [and others]; in general ed. V. S. Ponomarenko, "Modern information technologies and systems": monograph /– Kharkiv: 2022. – pp. 7–34. URL: http://repository.hneu.edu.ua/handle/123456789/29233. (in Ukrainian).