ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Innovative technologies and scientific solutions for industries. 2019. No. 2 (8)

UDC 681.518+37.013.32
DOI: https://doi.org/10.30837/2522-9818.2019.8.062

JU. MALIEIEVA, V. KOSENKO, O. MALYEYEVA, D. SVETLICHNYJ

CREATION OF COLLABORATIVE DEVELOPMENT ENVIRONMENT IN THE SYSTEM OF DISTANCE LEARNING

The **subject** of the study in the article is methods and tools for the creation and implementation of distance learning systems (DLS). The interactive capabilities used in the distance education system, programs and information delivery systems allow for feedback, dialogue and ongoing support that are not possible in most traditional learning systems. The **purpose** of the article is to create a collaborative environment for developing software products in the system of distance learning by forming an interactive platform with the help of Sphere Engine, which provides the ability to debug the code in many programming languages. At the same time, the following **tasks** are solved: a review and comparison of existing systems of distance learning; analysis of the functional and features of the Sphere Engine platform; development of the integration algorithm of the service with the system Moodle. **Methods**: methods of information processing, methods and information technologies for the creation of electronic distance learning courses. **Results.** As a result of the review and comparison of existing distance learning systems and software verification tools, the tools that best meet the needs for collaborative development environments, namely the Moodle DLS and the Sphere Engine service, are selected. The analysis of the functional and features of the Sphere Engine platform is carried out: a diagram of the work of the api compilers of the service is shown, which illustrates the process of execution of the program from the moment of its sending to the results of execution; The concept of a pool script in graphical form is provided. A prototype collaborative development environment created in the form of a web application is created, the prototype architecture is described. An algorithm for integration of the Sphere Engine service with the Moodle system is developed. In detail, the stages of the integration algorithm are considered on the example of testing the work environment of the development of the course "Web-technologies and web-design". **Conclusions**. An interactive module is created with the help of Sphere Engine, which allows you to perform laboratory work on programming disciplines and to carry out full control over their execution. Thus, the degree of automation of the educational process in institutions of higher education increases.

Keywords: distance learning; collaborative environment; integration algorithm; web service.

Introduction

With the development of information technology, the society faced an important problem, which is to create a new promising system of education, which should prepare society for life in the new conditions of civilization. This explains the emergence of a new form of education – distance, along with the forms already known and traditional – stationary, correspondence, external studies, etc [1].

To ensure the successful design, development and implementation of a distance learning system, it is necessary to create the infrastructure of the relevant educational institutions, train teachers, develop curricula, etc. An important part of distance learning is its implementation using information technology, namely Learning Management Systems, which are designed to develop, manage and distribute online teaching materials to ensure the sharing of many users. [2, 3].

Modern computer telecommunications can provide the transfer of knowledge and access to a variety of educational information on an equal, and sometimes more efficient, level than traditional learning tools. Experiments have confirmed that the quality and structure of training courses, as well as the quality of teaching in distance learning, is often much better than with traditional forms of learning [4, 5].

New e-technologies such as interactive disks, electronic bulletin boards, multimedia hypertext, accessible through the global Internet network, provide active involvement of students in the learning process. In addition, these technologies allow you to manage the learning process unlike most traditional learning environments. Integration of sound, movement, images and text creates a rich learning environment, with the development of which will increase the degree of involvement of students in the learning process.

The interactive capabilities used in the distance education system (DLS), programs and information delivery systems allow you to set up and even stimulate feedback, provide a dialogue and ongoing support that is not possible in most traditional training systems [6, 7].

Analysis of publications and distance learning systems

Many publications are devoted to the development of technologies and systems of distance learning. Among them are the works of C. Wademeyer, R. Harrison, D. Kigan, M. Moore, D. Schell, which testify that such systems operate in universities of many countries. DLSs are actively implemented abroad, in particular in the US by leading universities in Stanford, Harvard, Princeton, Massachusetts Institute of Technology (MIT) [8, 9]. R. Kulatta and M. Beynton note that users are offered a choice of free on-line courses on various technical disciplines.

In accordance with the concept of the development of distance education in Ukraine [10], there is now an increase in the number of higher education institutions that use the distance education. According to the results of the analysis of the development of DLS in institutions of higher education in Ukraine, conducted by Kravtsov G.M. [11], it can be argued that most higher education institutions develop their own distance learning systems (DLSs) that solve internal learning tasks, have specific document processing protocols, which implement individualized distance learning technologies.

In the dissertation, Pikulyak M.V. considers methods and instrumental means of constructing an adaptive system of distance education [12]. Dissertation work of Kuta V.I. is devoted to the development and formation of a complex of information technologies of distance learning systems for people with special needs [13].

Among the technologies used in the process of distance learning, the most widely used:

- case technology - a form of distance learning, based on the provision of educational information educational resources in the form of specialized sets of educational and methodological complexes using different types of information carriers (cases);

- Internet technology is a way of remotely transmitting information based on the use of global and local computer networks to provide students with access to information educational resources and to formulate a set of methodological, organizational, technical and software tools for implementation and management of the learning process, regardless of location its subjects;

- telecommunication technology – a technology based on the use of satellite and data transmission and broadcasts as well as global and local networks to provide interaction between students and teachers and the access of students to information educational resources presented in the form of digital libraries, video collections and other learning tools.

The main criterion for evaluating and implementing one or another educational technology (form, methodology) of education in the education system is its effectiveness.

We will conduct a comparative assessment of the existing systems of distance learning. Among them, the main criterion for evaluation is compliance with the standard (SCORM) (Sharable Content Object Reference Model). This is a set of specifications and standards developed specifically for the DLS at the international level. It contains requirements for the design of teaching materials, the formation of the learning process and other criteria that ensure the universality and scalability of the system. The relevance of the course to the format of SCORM provides the integration of the course into other systems and its multiple uses: all methodological materials are represented by blocks that can be used to create different courses, regardless of the environment in which they were created.

For the most part, the use of DLS is permissible only as part of the educational process, which involves the regular personal contact of students and teachers. Remoteness is used only as a tool for facilitating access to methodological materials and providing conditions for independent forms of educational work, which is especially relevant in connection with an increase in the number of hours allocated to this format in curriculum. This technology of interaction in the international space is called "Blended Learning" – a mixed learning. Within the framework of the given conception it is envisaged to obtain knowledge independently, for example, online, and internally - with a teacher. This approach allows you to control the time, place, pace and way of studying the material. Mixed education allows you to combine traditional techniques and topical technologies.

Comparison of the DLS, the appropriate condition for supporting the SCORM format, is presented in table 1, which clearly demonstrates the basic capabilities of the described systems.

