Virtual Environment for Internet of Things Technologies Studying

Yevhenii Yaremchenko[a], Anzhelika Parkhomenko[a], Artem Tulenkov[a], Andriy Parkhomenko[a], Yaroslav Zalyubovskiy[a], Aleksandr Sokolyanskii[a] and Olga Gladkova[a]

[a] *National University Zaporizhzhia Polytechnic, 64, Zhukovskogo str., Zaporizhzhia, 69063, Ukraine*

Abstract

For the development of Industry 4.0 educational course on Internet of Things we propose to implement the practical part of the course as a a cycle of laboratory works with the usage of modern online engineering technologies, in particular developed online Virtual Learning Environment. The application of this environment will provide students with knowledge and skills necessary for selection of components and protocols for IoT systems realization as well as for organization of users` interaction with IoT applications. The scientific novelty of the work lies in fact that the method of organizing user interaction with the Virtual Learning Environment has been improved through the introduction of remote user interface and remote programming method to provide new ways of obtaining data from the user, increasing the convenience of their processing and visibility of presentation.

Keywords

Internet of Things, online engineering, cloud platform, virtual learning environment, remote user interface, MQTT protocol

1. Introduction

The introduction of the Internet of Things (IoT) technologies in industrial production allows enterprises and companies to reduce operating costs, increase production uptime and, accordingly, increase production volumes and reduce its costs. For example, constant monitoring of production processes can ensure timely maintenance of equipment and improve the level of product quality control.

However, despite the growing popularity, the IoT technologies still have certain drawbacks and unresolved issues that hinder their comprehensive integration into industrial production. They are the problems of introducing standardization, ensuring an appropriate level of security and confidentiality, reducing the cost of the IoT system implementation, combining it with cloud technologies, etc. [1]. Therefore, the further development of the IoT technologies is an urgent task for both scientists and industrialists.

In the context of distance learning, mastering the IoT technologies is especially problematic, due to the complex organization of remote practical experiments with components, protocols, hardware and software platforms used during the implementation of the IoT systems. Therefore, the development and application of online engineering technologies and tools for the IoT systems studying are relevant [2].

CMIS-2022: The Fifth International Workshop on Computer Modeling and Intelligent Systems, Zaporizhzhia, Ukraine, May 12, 2022

EMAIL: yaremchenkoyev@gmail.com (Y. Yaremchenko); parkhomenko.anzhelika@gmail.com (A. Parkhomenko); aetulenkov@gmail.com (A. Tulenkov); andriy2872073@gmail.com (A. Parkhomenko); zalubovskiy.yar@gmail.com (Y. Zalyubovskiy); jorapobeditel@gmail.com (A. Sokolyanskii); gladolechka@gmail.com (O. Gladkova)

ORCID: 0000-0001-9971-5599 (Y. Yaremchenko); 0000-0002-6008-1610 (A. Parkhomenko); 0000-0003-4863-4144 (A. Tulenkov); 0000-0001-8265-0530 (A. Parkhomenko); 0000-0002-6847-8778 (Y. Zalyubovskiy); 0000-0002-3623-9019 (A. 6); 0000-0002-6834-2854 (O. Gladkova)

The goals of the work are research and practical implementation of online virtual environment for the IoT technologies studying.

2. Related works

Online engineering is a generic definition for a wide range of related technologies: remote engineering, virtual instrumentation, online modeling, cloud technologies, etc. [3].

The developed classification of online engineering technologies is presented in Fig. 1. As the studies have shown, these technologies can be successfully used for training in the field of the IoT by sharing engineering equipment and resources of remote laboratories (RL) [4], specialized software for virtual laboratories (VL) and virtual environments (VE) [5]-[6] as well as cloud platforms and services [7].

For example, IoTIFY [8] is a cloud platform designed to support the creation, validation and continuous monitoring of IoT systems. This platform allows to create realistic models of IoT devices based on various components and parameters that can be configured to simulate the interaction of devices with the environment and control this process dynamically through Application Programming Interface (API). IoTIFY supports the most popular data transfer protocols such as MQTT, HTTP, CoAP, AMQP, and LWM2M. With Raw UDP / DTLS and TCP / TLS support, it is also possible to simulate any other protocol [8]. IoTIFY offers two solutions: VL IoTIFY and IoT Simulator. VL IoTIFY is an online IoT learning platform that provides an interactive online workspace for creating electronic prototypes using a web browser. This laboratory allows to develop IoT applications in the browser based on the Arduino Uno, STM32 Nucleo F411RE and Raspberry Pi microcontroller platforms as well as various peripherals [9]. Using VL IoTIFY, students can design prototypes of IoT devices and learn embedded systems programming. VL IoTIFY also enables team collaboration on a project, as all components are available online via a web browser. VL IoTIFY is available under a university license for a minimum of 10 student licenses per year. For individual usage, first 2 credits are available (1 credit - 1 project for 1 hour of use), after that it is necessary to buy additional credits [8].

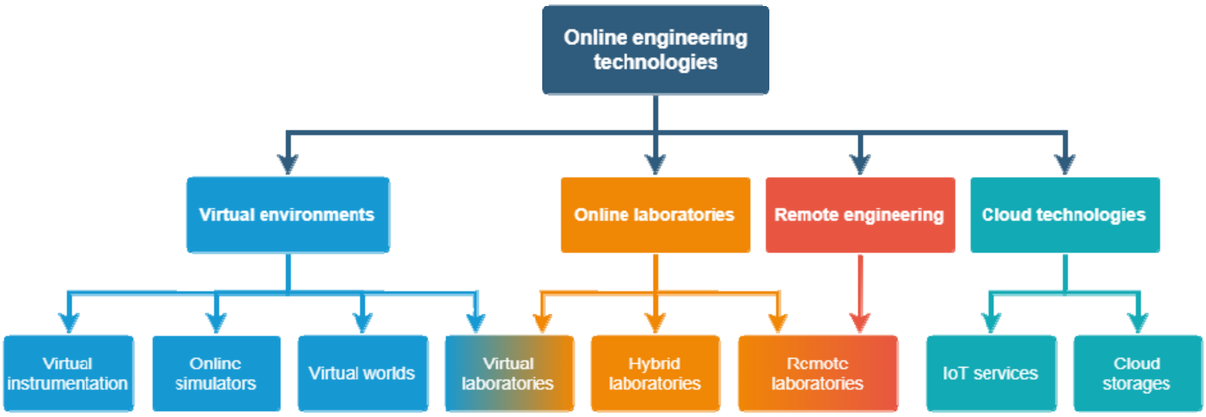


Figure 1: Classification of online engineering technologies

The IoT Simulator from IoTIFY solves the challenge of testing scalability and performance of the IoT system, uses virtualization and simulates thousands of IoT endpoints in the cloud. Thus, it is possible completely simulate the system before the product reaches the customer [10].

