International Journal of Research in E-learning
Vol. 6 (1), 2020, pp. 1–20

https://doi.org/10.31261/IJREL.2020.6.1.07

Mihail Petrov
Plovdiv University, Bulgaria
https://orcid.org/0000-0001-7144-5881

Vladimir Valkanov
Plovdiv University, Bulgaria
https://orcid.org/0000-0002-9862-3470

Asya Stoyanova-Doycheva
Plovdiv University, Bulgaria
https://orcid.org/0000-0002-0129-5002

Behaviour Analysis of Agents in Virtual Educational Space

Abstract

Modern teaching methods include both traditionally established approaches to the classroom and an entirely virtual approach aimed at learning from anywhere in the world at any time. Society is constantly witnessing increasingly popular platforms such as Coursera, Udacity, Udemy, and even YouTube, which are regarded as aggregators of new knowledge.

The problem that is not addressed, however, is to what extent learning is effective in the process not only of the particular course but also within several interrelated courses of study. This article will describe a conceptual model for analysing the behaviour of intelligent agents who acquire knowledge by using virtual tools to support the educational process, as well as ways to analyse the behaviour of the agents in question.

K e y w o r d s: e-learning, intelligent agents, ITL, virtual educational space, behaviour analysis

E-learning in the last decade has become an integral part of the overall process of acquiring knowledge and skills. The European Community is increasingly actively speaking about the so-called lifelong learning, which is an integral part of the acquisition of a broad spectrum of knowledge to be developed not only within the first years of professional development of future specialists but also later technological development.

The dynamic changes in technology, as well as the increasingly globalised reality of modern society, contribute to the increasing availability of knowledge and skills platforms, such as: Khan Academy, Udacity, Udemy and Coursera. What is common among these platforms is the combination of high teaching potential with the possibilities of modern internet space to provide short and meaningful educational materials presented in the form of a text but most often a video. Professional networks, that is, Pluralsign, Laracast, and Envato Tuts+ are also a specialised part of the development of knowledge and skills. These platforms specialise in aggregating and documenting useful practices, approaches, and lessons related to software product development.

The tendency for the development of e-learning does not go beyond the university institutions, which develop their platforms to support the education process of their alumni. Within this trend, a distributed Learning Center for e-learning within the Mathematics Faculty has been developed to provide a platform for Information Technology students in which they can learn, exercise, and consolidate their knowledge (Stoyanov, 2016, 72–82).

The abovementioned platforms successfully deal with problems of distance learning, reference and knowledge delivery at any time (Stoyanov, 2010, 11–16), however they often do not take into account the behaviour of trained agents and thus make it extremely difficult to track the effectiveness of the efforts made.

It is important to note that the common idea of e-learning is by itself not enough. The fact that education is using the Internet as a means of communication does not provide any value except for the fact that everyone can access a specific information resource remotely. This concept is already proven to provide the necessary level of utility but it does not work on its own. Technologies are developing fast enough to allow access to comprehensive tools. Some of the most recent examples of such capabilities are:

– real-time training and real-time communication online;
– use of hardware components to further build efficiency in online platforms also called the Internet of Things;
– analysis of learners' behaviour and needs;
– a personal advisor who enhances and improves the learning experience;
– augmented reality and virtual reality that materialises the subject right in front of the students' eyes.

As researchers, we cannot overlook the trends, which is why the DELC platform has slowly begun to transform into a project that brings together the concepts of the so-called virtual learning space – VES (Valkanov, 2015, 322–326). The foundation of the concept is blurring the line between the virtual and the physical world. The sense of physical surroundings is presented by living breathing agents interacting with a collection of sensors strategically positioned around them. The user interacts with the sensor actively or passively and feeds the information stream to the online platform that is responsible for the decision-making process.

The UniPlayground project is an independent platform built around the concepts defined by the VES, which aims to solve a range of problems that are essential to online education not only at the university level but overall. The platform aims to create a behaviour model relevant to the educational process agents' activities and to introduce a model based on it where the educator or the platform itself can act according to students' needs (Stoyanov, 2018, 20–28).

The main target audience of the project is students or agents interested in informatics and technology. Therefore, the platform is built around a set of programming tools that emulate a real programming environment, providing all of the necessary tools for the agents to exercise, learn, and revise their skills.

The following paper is dedicated to the problems of the modern e-learning process and the potential solutions that can be introduced in the face of the UniPlayground project.

E-learning Based on Web Information System

E-learning is not possible if the process is not implemented within a framework of a specific information system that allows supervisors to constantly monitor the behaviour process of learners (Popchev, 2019). There are several specifications describing processes where both training and verification of the learning material can be carried out. An example of such a concept is the SCORM standard that describes the technological aspect of verifying acquired knowledge and skills. Systems should not be viewed in isolation but they should always be classified in several ways (SCORM Explained).

