

## Article

# Integration of STEM Centers in a Virtual Education Space

Stanimir Stoyanov <sup>1,2,\*</sup> , Todorka Glushkova <sup>1</sup> , Veneta Tabakova-Komsalova <sup>1,2</sup> ,  
Asya Stoyanova-Doycheva <sup>1,2</sup> , Vanya Ivanova <sup>1</sup>  and Lyubka Doukovska <sup>2</sup> 

<sup>1</sup> Department of Computer Systems, University of Plovdiv, 4000 Plovdiv, Bulgaria; glushkova@uni-plovdiv.bg (T.G.); v.komsalova@uni-plovdiv.bg (V.T.-K.); astoyanova@uni-plovdiv.bg (A.S.-D.); vantod@uni-plovdiv.bg (V.I.)

<sup>2</sup> Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria; l.doukovska@mail.bg

\* Correspondence: stani@uni-plovdiv.bg; Tel.: +359-88-8318164

**Abstract:** Training in STEM centers is becoming increasingly relevant in order to meet the new educational needs caused mainly by the rapid speed with which new technologies have been entering our lives in recent years. Existing university e-learning systems can enhance the capacity of these centers by providing shared learning material. This paper presents a distributed educational platform that supports sharable usage of teaching material at university and in STEM centers in secondary schools. The architecture of the platform is also presented including two basic components. The university e-learning environment operates as a back-end and the front-end component is located in the STEM center. Furthermore, the implemented prototype of the platform is also discussed in the article. The use of the platform is demonstrated by two educational games. The platform is expanded with four educational robots to increase the attractiveness of the educational process.

**Keywords:** e-learning; game-based learning; STEM; integrated domains; virtual education space; software platform; personal assistants

**MSC:** 68T42; 68M14; 97U50



**Citation:** Stoyanov, S.; Glushkova, T.; Tabakova-Komsalova, V.; Stoyanova-Doycheva, A.; Ivanova, V.; Doukovska, L. Integration of STEM Centers in a Virtual Education Space. *Mathematics* **2022**, *10*, 744. <https://doi.org/10.3390/math10050744>

Academic Editors: Mikhail Noskov, Alexei Semenov and David Pugalee

Received: 22 December 2021

Accepted: 24 February 2022

Published: 26 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Educators are looking for more and more interesting and attractive forms of e-learning, especially in times like the ongoing crisis related to the COVID-19 pandemic. Creating meaningful and attractive teaching content that arouses interest and keeps learners' attention is a hard task. Therefore, the repeated use of ready-made teaching material tested in a real educational process and the ability to adapt it to various groups of learners and avoid the need for redevelopment are essential requirements for modern e-learning systems. Furthermore, game-based learning brings additional attractiveness to e-learning.

Our team from the DeLC Laboratory at the University of Plovdiv has been developing the VES (Virtual Education Space) learning platform [1] as a cyber-physical and social space for years. Taking into account these challenges, we decided to expand and modernize VES, which we use for real education of students at the university to support secondary school education and in particular for game-based education in STEM centers.

The new extended platform called ViToS (Virtual education space To Stem) implements the idea of integrated domains. Various advantages and benefits of implementing the concept of integrated domains can be pointed out. An important utility is the shared use of developed and well-functioning information resources—in this case, teaching material and e-learning tools. Such an integrated domain created by e-learning at university and e-learning in STEM centers in secondary schools is presented in this article. In the present case, the shared resources are the teaching material, created at the university, and the supported tools. Furthermore, educational robots are used to increase the attractiveness and also to insert the topics in the teaching material in the environment.

In this article, ViToS is presented. The next section provides a brief overview of the current situation with the construction and use of STEM centers around the world. The third section presents the Virtual Education Space and its components. The fourth section is dedicated to the ViToS platform. The fifth section demonstrates the usage of the platform by means of two examples. The sixth section presents the enhancement of the platform by educational robots. The article ends with a conclusion and a future work section.

## 2. STEM Centers

Training in STEM (Science, Technology Engineering, and Mathematics) centers is becoming increasingly relevant in order to meet the new educational needs caused mainly by the rapid speed with which new technologies have been entering our lives in recent years. STEM centers create an environment for integrated learning in various disciplines in specific curricula. STEM centers organize various interest clubs, which include training on topics related to robotics, computer science, science, and mathematics. In recent years, various international projects have been implemented at European and global level, aimed at supporting the training of students in school STEM centers. Various associations and organizations are being set up to achieve this goal [2]. In this context, the European Community pays attention to projects such as Erasmus + “DO WELL SCIENCE” [3], the purpose of which is to develop a software architecture designed to build tools such as e-learning environments, learning management systems, and content management systems supporting training, assessment, and validation of students’ competencies. Furthermore, it is possible to update the request for wide representation and dissemination of knowledge and experience in innovation in different schools. The “Do Well Science” project is in line with the horizontal priorities identified by the European Commission. It seeks to validate an innovative method and build a database of exercises, problems, and tasks related to the disciplines of mathematics, physics, and science to improve students’ skills, as well as to share results with other students who are also from across the European community. #EuroFactor [4] of the European Commission and the European Parliament is another project that seeks to stimulate young people to study science, technology, and information technology, given the new job opportunities and skills required by the market. A third example is the EU STEM coalition [5], which is a European network of national STEM platforms aimed at close cooperation between the government, education, and industry and it has strongly regionalized its implementation. STEM platforms are organizations usually set up by governments to increase the number of STEM graduates and reduce skills gaps. In the long run, the EU STEM Coalition aims to reduce the skills gap by having a national STEM strategy in each of the EU member states.

Currently, there is no completed policy regulating the implementation of STEM education in Bulgaria. However, Bulgaria participates in the EU STEM coalition. According to the Bulgarian Ministry of Education and Science, the Bulgarian education system traditionally supports STEM, providing students with many opportunities to expand their experience in the field of STEM outside the educational program. The Ministry of Education and Science has established a National Program “Building a school STEM environment” (2020) aimed at increasing students’ interest in science and technology by supporting the establishment of school centers with a focus on STEM (Science-Technology-Engineering-Mathematics). They will provide all the necessary conditions for conducting modern and quality STEM training in school as well as create the basis for the future achievement of young scientists. STEM centers represent the modern rethinking of specialized rooms and computer labs in today’s schools, as well as the creation of new spaces according to the requirements of modern technology industries. The main goal of the program is the creation of a unified learning environment in Bulgarian schools, motivation, and support of innovations in the teaching and learning of natural sciences, creativity, and research (STEM). The national program will form STEM-schools, which will be models for the implementation of the investment, including teacher training, integrated content in science, technological inclusion, and management of electronic resources.

In Russia, STEM education is associated primarily with the system of additional and extracurricular education, and it is developing very dynamically today [6]. In the immediate practice of education in Russia so far, we are aware of two educational projects that are aimed at STEM. The Institute for the Study of Childhood, Family, and Education is developing the “STEM education for children of preschool and primary school age” program [7]. A unique format of science education is designed and implemented at the ЧОУ «Хорошевская школа» (Horoshkola) thanks to the special educational space and the non-traditional organization of the educational process. Creating different types of conditions, in which research in natural sciences is conducted, makes possible a significant restructuring of the learning process and its content. The aim of the science education of Horoshkola is to form the ability to independently solve various life tasks in a modern, rapidly changing, high-tech world in accordance with the research base, a project approach, 21st century skills, and the possession of basic scientific concepts and tools.

The STEM movement has spurred the impetus for new and effective professional development in the United States over the past two decades. Research in Canada concentrates on teachers’ perceptions of what constitutes effective professional development in STEM education. If instructors need to become confident and comfortable in teaching STEM subjects, it is important that they understand how to create professional development opportunities that are consistent and meaningful.

STEM centers use various new attractive forms of training by means of digital technologies such as virtual and remote science labs and educational games. The results of this training (IBSE) are achieved through the introduction of innovative training with the active participation of students in science [8]. Innovations in education are supported by information technology and IBSE approaches, and they accelerate change in educational institutions [9]. Schools in Europe use and integrate Internet applications and virtual laboratories in their education [10]. The introduction of game approaches in the educational environment [11,12] motivates and stimulates learning. Increasing students’ interest in their further development [13] is a major goal, and the so-called “serious games” are used to achieve the learning objectives [14].

Training games in all countries considered include the following possible advantages:

- Giving students the feeling that they have mastered the learning process and the freedom to make mistakes and try again without negative consequences [15];
- Opportunities to work on identity by taking different roles [16];
- Chances for more enjoyment and fun in the classroom;
- Opportunities for differentiated learning, whereupon learning becomes visible [17];
- Providing a manageable package of tasks and sub-tasks;
- Inspiring students to find intrinsic motivators for learning [18];
- Motivating students with dyslexia and low morale [19].

### 3. Virtual Education Space

The Virtual Education Space (VES) supports various forms of e-learning for students—self-paced, blended, and lifelong learning. Taking into account the surrounding physical world is an essential aspect of an e-learning system, especially for disadvantaged learners. VES consists of two basic components:

- DeLC 2.0—directly to support e-learning and distance learning of university students.
- ViSMod—an environment for modeling scenarios and processes in various domains (smart agriculture, smart environment, smart city), which is developed mainly for project-oriented training of students in the discipline of “Artificial Intelligence”, i.e., the environment is created in practical student projects.

#### 3.1. DeLC 2.0

DeLC is an e-learning portal that has actually been operating at the University of Plovdiv “P. Hilendarski” since 2010. The portal was built within the Distributed eLearning Center project [20], which was developed for more than 10 years. A second version of the

portal called DeLC 2.0 is currently in use, which provides basic educational services. The main educational services offered by DeLC will be briefly presented below [21].

The library is a plugin module for DeLC 2.0, which implements the functions of management and handling of static resources (Figure 1). This module provides an opportunity for hierarchical organization in sections of different types of files and maintains meta information about them (description, author, creation date, date of last change, etc.). In addition to static files, the library structure is designed to support information about SCORM packages [22] and QTI tests [23]. Access to the resources stored in the library is defined at the section level and can be set more generally—for a group or role, as well as for a specific user. The rights are cascaded down the hierarchy, with these assigned to a specific section taking precedence over those inherited from the parent sections.