Virtual IoT is a VE for collaborative learning in IoT technologies [11]. In order to start using the VE Virtual IoT, it is necessary to register on the project portal as a teacher. After completing the registration process, the teacher can add students to the class. Students need to download the VE to their computers and log in using the mailbox that the teacher has used to invite the student [12].

The training course consists of three levels with three tasks for each of them. The student's goal is to pass the level, getting as many points as possible for completing the tasks. The transition to the next level is possible only after completing all the tasks at the current level. The training material is presented in the form of movies, which can be viewed in the room located before the task room.

Currently, Virtual IoT is in the process of development, so, there are still certain problems with the displaying of training movies, chat and with the virtual surroundings [12].

Wokwi [13] is an online VL for simulating Arduino and other electronics. The goal of the project is to provide a convenient and powerful tool for learning programming based on the Arduino UNO, Mega, Nano and other platforms for creating prototypes of IoT devices. Wokwi allows to simulate the operation of various devices, in particular, LEDs, shift registers, sensors (ultrasonic, temperature, humidity, etc.), SSD display, a keyboard, etc.

An important idea of the project is to enable users to share their ideas with other developers, because the results of the discussions, suggestions and comments may help to improve the projects. In addition, VL Wokwi offers many ready-made examples of the implemented devices based on various hardware platforms and measuring instruments. The development of documentation for this VL is still in its early stages, so there is a lot of missing information. If necessary, it is possible to contact the project server for help via the Discord communication platform [14].

The work [15] describes the usage of 3D virtual world Second Life with elements of social network for the development of VE for teaching in the field of engineering. Second Life is a powerful system for VE development with the ability to render high-quality 3D graphics. It gives the possibility to create collaborative VE, where many users can interact with each other. There is a disadvantage of this system: it can be noted that it is impossible to create an HTML5-compatible program for usage in a regular browser, because a special browser must be used to access the developed virtual worlds.

Thus, as the studies have shown, teaching methods and learning resources based on online engineering technologies can be used successfully for training in the field of the IoT. Each of them has its own functionality, advantages and disadvantages. The choosing of appropriate learning resources is quite complicated and requires consideration of many criteria, so it is important to use recommendation systems, in particular for selection the appropriate RL and VL [16] as well as the IoT platforms [7].

Existing VLs offer experiments focused on specific programming problems for IoT devices based on different hardware platforms. These VLs are mostly presented as web clients, as such systems are fairly easy to implement for usage in a web browser. Existing VEs offer more possibilities to the user based on the free exploration of the virtual world and performing various tasks in it. Most of them are presented as desktop clients, so they require installation on users' computers.

Thus, the task of further development of online learning environment for organizing virtual experiments in the area of the IoT technologies is relevant.

3. Main and anticipated findings

3.1. Method of user interaction with online Virtual Learning Environment

The developed Virtual Learning Environment (VLE) for the IoT technologies studying can be implemented as a non-immersive online system in the form of a web client, i.e. HTML5-compatible program for using in a regular browser. This approach does not require the application of additional control devices, high computing power and installation of software on users' computers.

To organize the interaction of users with online VLE, it is advisable to implement GUI and WIMP paradigm methods: point and click, select and highlight [17] - [19]. This will reduce the number of steps required to take action and obtain information about the VLE.

The selection of items is carried out using the "point and click" method. The selected items are highlighted and icons and widgets with additional information are visualized. In addition, it is necessary to use specific for VLE interaction methods:

- Heads-up display (HUD), on which the actual data, error messages, information widgets and widgets for performing certain actions are visualized. HUD is used in all VLEs, because it is the most user-friendly ways of displaying important information to the user [20] - [21].
- Perspective of observing an agent is top-down. Although the usage of the first-person perspective makes the user more immersed in the VLE, it is more convenient to use the top-down perspective for performing experiments and complex research of VLE. In this case, the user can see what is happening in all or almost all areas of VLE and the agent surveillance camera is behind him [21] - [22].

- Computer mouse to organize agent movement inside VLE. The user is able to indicate the place to which the agent should reach using the "point and click" method. The agent is able to perform only simple movements in order to get to the target, in particular, he cannot jump. Such set of available movements was defined based on investigated works (for example, Virtual IoT). This approach to the organization of agent movement is since to move the game character the user needs to use only a computer mouse without the need to use the keyboard, which simplifies the process of interacting with VLE [23].

However, a set of basic methods for organizing user interaction with the VLE should be improved by introducing a remote user interface and a remote programming method (Fig. 2).

A remote user interface is an interface that allows a user to interact with a remote environment using a variety of communication methods, one of which is the remote programming method.

The remote programming method is a method of user interaction with the system, which consists in using a programming language to send commands to the system based on online technologies. The application of the remote programming method for interaction with the system allows the user to develop external programs that can control the full range of the IoT system functions.

A typical approach to this method implementation is the usage of TCP / IP sockets and any programming language with built-in capabilities or methods for working with them. In the system, that implements this method of interaction, the target device can be remotely controlled and it provides API for external applications or devices with which they can send commands and receive results.

Based on this method, various interfaces for the developed VLE can be supported using one API. For example, interactions between VLE and the user based on ready or user-created hardware or software with sensors and actuators, PCs, web clients, smartphones, virtual instruments, etc. [24].