Table 1. *Comparative characteristics of the DLS*

DLS	**License type**	**Architecture**	**Testing**	**Interface (10 Types of Questions)**	**Management**	**Communications**	**Online Conference**	**Development**	**Extras**
Infotechno	Commercial	Closed	+	+	Group	Message, chat, forum	+	Web GUI	Document flow, blackboard
Docent	Commercial	Closed	-	+	Calendar + Group	Chat, forum	+	Web GUI	APM, emulation
Web Tutor	Commercial	Closed	-	+	Calendar	Message, chat, forum	-	GUI	Virtual classroom
Prometheus	Commercial	Closed a	+	+	Calendar	Message, chat, forum	-	Web GUI	Team work, 3 test modes
Competent Shared Knowledge	Commercial	Closed	+	+	Group	Message, chat, forum	+	Web GUI	Integration with MS SharePoint
eLearning Server	Commercial	Open	-	+	Calendar + Group	Message, chat, forum	+	Web GUI	Electronic library
Redclass	Commercial	Closed	-	-	Group	Chat, forum	-	GUI	Emulation, virtual laboratory, electronic textbook
Moodle	GNU GPL	Open	+	+	Calendar + Group	Message, chat, forum, blogs	+	Web GUI	A large number of additional modules and the possibility of self-creation depending on the needs

There are a number of similar software products, such as Echo-360, e-Manual, IBM Simulation Producer, Itrain, UPK, Courselab and others. During the study, only those systems that are most widely represented in the market of educational services were considered, and all of the above systems are in demand, mainly in corporate education.

According to the available functionality, all systems are similar, except for individual modules. Each of them is intended for full-fledged provision of electronic resources for training; systems represent a complete set of functions necessary for managing the learning process. Each solution provides modules for creating training courses. There is an opportunity of individual communication with students. At different levels of implementation, administrative control systems have been introduced. Functional representation of the systems fully meets the requirements for organizing e-learning and provides a high level of educational process.

Some systems have unique components, such as the DLS Docent has an APM and Infotechno contains a virtual classroom board that may be important for the certain educational projects, but no significant differences have been identified.

The most popular among the existing distance learning systems is the Moodle system, which has a number of obvious advantages [14–19]:

- convenient intuitive interface;
- the possibility of organizing a large number of educational work;
- accessibility for users and developers;
- "flexible" course setup to suit the needs.

In the electronic system, Moodle may create a training framework for disciplines that do not require the use of third-party software. Moodle also has its own set of APIs for interfacing with external services.

It should be noted the importance of creating collaborative learning environments. This is emphasized by the widespread dissemination of e-learning facilities at a high school, which opens up broad prospects for achieving a qualitatively new level of cooperation among the participants in the educational process. Using a collaborative development environment, you can achieve a new level of teaching and learning programming. For example, remote programming workshops using a collaborative development environment can provide better perceptions than using utilities to stream current screen content. The created collaborative development environment will be a major addition to the existing tools for e-learning programming.

For full migration from classical integrated development environments to collaborative environments, it is necessary to transfer the entire functionality available in desktop systems. This process is complicated by the fact that collaborative development environments change the social aspects of software development organization. Thus, the development of the created prototype and its implementation will be an important step in the education of professional ethics of future software developers in terms of successful collective creativity. The results of the annual student survey confirm the feasibility and relevance of group learning projects. Therefore, the introduction of the facilities created in the learning process will enable an analysis of the effectiveness of the use of collaborative environments for e-learning systems for group programming, in particular, for example, the design of web applications.

Most DLSs have a disadvantage: it is difficult to verify the code of the training software product. But for IT disciplines (such as "web-technologies and web-design"), it is possible to create an interactive platform with the help of Sphere Engine, which provides the ability to debug the code in many programming languages (including php, JavaScript, CSS, HTML) and provides a wide range of APIs for interfacing with external services.

Therefore, the purpose of the article is to create a collaborative software development environment in the system of distance learning by forming an interactive platform with the help of Sphere Engine, which provides the ability to debug the code in many programming languages. This solves the following tasks:

- analysis of the functional and features of the Sphere Engine platform;
- development of the integration of the Sphere Engine service with the Moodle system.

Materials of research

For a well-functioning collaborative development environment requires the use of a third-party external service, which will be an online compiler with a large set of programming languages. In addition, such a service must work with the Moodle system by integrating it. Today many services allow the use of compilers directly in the browser. Only one service provides a wide list of api-functions for working with it from external applications - this is Sphere Engine (https://sphere-engine.com). With the use of this service, developers of the collaborative environment eliminate the need to deploy it on servers to support compilation, configuration and API creation. This service also has widgets for integration into electronic learning systems [20–22].

Sphere Engine is a platform that supports training and testing of IT skills. This technology allows you to check automatically the programming skills in real time. In addition, the service allows you to execute remotely the computer programs in a safe environment.

Sphere Engine can be used in many areas, including:

- IT education (programming for learning);
- courses of e-learning disciplines, which are accompanied by interactive examples and exercises);
- when testing the programming skills as a platform for conducting interviews while recruiting staff with the ability to run code written by candidates;
- hakatons and programming competitions as a system for sharing the executable source code;
- in the cloud environment of the IDE, etc.

The Sphere Engine service allows:

- download the code;
- launch a program with incoming data on the server with support for most popular programming languages;
- receive the result of execution.

API Sphere Compiler Engine offers the ability to execute programs in dozens of programming languages, such as: C ++, C #, Go, Haskell, Java, Kotlin, Node.js, PHP, Python, Ruby, Scala or Swift and hypertext markup languages html.

The system consists of two modules:

1. Compilers – allows you to remotely execute computer programs and download the results of their execution, control the implementation of the source code of the program, add the value of input data.

2. Problems – Carries out an automatic check of correctness of computer programs in the context of programming problems.

Integration of both modules is possible with any of the following methods:

1) widget integration - a quick and easy way to integrate, recommended for expansion of existing projects;

2) RESTful API - a way to integrate with a wide selection of options recommended for creating specialized applications "from scratch".

Data transfer through API is performed using HTTP parameters. Therefore, any client library that supports HTTP can be used to integrate with the Sphere Engine service.

The integration of the Problems Sphere Engine module into the learning management system (in our case, Moodle) allows you to embed interactive programming tasks in them.

Fig. 1 shows the scheme of the Api Compilers service, which illustrates the process of execution of the program from the moment of its sending, and the results of execution.

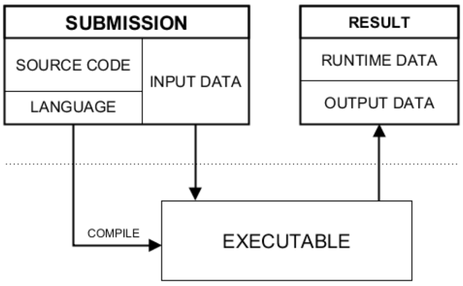


Fig. 1. Scheme of work of Api Compilers service

Data available after execution of the program include: execution time, memory consumption, output signal and data streams (source data generated, execution errors, warnings and compilation errors).