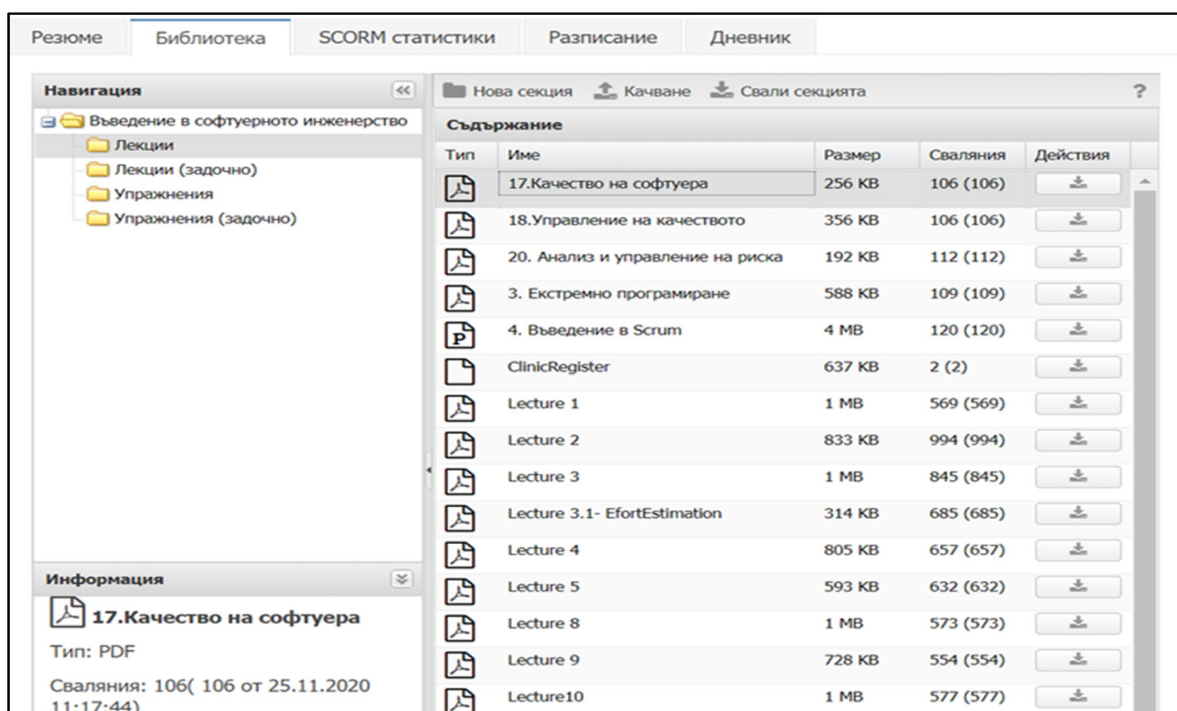


Figure 1. Library module in DeLC 2.0 (in Bulgarian).

The SCORM Player (Figure 2) in DeLC 2.0 is built according to the specification SCORM 2004 R4 and is inherited from the first version of DeLC, where it is implemented as portlets, which are:

- Content management portlets that are intended for use by trainers. These portlets do not include a functionality for creating SCORM content, but only for its management: loading ready-made SCORM packages, defining rules for accessing them, changing and removing them. These portlets also manage information that is outside the SCORM standard but specific to the digital library: metadata, classification, and content annotation. SCORM content creation functions have been delegated to SCORM content creation editors such as SELBO 2.0, which will be presented in the article.
- Portlet for delivering SCORM content to learners (the so-called SCORM Player). A key element in this portlet is the Sequencing and Navigation module, which interprets the rules defined in the respective SCORM package and, based on the learner's behavior, determines the specific content to be delivered at any given time.

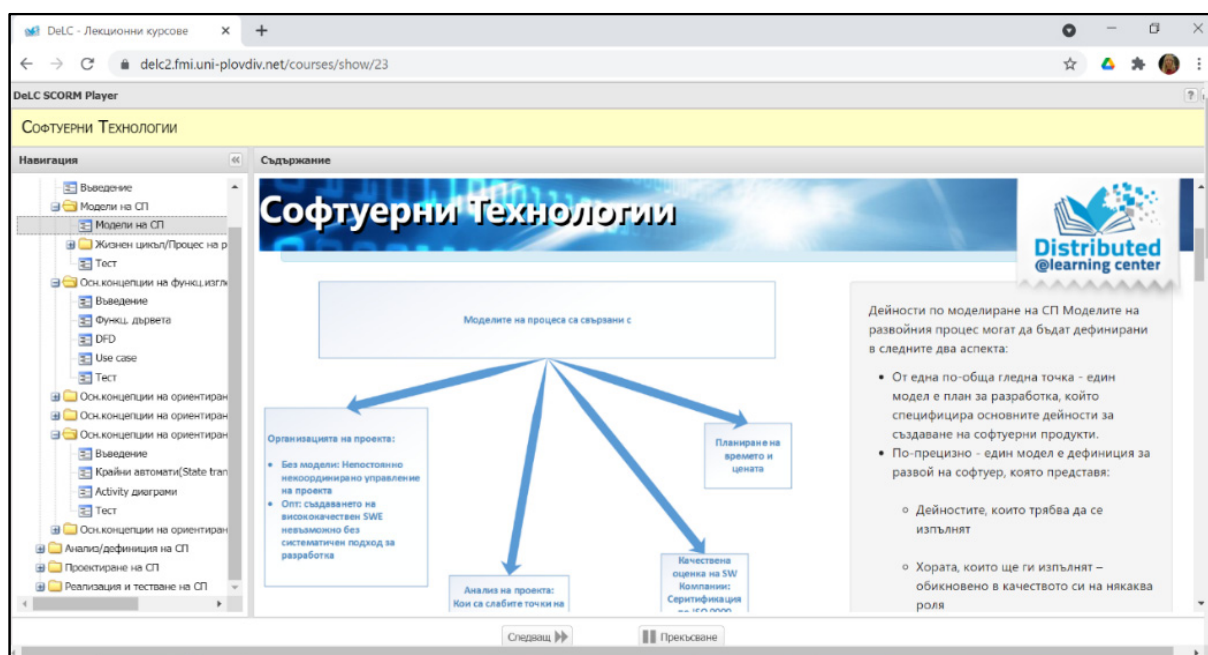


Figure 2. SCORM Player in DeLC 2.0 (in Bulgarian).

The learning content is organized in a tree structure, which is presented in the form of a package—a SCORM content package, in a given textbook or discipline. It contains certain contents defined by the SCORM standard. The file is known as a Package Interchange File (PIF) and it contains all files needed to deliver the content package via a SCORM run-time environment and/or learning management system (LMS). SCORM packages contain two mandatory elements:

- A manifest file—it can describe part of a course, an internal course, a set of courses, or just some content that needs to be transferred from one system to another.
- The content describes the physical files that make up the package. The package must be self-contained, i.e., when unpacked, it must contain all the information necessary for the training. The package presents a training unit, which can be a course or part of a course that is delivered independently.

Each package consists of smaller units called Sharable Content Objects (SCO) and Aggregations. Aggregations, in turn, may consist of SCO elements. A sequence of execution and navigation is set between the individual elements in the package. In the SCORM training packages in DeLC 2.0, tests are created as content elements that participate in content management—if the student answers all the test questions correctly, the next element in the package is unlocked, otherwise the SCORM Player returns the learners to read the content which they have done wrong.

One of the benefits of the SCORM 2004 content is that we can follow student progress in real time. Figure 3 presents the general statistics for the students who can start the study content in the specific semester, how many of them have started it and how many have completed it. In addition, the diagram shows the names of the students and what percentage of the textbook material has been passed by each one of them at a time.

The feedback that the SCORM Player allows is extremely useful for the teacher because he/she can use the data to improve the quality of the learning content and track the progress of the students during the course. Students, in turn, can use the textbook for self-preparation as each one has their own pace of passing through the study material, which contributes to the personalization of the learning process.

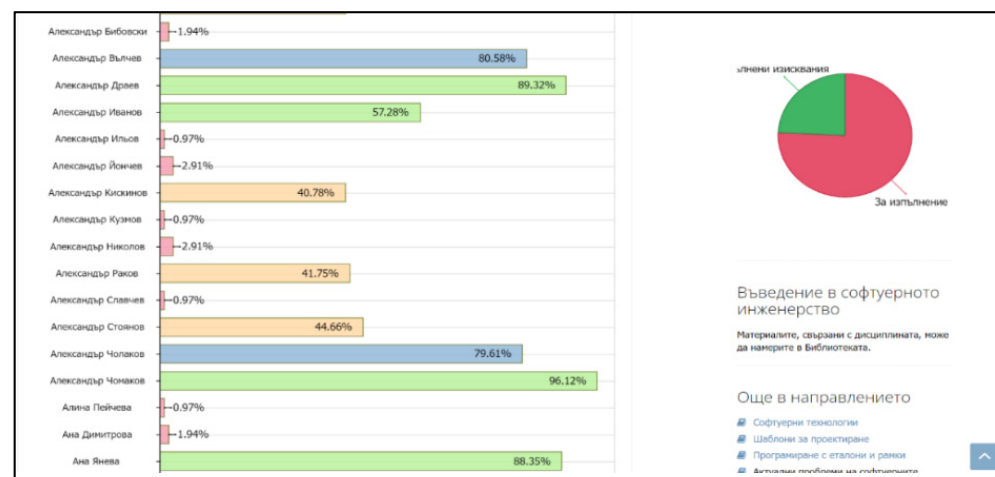


Figure 3. Statistics in the SCORM Player (in Bulgarian).

A very important part of using the SCORM Player module is to provide teachers with an easy and convenient way to create SCORM content. For this purpose, we have developed our own editor—SELBO 2 [24]. Lecturers can use this system to create interactive teaching material. This system is implemented as individual editors connected to each other by a single user interface (Figure 4). Exposing a plug-in architecture, these editors, in turn, can be dynamically added, removed, and replaced without modifying the rest of the system. Being an element of the graphical user interface, the editors are able to prepare various electronic resources. The editors belong to two groups. The first group is for standard formats such as the text editor and the HTML editor. The second group consists of specialized editors relevant to a specific domain, for example to the “Software Engineering” UML editor, the editor of the source code, and others. An important specialized editor is used to create standard electronic content that is compatible with SCORM 2004. SCORM electronic packages can be provided for further processing (update, modification, integration with other components present in the environment, etc.) in the “workplace” of the teacher and integrated in the educational portal.