Passive and Active Systems

Depending on the level of interaction between the information system, it can be divided into active and passive. Passive systems integrate only components aimed at acquiring knowledge, regardless of additional activities related to user interaction and obtaining additional information from its actions.

General-purpose and specialised. If the system is aimed at users with a broad knowledge base and does not imply a specific approach to knowledge verification, then this system meets the general conditions and cannot be labelled general-purpose. The specialised approach requires the provision of processes, whose understanding is part of the professional training of a specific professional group or a precisely specified set of pre-trained agents who can cope with the knowledge.

Behaviour-driven and data-driven. The systems that are developed within our database are called data-driven. They develop an asset of knowledge that depends only on the efforts of the service agents. The change of knowledge will affect already active information units by expanding or updating them. Systems based on consumer behaviour develop this idea by adding the unit to study the behaviour of learners as well as training agents. With the development of knowledge and skills, the system learns and adapts to the current agent's approach and provides a different experience depending on a set of achievements over time.

Problems in the Analysis of Educational Results

When we talk about such an analysis of the results, we have to ask a few questions: (1) What are the characteristics of the agents that work with the system?; (2) What kind of problems are solved?; (3) What is the agents' activity?; (4) When is the interaction with the system and what is its duration? The answer to each of these questions can give us key characteristics of the necessary conditions we have to achieve to say that we can judge whether the behaviour of the agents can be adequately analysed (Kehayova, 2016, 784–788).

What are the Characteristics of the Agents that Work with the System?

It is a fundamental question we need to ask before we start analysing behaviour. In the context of software education, the concept that is being studied is often the study of technology by writing a software code using language constructs that we call programming language (Stoyanova-Doycheva, 2019). The understanding of this matter can be enhanced or deepened if the agent whose behaviour we analyse is studying the subject. Then this activity can be reduced but is not limited to:

– studying the material at school;
– active work on projects related to building software products;
– interest in areas of knowledge that have a strong correlation with the relevant problem, such as the field of mathematics or other engineering sciences.

Knowledge of such activities in the agent's past enables us to assess to what extent the complexity of the outcome that would be achieved by the agent within the course would be due to the knowledge gained during the course of engagement with the subject matter or experience before the course. Such a definition is also a point of departure in the analysis of agent behaviour within related courses, thus making it very clear whether the results of past activities have been maintained and used adequately in current ones.

How can we get information about the agents' knowledge? The technical problem of finding preliminary agent data can be divided into two parts, namely, collecting preliminary knowledge and gathering knowledge within their activity. Within the course, the activity of the agents is assessed by means defined by the supervisor of the discipline, but within the preliminary stage when knowledge is minimal, two approaches can be used.

Preliminary knowledge survey. This is a set of questions aimed at directing the supervisor to the course in which it will be most appropriate for the trained agent's purposes (Todorov, 2016, 753–757). The system used for training can be based on the agents' responses as well as a small sample of their core skills. It can provide activities appropriate to the agent's level. The point is that not all agents have an equal start, experience or educational background. The ability to learn is a complex set of skills that is further supported by the agent's professional experience or the time spent studying the relevant problems in high school or university (Stoyanova-Doycheva, 2019).

Let us approach the problem with an example. Student A and Student B have the same educational statistics; they have the same academic background and at this moment in their life they are trying to learn a specific programming technology. Student A spends his/her time attending lectures or learning from notes and watching video tutorials and explanations, trying to make sense of the problems. Student B spends the majority of his/her time outside of school and does not attend any classes. The result is that the score of the two students is nearly identical. Why does Student B score a nearly perfect score? The answer to this question requires an in-depth analysis. Before we answer, we should ask the following: what is the subject that the two students are learning? and what is the background of Student A and Student B?

It turns out that the technical subjects that suggest a lot more practical skills based on experience and repetitions are handled better by students that are currently working or already have experience with them. It is no secret that the IT industry is expanding at an incredible pace and the majority of the employees are young people who have daily working experience with technology. So, in this example, the experience of Student B is giving him/her a head start over the majority of the efforts that Student A will make during a fixed period. This example also outlines another problem that the survey can solve: namely, the problem of wasting the time and effort of a student who has prior knowledge regarding the subject. For the system to be fair, we must challenge the students regardless of their level of knowledge. Therefore, having a piece of information in advance, we can approach the problem intelligently and be very flexible in the assignments that the supervisor gives to every single student.

Data derived from previous courses. As previously mentioned, the analysis of agent behaviour helps provide information within related courses. This information is much more valuable than the pre-order because it provides an up-to-date and completely objective starting point for the agent's level based on the knowledge and skills shown at this stage of his or her training.

What Kind of Problems They Solve?

Not every problem can be adequately addressed and classified when the agents actively communicate with an information system; the responsibility for their assessment can be divided into three parts.