All URLs that are related to the Sphere Engine Compilers API have a common base in the form http: // <endpoint> / api / v4 /, which is called the endpoint. The Sphere Engine service allows you to use an access token for each use of the API method.

For example, the test method available at the relative address / test must be executed using the next address along with the access token: http: // <endpoint> / api / v4 / test? Access_token = <access_token>.

The API Sphere Engine Compiles is compatible with the REST architecture. This is due to the following characteristics:

- each API uses an HTTP request;

- the HTTP request method defines the type of API method (GET – receiving data, POST – creating a new resource, PUT – updating the resource data, DELETE – deleting the resource).

The answer is always returned in JSON format. In the case of an error when executing the API method (that is, the response code, starting with the number 4), the structure of the response has the following form:

```
{
  "error_code": xxx
  "message": " message that describes the error "
}
```

The error_code field contains an integer that identifies the error and allows you to easily interpret the error.

The program that is sent for execution in the Sphere Engine system is called a submission. The training consists of the source code of the program, the programming language and (optionally) the input data that is processed by the program. The response of the API method to obtain submission data (ie GET / submissions /: id) contains two fields that determine the current state of the submission.

The Binary Run field helps you quickly check whether the system processes the submission. Can take the following values:

1) true – representation in the process of processing the system, and the data created by the program are not yet available.

2) false - submission already left part of Sphere Engine system responsible for executing programs, and all information is finally installed and available.

After creating a new request, the system needs some time to execute it. The following scenario is recommended: create a request using the POST / Submissions method; waiting 2 seconds; Receive sending data using GET / submitings /: id; representation of the current state of presentation to the user (optional). If executing = true, you need to go back to the previous step. Fig. 2 presents the concept of this scenario in graphical form.

The disadvantages of the API Sphere Engine include the fact that it works only with single-file downloads, that is, it is not possible to download a large structured project. In addition, it is not possible to set breakpoints for its debugging.

In the case of an intense load on the server (that is, if there is no time interval between requests), in anticipation of processing, this can be recognized as a DoS / DdoS attack and can lead to a reaction of the API's Sphere Engine's security system. In this case, the Sphere Engine servers may refuse further access to the service.

Integration of the Sphere Engine service into Moodle system

The use of a third-party external service, which is an online compiler with a large set of programming languages, is illustrated by the example of monitoring the implementation of laboratory work on discipline "Web-technology and web-design". The specified service interacts with the Moodle system by integrating it. An important point is also the availability of API functions to access certain elements of the service for proper setup.

Fig. 3 presents a generalized algorithm for integrating the Sphere Engine service into the Moodle system.

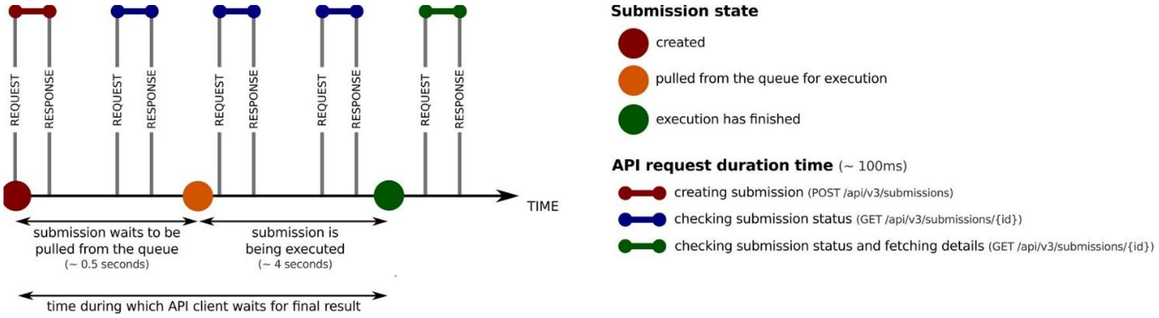

REQUEST
RESPONSE
TIME
submission waits to be pulled from the queue (~ 0.5 seconds)
submission is being executed (~ 4 seconds)
time during which API client waits for final result
Submission state
created
pulled from the queue for execution
execution has finished
API request duration time (~ 100ms)
creating submission (POST /api/v3/submissions)
checking submission status (GET /api/v3/submissions/{id})
checking submission status and fetching details (GET /api/v3/submissions/{id})

Fig. 2. Illustration of Pool Submission Script

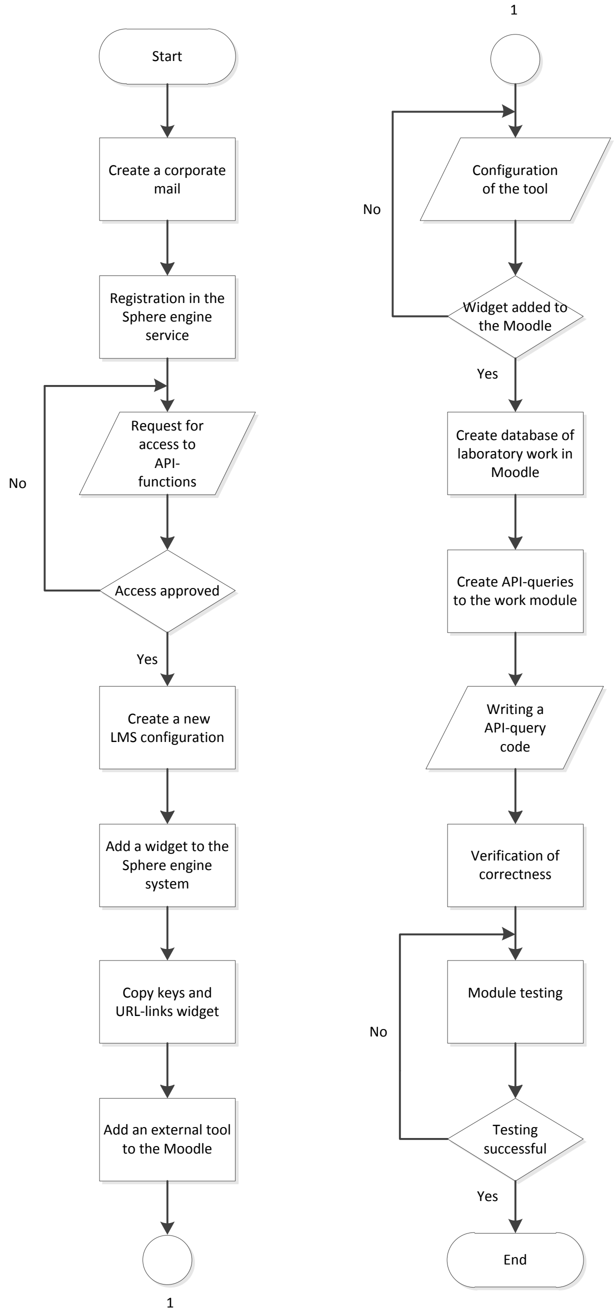

Start
Create a corporate mail
Registration in the Sphere engine service
Request for access to API-functions
Access approved
No
Yes
Create a new LMS configuration
Add a widget to the Sphere engine system
Copy keys and URL-links widget
Add an external tool to the Moodle
1
1
Configuration of the tool
Widget added to the Moodle
No
Yes
Create database of laboratory work in Moodle
Create API-queries to the work module
Writing a API-query code
Verification of correctness
Module testing
Testing successful
No
Yes
End