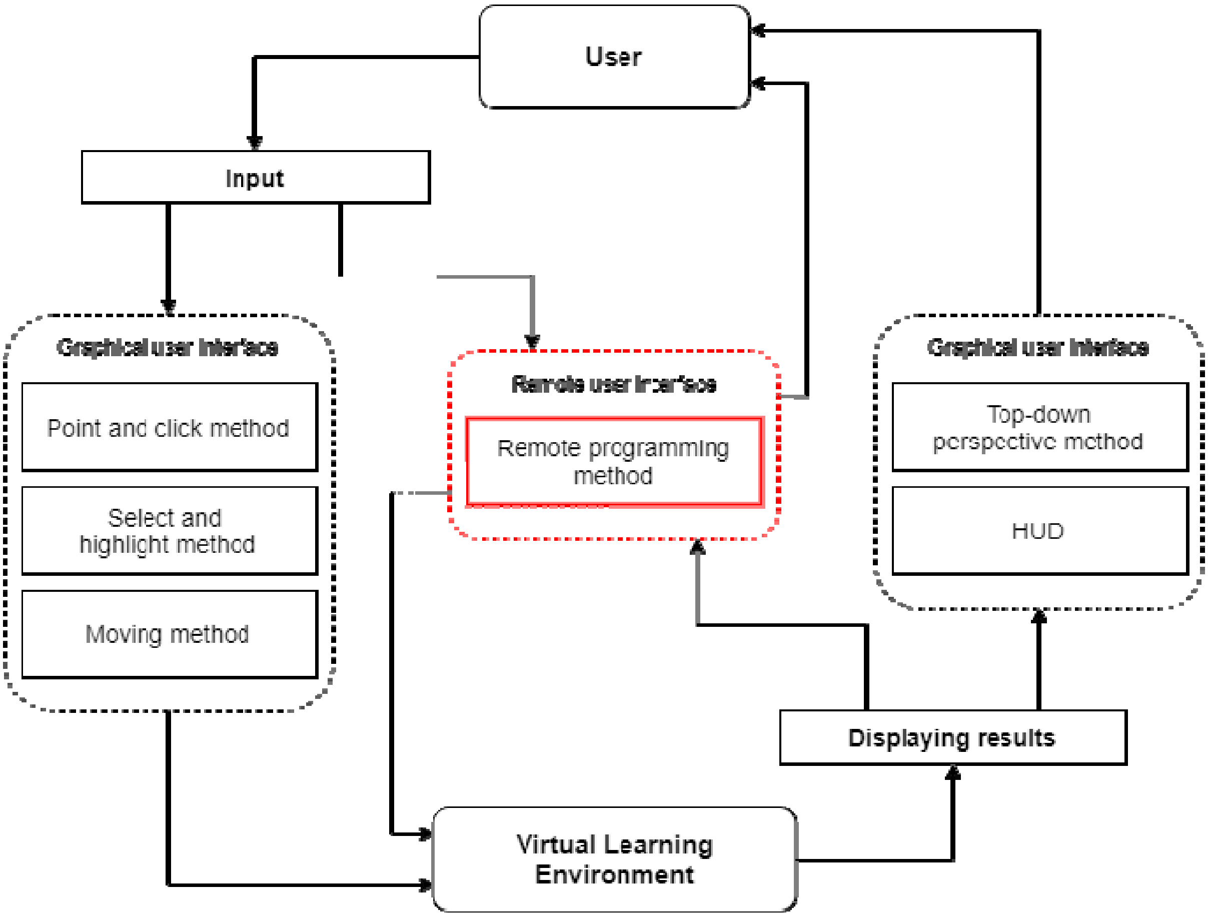


Figure 2: Improved method of organizing user interaction with VLE

3.2. Implementation of online Virtual Learning Environment

The developed architecture of VLE is presented in Fig. 3. To ensure the possibility of remote interaction with the developed system, the data transmission protocols used in the IoT systems were analysed. The investigations [25] - [26] have shown that protocols are an integral part of the technology stack used in such systems.