Evaluation of the system itself. Evaluation of the system itself by means of techniques such as artificial intelligence, machine self-learning, and temporal analysis. This analysis may be sufficiently objective because it takes into account only the input data of the agent without calling into question characteristics such as racial classification, sympathy for the agent, other characteristics that may increase or decrease the assessment coefficient by its final or intermediate performance.

Teacher assessment. Teacher assessment, based on the impressions of one or several course managers, is primarily subjective because of the factors mentioned above, however, with much more weight than the machine-generated one, due to the specific circumstances which are important for the course, defined and conducted by the same lecturer.

Assessment from another student. This is another course participant who has faced the same or a similar set of problems and can give an adequate assessment of the work of an equal subject. The advantage of this activity is that usually the weight of the final score is distributed among a group of active agents, and thus the average score has a great potential to be closer to reality than the subjective assessment of the other two parameters. The main negative aspect is the need for a fine calibration in which it is quite often necessary to intervene in the above two agents' activities to obtain a definitive result and to minimise the injury to the evaluated agent. To classify the behaviour of the agent, the system must keep a record of the actions taken, as well as information on the evaluation of each agent. The examples that will be given in this article are related to the development of software products within the context of writing and analysing software code.

What is the Agents' Activity?

At the core of the behavioural analysis is, of course, the agents' activity. To make the correct evaluation, the theory of interval temporal logic is used, which is a flexible methodology for describing processes that occur over time. Interaction of agents with the system is done through a set of activities that can be classified in general in the following categories:

- getting knowledge: Activities related to the perception of information in any format, text, illustrative or video content.
- knowledge check: Agents solve a problem defined by their supervisor or system.
- assessment of knowledge: Agents assess the problems solved and consolidate their knowledge by classifying the work of other agents based on their current ones.

The process is collaborative, with a system that records a file of all committed actions. To be effective and reliable, the information collected should be often obtained and sufficiently complete, and, in particular, not compromised by technical problems or negative behaviour of the calibrating agents.

UniPlayground Project Architecture

The system is conventionally divided into two large modules that manage the main application capabilities. Administration and supervision is a component that contains the main interfaces to manage ongoing courses and projects that a user interacts with. At the same time, it also combines administrative interfaces to create new or up-to-date current learning processes.

Jobseekers are in continuous communication with the other modules of the system, and they play the role of a client application. It is important to emphasise that the client part communicates with the other components using a set of system components called supervisors that support the application management process between the two layers of the application.

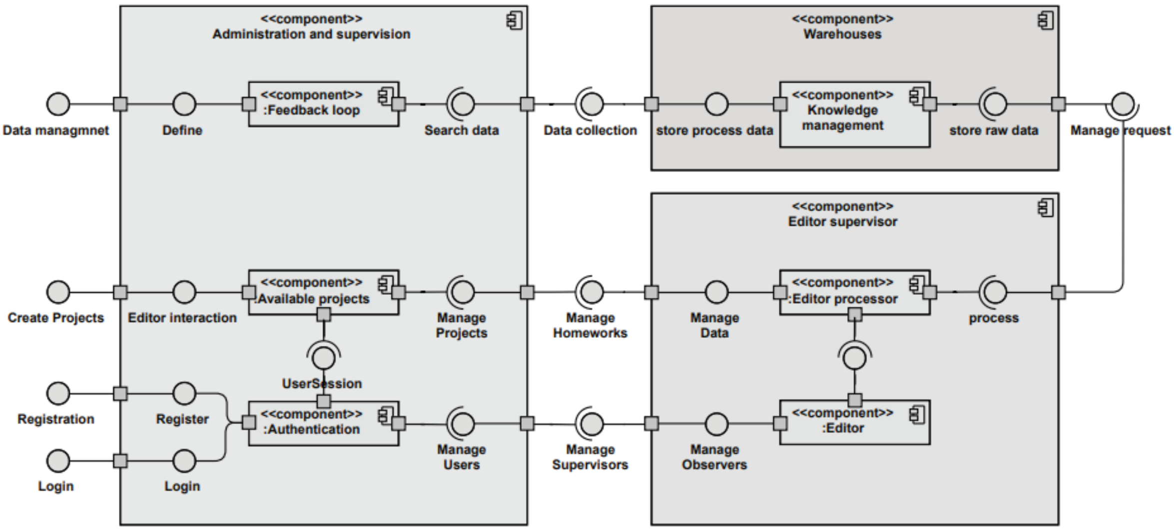

Figure 1. Communication process in UniPlayground.
S o u r c e: the author's own work

This division allows flexibility when introducing different client implements within the platform. At present, the development is based on a web application due to the specifics of the sessions, but without any logic. Logic can be delivered within a mobile application or even an Internet of Things application, that is, a stand-alone hardware component that interacts with the environment.