Fig. 3. A generalized algorithm for integrating the Sphere Engine

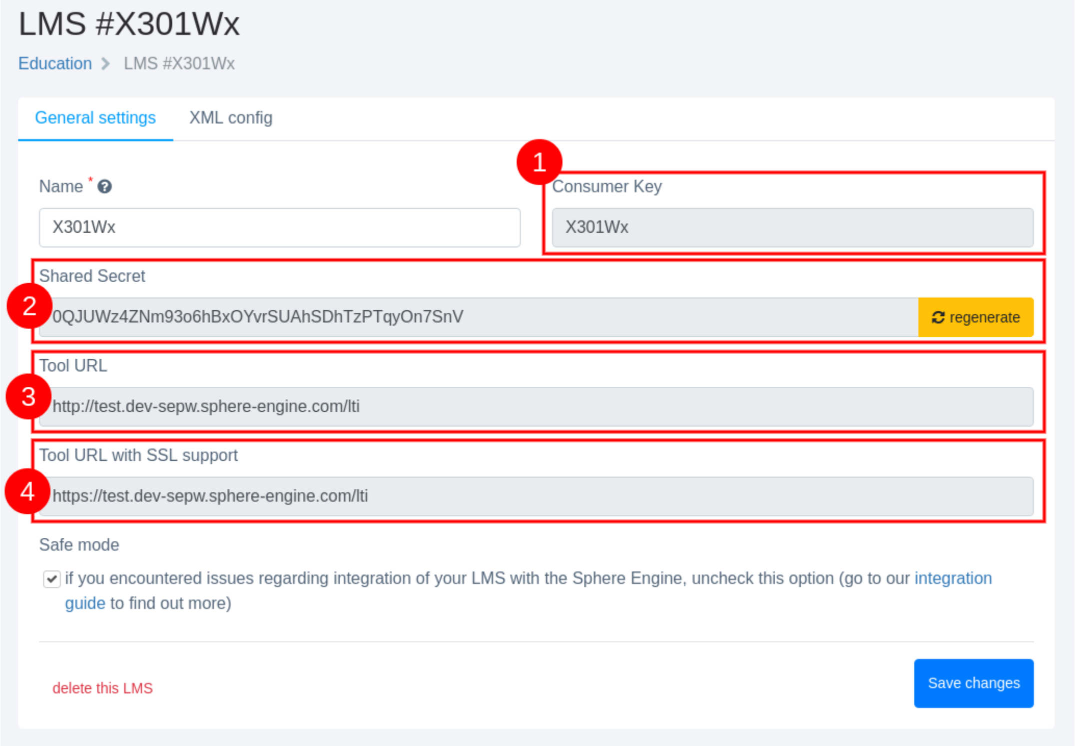
Let us consider the main stages of the developed algorithm.

To access the API functions that will be used to access the functionality of the service from Moodle, you must create a corresponding request.

When integrating the Sphere Engine, a configuration set for the LMS system is created that integrates with the Sphere Engine for education in a manner that is consistent with LTI (fig. 4): consumer key (1), general key (2), tool URL (HTTP (3) or HTTPS (4)).

Fig. 5 shows the XML (5) data needed for an automated configuration that can be used in two ways: through a special URL (6) by copying it directly from the text field (7).

A separate configuration has been used for each LMS (fig. 6).



LMS #X301Wx

Education > LMS #X301Wx

Name * X301Wx

Consumer Key X301Wx

Shared Secret 0QJUWz4ZNm93o6hBxOYvrSUAhSDhTzPTqyOn7SnV

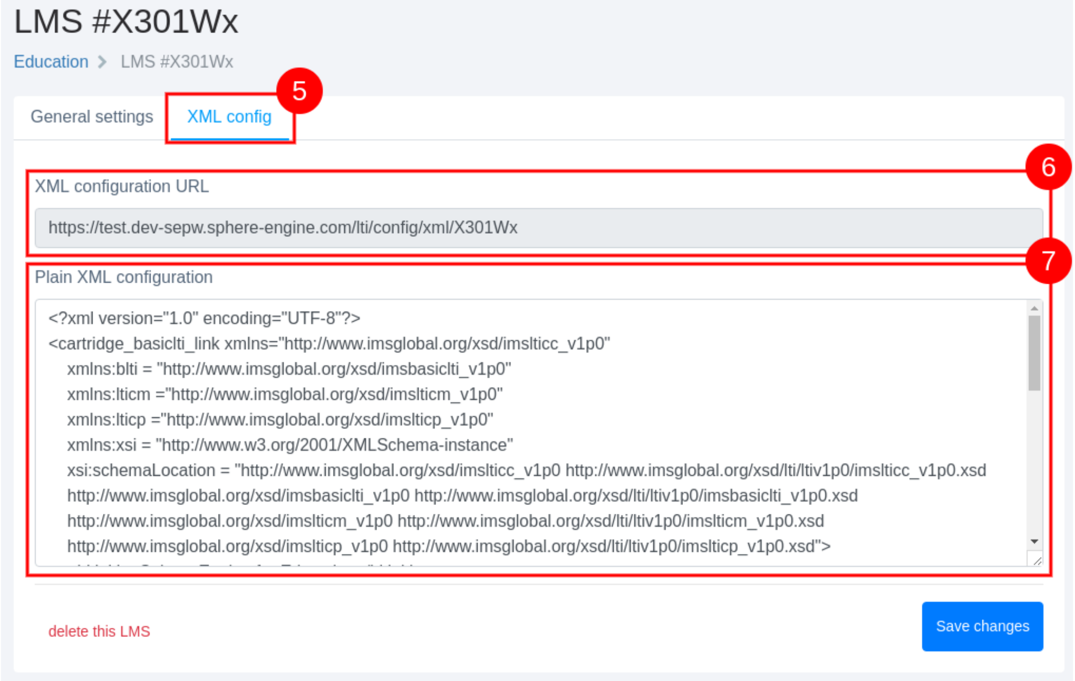
Tool URL http://test.dev-sepw.sphere-engine.com/lti