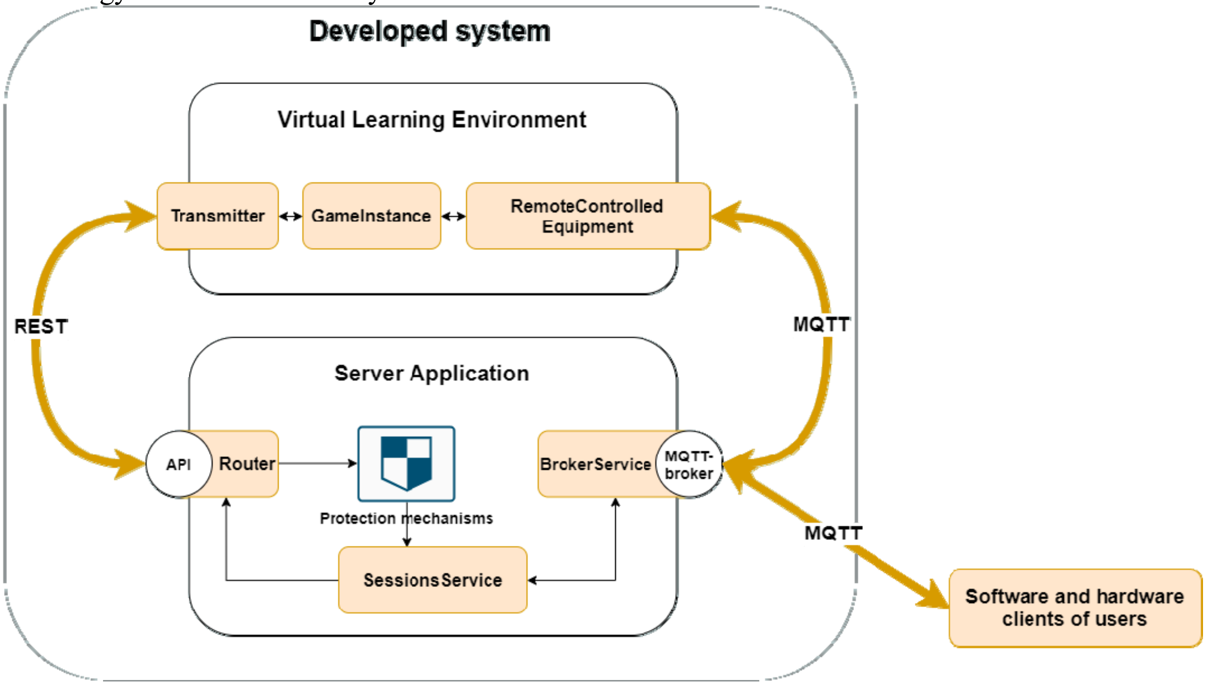


Figure 3: Architecture of developed system

The stack of protocols within the OSI model, which are most often considered in the context of IoT technologies, is the next:

- Application layer protocols (IETF CoAP, IBM MQTT, XMPP and AMQP) are used to connect Smart Things as well as HSC of the end users to the Internet.
- Transport layer protocols (UDP and TCP) are designed to transfer data between HSC.
- Network layer protocols (IPv6, ROLL RPL and 6LoWPAN) provide network routing and encapsulation capabilities.
- BLE, Z-Wave, ZigBee, HomePlug GP and Dash7 are five standard low-power channel layer protocols for short-distance wireless communication within a single LAN.
- At the physical level, the main protocols are IEEE 802.11 (Wi-Fi), IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 (LR-WPAN).

In the context of the development of VLE, which should provide the opportunity to gain practical skills in developing hardware and software for working with IoT systems, it was necessary to analyse application layer data transmission protocols, because users and HSC directly interact with them.

Comparisons of application layer data protocols used in IoT systems are presented in Table 1. The investigations [25] - [26] have shown that the MQTT data transfer protocol can provide reliable and secure routing for small and cheap devices with low power consumption and memory in unstable and low bandwidth networks.

Therefore, the MQTT application-level data transfer protocol can be recommended for organizing remote interaction between VLE and users' HSC, because this popular and reliable protocol is often used in the IoT systems.

The system is built on the basis of a client-server architecture that combines a server and VLE. The developed server class diagram is presented in Fig. 4. The Node.js platform, the Express.js framework, the JavaScript and TypeScript programming languages and the Visual Studio Code IDE were chosen as the system server development toolkit. To implement online VLE the game engine Unreal Engine 4 was chosen, which provides a high level of graphic capabilities, ready-made templates and models, textures and materials for them. It is important that with the help of this game engine it is possible to create applications for web browsers that is HTML5-compatible programs [27].

For the server, the structure of entry points for communication between the server and VLE was developed. The naming rules and the structure of MQTT equipment topics as well as the rules for publishing in them were determined for VLE.

Table 1
Comparison of application layer data protocols used in IoT systems

Criterion \ Protocol	CoAP	MQTT	XMPP	AMQP
Goal	Reduce the overhead of messages, their size and the need for their fragmentation	Ensure reliable communication in networks with limited bandwidth or unstable communication	Provide users with the ability to connect to multiple instant messaging systems	To transfer business messages between HSC
Messaging mode	Synchronous and asynchronous	Asynchronous	Synchronous and asynchronous	Asynchronous
Works in real time	+	+	+	-
Reliability	Built-in mechanism for ensuring the reliability of messaging	Reliability of message delivery through different levels of QoS	TCP guarantees reliability	The intermediate storage function ensures reliability
Security	DTLS	SSL / TLS	SSL / TLS	SSL / TLS
Quality of Service (QoS) option support	+	+	-	+
Architecture	Request / response	Publication / subscription	Publication / subscription, request / response	Publication / subscription
Transport layer protocol	UDP	TCP	TCP	TCP
Energy consumption	Reduces energy consumption	Reduces energy consumption	Reduces energy consumption	Consumption is higher than in MQTT
Network bandwidth	Increases network bandwidth	Optimizes network bandwidth	Significantly reduces network bandwidth	Reduces network bandwidth

Implemented VLE is the non-immersion system in the form of a web client to run in Google Chrome and Mozilla Firefox web browsers. The developed software provides the functions of simulating VLE and its equipment (using the example of home automation system) as well as the possibilities for user interaction with them. This VLE initiates actions on the session by sending requests about this to the server using its REST API. It also displays the parameters for connecting the users' HSC, the message exchange log and other information to the user (Fig. 5). The main areas of HUD are:

- A - Log of events;
- B - Session timer;
- C - Data for connecting users' software and hardware clients;
- D - Log of MQTT messages published by the equipment.