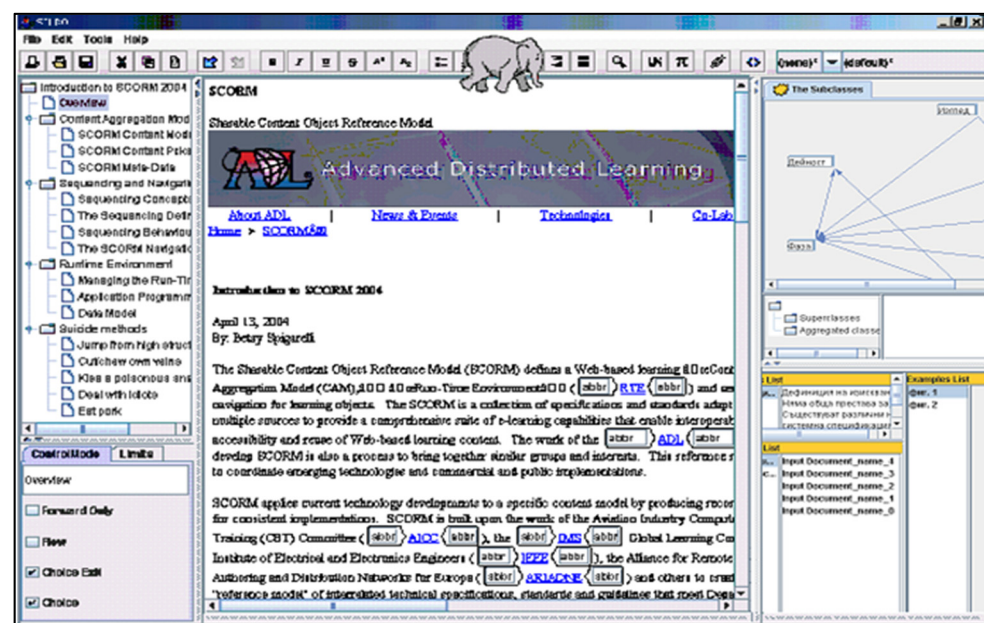


Figure 4. Selbo 2 editor (in Bulgarian).

The purpose of this module is to provide tools for managing electronic tests in accordance with the QTI 2.1 standard. The test editor provides the ability to perform basic test operations—view, create, edit, delete (Figure 5).

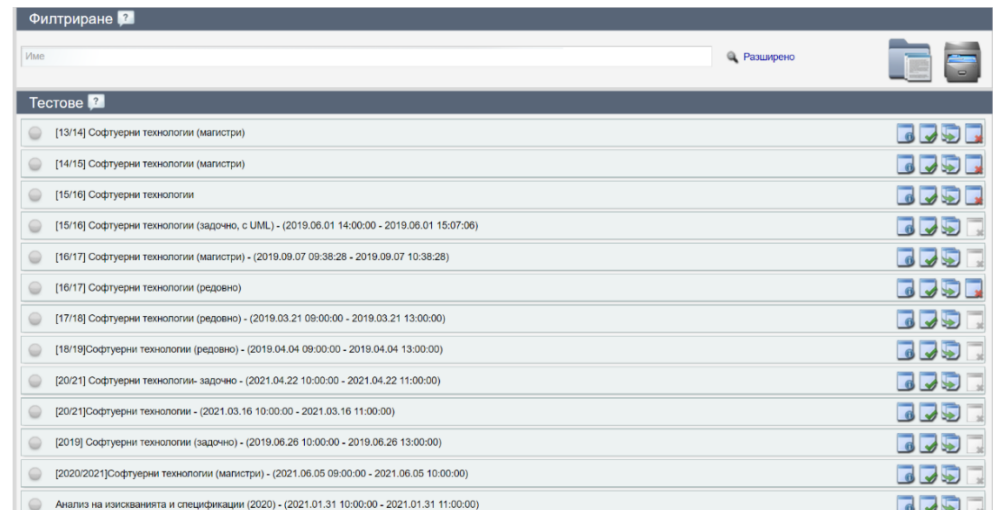


Figure 5. Test Editor (in Bulgarian).

What is specific to the test editor is that the Questions section is divided into two parts. An excerpt from the Question Library is displayed on the left and the selected questions and templates for the respective test are shown on the right. The selection of a question is completed by simply dragging and dropping the respective question from the Library area in the area with the selected questions.

For the convenience of the trainer, a function is available to visualize a sample test generated by the current state of the test, i.e., from its templates and/or fixed questions.

Manipulating questions in a library is performed with the help of a question editor. When creating a new question, the editor displays a series of dialogs to define the main attributes of the new question (Figure 6).

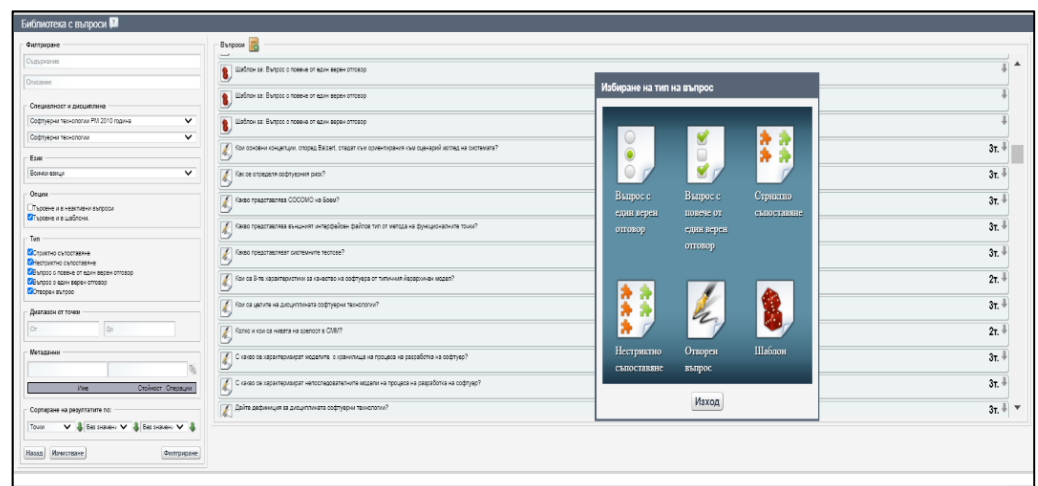
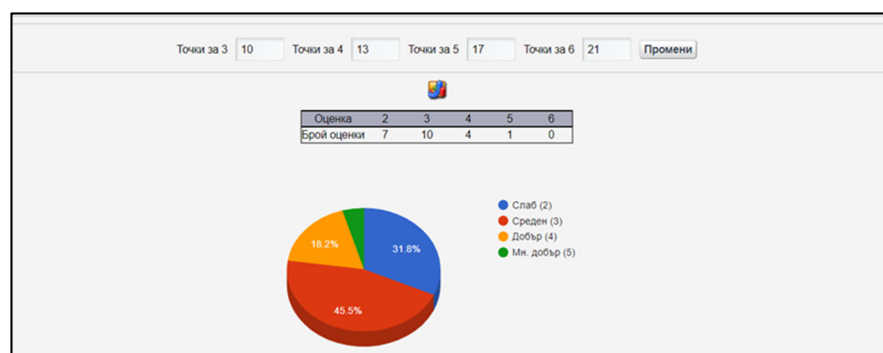


Figure 6. A Question Editor (in Bulgarian).

Depending on the type of question, the editor opens a specific form for entering the data for the respective type of question. Each type of question has its own rules for the validity of the data entered. The general requirement is that the question has content (the part that asks the question itself) and points. In the case of a multiple-choice question with one correct answer, it is also obligatory to define at least one answer and mark exactly one

answer option as correct. In the multiple-choice type of question with more than one correct answer, it is mandatory to define at least one answer and at least one answer option to be marked as correct. For multiple choice questions with strict and non-strict comparison, it is necessary for each of the listed sub-questions to determine which of the set of possible answers is the correct one. There are no additional requirements for long and short open questions, as they are not evaluated automatically. In the case of a short open-ended question with a fixed answer, it is required to enter the word or phrase that represents the correct answer. The mechanism for assessing this type of question is intelligent, and, comparing the data with the input entered by the learner, techniques are used to deal with typos and other similar errors. The question editor also provides the ability to define a virtually unlimited number of metadata for each question, regardless of its type. The assessment of the tests is performed by the trainer after the trainee has completed the test. When assessing an open-ended question, it is possible to enter a comment, which is then displayed to the learner together with his/her assessment. In this case, it is a one-way communication, but the aim is to provide an opportunity to justify the assessment, and not to start a dialogue. After assessing all open questions for all learners, the trainer has the opportunity to see summary information about the success of students for the current test (Figure 7) and, if necessary, he/she may change the grading scale, which updates the students' grades and the summary reference for test success. When saving the grades, the currently calculated marks are saved in the database and each student can see only his/her own grade from their profile.

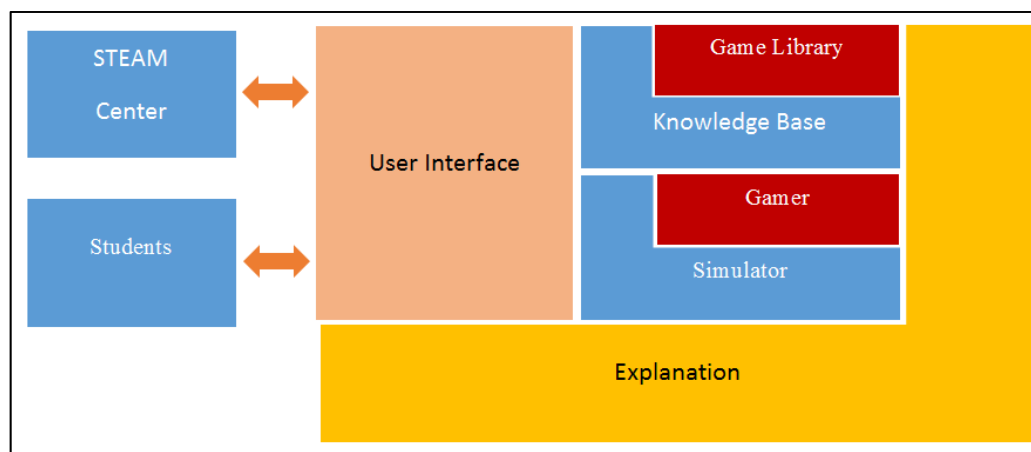


**Figure 7.** Protocol and statistics about the test results (in Bulgarian).

DeLC 2.0 has been successfully used at the FMI of Plovdiv University for over 30 academic disciplines. The library and test system are the most commonly used services by teachers and students. The SCORM content is increasingly applicable in the COVID-19 situation due to the need for online training. Students using SCORM textbooks can be navigated by the SCORM Player, which creates the feeling of controlled presentation of the material much like when teaching it in a real environment.