Tool URL with SSL support https://test.dev-sepw.sphere-engine.com/lti

Safe mode

☑ if you encountered issues regarding integration of your LMS with the Sphere Engine, uncheck this option (go to our integration guide to find out more)

Fig. 4. Configuration data



LMS #X301Wx

Education > LMS #X301Wx

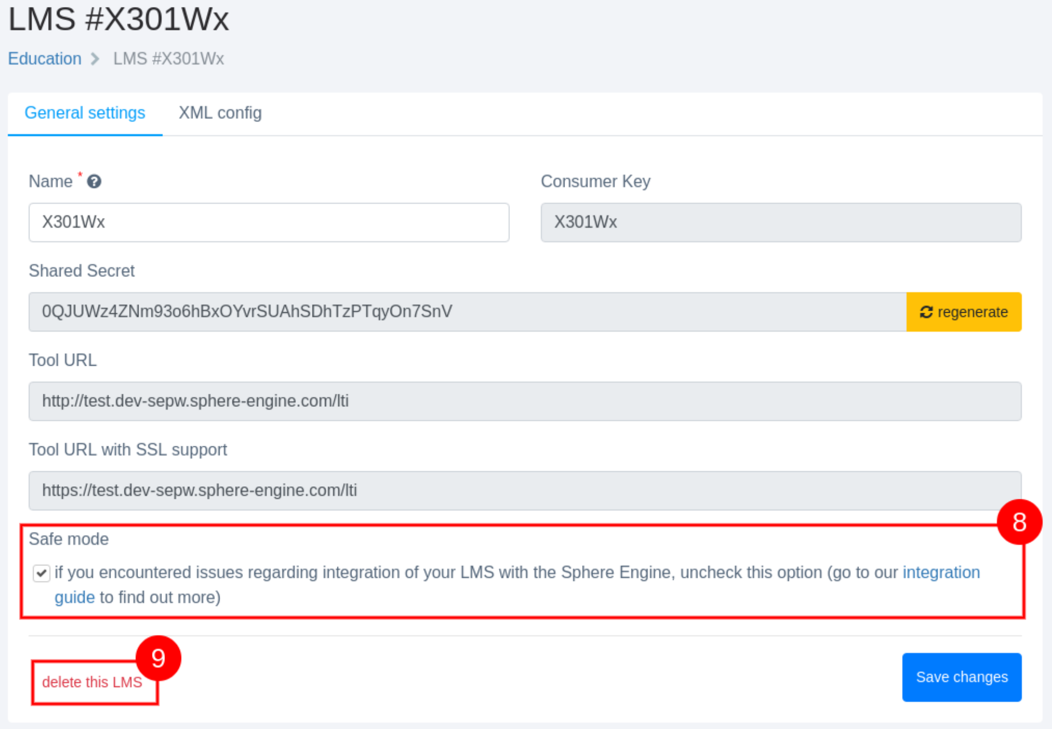
XML configuration URL https://test.dev-sepw.sphere-engine.com/lti/config/xml/X301Wx