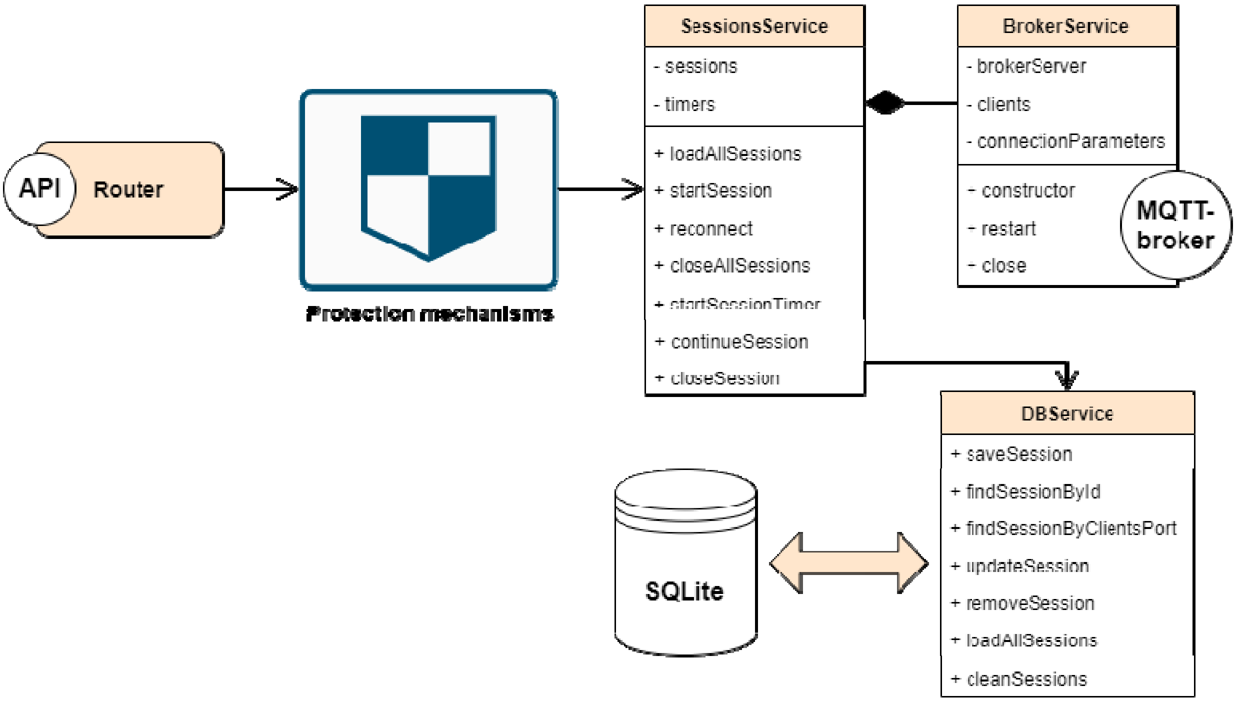
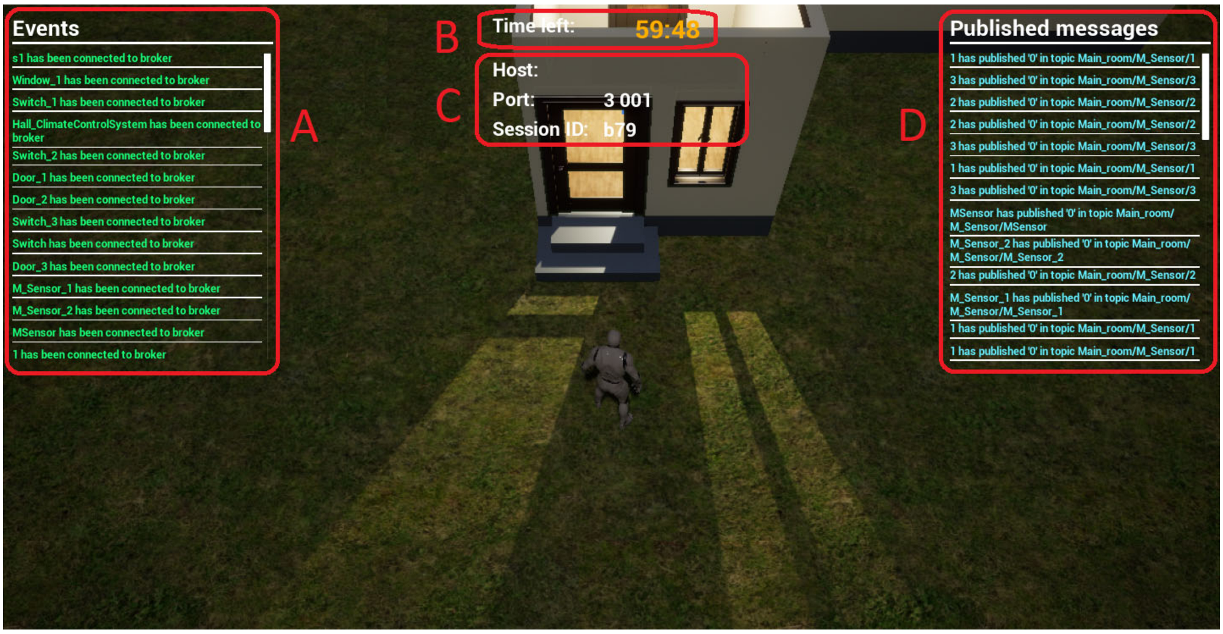


Figure 4: Server class diagram



Events

```
s1 has been connected to broker
Window_1 has been connected to broker
Switch_1 has been connected to broker
Hall_ClimateControlSystem has been connected to broker
Switch_2 has been connected to broker
Door_1 has been connected to broker
Door_2 has been connected to broker
Switch_3 has been connected to broker
Switch has been connected to broker
Door_3 has been connected to broker
M_Sensor_1 has been connected to broker
M_Sensor_2 has been connected to broker
MSensor has been connected to broker
1 has been connected to broker
```

Time left: 59:48

Host:
Port: 3 001
Session ID: b79

Published messages

```
1 has published '0' in topic Main_room/M_Sensor/1
3 has published '0' in topic Main_room/M_Sensor/3
2 has published '0' in topic Main_room/M_Sensor/2
2 has published '0' in topic Main_room/M_Sensor/2
3 has published '0' in topic Main_room/M_Sensor/3
1 has published '0' in topic Main_room/M_Sensor/1
3 has published '0' in topic Main_room/M_Sensor/3
MSensor has published '0' in topic Main_room/M_Sensor/MSensor
M_Sensor_2 has published '0' in topic Main_room/M_Sensor/M_Sensor_2
2 has published '0' in topic Main_room/M_Sensor/2
M_Sensor_1 has published '0' in topic Main_room/M_Sensor/M_Sensor_1
1 has published '0' in topic Main_room/M_Sensor/1
1 has published '0' in topic Main_room/M_Sensor/1
```

Figure 5: HUD of VLE

3.3. User interaction with the developed Virtual Learning Environment

In order to be able to control the events that take place in the home automation system reproduced in VLE as well as to create them, the structure of MQTT-topics was implemented (Table 2).