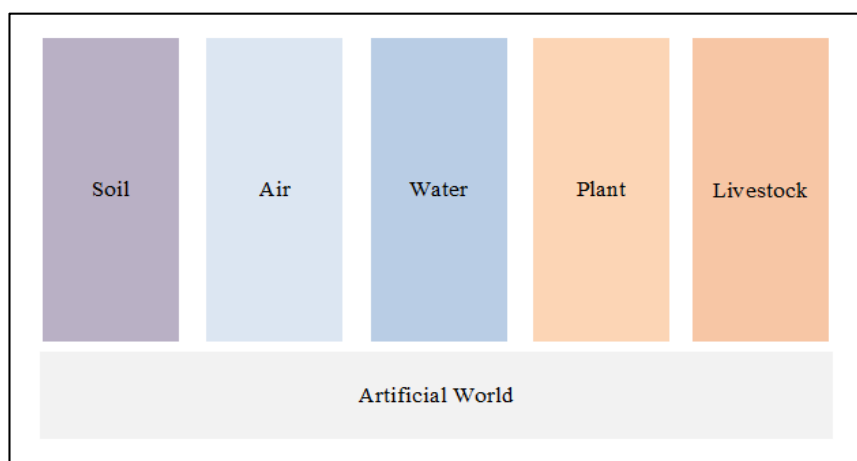
### 3.2. ViSMod

ViSMod is the second basic component of VES, which provides tools for modeling processes, scenarios, and systems from different application areas. The aim is for the modeling tools to be developed by students within the framework of their training in the disciplines of Artificial intelligence and Logical programming. At the same time, in the infrastructure of the integrated domain presented in the article, the ViSMod environment could be used for training students in STEM centers. An appropriate approach to such training can be e-learning in various forms (for example, game-based learning), especially in the subjects of Artificial intelligence, Biology, Ecology, and Agriculture. ViSMod is being developed as an expert system containing the following four components—a knowledge base, simulator, explanation, and user interface (Figure 8).



**Figure 8.** ViSMod Architecture.

The knowledge base stores specific knowledge for domains from the natural and artificial worlds. In the current version of the environment, we are developing the following five knowledge bases for the natural world—soil, air (the climate), water, plants, and animals, as well as a base of crafts for the artificial world (Figure 9). The library storing games is one of the components of the knowledge base. It is supported by a specialized editor that can be used to create libraries with domain-specific building blocks. These building blocks are used in the user interface to prepare specific game scenarios. For a smart city, the building blocks can be buildings, streets, traffic lights, cars. Similarly, plants, farm animals, irrigation system elements (water pumps, valves, hosepipes) are examples of building blocks for agriculture. For both domains, there are shared building blocks (sensor and communication networks, drones, robots) from the artificial world.



**Figure 9.** Structure of the Knowledge Base.

Each game in the game library is characterized by the following three parts:

- The content and the rules of the game;
- Information for the Personal Assistant—used to manage and control the game;
- Building blocks presenting elements of game scripts.

The simulator is an inference machine that works with the knowledge base of the environment. A Gamer built into the simulator is also, in principle, an inference machine that specializes in interpreting games presented as rules. The core of the Gamer is a Personal Assistant, presented in Section 3.3.

The player (the Personal Assistant), managing the game, keeps a record of the course of the game (the dialogue with the player and the mistakes made by him/her), which is

recorded in the explanatory module. By recording this protocol, the teacher has access to the player's data and can properly analyze it.

A user interface is used by the users to interact with ViSMod. Users from the university have direct access to the system as well as users from schools through the Front-End module of the platform (described in the next section).

### 3.3. Personal Assistants

The active components of VES are assistants implemented as BDI (Beliefs-Desires-Intention) agents. A BDI-agent exhibits a goal-oriented behavior that is based on its Beliefs (the agent's perception of the environment) and desires (possible potential options of the agent) [25]. Operational assistants are usually server agents that implement a certain functionality. Personal Assistants are used to interact with and support users.

VES is developed as a CPSS-like (Cyber-Physical-Social) infrastructure, according to which the users are in the spotlight [26]. Therefore, from an architectural point of view, the central components of VES are the Personal Assistants (PAs), directly dedicated to assisting the users. Generally, a PA has to perform two functions:

- Assistance—operational support to the user in his/her everyday activities.
- Prevention—in certain cases, it is necessary to carry out preliminary activities to provide optimal conditions for the user to participate in (and benefit from) the upcoming event, including (especially for disabled learners) ones related to movement in the physical space and over time.

In our space, the personal assistants make decisions mostly in the virtual space, taking into account the physical world that the user resides in. Since various groups of users can work with the space, a reference architecture has been developed that can be adapted to implement specific instances of the PA. The work of the PAs is based on the event model, while the states (situations), in which the user has to be assisted, are presented as domain events. Moreover, in performing its duties, the PA interacts with the other components of VES.

In general, at an abstract level, the operation of a referent PA can be represented as the triad  $\langle \varepsilon, E, \delta \rangle$ , where:

- $E$  is an event identifier to which the PA will respond.
- $\varepsilon$  is an interval preceding the event, when the PA has to carry out prevention (preparation). This interval depends on the type of event and can be configured for the current application and customized for the particular user.
- $\delta$  is an interval, in which the event is active.

As a general rule, intervals may be different. Typically, they are temporal, but they can also be locations (e.g., the event begins or ends when a certain location is reached), or something else. The operation of a referent PA is closely related to a structure called Personal Timetable (PTT), which consists of two main elements:

- Schedule—it contains the actions of the user for a certain period of time, in which he/she will be assisted by a PA. In our case, these actions are presented as domain events.
- Index—it assists the PA to identify the actual action (a record in the Schedule), to which it has to react. In our case, the indices are presented as basic or system events. For example, if the PA receives a message from an operative assistant that a student has failed an exam, then it has to enter a new record in the PTT.

The primary way of operating a referent PA will be demonstrated with an example. A student in a wheelchair has to attend a lecture. The corresponding response of the PA can be represented as, for instance,  $\langle \varepsilon = 15, E = \text{Lecture}, \delta = 90 \rangle$ . Within the Prevention Interval (we assume that it will take 15 min to find a suitable route to the hall), the PA has to determine an appropriate route and assist the student to move to the lecture hall. At the scheduled time, the lecture begins. Then, in the interval of an active event (in this case a lecture that we assume continues for 90 min), the PA assists the student, for example, by

establishing a link to the blended learning system and choosing the specific study material or control tests.

Another structure that is essential for the operation of PAs is the user profile. Profiles contain application-specific and user-specific information, and an individual profile is maintained for each user. For each application area, a specific profile template can be configured, from which the personal profiles are generated. The profiles can be stored and managed in different ways, for example:

- Entirely in the supporting in-house system.
- In the supporting in-house system and in the PA's local memory-distributed.
- Entirely in the PA's local memory.

Typically, the PA interacts with some sort of in-house system. For instance, in the e-learning application, the PA retrieves information about curricula, schedules, and examination sessions from the integrated university system. As an example, in order to enter a new record for a resit exam in the PTT, the PA has to retrieve the planned date from the university system. In the next section, the interaction between the PA and the university in-house system is described in more detail.

The PA is a component of a CPSS-like infrastructure [26] such as VES. The assistant itself operates in the virtual world. A major problem, however, is the interaction with the physical world. The PA interacts with the physical world in the following two ways:

- Through the Event Engine—the events from the physical world are registered by the guards. They transmit information to the Event Engine about what is happening in the physical world. The Event Engine, in turn, uses this information to identify the type of the event. When the event is relevant to the PA, it notifies the Event Engine, as explained above.
- Through the PA's sensors—the PA can use its own sensors to obtain information about what is going on in the physical world.

Regarding the reference architecture of a Personal Assistant, there are different adaptations of the referent PA for different groups of users, such as:

- LISSA—a Personal Assistant to a student [27];
- Gamer—a Personal Assistant of a playing student (presented below);
- Personal Assistants at BLISS—also presented in the next section.

The PA is a central module of the system Gamer. This module is implemented as an intelligent autonomous agent exposes reactive and proactive behavior. Furthermore, it is able to communicate with other agents. This personal assistant accomplishes interaction between various user groups and the platform. In addition, it is able to promote, guide, publicize, and offer educational games. The PA, being proactive, directs the player's progress and, if necessary, activates a dialogue to help the learner. The PA keeps a record of the course of the game. This information can be used by teachers for future analysis of student performance. From an architectural point of view, the Personal Assistant is a multi-agent system. The individual agents are deployed on the server and the mobile client is deployed in the STEM center.

The server PA is the operative engine of the platform. It is able to analyze the player's behavior and to monitor not only the points scored but the control time as well. Due to the assessment being proactive, the PA prepares and sends prompts and warnings. A simple example of this is the case where learners have been inactive for several days. Then it will send a kind reminder that it is time to play and learn.