As mentioned before, the virtual learning space is a concept that blurs the boundary between the physical world and the virtual world, making it possible for interfaces to be replaced by devices that are not traditionally characteristic of day-to-day operation with information systems such as cameras, motion switches, and motion sensors.

Knowledge Supervisor

The supervisor module editor is the main system for managing the interaction between the learner and the system. There are the following important tasks:
Educational tasks
– management of the learning process;

– tutorials;
– returning information.

Analytical tasks
– loop CPU management;
– managing a bug box;
– interface communication;
– communication with the analytical module.

Data Supervisor

Considering the consistency of data transfer, not all information that is generated is equally important for the final analysis of the agent behaviour. For example, a majority of the information that is generated by operations regarding search and reading of publications is classified as important because it gives the analysing algorithm a context for measuring user expectations. The listed component flags the input data generated by the collection of event processors and labels it by categories. The classified data is then sent to a queue which is going to be resolved in a later state of the data process life-cycle.

Knowledge Supervisor

The knowledge management and analysis of user behaviour are done in the so-called Data analysis and Warehousing module of the platform. Information is not analysed in real-time but is subject to storage and processing over a long period. The main tasks of the module are locked into the following activities:
– storage of non-normalised information resources;
– classification of input data from solving information problems;
– classification of input data received when completing survey resources as well as user behaviour within the system, reading articles, links, and other information resources;
– processing of temporal activities that are viewed at a given interval in time and space;
– the output of this module builds profiles of all participants in the training process. Profiles serve both the individual guidelines of each agent and the validation of the flow of actions the teacher takes in the learning process.

Example

If it is necessary to conduct a course for beginners, and the level set by the supervisor is too high, the results will be reflected in the final report, taking into account the previous experience of all the students and thus easily defining a model to serve as the basis of evaluation for the actions taken and their validity towards the environment. Because of that, a model of the training process that is valid for the system parameters will be created. Data collection for the analysis goes through several important phases in which different components of the system are responsible for the correct distribution of the received information resource. The process is divided into four parts:

– data collection – data collection is possible through an application interface. The common interface for approaching behaviour analysis is a web or desktop-centric application that feeds itself with user input. But the modern approach to this problem can be achieved using mobile applications for tracking data activity and also an Internet of Things-based approach for storing data based on user events/movement and operations with the surrounding environment.
– data transformation;
– data classification;
– data analysis and processing.

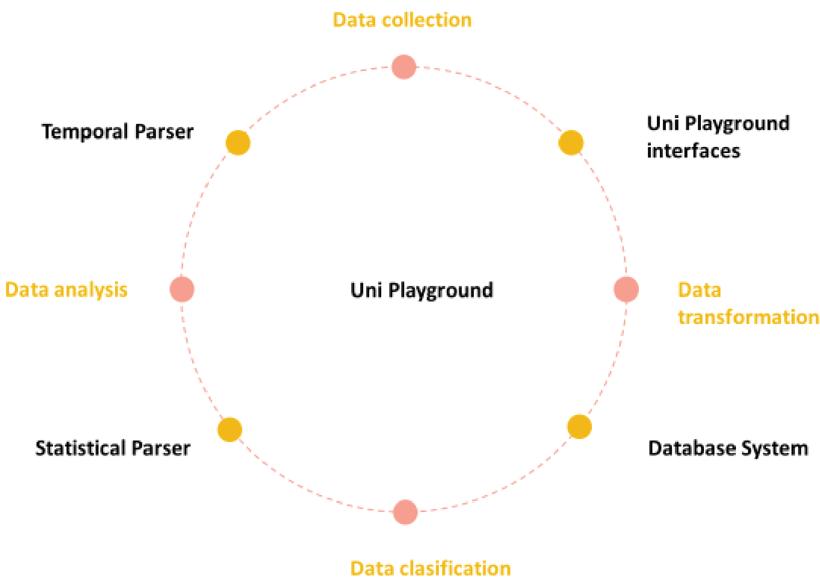

Figure 2. Data analysis and collection cycle
S o u r c e: the author's own work

Implementation of the UniPlayground Project

The implementation of the project is divided into three stages:

– development of the application core;

– development of application modules providing an environment for conducting a specific discipline;
– development of a temporary parser to analyse the obtained data results.

The UniPlayground information system is organised as a set of information services, combining a poly-technological approach to solving a wide range of issues. In its overall model, the system consists of three components:

– client part – providing system interfaces;
– administration layer – provides functionality for both the teaching process management and the overall process of analytical tool definition for analysing activities specific to the supervisor;
– aggregation layer – receives, processes, and transforms received data from the system;
– temporal layer – takes care of the classification and analysis of the processed data;
– service layer – implements the functionalities of the individual training modules as well as the general logic of the information system.