The server adheres to the following rules for naming and publishing MQTT-topics of equipment:

- MQTT case-sensitive topics;
- MQTT-topics are structured in a hierarchy similar to folders and files in the file system using a slash (/) as a delimiter;
- only UTF-8 is used in topic names;
- $SYS is a reserved topic and is used to publish broker information;
- # (hash symbol) - multilevel substitution symbol;
- the client can publish only in a separate topic. That is, the use of wildcards during publication is prohibited. For example, to publish messages in two topics, the client needs to publish messages separately in each of them.

Table 2
The structure of MQTT-topics of VLE equipment

Topic	Structure
'Hello'-topic	session_id/hello
$SYS-topic	session_id/$SYS
Equipment	session_id/room_id/equipment_type/equipment_id/…
Subscription on all non-$SYS topics	session_id/#
Subscription on all equipment topics in specific room	session_id/room_id/#
Subscription on all topics of specific equipment type in specific room	session_id/room_id/equipment_type/#

UML diagram of the algorithm of user interaction with the developed system is presented in Fig. 6.

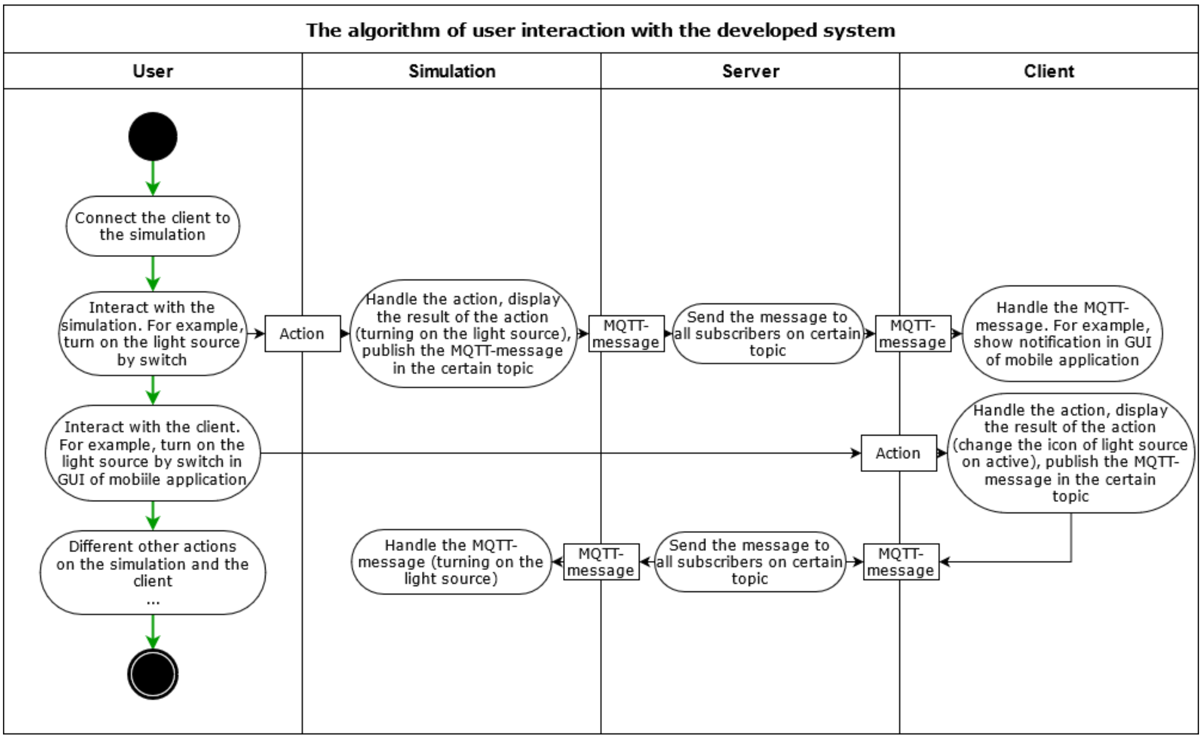

The algorithm of user interaction with the developed system
User
Simulation
Server
Client
Connect the client to the simulation
Interact with the simulation. For example, turn on the light source by switch
Action
Handle the action, display the result of the action (turning on the light source), publish the MQTT-message in the certain topic
MQTT-message
Send the message to all subscribers on certain topic
MQTT-message
Handle the MQTT-message. For example, show notification in GUI of mobile application
Interact with the client. For example, turn on the light source by switch in GUI of mobiile application
Action
Handle the action, display the result of the action (change the icon of light source on active), publish the MQTT-message in the certain topic
MQTT-message
Send the message to all subscribers on certain topic
MQTT-message
Handle the MQTT-message (turning on the light source)
Different other actions on the simulation and the client
...

Figure 6: The algorithm of user interaction with the developed system

After connection of HSC to VLE user can interact with both and the result of such interaction affects on another part. For example, user turns on the light source by switch in VLE. VLE handles this action and, in this example, display the turning on of the desired light source as well as publish the MQTT-message about this change in the certain topic. The server sends the received message to all subscribers on the certain topic. HSC should handle the received message. For example, show notification in GUI of mobile application.

In case of user interaction with HSC, which is developed by himself, HSC should handle the action. For example, user turns on the light source by switch in GUI of mobile application. This application should display the result of the action (in this case, change the icon of the light source on active) as well as publish the MQTT-message about this change in the certain topic. The server sends the received message to all subscribers on the certain topic. VLE handles this action and, in this example, display the turning on of the desired light source.