### 3.4. BLISS

The first attempt to integrate the university educational platform VES for the secondary school was made by implementing the BLISS environment where VES operates as a back-end [28]. BLISS is mainly designed to provide a comfortable and supportive learning environment for students in independent learning and lifelong learning. These students are often adults who have different professions and commitments, live in different localities

and need a Personal Assistant to inform them about upcoming events and to assist them in organizing their training process. In essence, BLISS is a multi-agent system, at the core of which are the Personal Assistants of all participants in the learning process. The environment of the agents consists of two parts—a BLISS server and an event-managed school diary, realized through Blockchain technology. The integration of this technology in the multi-agent environment creates additional conditions for the reliability of the processed information. Some of the information stored in the blockchain is used for other services such as issuing certificates and diplomas to graduating students. BLISS has developed new types of specific Personal Assistants as a GPA instance. During the development of BLISS, we have defined the following Personal Assistants:

- The PASTudent supports and assists students in the performance of their daily duties. It also has the task of informing them about all upcoming events such as exams, lessons, training sessions, consultations, and others. In addition, the PA stores profile information about the specific student, monitors and reminds him/her of events that correspond to this profile. For example, if a notification is received for an upcoming event (a math exam)  $L1 = \langle \text{Math, Exam, attr (20 December 2021, time (10:00,10:45))} \rangle$ , PASTudent must not only inform the respective user, but also be able to take proactive action, according to the prevention interval set in its knowledge base (e.g., 15 days). It must draw up a plan on how to facilitate the student to participate successfully in the event. In this case, it can provide him/her with the necessary resources for self-preparation, remind him/her periodically, and assist him/her in the learning process. This assistant is able to prepare analyses of the results of the student's participation in the learning process, as well as to communicate and interact with other assistants in the space.
- The PATeacher is designed for teachers. This assistant also aims to remind them of upcoming events and the necessary preparation. Its main function, however, is to help teachers track and analyze the participation, progress, and performance of their students. Analyses can be used to improve learning.
- According to the profile characteristics of the school principal, the purpose of the PAPrincipal is to support the effective management of the school institution and all processes of planning, conducting, and controlling the learning process.

The environment of the various assistants is the BLISS server and the school diary. In the BLISS server, all information objects (such as schedules, lessons, exams, consultations, self-preparation, meetings) are presented as domain-specific events. The server stores these events and controls access to them. Any change in the information on the server is automatically perceived by all the concerned assistants.

### 3.5. Technology

The educational space is implemented by means of the multi-agent programming framework JaCaMo [29]. In addition to supporting the BDI agent approach, the framework provides the opportunity to use three appropriate and powerful programming languages, namely Java, Prolog, and XML. JaCaMo combines three separate technologies, each of which is well-known in itself and developed over the years, so they are quite robust and complete. JaCaMo is a combination of:

- Jason—for programming autonomous agents;
- Cartago—for programming the agents' environment in the artifact paradigm;
- Moise—for specifying various societies of agents.

In this way, the framework encompasses all levels of abstraction that are necessary for the development of advanced multi-agent systems.

## 4. ViToS Platform

In addition to BLISS, which is supported by university education systems, we want to expand the architecture by including STEM centers. Practically, a STEM center is a new

domain. Therefore, the new version implements our concept of “shared domains”, where the needed integration is performed through Operative and Personal Assistants that can also be deployed on third-generation Robobo educational robots.

#### 4.1. General Architecture

The model and the general architecture of the platform ViToS are based on the concept of integrated domains. This is how the idea of integrated domains arose, which will be presented through the “point of view” of the Personal Assistant (PA). Here, we present an initial idea for a formal model of an integrated domain, which will be the basis for the architecture of the supporting software.

Let there be given a set of  $E$  environments. Let us look at  $n$  domains and  $n$  PAs, respectively. Each  $PA_i$  operates in a domain-specific environment  $E_i$ ,  $i = 1, \dots, n$  as  $E_i \in E$ . Each environment in turn includes many artifacts of different nature, depending on the specifics of the domain. We define the term integrated domain as follows:

$$IE = \bigcup_{i=1}^n A_i$$

For effective management of the integrated domain, each PA must know the type of integrated environment ( $IE$ ), in which it will operate. Using the notation in [30], we can define the following  $IE$ s:

- Disjoint ( $IE$ )  $\Leftrightarrow (\forall E_1, E_2 \in IE \wedge E_1 \neq E_2 \Rightarrow \text{Intersection}(E_1, E_2) = \emptyset)$
- ExhaustiveDecomposition ( $IE, E$ )  $\Leftrightarrow (\forall A_i, A_i \in E \Leftrightarrow \exists E_1, E_1 \in IE \wedge A_i \in E_1)$
- Partition ( $IE, E$ )  $\Leftrightarrow \text{Disjoint}(IE) \wedge \text{ExhaustiveDecomposition}(IE, E)$ .

These definitions allow us to plan what PAs are needed depending on the type of integrated domain. In our case, the  $IE$  is of the Partition type because the integrated domain contains the artifacts of all four components, but the individual domains do not have common artifacts. In this case, four PAs are needed, three of which are implemented and one is missing (in ViSMod).

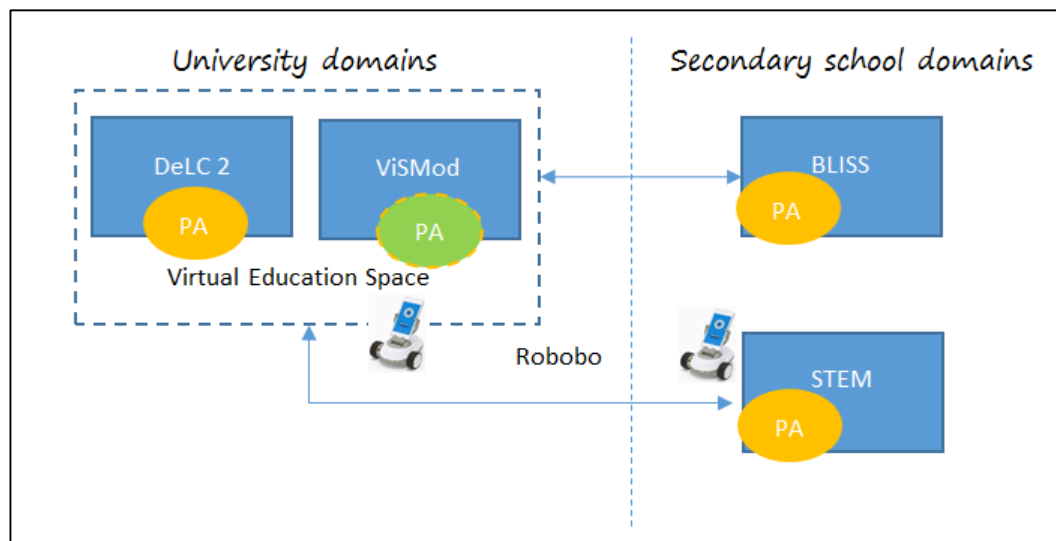
Two domains can be identical, i.e.,  $E_k, E_m \in E, E_k \equiv E_m \Leftrightarrow (\forall A_i \in E_k \Leftrightarrow A_i \in E_m)$ . However, it is possible for an artifact from identical domains to be interpreted in the same way or differently by different PAs. To deal with this problem (from an architectural point of view), we introduce the notion interpretation  $IPA: EPA \rightarrow \text{BeliefsPA}$ , i.e., the interpretation of artifacts from the environment of a Personal Assistant results in its beliefs. In the case of an equal interpretation, the respective interpreters can be located in the identical domains (outside PAs). Otherwise, the interpreters have to be specific to each PA and be part of its architecture.

When building integrated domains, we can also use ontologies that represent the relationships between different types of artifacts in an integrated environment. These ontologies will also be part of the  $IE$ ; in other words, it would be good for agents to be able to view ontologies as artifacts as well.

The overall architecture of the ViToS platform consists of two parts (Figure 10):

- Back-end—practically, a back-end of the platform operates VES including both the university domains DeLC2 and ViSMod, presented above.
- Front-end—the front-end was built like this by the inherited BLISS domain and the newly built STEM center domain, which is described below.

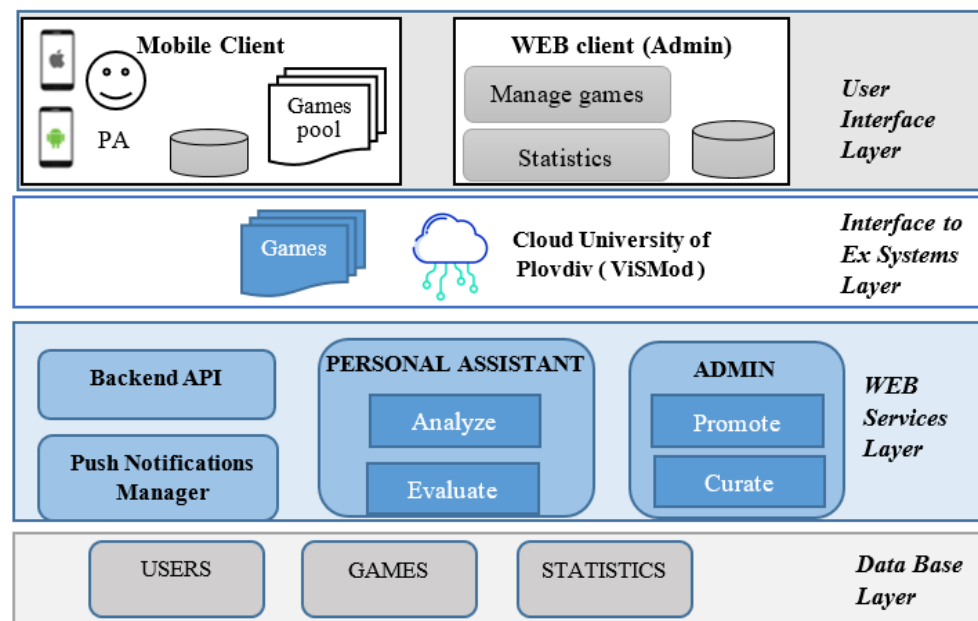
Domain integration is not unconditional; it is not completed through interfaces, but rather through protocols for interaction between Personal Assistants. If an appropriate protocol (and possibly a suitable learning content transformer) can be specified and implemented, then integration is possible.



**Figure 10.** ViToS Architecture.

#### 4.2. The STEM Domain

The front-end module is the school STEM center. It is created as a game-based training system. The architecture of the module has four layers (Figure 11).



**Figure 11.** STEM Domain Architecture.

- Data Base Layer retains information about users, games, and statistics. The users are separate portable appliances. In this way, personal data are not gathered, and registration is not necessary, because that would be unacceptable for an application suitable for children.
- The Web Services Layer comprises the main business logic of the system with the following modules—Administrative (Admin), Personal Assistant (PA), Application Programming Interface (Backend API), and Notifications Manager (Push Notifications Manager). The Admin module includes a user interface for functionalities performed by users with the rank of administrators and super users such as: adding and configuring games, providing management, granting access, permissions and prohibitions, etc. In addition, they provide statistics for players. Backend API is a standard REST

API interface for communication with the server. It should be noted that the statistics for the student are not recorded directly in the database but are transmitted to the intelligent PA for preliminary analysis and evaluation. The main task of Push Notifications module is to send personalized messages to a mobile client via Google Cloud Messaging and Apple Push Notifications Server.