Plain XML configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<cartridge_basiclti_link xmlns="http://www.imsglobal.org/xsd/imslticc_v1p0"
   xmlns:blti = "http://www.imsglobal.org/xsd/imsbasiclti_v1p0"
   xmlns:lticm ="http://www.imsglobal.org/xsd/imslticm_v1p0"
   xmlns:lticp ="http://www.imsglobal.org/xsd/imslticp_v1p0"
   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation = "http://www.imsglobal.org/xsd/imslticc_v1p0 http://www.imsglobal.org/xsd/lti/ltiv1p0/imslticc_v1p0.xsd
   http://www.imsglobal.org/xsd/imsbasiclti_v1p0 http://www.imsglobal.org/xsd/lti/ltiv1p0/imsbasiclti_v1p0.xsd
   http://www.imsglobal.org/xsd/imslticm_v1p0 http://www.imsglobal.org/xsd/lti/ltiv1p0/imslticm_v1p0.xsd
   http://www.imsglobal.org/xsd/imslticp_v1p0 http://www.imsglobal.org/xsd/lti/ltiv1p0/imslticp_v1p0.xsd">
```

Fig. 5. XML data for automated configuration

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)



LMS #X301Wx

Education > LMS #X301Wx

General settings | XML config

Name *
X301Wx

Consumer Key
X301Wx

Shared Secret
0QJUWz4ZNm93o6hBxOYvrSUAhSDhTzPTqyOn7SnV
regenerate

Tool URL
http://test.dev-sepw.sphere-engine.com/lti

Tool URL with SSL support
https://test.dev-sepw.sphere-engine.com/lti

Safe mode
☑ if you encountered issues regarding integration of your LMS with the Sphere Engine, uncheck this option (go to our integration guide to find out more)

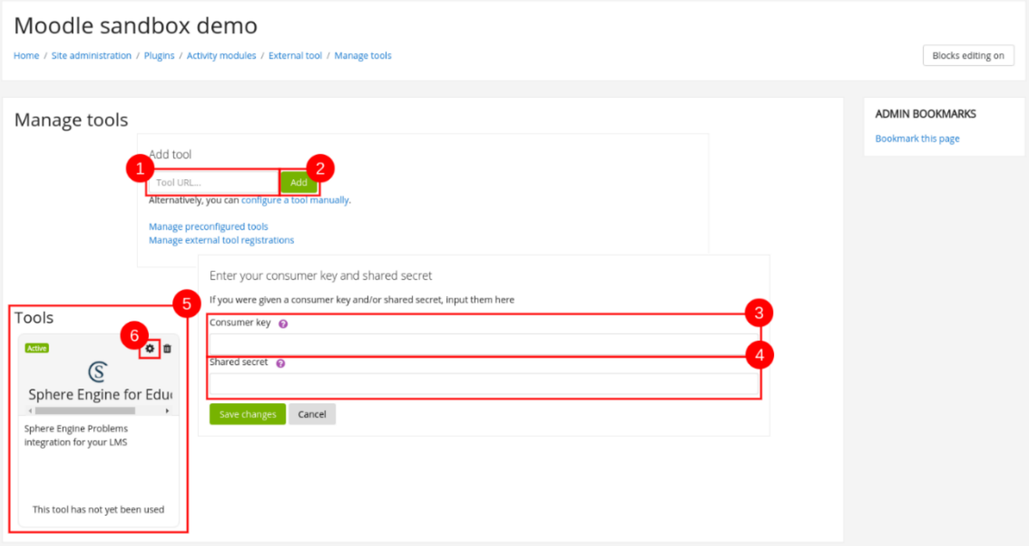
delete this LMS

Save changes

Fig. 6. Additional settings

The Moodle system offers the ability to configure an external tool (for example, Sphere Engine for Education) in such a way that the program becomes public in the resource list. The settings required for this process require administrator privileges and are available in Site Administration> Plugins> Activity modules> External tools> Manage tools. If admin use is not possible, an alternative configuration method allows you to use Sphere Engine for Education with the course configuration.

Moodle provides an automatic configuration process. The configuration window is shown in fig. 7.



Moodle sandbox demo

Home / Site administration / Plugins / Activity modules / External tool / Manage tools

Blocks editing on

Manage tools

Add tool
Tool URL...
Add
Alternatively, you can configure a tool manually.

Manage preconfigured tools
Manage external tool registrations

Enter your consumer key and shared secret
If you were given a consumer key and/or shared secret, input them here
Consumer key
Shared secret
Save changes
Cancel

Tools
Active
Sphere Engine for Edu
Sphere Engine Problems integration for your LMS
This tool has not yet been used

ADMIN BOOKMARKS
Bookmark this page

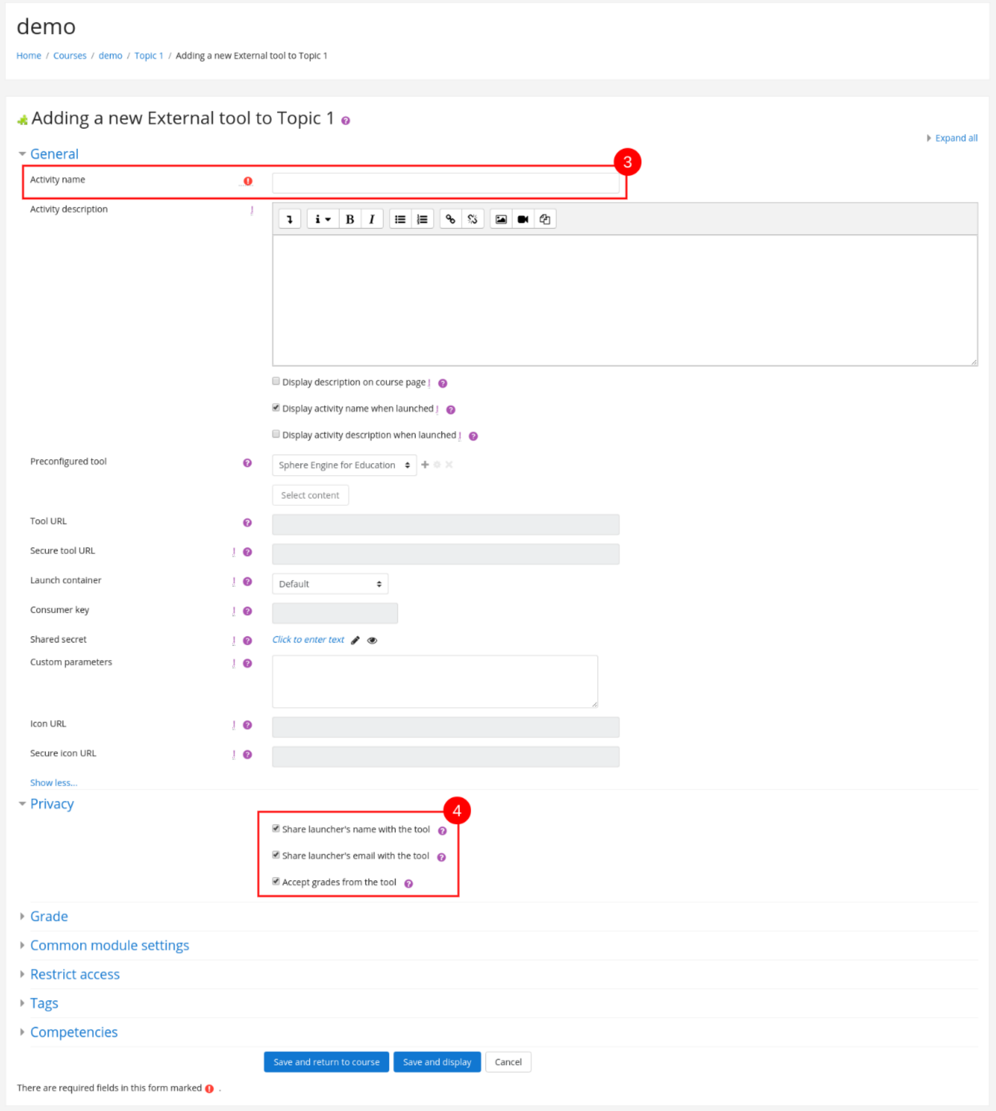
Fig. 7. Adding of the LTI configuration to Moodle system

In Moodle, the URL of the tool (1) contains the URL that is available in the appropriate field of the "XML Configuration" tab (Sphere Engine), and the Add button (2) (Moodle) is clicked. For security reasons, you must enter a consumer key (3) and a general key (4). Configure the configuration (6) for which Sphere Engine for Education will be present when adding resources.

When adding and adjusting the resource for the course (fig. 8), the required parameters are specified (resource name (3) and privacy settings (4)).

After adding the instrument to the discipline topic, it is available for use. Then using the Sphere Engine API:

- to transfer the text of the task of the lab work, which will be stored in the element of the course "Database";
- setting the time for the task.



demo

Home / Courses / demo / Topic 1 / Adding a new External tool to Topic 1

Adding a new External tool to Topic 1

General

Activity name

Activity description

□ Display description on course page

☑ Display activity name when launched

□ Display activity description when launched

Preconfigured tool: Sphere Engine for Education

Tool URL

Secure tool URL

Launch container: Default

Consumer key

Shared secret: Click to enter text

Custom parameters

Icon URL

Secure icon URL

Privacy

☑ Share launcher's name with the tool

☑ Share launcher's email with the tool

☑ Accept grades from the tool

There are required fields in this form marked .

Fig. 8. Configuration of the resource

Conclusions

Because of the review and comparison of existing distance learning systems and software verification tools, the tools that best meet the needs for a collaborative development environment, namely the Moodle DLS and the Sphere Engine service, are selected. The analysis of the functional and features of the Sphere Engine platform allowed to implement the integration of the Sphere Engine service with the Moodle distance learning system.

For the course "Web-technologies and web-design" a separate interactive module has been created with the help of Sphere Engine, which allows to perform laboratory work in this discipline. With the help of the integrated service, you can carry out full control over the progress of laboratory work. Thus, the degree of automation of the educational process in institutions of higher education increases.

References

1. Bikkulova, G. R. (2015), "The principles of distance learning as a reflection of modern requirements for the educational process" ["Printsipy distantsionnogo obucheniya kak otrazheniye sovremennykh trebovaniy k obrazovatel'nomu protsessu"], *News of the Russian State Pedagogical University A. I. Herzen*, No. 60, P. 332–335.

2. Voronkin, A. (2013), "Problems of the introduction of distance learning in domestic educational institutions" ["Problemy vnedreniya distantsionnoy formy obucheniya v otechestvennykh uchebnykh zavedeniyakh"], *E-Learning World*, No. 2 (28).

3. Kukyan, V. N., Photina, O. V. (2012), "Actual problems of distance education", *Open and distance education*, No. 4 (48), P. 86–89.

4. Galustyan, O. V (2013), "E-learning technology in the educational process" ["Tekhnologiya E-learning v obrazovatel'nom protsesse"], *Innovation in education*, No. 5, P. 128–129.