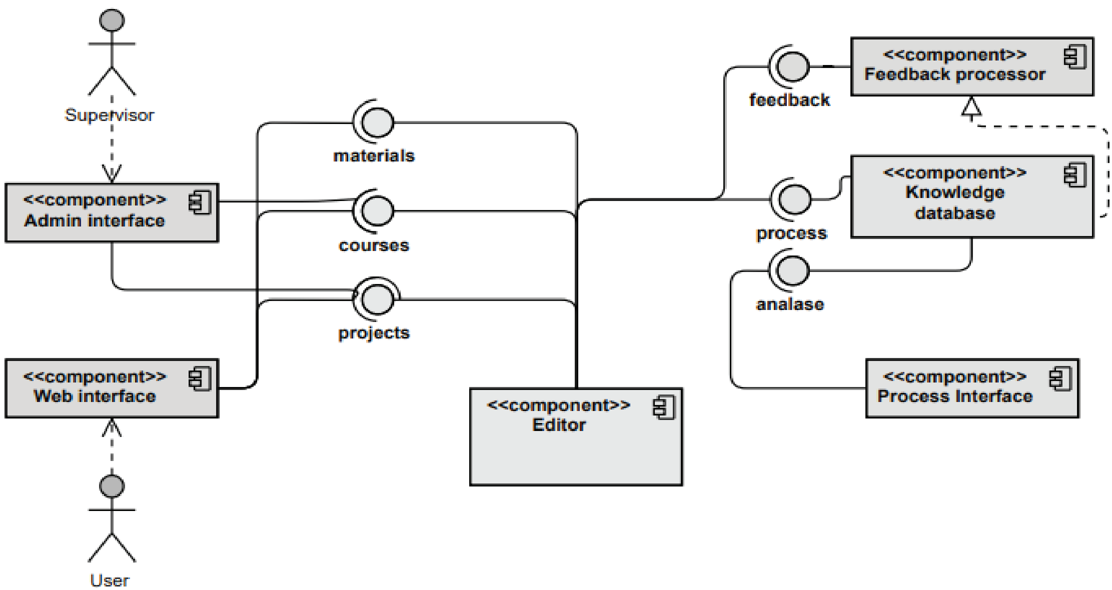

Figure 3. Process of user communication.
S o u r c e: the author's own work

The Multi-Layer Division provides exceptional flexibility in distributing applications, both between individual information infrastructures, but also across multiple different types of client environments. Learning in a virtual environment is an interactive process that involves the use of devices of any kind, both static and mobile. The client part of the application is completely independent of system logs and can be subject to both the specific needs of the school or academic establishment if the classes are held at school, as well as the specific needs of the users if the classes are open to the general public. The main component of the so-called interfaces is graphic editors. Focusing primarily on the needs of software training, the system aims to provide a specific implementation that addresses the needs of learning in a particular programming language. The graphics editor aims to organise a set of controls to communicate with the service layer of the system and provide the necessary information to the users. Actions typically associated with learners' work with graphic editors can be divided into the following categories:

– defining a program component;
– program component execution;
– obtaining system results.

Each program component is associated with a specific graphics editor that aims to communicate with the service module specified for that purpose. The service modules form the overall functional model of the application. They are heterogeneous in terms of their programming logic and can be defined in any programming language. They are organised around the concept of separating the tasks of independent components and thus avoid the monolithic approach at the expense of easy and independent distribution within any information infrastructure. The static processes are:

– authentication and authorisation of user agents;
– managing educational processes;
– defining content;
– managing interface communication – graphic editors.

Dynamic processes are all processes that aim to retrieve information about a particular subject or group of subjects based on changes in the environmental characteristics.

Collecting and Using Results Based on the Specific Time Interval

An important part in the process of managing consumer behaviour plays the layer analysing the evolution and revolution of user actions over time. Such an analysis enables the interested units in the system to gain a clear idea regarding the movement of the learner in the educational process as part of their training, while also enabling the trainee's potential to develop their knowledge within the framework of a variety of courses involving related content that builds on existing ones.

For the needs of the application, it is necessary to build a formal model and on its basis to develop an algorithm for monitoring and classifying such behaviour. The actions of the user will always be considered at some time, with predefined characteristics of their knowledge and skills. For this reason, we can use the concepts of temporal logic and, in particular, temporal logic intervals, which offer a convenient formality, providing a set of logical constructs that help us achieve this task.

At present, the integration of time interval logic management tools can only be linked to the development of Ben Moskowski and his Tempura interpreter, which can process complex ITL expressions and produce a logical final result. The demonstration of whether a system is valid or not can be found in the answer to the logical statements aimed at checking a set of facts that are considered at the time of the analysis (Moszkowski, 1985).