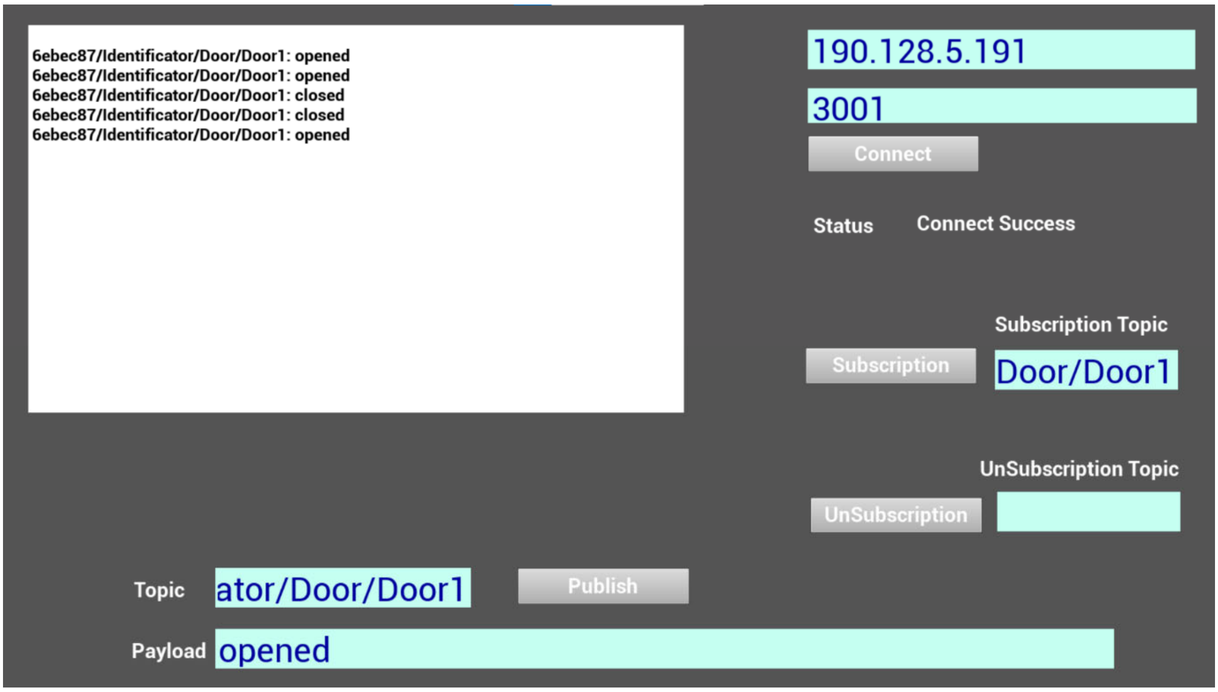
4. Discussion

Analysis of the possibilities of using the product showed that the user can interact with the developed VLE using hardware as well as self-developed or ready-made software applications. To ensure compatibility between clients and the server of the system, students' clients must be able to work with TCP/IP sockets using the MQTT data transfer protocol.

A software client is software that allows the user to perform actions in a VLE by passing commands to it. Due to the active development of smartphones and other mobile devices, the most common type of software clients are mobile applications. They have opportunities for informative visualization of the VLE state as well as for active interaction with it.

A hardware client is a device that provides various physical tools to interact with VLE. Such tools include simple buttons as well as various sensors (lighting, temperature, etc.), based on which the software component of such clients can interact with VLE. For example, a hardware client can implement behaviour, when based on the measurement of the surrounding room temperature in case of exceeding the normal temperature, will transmit a VLE command to turn on the climate control system.

In addition to developing their own applications, the user can use existing solutions, such as a free MQTT client, which is presented in Fig. 7 [28].



```
6ebec87/Identificator/Door/Door1: opened
6ebec87/Identificator/Door/Door1: opened
6ebec87/Identificator/Door/Door1: closed
6ebec87/Identificator/Door/Door1: closed
6ebec87/Identificator/Door/Door1: opened
```

190.128.5.191

3001

Status Connect Success

Subscription Topic: Door/Door1

UnSubscription Topic:

Topic: ator/Door/Door1

Payload: opened

Figure 7: Free MQTT client to work with the developed VLE

The experimental representation of the system data is presented in Table 3 [29]. For today, the number of implemented standard Smart Houses subsystems according to the Table 3 is limited. This is due to the complexity and duration of the development of virtual models of equipment, which often requires the implementation of a mathematical model of physical processes (for example, a mathematical model of room temperature measurement result changing by the influence of the climate control system taking into account room configuration).

5. Conclusion

As a result of the performed research, the features of existing methods and tools of online engineering for studying IoT technologies were identified.

The developed online VLE is intended for teaching and learning the development of software and hardware for interacting with the IoT systems, conducting remote experiments on developing clients for working with the IoT systems as well as creating and testing the operability and efficiency of control scenarios.

Table 3

Experimental representation of the system data

Topic	Payload	VLE action / reaction
f84d592/Room1/Door/Door1 f84d592/Room1/Window/Window1	opened / closed	Open / close the door Door1 / the window Window1 in the room Room1
f84d592/Room1/LightSource/LS1	turn_on / turn_off	Turn on / off the light source LS1 in the room Room1
f84d592/Room1/LightSource/LS1/color	58,49,91	Set the color of the light source LS1 to yellow (HSV color format)
f84d592/Room1/CCS/CCS1	turn_on / turn_off	Turn on / off the climate control system CCS1 in the room Room1
f84d592/Room1/CCS/CCS1/humidity	60	Set the humidity for the climate control system CCS1 to 60%
f84d592/Room1/CCS/CCS1/temperature	20	Set the temperature for the climate control system CCS1 to 20°C
f84d592/Room1/L_Sensor/LSens f84d592/Room1/M_Sensor/MSens f84d592/Room1/T_Sensor/TSens f84d592/Room1/H_Sensor/HSens	value	Send to the broker information about measurement results of the specific sensor

The practical value of the proposed solutions is determined by the fact that implemented VLE which is based on improved method, provides a tool for developing practical skills for programmable interaction with the IoT systems using hardware and software clients (HSC). User develops these clients independently and he can connect them to VLE for its control via API. Such system offers a high level of interactivity, because users` actions in VLE affect the state of the HSC developed by him and vice versa.

In the future, it is planned to increase the number of available components of VLE, in particular for industrial IoT systems as well as to implement new possibilities for user interaction with VLE.

6. Acknowledgements

This work is partly carried out with the support of Erasmus + KA2 project WORK4CE “Cross-domain competences for healthy and safe work in the 21st century” (619034-EPP-1-2020-1-UA-EPPKA2-CBHE-JP).