5. Omarov, M., Muradova, V. (2018), "Infological support of the information technology for management of distance education", *Innovative Technologies and Scientific Solutions for Industries*, No. 4 (6), P. 146–153. DOI: https://doi.org/10.30837/2522-9818.2018.6.146

6. Haidai, B., Artiukh, R., Malyeyeva, O. (2018), "Analysis and modelling the preferences of social networks users", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (3), P. 5–12. DOI: https://doi.org/10.30837/2522-9818.2018.3.005

7. Henry J., Meadows, J. (2008), "An Absolutely Riveting Online Course: Nine Principles for Excellence in Web-based Teaching", *Canadian Journal of Learning and Technology*, Vol. 34 (1).

8. "The Top eLearning statistics and facts for 2015", *eLearning Industry*, available at : https://elearningindustry.com/elearning-statistics-and-facts-for-2015

9. "MOOCs: Top 10 Sites for Free Education With Elite Universities", available at : http://www.bdpadetroit.org/portal/index.php ?option= com_content&view=article&id=57

10. "Concept of development of distance education in Ukraine" ["Kontseptsiya rozvytku dystantsiynoyi osvity v Ukrayini"], available at : http://www.osvita.org.ua/distance/pravo/00.html

11. Kravtsov, G. M. (2005), "Conceptual tasks of development of systems of distance learning and their implementation technology" ["Kontseptualni zadachi rozrobky system dystantsiynoho navchannya ta tekhnolohiyi yikhnoyi realizatsiyi"], *Scientific journal of NPU named after M. P. Drahomanov, Series No. 2. Computer-oriented systems of training*, Kyiv : NPU them. MP Drahomanov, No. 2 (9), P. 294–305.

12. Pikulyak, M. V. (2016), Methods and instrumental means of constructing an adaptive system of distance education" [Metody ta instrumental′ni zasoby pobudovy adaptyvnoyi systemy dystantsiynoyi osvity] : Abstract abstract Diss., Ternopil.

13. Kut, V. I. (2013), "Information technologies of distance learning systems for people with special needs" ["Informatsiyni tekhnolohiyi system dystantsiynoho navchannya osib z osoblyvymy potrebamy"], Ternopil, available at : http://elartu.tntu.edu.ua/bitstream/123456789/2036/2/Thesis-Kyt_V_I-Information_technology_for_distance_learning__2013.pdf

14. Andreev, A. B. Andreeva, S. V., Dotsenko, I. B. (2014), *Practice e-learning using Moodle* [*Praktika elektronnogo obucheniya s ispol'zovaniyem Moodle*], Taganrog : TIT SFU.

15. Demida, B., Sagaidak, S., Kopil, I. "Systems of distance learning: review, analysis, selection" ["Systemy dystantsiynoho navchannya: ohlyad, analiz, vybir"], *National University "Lviv Polytechnic"*, available at : http://ena.lp.edu.ua/bitstream/ntb/10662/1/14.pdf

16. Anisimov, A. M. (2015), *Work in the Moodle distance learning system* [*Robota v systemi dystantsiynoho navchannya Moodle*], Kharkiv, 213 p.

17. Khudyakova, A. V. (2013), *Designing a distance course on the Moodle platform 2.7* [*Proyektirovaniye distantsionnogo kursa na platforme Moodle 2.7]*, Perm : PGPPU, 32 p.

18. Kravchenko, G.V. (2013), *Construction of a distance course and organization of training for students of institutions of higher education in the Moodle system* [*Pobudova dystantsiynoho kursu i orhanizatsiya navchannya studentiv zakladiv vyshchoyi osvity v systemi Moodle]*, Altai, 122 p.

19. Trius, Yu.V. (2018), *Organization of certification of electronic training courses in ZVO by means of the system Moodle* [*Orhanizatsiya atestatsiyi elektronnykh navchal′nykh kursiv u ZVO zasobamy systemy Moodle*], Cherkasy, 312 p.

20. Donets, V., Kuchuk, N., Shmatkov, S. (2018), "Development of software of e-learning information system synthesis modeling process", *Advanced Information Systems*, Vol. 2, No. 2, P. 117–121. DOI: https://doi.org/10.20998/2522-9052.2018.2.20

21. Zykov, I., Kuchuk, N., Shmatkov, S. (2018), "Architecture synthesis of the computer system of transaction control e-learning", *Advanced Information Systems*, Vol. 2, No. 3, P. 60–66. DOI: https://doi.org/10.20998/2522-9052.2018.3.10

22. Official site of the service Sphere engine, available at : https://sphere-engine.com.

Received 31.05.2019

Відомості про авторів / Сведения об авторах / About the Authors

Малєєва Юлія Анатоліївна – кандидат технічних наук, доцент, Національний аерокосмічний університет імені М. Є. Жуковського "ХАІ", доцент кафедри комп’ютерних наук та інформаційних технологій, Харків, Україна; e-mail: juliabelokon84@gmail.com; ORCID: https://orcid.org/0000-0003-3553-9156.

Малеева Юлия Анатольевна – кандидат технических наук, доцент, Национальный аэрокосмический университет имени Н. Е. Жуковского "ХАИ", доцент кафедры компьютерных наук и информационных технологий, Харьков, Украина.

Malieieva Julia – PhD (Engineering Sciences), Associate Professor, National Aerospace University "Kharkiv Aviation Institute", Associate Professor of the Department of Computer Sciences and Information Technologies, Kharkiv.

Косенко Віктор Васильович – доктор технічних наук, доцент, Державне підприємство "Харківський науково-дослідний інститут технології машинобудування", директор, Харків, Україна; e-mail: kosv.v@ukr.ua; ORCID ID: https://orcid.org/0000-0002-4905-8508.

Косенко Виктор Васильевич – доктор технических наук, доцент, Государственное предприятие "Харьковский научно-исследовательский институт технологии машиностроения", директор, Харьков, Украина.

Kosenko Viktor – Doctor of Sciences (Engineering), Associate Professor, State Enterprise "Kharkiv Scientific-Research Institute of Mechanical Engineering Technology", Chief Executive Officer, Kharkiv, Ukraine.

Малєєва Ольга Володимирівна – доктор технічних наук, професор, Національний аерокосмічний університет імені М. Є. Жуковського "ХАІ", професор кафедри комп’ютерних наук та інформаційних технологій, Харків, Україна; e-mail: o.maleyeva@khai.edu; ORCID: https://orcid.org/0000-0002-9336-4182.

Малеева Ольга Владимировна – доктор технических наук, профессор, Национальный аэрокосмический университет имени Н. Е. Жуковского "ХАИ", профессор кафедры компьютерных наук и информационных технологий, Харьков, Украина.

Malyeyeva Olga – Doctor of Sciences (Engineering), Professor, National Aerospace University "Kharkiv Aviation Institute", Professor of the Department of Computer Science and Information Technologies, Kharkiv, Ukraine.