Direct integration of the temporal interpreter is not an adequate solution, given that data analysis can be limited to reducing all input arguments to a set of interval temporal expressions. For this purpose, a tool for the runtime verification of systems using Interval Temporal Logic (ITL) has been developed. It provides syntactic constructs that are equally feasible by the Tempura interpreter. The basic concept provided by the tool is the ability to dynamically verify the system's status.

The main problem with the development is that they do not provide adequate tools for easy and fast development of system verification conditions, namely, the required statements whose validity we are testing are subject to research by the developer. Unfortunately, they are not at all trivial for development and quite often involve knowledge of the whole system, which can hardly be obtained if we need to analyse a small data set.

For this reason, it was necessary to create a suitable interface for the already existing technologies for which the following features are required:

- easy integration within existing systems without the need to write additional source code;
- ability to access knowledge databases that can be used directly in expressions for validation and proof of status of the system under consideration;
- ability to add external services, libraries, and scripts to classify data arrays;
- introduction of an easy syntax to define expressions that closely resemble a high-level programming language that is suitable and convenient to work by both software developers and agents not familiar with the logic of interval time processes.

To fulfil those conditions, it was necessary to define an additional layer of communication called System Tempura, that is, an independent interpreting mechanism managing the interval temporal processes into the Web-based infrastructure. Figure 3 shows a sample diagram of the communication between the System Tempura module and the software solutions on which the tool is based.

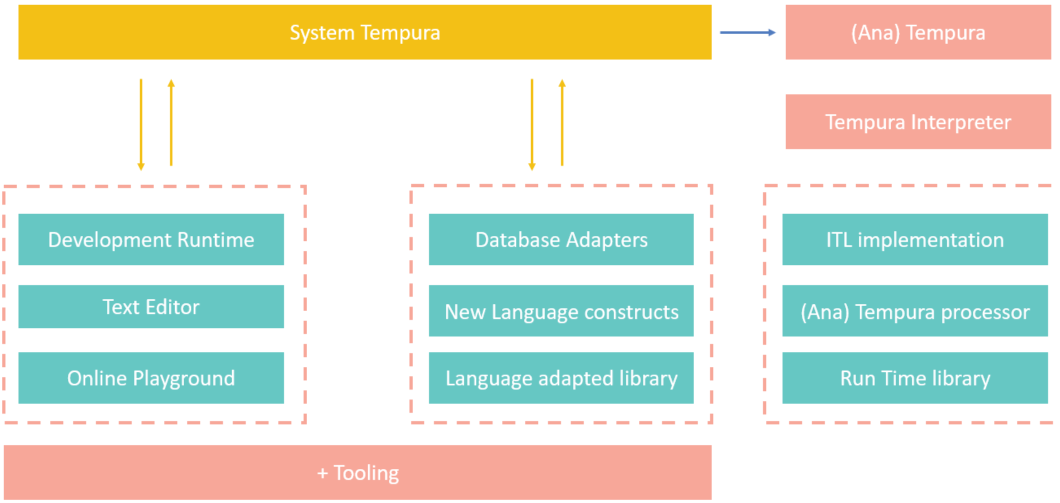

Figure 4. Communication process in System Tempura.
Source: the author's own work.

The communication between different layers of the interpreter mechanism is accomplished by HTTP communication, which makes the integration of the component fully flexible and does not interfere with existing systems. The Interpretation Mechanism aims to analyse expressions and sequences called formulas that describe interval logic aimed at analysing the behaviour of one or a group of learning agents in the context of the UniPlayground environment or any other one that meets similar information requirements. To organise a proper communication environment, the System Tempura interpreter manages its input process using additional tools embedded in the UniPlayground information system. Such tools are editors to manage language formulas, as well as a management environment for the formula implementation process. All processed queries are transcribed to an equivalent AnaTempura code and serve as an entry point for the interpretation of related processes in the context of which the final result of the defined formula is reached.

Behaviour Analysis in E-learning Platforms

Systems generating information on the behaviour of trainees provide an adequate and fairly detailed view of the educational behaviour of anyone actively working with them. Lack of data itself is also a valid statistic that can easily show dissatisfaction, lack of time or misunderstanding of material. The processing of this information over time can lead to the disclosure of interesting templates and relationships between the context of the environment, the teaching material, and the qualities of the agents currently being trained in a particular subject or matter.

The UniPlayground system examines a set of concepts described by enabling the training staff to define and diversify the system to collect learner behaviour data in a way that does not compromise the overall interaction with the system and the other agents in the system. The results of this research are going to be processed over three years of education and internal training classification of the work and development of students at Plovdiv University. Any valid statistical information is going to be published in order to present a complete theoretical model of behaviour analysis of agent behaviour based on a specific educational topic and representative sample.