- The Interface to External Systems Layer provides connection to external systems, one of which is the cloud structure of Plovdiv University. Thanks to this, a connection is established with the back-end module and as a result it is realized as a content delivery network (CDN). In school STEM centers, this interface delivers the game library, which is stored in ViSMod. Another external system used in the system is Google Cloud.
- The User interface incorporates two client applications, which are a mobile client for players (students) and a web client for administrators (teachers, facilitators). Students can participate in the game by means of the interface that serves as a PA, demonstrated below. Teachers can manage the games and observe player statistics. The user interface supplies an API for an administrator web client. Its major purpose is to provide mechanisms for the management of data and metadata. This module is accountable for gathering data exhibited to the mobile customer and for ensuring adherence to the terms of use. There is an integrated promotion module too within the administration module that informs end users of new games supplemented to the game library.

#### 4.3. Personal Assistant Gamer (Client Side)

The player's Personal Assistant is an important element of the architecture of the STEM center. It is the tool through which users communicate with the learning environment. Owing to it, learners can choose educational games and the Personal Assistant, in its turn, can monitor and record the progress of the user in a particular game or numerous games. The assistant demonstrates inherent human behavior by greeting, asking questions, or offering a continuation of a game or rest to users (Figure 12). Some of the functions of the Personal Assistant are related to the personalization of the environment to the users—a change of the system interface (color scheme, fonts, etc.), change of the time for rest, or deactivation of the assistant, if the player wishes so. In addition, the Personal Assistant can provide users (teachers or parents) with various statistics about the game of students, their progress in a particular game, how they coped, how long the application was active for in a certain period of time, and others. In the future, the assistant can be improved with more functionalities for statistics and tracking of the student's game, such as suggestions for starting an educational game in an area that is not initiated by the student or if the result of his/her game is low, etc.

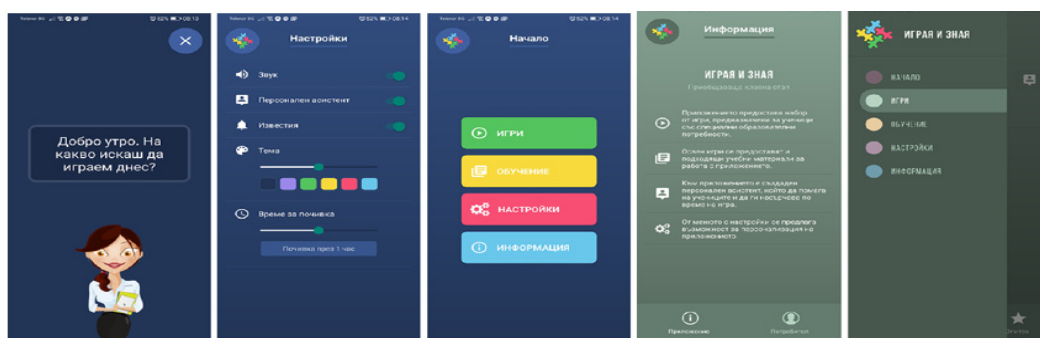


Figure 12. User interface—mobile application (in Bulgarian).

#### 5. Educational Games

Over the past few years, because of the state of affairs related to the COVID-19 pandemic, the infrastructure suggested in the article has been shown to be appropriate to support learning in the school. Interdisciplinary lessons are scheduled with groups of students in the STEM center, including subjects from mathematics, technology, the natural

sciences, and other scientific fields. Game-based learning is a didactic technology, in which games are used to increase the effectiveness of students' learning. Here the learning process is realized by increasing the activity and motivation of students and encourages critical thinking and problem-solving skills. In addition, when games are applied in the cyber-physical education space, they are combined with other didactic methods such as learning by doing, learning by experiment, etc. The interaction of students with their personal assistants turns learning into solving a real case study, supported by the teacher in his/her role of mentor and facilitator.

For demonstration of the described infrastructure, we will present two sample lessons—one from a smart city and another from smart agriculture. This training tries to strike a balance between the school subjects in the purpose of the game and the ability of the player to put into practice in the real world what he/she has learned. In the STEM center, by way of cooperation between trainers and students, depth and perspective are added to the experience obtained from the game. The benefits that game-based learning brings are common knowledge. In this context, learning games increase motivation and provide a safe almost real model of the world in the classroom, improve learning success, stimulate creativity and promote self-discipline and control, etc.

### 5.1. Educational Game That Helps to Learn the Rules of the Road

The topic related to road safety training (RS) is an essential part of the compulsory training and socialization of students. This training is important for the development of students' knowledge and specific skills as road users, especially in the context of the constraints imposed by COVID-19. In this sense, road safety training is fundamental and takes place in each European country at different levels of secondary education. In Bulgaria, education is mandatory in all types of schools—kindergartens, schools and centers for personal development. The planned number of training lessons is very limited. Data and research show that the active management of such training has a positive effect on the attitude and behavior of students as participants in traffic and reduce road accidents. Therefore, we believe that through the games in the STEM centers in the free time of the students the main goals of the training will be achieved.

One of the games offered in the Game pool is related to the study of traffic rules. The game offers a specialized visual editor, with which users (players or teachers) can create different educational scenarios related to traffic in a particular neighborhood or to create a fictional road environment (Figure 13). The editor provides a toolbar with various virtual objects such as buildings, roads, plants, road signs, etc., through which users can build their virtual world in the editor. The toolbar with virtual objects can be improved with additional ones, which in turn will allow the construction of even more interesting scenarios. When creating educational scenarios, users are given the opportunity to determine the entry point in the scenario for the players and to define various educational questions for the signs, roads, and others. The game can be adapted for other domains by changing the library with virtual objects in the editor—for example, in the field of botany, these can be different types of plants, about which different questions will be supplied for the users to answer.

- The gaming platform provides two modules—web-based gaming environment and mobile gaming environment. The first one is built using the Backbone.js framework and is used through a browser. The React Native framework is used for the development of the mobile environment, which allows exportation of the mobile application to the two mobile platforms—Android and iOS.

The statistics made for the actual use of the game showed high commitment and interest. Due to the virtual presentation of the scenarios of the road conditions and the competitive nature of the game, the success is very high. The statistics about the mistakes are interesting. The majority of the errors are made on easier and more preferred questions by students (questions for pedestrians and bicycles), while on more difficult topics the mistakes are fewer. This is probably due to the use of virtual reality, in which students find it easier to master otherwise abstract topics, and to the increased level of concentration

during the game. It is also noted that in each subsequent game the students make fewer and fewer errors (Figure 14).



Figure 13. A visual editor for generating a virtual city model and a game scenario (in Bulgarian).

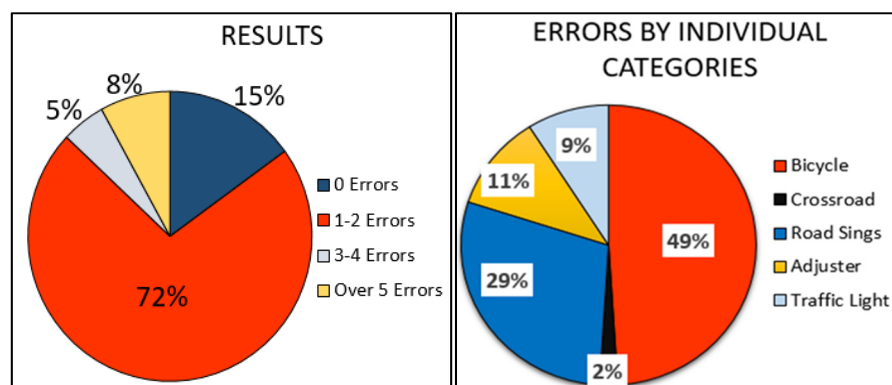


Figure 14. Results from the experimental training with the game environment in STEM centers.

The proposed environment integrates knowledge from different domains—civic education, physics, transport, mathematics, etc. It can be adapted for other areas such as biology, ecology, geography, history, and so on, for which purpose the virtual world and the library with virtual objects must be changed.

### 5.2. Educational Game That Helps Learn about Crops

In connection with intelligent agriculture, a second scenario has been developed, which includes the field of natural sciences studied at school. The curricula of natural sciences are aimed at understanding the relationships between man and the environment, sustainable development and a healthy lifestyle, making decisions about human health and protecting the environment. In the technical sciences, a major focus is the formation of technological competence related to the development of projects that use affordable digital tools and software products. Students' interest and position for a healthy lifestyle is formed through the biological knowledge embedded in the curricula related to the organisms and biodiversity of the Earth, protection of human health, and its relationship with nature.

The second scenario was created on the basis of the game, popular in the United States in the 19th century "Game of Twenty Questions", which motivates deductive decisions and creativity. The version of the game that recognizes an object (an animal, plant, or a

mineral) is the most popular. Our scenario is called Plant Recognition. The plot is very simple—through conducting a dialogue, the game tries to guess the plant that the player has in mind. For this purpose, the game uses the answers of the players and its base of primary (initial) knowledge.

We use two approaches to realize the presented scenario of the game. In the first one, the student initially formulates a certain hypothesis, so we provisionally call it “The game teaches the student”. The system, founded on its knowledge, consistently checks and accepts or rejects a hypothesis on the grounds of the given and respectively received answers to questions. When rejecting a hypothesis, the inference machine looks for another possible hypothesis (Figure 15). The game ends with a negative score if the Gamer does not find possible hypotheses based on its knowledge. By answering questions from the game, students can test an assumption (hypothesis) for a particular type of crop (such as fruits or vegetables). We associate the game with training focused on agriculture. In this scenario, the “teacher” is the university domain, and the “teacher” domain is the school domain. The student, working in a STEM center (for example, a vegetable garden), can define a hypothesis and through his/her Personal Assistant will gain access to the ViSMod library with games, realizing a dialogue between the student and ViSMod.