Світличний Дмитро Вікторович – Національний аерокосмічний університет імені М. Є. Жуковського "ХАІ", студент кафедри комп’ютерних наук та інформаційних технологій, Харків, Україна; e-mail: d48686347@gmail.com; ORCID: https://orcid.org/0000-0002-1323-8996.

Светличный Дмитрий Викторович – Национальный аэрокосмический университет имени Н. Е. Жуковского "ХАИ", студент кафедры компьютерных наук и информационных технологий, Харьков, Украина.

Svetlichnyj Dmitrij – National Aerospace University "Kharkiv Aviation Institute", Student of the Department of Computer Sciences and Information Technologies, Kharkiv, Ukraine.

СТВОРЕННЯ КОЛАБОРАТИВНОГО СЕРЕДОВИЩА РОЗРОБКИ В СИСТЕМІ ДИСТАНЦІЙНОГО НАВЧАННЯ

Предметом дослідження в статті є методи й інструментальні засоби створення та впровадження систем дистанційного навчання (СДН). Інтерактивні можливості, які використовуються в системі дистанційного навчання, програми і системи доставки інформації, дозволяють налагодити зворотний зв'язок, забезпечити діалог і постійну підтримку, які неможливі в більшості традиційних систем навчання. **Метою** статті є створення колаборативного середовища розробки програмних продуктів в системі дистанційного навчання шляхом формування інтерактивної платформи за допомогою сервісу Sphere Engine, який надає можливості налагоджувати код на багатьох мовах програмування. При цьому вирішуються наступні **завдання**: огляд та порівняння існуючих систем дистанційного навчання; аналіз функціоналу та особливостей застосування платформи Sphere Engine; розробка алгоритму інтеграції сервісу з системою Moodle. **Методи**: методи обробки інформації, методи й інформаційні технології створення електронних курсів дистанційного навчання. **Результати**. За результатами огляду та порівняння існуючих систем дистанційного навчання та засобів для перевірки програмного коду обрано інструменти, які найбільше відповідають потребам створення колаборативного середовища розробки, а саме СДН Moodle та сервіс Sphere Engine. Проведено аналіз функціоналу та особливостей застосування платформи Sphere Engine: приведено схему роботи api compilers сервісу, яка ілюструє процес виконання програми з моменту її відправки до результатів виконання; надано концепцію сценарію пулу у графічній формі. Створено прототип колаборативного середовища розробки, який доступний у вигляді веб-застосунку, описано архітектуру прототипу. Розроблено алгоритм інтеграції сервісу Sphere Engine з системою Moodle. Детально розглянуто етапи алгоритму інтеграції на прикладі тестування роботи середовища розробки курсу "Web-технології та web-дизайн". **Висновки**. Створено інтерактивний модуль за допомогою сервісу Sphere Engine, який дозволяє виконувати лабораторні роботи з дисциплін програмування і здійснювати повний контроль за ходом їх виконання. Таким чином підвищується ступінь автоматизації навчального процесу в закладах вищої освіти.

Ключові слова: дистанційне навчання; колаборативне середовище; алгоритм інтеграції; веб-сервіс.

СОЗДАНИЕ КОЛАБОРАТИВНОЙ СРЕДЫ РАЗРАБОТКИ В СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Предметом исследования в статье являются методы и инструментальные средства создания и внедрения систем дистанционного обучения (СДО). Интерактивные возможности, которые используются в системе дистанционного обучения, программы и системы доставки информации, позволяют наладить обратную связь, обеспечить диалог и постоянную поддержку, которые невозможны в большинстве традиционных систем обучения. **Целью** статьи является создание колаборативной среды разработки программных продуктов в системе дистанционного обучения путем формирования интерактивной платформы с помощью сервиса Sphere Engine, который предоставляет возможности отладки кода на многих языках программирования. При этом решаются следующие **задачи**: обзор и сравнение существующих систем дистанционного обучения; анализ функционала и особенностей применения платформы Sphere Engine; разработка алгоритма интеграции сервиса с системой Moodle. **Методы**: методы обработки информации, методы и информационные технологии создания электронных курсов дистанционного обучения. **Результаты**. По результатам обзора и сравнения существующих систем дистанционного обучения и средств проверки кода избрані инструменты, которые наиболее соответствуют потребностям создания колаборативной среды разработки, а именно СДО Moodle и сервис Sphere Engine. Проведен анализ функционала и особенностей применения платформы Sphere Engine: приведена схема работы api compilers сервиса, иллюстрирующая процесс выполнения программы с момента ее отладки до результата выполнения; придставлена концепция сценария пула в графической форме. Создан прототип колаборативной среды разработки, который доступен в виде веб-приложения, описана архитектуру прототипа. Разработан алгоритм интеграции сервиса Sphere Engine с системой Moodle. Подробно рассмотрены этапы алгоритма интеграции на примере тестирования работы среды разработки для курса "Web-технологии и web-дизайн". **Выводы**. Создан интерактивный модуль с помощью сервиса Sphere Engine, который позволяет выполнять лабораторные работы по дисциплинам программирования и осуществлять полный контроль за ходом их выполнения. Таким образом, повышается степень автоматизации учебного процесса в учреждениях высшего образования.

Ключевые слова: дистанционное обучение; колаборативная среда; алгоритм интеграции; веб-сервис.

Бібліографічні описи / Bibliographic descriptions

Малєєва Ю. А., Косенко В. В., Малєєва О. В., Світличний Д. В. Створення колаборативного середовища розробки в системі дистанційного навчання. *Сучасний стан наукових досліджень та технологій в промисловості*. 2019. № 2 (8). С. 62–71. DOI: https://doi.org/10.30837/2522-9818.2019.8.062.

Malieieva, Ju., Kosenko, V., Malyeyeva, O., Svetlichnyj, D. (2019), "Creation of collaborative development environment in the system of distance learning", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (8), P. 62–71. DOI: https://doi.org/10.30837/2522-9818.2019.8.062.