The Study Results and Discussion

The experimental application testing was conducted by the use of one base group and two experimental groups in the duration of one academic year. The base group consisted of students with different experience in computer science and a different level of knowledge and educational background. The base group experienced the educational process by using only guidance directly influenced by the core facilitator team of the course. The other two control groups experienced the same education process but using only the application processes developed in UniPlayground by the same team. The resource utilisation and application exercise analysis was approximately three times greater compared to the traditional educational process. Information regarding the parameter of the base and the control group is presented in Table 1.

Table 1
Base experimental group parameters (the author's own work)

Academic year	Total participants	Participants with IT experience	Participants with no previous knowledge in IT
First	2	1	1
Second	10	5	5
Third	10	5	5
Fourth	10	5	5

Table 2
Control experimental group parameters (the author's own work)

Academic year	Total participants	Participants with IT experience	Participants with no previous knowledge in IT
First	2	1	1
Second	10	5	5
Third	10	5	5
Fourth	10	5	5

The time spent facilitating already developed exercises was cut in half thanks to the built-in analytics process. The single most important improvement was found in the overall track record of incorrect behaviour. For every single student, the facilitator had access to the full range of mistakes done over the course of their education and also the range of improvement based on faculty guidance.

There is no doubt that for a correct assertion of the provided results, it is necessary to have the corrected data spawn across a significant period for the results to be critically analysed and assessed. Using the long-term data classification process is necessary for predicting a range of behaviour that is going to be observed if the control group consists of students that exhibit a specific range of behaviour or have a collection of specific strengths and weaknesses necessary to achieve results over a specific course. The observed results can be generalised and used for a broad range of online education and electronic education at the university or even at a company level.

Acknowledgements

The research is partly supported by the project FP19-FMI-002 "Innovative ICT for Digital Research Area in Mathematics, Informatics and Pedagogy of Education" of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

Conclusion

One of the most challenging topics related to e-learning today is that of evaluating the work and competences of learners within the course. In this line of thought, the involvement of the teaching staff as well as of the supervisors and assistants responsible for conduct or successful facilitation must be taken into account. The answer to this problem will enable us to develop adequate tools that will give us more accurate information about the statistical status of each agent as well as the overall commitment of course facilitators to the work they have done.

References

Alfred, A. (2014). *Compilers: Principles, techniques, and tools*. New York: Pearson Education Limited.

Allen, J., & Ferguson, G. (1994). *Actions and events in interval temporal logic technical report*. Rochester, NY: University of Rochester.

Cau, A., Moszkowski, B., & Zedan, H. (2006). *Interval temporal logic*. Leicester: De Montfort University.

Dutertre, B. (1995). Proceedings of Tenth Annual IEEE Symposium on Logic in Computer Science. Complete proof systems for First Order Interval Temporal Logic, IEEE, https://doi.org/10.1109/LICS.1995.523242

Glushkova, T., Stoyanova, S., & Stoyanova-Doycheva, A. (2018). Internet of Things ecosystem supporting e-learning. In E. Smyrnova-Trybulska (Ed.), E-learning, vol. 10: E-learning and Smart Learning Environment for the Preparation of New Generation Specialists, (pp. 409–425). Katowice–Cieszyn: Studio-Noa for University of Silesia Press.

Kehayova I., Malinov P., Valkanov V., & Doychev E. (2016). Architecture of a module for analyzing electronic test results. Proceedings of 2016 IEEE 8th International Conference on Intelligent Systems, Sofia, Bulgaria, pp. 784–788, https://doi.org/10.1109/IS.2016.7737402

Moszkowski, B. (1985). *Executing temporal logic program*. Cambridge: University of Cambridge.

Popchev I., Orozova D., & Stoyanov S. (2019). IoT and Big Data Analytics in E-Learning, Big Data, Knowledge and Control Systems Engineering, BdKCSE 2019, Sofia, Bulgaria, 21–22 November, https://doi.org/10.1109/BdKCSE48644.2019.9010666.

Stoyanov S., Valkanova V., Ganchev I., & O'Droma M. (2010). An approach and architecture supporting context-aware provision of mLearning services, 2nd International Conference on Mobile, Hybrid, and On-Line Learning, eL and mL 2010, Saint Maarten, Netherlands, pp. 11–16, https://doi.org/10.1109/eLmL.2010.14.

Stoyanov, S., Stoyanova-Doycheva, A., Glushkova, T., Doychev, E., & Todorov, J. (2018). A reference architecture of internet of things ecosystem. *Компютърни науки и комуникации* [Computer Science and Communication], vol. *7*(1), 20–28.

Stoyanov, S. (2012). Context-aware and adaptable e-learning systems (PhD Thesis). Software technology research laboratory, De Montfort University, Leicester, UK.

Stoyanov, S. (2016). A virtual space supporting e-learning. Proceedings of the Forty Fifth Spring Conference of the Union of Bulgarian Mathematicians. Pleven, April 6–10, 2016, 72–82.