7. References

[1] What are the main advantages and disadvantages of the Internet of Things? URL: https://www.imd.org/iot/iot-refections/pros-and-cons-of-iot.

[2] D. Pirrone, C. Fornaro, D. Assante, Open-source multi-purpose remote laboratory for IoT education, in: Proceedings of the 2021 IEEE Global engineering education conference, EDUCON, IEEE, Vienna, Austria, 2021, pp. 1462-1468. doi:10.1109/EDUCON46332.2021.9454034.

[3] International journal of online and biomedical engineering. About the journal. URL: https://online-journals.org/index.php/i-joe/about.

[4] A. Parkhomenko, A. Tulenkov, A. Sokolyanskii, Y. Zalyubovskiy, A. Parkhomenko, Integrated complex for IoT technologies study, in: M. Auer, D. Zutin (Eds.), Online engineering & Internet of Things, volume 22(31) of Lecture notes in networks and systems, Springer, Cham, 2017, pp. 322–330. doi:10.1007/978-3-319-64352-6_31.

[5] B. S. Wästberg, T. Eriksson, G. Karlsson, M. Sunnerstam, M. Axelsson, M. Billger, Design considerations for virtual laboratories: A comparative study of two virtual laboratories for learning about gas solubility and colour appearance, education and information technologies 24 (2019) 2059–2080. doi:10.1007/s10639-018-09857-0.

[6] Y. Francillette, E. Boucher, A. Bouzouane, S. Gaboury, The virtual environment for rapid prototyping of the intelligent environment, Sensors (Basel) 17.11 (2017). doi:10.3390/s17112562.

[7] A. Tulenkov, A. Parkhomenko, A. Sokolyanskii, Evaluation and selection of IoT service for Smart House system big data processing, in: Proceedings of the IEEE 14th International scientific and technical conference on Computer sciences and information technologies, CSIT, IEEE, Lviv, Ukraine, 2019, pp. 124-129. doi:10.1109/STC-CSIT.2019.8929810.

[8] IoTIFY. URL: https://iotify.io/.

[9] Virtual lab. URL: https://docs.vlab.iotify.io/.

[10] IoTIFY. IoT simulator. URL: https://iotify.io/iot-network-simulator/.

[11] Virtual IOT. URL: http://www.virtualiot.net/.

[12] Virtual IOT. How to use. URL: http://www.virtualiot.net/how-to-use/.

[13] Wokwi. URL: https://wokwi.com/.

[14] Wokwi docs. URL: https://docs.wokwi.com/.

[15] M. J. Callaghan, K. McCusker, J. Lopez Losada, J.G. Harkin, S. Wilson, Engineering education island: teaching engineering in virtual worlds, Innovation in teaching and learning in Information and computer sciences 8.3 (2009) 2–18. doi:10.11120/ital.2009.08030002.

[16] A. Parkhomenko, O. Gladkova, A. Parkhomenko, Recommendation system as a user-oriented service for the remote and virtual labs selecting, in: M. Auer, T. Tsiatsos (Eds.), The challenges of the digital transformation in education, volume 917 of Advances in intelligent systems and computing, Springer, Cham, 2019, pp. 600-610. doi:10.1007/978-3-030-11935-5_57.

[17] K. Hinckley, D. Wigdor, Input Technologies and Techniques, in: J. A. Jacko (Ed.), The human-computer interaction handbook, 3rd. ed., CRC Press, Boca Raton, FL, 2012, pp. 95–132. doi:10.1201/b10368-12.

[18] K. M. Inkpen, Drag-and-drop versus point-and-click mouse interaction styles for children, ACM Transactions on Computer-Human Interaction 8.1 (2001) 1–33. doi:10.1145/371127.371146

[19] Selection. URL: https://material.io/design/interaction/selection.html.

[20] L. Caroux, K. Isbister, Influence of head-up displays' characteristics on user experience in video games, International journal of human-computer studies 87 (2016) 65–79. doi: 10.1016/j.ijhcs.2015.11.001.

[21] L. Caroux, K. Isbister, L. Le Bigot, N. Vibert, Player–video game interaction: A systematic review of current concepts, Computers in human behavior 48 (2015) 366–381. doi: 10.1016/j.chb.2015.01.066.

[22] J. Garcia, Perspectives and points of view, 2020. URL: https://www.ign.com/wikis/video-game-dictionary/Perspectives_and_Points_of_View.

[23] C. Compton, Run, jump and climb: designing fun movement in games, 2019. URL: https://www.gamedeveloper.com/audio/run-jump-and-climb-designing-fun-movement-in-games.

[24] R. M. Prades, P. J. Sanz, P. Nebot, R. Wirz, A multimodal interface to control a robot arm via the web: a case study on remote programming, IEEE Transactions on industrial electronics 52.6 (2006) 1506–1520. doi:10.1109/TIE.2005.858733.

[25] C. Sharma, N. K. Gondhi, Communication protocol stack for constrained IoT systems, in: Proceedings of the 3rd. International conference on Internet of Things: smart innovation and usages, IoT-SIU, IEEE, Bhimtal, India, 2018, pp. 1–6. doi:10.1109/IoT-SIU.2018.8519904.

[26] T. M. Tukade, R. Banakar, Data transfer protocols in IoT an overview, International journal of pure and applied mathematics 118 (2018) 121–138.

[27] P. Gestwicki, Unreal Engine 4 for Computer Scientists, Journal of computing sciences in colleges 35.5 (2019) 109–110.

[28] BP_Mqtt. URL: https://github.com/damody/BP_MQTT_Demo/releases/download/1.6/BP_Mqtt.zip.

[29] Y. Yaremchenko, J. Nau, D. Streitferdt, K. Henke, A. Parkhomenko, Virtual environment Smart House for hybrid laboratory GOLD, in: M. E. Auer, H. Hortsch, O. Michler, T. Köhler (Eds.), Mobility for Smart cities and regional development - challenges for higher education, volume 389 of Lecture Notes in Networks and Systems, Springer, Cham, 2022, pp. 250–257. https://doi.org/10.1007/978-3-030-93904-5_25