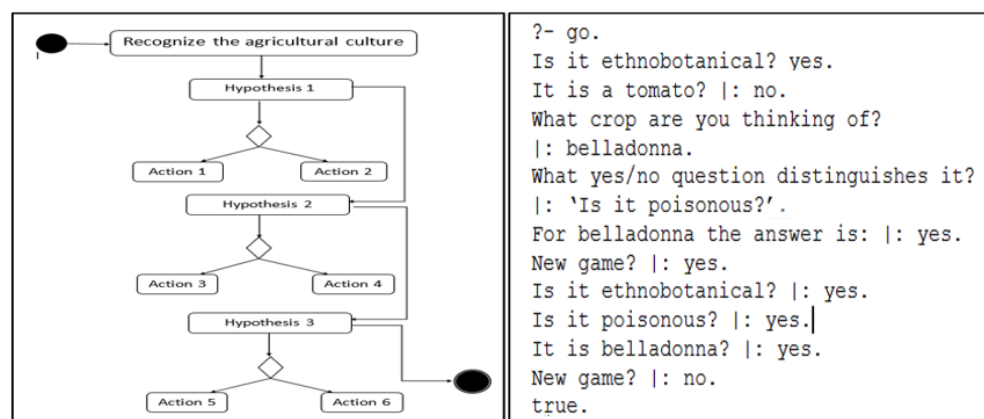
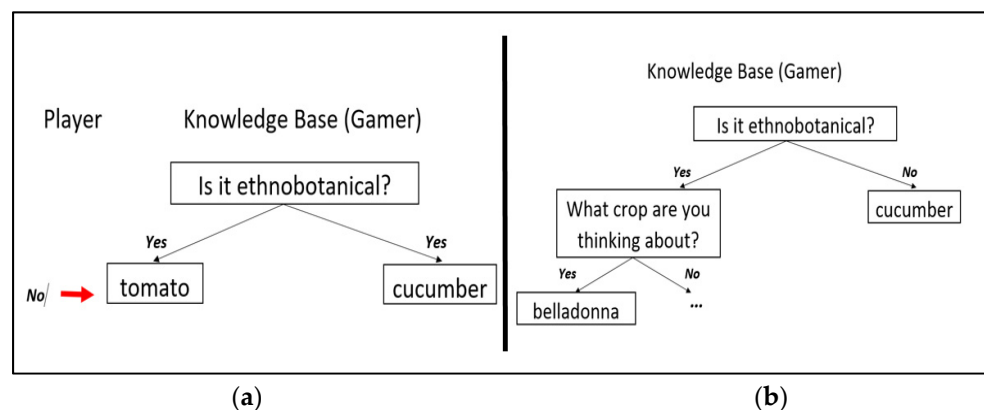


Figure 15. Game control scheme by the Gamer and its implementation.

In the second approach, which is called “The student teaches the game”, the school domain is the “teacher” and the university domain is the “learner”, i.e., the roles are exchanged. In this scenario of the game the students teach the game by having a dialogue with it. Dynamically changing tree structures are used in this game scenario by the Gamer. When activating the game, a dialogue is started with the player to reveal what culture (plant) the student has in mind. If the student answers a question in the negative, the Gamer needs to expand its knowledge with a new hypothesis so for this purpose it asks the student questions. Therefore, the game becomes more “informed” by dynamically expanding the game tree with the new hypothesis.

We will demonstrate the conduct of the game with a specific example. We assume that the following knowledge is included in the knowledge base: “Potato plants are widely used by humans. They are an important source of food, spices, and medicines. Not all these plants are suitable for consumption, some are poisonous. Even in small amounts, they are rich in alkaloid glucosides, which, in terms of toxicity to humans and animals, can range from mildly irritating to fatal consequences. The following crops are included in the potato family: eggplant, tomato, mandrake, pepper, tobacco, belladonna, tatul, petunia.” When starting the game, the Gamer initiates the dialog presented in Figure 16.



**Figure 16.** Dynamic construction of the game tree—building a tree (a), adding a subtree (b).

First, the Gamer builds a game tree based on the Knowledge Base data shown in Figure 16a. In the dialogue itself, in the tomato node, it finds a discrepancy between the background knowledge and the player’s reaction. In such a situation, the Gamer requires additional information and subsequently replaces the tomato node with a subtree (Figure 16b). Upon future launch of the game, the Gamer applies the new knowledge presented in the game tree, which can be further developed.

The second game scenario was implemented using the logical programming language Prolog (SWI-Prolog version [31]). The prototype is being developed (and will be expanded) by university students as course projects, although the game is designed primarily for schoolchildren. Thus, students will have the opportunity to learn logical programming, knowledge representation, and work with recursive structures and algorithms. A relatively simple user interface has been implemented to use the system.

## 6. Insertion of Robots

We are expanding the platform presented in the article with training robots. Our motivation is to make the integrated domain more attractive and, furthermore, to initiate the study of new topics related to robots. Four robots are included in the current implementation. The front-end (STEM centers) uses the Scratch programming language, which is studied in Bulgarian schools, while in VES, robots are programmed on ROS (Java and Android). The Robobo project was created for use by STEM or STEAM (Science-Technology-Engineering-Arts-Mathematics) laboratories, and it aims to bring educational robotics from primary and secondary school to real world applications [32]. It is based on a hardware platform called Robobo, a very flexible programming environment, and a set of STEM tutorials for their integration. The Robobo robot consists of a mobile platform and a tilt panel (PAN-TILT device) that supports a smartphone. The platform is powered by two wheels, which, in turn, are powered by two motors and allow Robobo to reach speeds of 1 m/s with a torque of 40 N cm. Both wheels can move independently forward or backward, allowing the robot to rotate. It consists of seven LEDs and eight infrared sensors. Through the sensors, the robot detects obstacles up to 10–15 cm, depending on the color of the obstacle and precipices, and it avoids falling and collision [33].

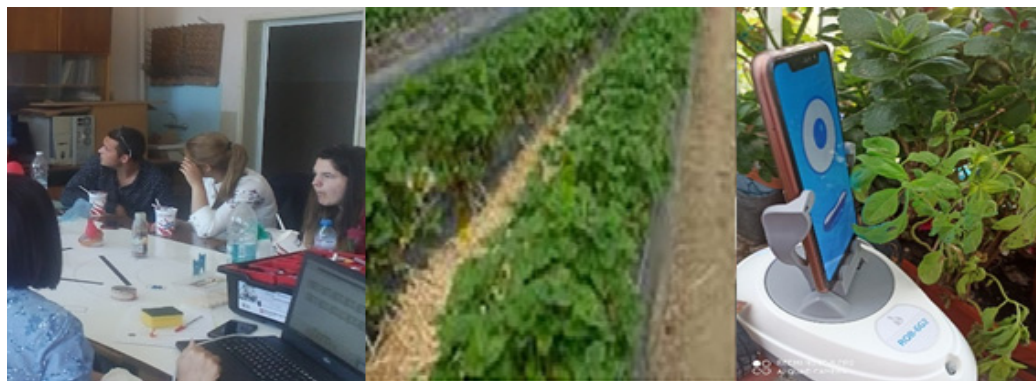
The Robobo base is the body, while the smartphone is the robot’s “brain”, which attaches to the tilt device. The communication between the body and the smartphone is via Bluetooth. Thanks to the tilting device, the robot can perform various movements such as shaking the “head”, bending, turning, and more. The robot can express and convey emotions and feelings when combined with images displayed on the smartphone screen, with the sounds that it produces, by shifting the main platform and using different colored LEDs. Tilting the device allows hemispherical observation around the robot using the smartphone’s camera, which provides information about its environment. A computer or tablet is required to program the robot. The device on which the application is created is the smartphone, which must be in the same Wi-Fi network as the computer. Depending on

the age, knowledge, and competence of the students, different programming languages are supported. Using the Robobo framework, users can create custom Android applications that control the robot's behavior. The applications use the interfaces for programming the Robobo modules, for accessing the various functionalities of the robot, for building its new behavior, as well as for solving various tasks.

The usage of a smartphone as the robot's "head" allows the development of complex applications using its computing power, which include image recognition, speech, connection to remote data warehouses, use of multi-agent systems, and others. With each new generation of smartphones, their sensor network and computing power improve.

Practically, through Robobo robots, students from STEM school centers can use information from different domains stored in the VES digital libraries, as well as participate in game-based learning in the system.

Let us look at the following learning scenario: A training greenhouse has been set up in the school STEM center, where different types of herbs are grown. The digital libraries of the Virtual Education Space store information about herbs as plant species, their application for healthy living, and the technology for their cultivation and use. Through a specialized interface designed by us, the Robobo robots in the experimental STEM center periodically communicate with the VES Robobo robots and receive information about the technology of growing various herbs. The students in the STEM groups use this information and organize the activities for growing the plants. In the current conditions of learning in an electronic environment due to COVID-19, the need to remotely monitor the development of herbs in the training greenhouse proved to be of particular importance. Through a special mobile application, the robot can be given the task to visit a certain plot of the greenhouse and photograph the herbs in it. Thus, by analyzing the obtained images, the students from the STEM group are able to decide, for example, whether irrigation is necessary and to start the drip irrigation system (Figure 17).



**Figure 17.** Application of the Robobo robot in a school STEM greenhouse.

Robobo robots can also participate in game-based learning of other STEM student groups. For example, a robot can move to any plant in the training greenhouse, capture a specific plant, and determine its appearance by recognizing images. Then, it can start the game "Guess the plant" through two-way communication with the student's Personal Assistant.

Another aspect of the application of robots in the STEM center is their programming by IT groups. Student-created mobile apps can control the robot's behavior and extend its functionality. An example is the creation of games such as "Question games", which generate random questions of knowledge structured in the VES ontologies [34]. The programming is implemented in one of the programming languages studied at school (Figure 18) and the created application is launched on a mobile device (Robobo's head). According to the answer received, the robot performs different actions and expresses different emotions—it is happy and it dances, or it is dissatisfied and unhappy. In addition,

due to the peculiarities of its structure and the capabilities of the mobile device, Robobo can talk, ask questions, give advice, etc.



**Figure 18.** Game-based learning with ROBOBO in the experimental STEM center.

We have conducted a survey of students' work satisfaction regarding the whole period of work with Robobo robots in the STEM experimental school center. As can be seen from Figure 18, over 83% of students express a very good and positive attitude towards this approach to learning. The results show that the proposed method stimulates the students' interest and desire to study and increases their motivation and activity.