Stoyanov S., Valkanov V., Popchev I., Stoyanova-Doycheva A., & Doychev E. (2014). A model of context-aware agent architecture. *Comptes rendus de l'Académie bulgare des sciences: sciences mathématiques et naturelles, 67*(4), 487–496.

Todorov J., Stojanov S., Valkanov V., Daskalov B., & Popchev I. (2016). Learning Intelligent System for Student Assistance – LISSA, Proceedings of IEEE 8th International Conference on Intelligent Systems, Sofia, Bulgaria, pp. 753–757, https://doi.org/10.1109/IS.2016.7737397.

Valkanov V., Stoyanov S., & Valkanova V. Building a virtual education space, WMSCI 2015 – 19th World Multi-Conference on Systemics, Cybernetics and Informatics, Proceedings Vol. 1, 2015, pp. 322–326, ISBN: 978-194176329-2.

Mihail Petrov, Vladimir Valkanov, Asya Stoyanova-Doycheva

Analiza zachowania agentów w wirtualnej przestrzeni edukacyjnej

Streszczenie

Nowoczesne metody nauczania obejmują zarówno tradycyjnie ustalone podejście do klasy, jak i całkowicie wirtualne podejście mające na celu uczenie się z dowolnego miejsca na świecie w dowolnym momencie. Społeczeństwo stale obserwuje coraz bardziej popularne platformy, takie jak Coursera, Udacity, Udemy, a nawet YouTube, które są agregatorami nowej wiedzy.

Problemem, który nie został rozwiązany, jest jednak to, w jakim stopniu uczenie się jest skuteczne w obrębie nie tylko danego kursu, ale także w ramach kilku powiązanych ze sobą kursów. W artykule opisano model koncepcyjny, służący do analizy zachowania inteligentnych agentów, którzy zdobywają wiedzę za pomocą narzędzi wirtualnych w celu wsparcia procesu edukacyjnego. Opisano także sposoby analizy zachowania danych agentów.

W artykule przedstawiono koncepcję środowiska w analizie zachowań agentów UniPlayground, a także różne obszary wiedzy korzystające z tego środowiska.

Behaviour Analysis of Agents in Virtual Educational Space

S ł o w a k l u c z o w e: e-learning, intelligent agents, ITL, wirtualna przestrzeń edukacyjna, analiza zachowań

Mihail Petrov, Vladimir Valkanov, Asya Stoyanova-Doycheva

Поведенческий анализ агентов в виртуальном образовательном пространстве

А н н о т а ц и я

Современные методы обучения включают в себя как традиционно установленные подходы к занятиям в классе, так и полностью виртуальный подход, направленный на обучение в любой точке мира в любое время. Общество постоянно отслеживает все более популярные платформы, такие как Coursera, Udacity, Udemy и даже YouTube, которые являются своего рода агрегатором новых знаний.

Однако проблема, которая не решается, заключается в том, в какой степени обучение эффективно в процессе не только конкретного курса, но также и в рамках нескольких взаимосвязанных курсов обучения. В этой статье будет описана концептуальная модель для анализа поведения интеллектуальных агентов, которые получают знания с использованием виртуальных инструментов для поддержки образовательного процесса, а также способы анализа поведения рассматриваемых агентов.

В статье будет описана концептуальная архитектура среды для анализа поведения агентов UniPlayground, а также различных областей знаний, использующих эту среду.

К л ю ч е в ы е с л о в а: электронное обучение, интеллектуальные агенты, ITL, виртуальное образовательное пространство, анализ поведения.

Mihail Petrov, Vladimir Valkanov, Asya Stoyanova-Doycheva

Análisis de comportamiento de agentes en espacio educativo virtual

R e s u m e n

Los métodos de enseñanza modernos incluyen enfoques del aula tradicionalmente establecidos y un enfoque completamente virtual destinado a aprender desde cualquier parte del mundo en cualquier momento. La sociedad observa constantemente plataformas cada vez más populares, como Coursera, Udacity, Udemy e incluso YouTube, que son una especie de agregador de nuevos conocimientos.

Sin embargo, el problema que no se aborda es en qué medida el aprendizaje es efectivo en el proceso no solo del curso en particular sino también dentro de varios cursos de estudio interrelacionados. Este artículo describirá un modelo conceptual para analizar el comportamiento de los agentes inteligentes que adquieren conocimiento utilizando herramientas virtuales para apoyar el proceso educativo, así como las formas de analizar el comportamiento de los agentes en cuestión.

El artículo describirá una arquitectura conceptual para un entorno para analizar el comportamiento de los agentes de UniPlayground, así como las diferentes esferas de conocimiento que utilizan este entorno.

Palabras clave: eLearning, agentes inteligentes, ITL, espacio educativo virtual, análisis de comportamiento.