In the last two years, our team has been working in the field of training in artificial intelligence in high schools in Bulgaria [35]. The results of the theoretical training as well as the integrated work of STEM centers and VES give us reason to claim that this approach will encourage students to more systematic and in-depth study of the theoretical and practical aspects of classical and modern artificial intelligence in school education.

## 7. Conclusions

This paper presents a prototype of a distributed educational platform supporting shareable use of teaching material at university and in STEM centers in secondary schools. The platform implements the concept of integrated domains. As a multi-agent, it provides effective using, adapting, transfer, and sharing of the learning process. The implemented prototype of the platform is also described. The represented tools and services such as the use of personal assistants, electronic diary, specialized operational assistants for statistical processing of students' knowledge, etc., provide an opportunity for monitoring and partial management of the learning process in individual interest clubs in experimental STEM centers.

The use of the platform is demonstrated by two educational games. The first example is implemented directly in selected secondary schools. The training of students in STEM centers provides an opportunity to apply new approaches related to the use of virtual and augmented reality, as well as to experiment with models presenting an almost real world.

Two groups of users are trained with the platform. At the university, students are involved in the development of game-based learning content. In the STEM centers, the students use educational games as a means for acquiring knowledge in the respective subject.

**Author Contributions:** Methodology, T.G.; Writing—original draft, V.T.-K., A.S.-D. and L.D.; Writing—review & editing, S.S. and V.I. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Acknowledgments:** This work is supported by the Bulgarian Ministry of Education and Science under the National Research Program “Smart crop production” approved by Decision of the Ministry Council №866/26.11.2020. We acknowledge the provided access to the e-infrastructure of the Centre for Advanced Computing and Data Processing, with the financial support of the Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program (2014–2020) and co-financed by the European Union through the European structural and Investment funds.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Doychev, E.; Stoyanova-Doycheva, A.; Stoyanov, S.; Glushkova, T.; Ivanova, V. An IoT Virtual eLearning Space. In *Hybrid Learning and Education*; Springer Science and Business Media LLC: Basel, Switzerland, 2020; pp. 148–169.
2. NSEC (Network of STEAM Educational Centers). Available online: <https://serc.carleton.edu/StemEdCenters/index.html> (accessed on 18 December 2021).
3. Erasmus+ Program Call 2017–KA2, “Do Well Science”, 2017-1-IT02-KA201-036780. Available online: <https://www.dowellscience.eu/project/> (accessed on 15 December 2021).
4. #EuFactor Project. Available online: [www.nextadiv.it/project/eufactor-il-genio-e-dentro-di-te](http://www.nextadiv.it/project/eufactor-il-genio-e-dentro-di-te) (accessed on 12 December 2021).
5. EU STEM Coalition. Available online: [www.stemcoalition.eu](http://www.stemcoalition.eu) (accessed on 12 December 2021).
6. Научно-практическое образование, исследовательское обучение, STEM-образование: новые типы образовательных ситуаций, Сборник докладов IX Международной научно-практической конференции «Исследовательская деятельность учащихся в современном образовательном пространстве», Москва. 2018. Available online: <http://www.rdit.kg/files/books/STEM-%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/222421729.pdf> (accessed on 12 December 2021).
7. Волосовец, Т.В.; Маркова, В.А.; Аверин, С.А. «STEM образование для детей дошкольного и младшего школьного возраста» парциальная модульная программа развития интеллектуальных способностей в процессе познавательной деятельности и вовлечения в научно-техническое творчество. М.: [б.и.], 112 с. 2017. Available online: <http://gel-school-12.ru/wp-content/uploads/2019/01/prog.pdf> (accessed on 18 December 2021).
8. Rocard, M. *Science Education NOW: A Renewed Pedagogy for the Future of Europe*; Office for Official Publications for the European Commission: Luxembourg, 2007. Available online: <https://www.eesc.europa.eu/sites/default/files/resources/docs/rapportrocardfinal.pdf> (accessed on 20 December 2021).
9. Sampson, D. *Instructional Design*; Course Lectures; University Piraeus: Piraeus, Greece, 2010.
10. Dikke, D.; Tsourlidaki, E.; Zervas, P.; Cao, Y.; Faltin, N.; Sotiriou, S.; Sampson, D.G. Towards a federation of online labs for inquiry based science education at School. In *Proceedings of the 6th International Conference on Education and New Learning Technologies (EDULEARN 2014)*, Barcelona, Spain, 22–24 July 2014.
11. Kapp, K.M. *The Gamification of Learning and Instruction: Gamebased Methods and Strategies for Training and Education*; John Wiley & Sons: Hoboken, NJ, USA, 2012; ISBN 978-1-118-09634-5.
12. Shatz, I. Using Gamification and Gaming to Promote Risk Taking in the Language Learning Process (PDF). In *Proceedings of the MEITAL National Conference*; Technion: Haifa, Israel, 2015; pp. 227–232.
13. Huang, W.H.-Y.; Soman, D. *A Practitioner’s Guide to Gamification of Education (PDF) (Report)*; Research Report Series Behavioural Economics in Action; Rotman School of Management, University of Toronto: Toronto, ON, Canada, 2013.
14. Lugmayr, A.; Sutinen, E.; Suhonen, J.; Sedano, C.I.; Hlavacs, H.; Montero, C.S. Serious storytelling—A first definition and review. *Multimedia Tools Appl.* **2017**, *76*, 15707–15733. [CrossRef]
15. Pavlus, J. The Game of Life. *Sci. Am.* **2010**, *303*, 43–44. [CrossRef]
16. Klopfer, E.; Osterweil, S.; Salen, K. *Moving Learning Games forward (PDF) (Report)*; The Education Arcade/Massachusetts Institute of Technology: Mountain View, CA, USA, 2009.
17. Lee, J.J.; Hammer, J. Gamification in education: What, how, why bother? *Acad. Exch. Q.* **2011**, *15*, 146.
18. Li, C.; Dong, Z.; Untch, R.H.; Chasteen, M. Engaging Computer Science Students through Gamification in an Online Social Network Based Collaborative Learning Environment. *Int. J. Inf. Educ. Technol.* **2013**, *3*, 72–77. [CrossRef]
19. Gooch, D.; Vasalou, A.; Benton, L.; Khaled, R. Using Gamification to Motivate Students with Dyslexia. In *Proceedings of the CHI’16: CHI Conference on Human Factors in Computing Systems*, San Jose, CA, USA, 7–12 May 2016; pp. 969–980.
20. Stoyanov, S.; Popchev, I.; Doychev, E.; Mitev, D.; Valkanov, V.; Stoyanova-Doycheva, A.; Valkanova, V.; Minov, I. *DeLC Educational Portal, Cybernetics and Information Technologies (CIT)*; Bulgarian Academy of Sciences: Sofia, Bulgaria, 2010; Volume 10, pp. 49–69.
21. Doychev, E. Environment for E-Learning Services. Ph.D. Thesis, Plovdiv University, Plovdiv, Bulgaria, 2013.
22. SCORM 2004. Available online: <https://www.elearninglearning.com/adl/scorm/> (accessed on 12 December 2021).
23. QTI. Available online: <https://ed.link/community/ims-question-test-interoperability-qi-specification/> (accessed on 12 December 2021).
24. Mitev, D.; Stoyanov, S.; Popchev, I. *Selbo2—An Environment for Creating Electronic Content in Software Engineering, Cybernetics and Information Technologies (CIT)*; Bulgarian Academy of Sciences: Sofia, Bulgaria, 2009; Volume 9, pp. 96–105.
25. Wooldridge, M. *An Introduction to MultiAgent Systems*; Wiley: Chichester, UK, 2009.

26. Wang, F.-Y. The Emergence of Intelligent Enterprises: From CPS to CPSS. *IEEE Intell. Syst.* **2010**, *25*, 85–88. [[CrossRef](#)]
27. Todorov, J.; Valkanov, V.; Stoyanov, S.; Daskalov, B.; Popchev, I.; Orozova, D. Personal Assistants in a Virtual Education Space. In *Practical Issues of Intelligent Innovations*; Sgurev, V., Jotsov, V., Kacprzyk, J., Eds.; Springer Book Series “Computational Intelligence”; Springer: Berlin/Heidelberg, Germany, 2018; Chapter 6, pp. 131–153.
28. Todorov, J.; Krasteva, I.; Ivanova, V.; Doychev, E. BLISS-A CPSS-like Application for Lifelong Learning. In Proceedings of the 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 3–5 July 2019; pp. 1–5.
29. Boissier, O.; Bordini, R.H.; Hübner, J.F.; Ricci, A.; Santi, A. Multi-agent oriented programming with JaCaMo. *Sci. Comput. Program.* **2013**, *78*, 747–761. [[CrossRef](#)]
30. Russel, S.; Norvig, P. *Artificial Intelligence. A Modern Approach*; Pearson: London, UK, 2021.
31. SWI Prolog. Available online: <https://www.swi-prolog.org/> (accessed on 24 October 2021).
32. Alimisis, D.; Moro, M. Special issue on educational robotics. *Robot. Auton. Syst.* **2016**, *77*, 74–75. [[CrossRef](#)]
33. Bellas, F.; Naya, M.; Varela, G.; Llamas, L.; Prieto, A.; Becerra, J.C.; Bautista, M.; Faiña, A.; Duro, R. The Robobo Project: Bringing Educational Robotics Closer to Real-World Applications. In *Advances in Human Error, Reliability, Resilience, and Performance*; Springer Science and Business Media LLC: Basel, Switzerland, 2017; Volume 630, pp. 226–237.
34. Stoyanova-Doycheva, A.; Stancheva, N.; Ivanova, V.; Stoyanov, S. Structure of an ontology used in a test generation environment. *Thermophys. Basis Energy Technol. (Tbet 2020)* **2021**, *2333*, 050005. [[CrossRef](#)]
35. Glushkova, T.; Stoyanov, S.; Tabakova-Komsalova, V.; Grancharova-Hristova, M.; Krasteva, I. An Approach to Teaching Artificial Intelligence in School, Innovative Educational Technologies, Tools and Methods for E-learning Scientific Editor Eugenia Smyrnova-Trybulska “E-learning”, 12, Katowice–Cieszyn. 2020, pp. 257–267. Available online: <http://studio-noa.pl/doi/e-learning/12/el-2020-12.pdf> (accessed on 20 December 2021).