

UDC 004.588

DOI: 10.15827/2311-6749.20.2.1

Development of a multi-user software simulator based on web-technologies

V.S. Ivanov ¹, student, mrcitron@mail.ru

S.A. Belyaev ¹, Ph.D. (Engineering), Associate Professor, bserge@bk.ru

¹ Saint Petersburg Electrotechnical University "LETI", 197376, Saint Petersburg, Russian Federation

The paper describes the relevance of using software simulators in the operator training process. There is a problem of developing a multi-user software simulator based on web technologies in the paper. The set of requirements proposed for the developed solution, which are necessary for learning how to perform technological process operations and developing appropriate action sequences both in the instructor presence and without supervision.

The paper analyzes the available solutions and proposes the developed system architecture, presents its ER-diagram with key description points that affect its operation.

There is a simulator used by a teacher and a learner. There are differences between the training modes, including a mode, designed to learn emergency handling. To show the system capabilities, the authors implemented a few exemplary control panels for training in performing process operations.

The proposed solution has been tested as the «Gazprom Classes» project part. The main advantages of the proposed solution are the development simplicity and the capability to design the control panes with a visual interface.

Keywords: software simulator, technological process operator, scenario, emergency situation, web-technology, teamwork.

Nowadays, the *technological process* (TP) operator's work is gradually becoming harder and requires solving more and more difficult management problems [1]. Working with specialized equipment in addition to theoretical knowledge often requires practical skills. Training at real facilities can be difficult to implement, for example, due to the limited resource of complex technical systems. The solution to this problem is the computer simulator use, which is being improved along with the information technology growth [2]. Hardware-software simulators allow you to develop the student's motor skills, exclusively software versions provide only the study and the action sequence development. At the same time, the more realistic the controls look, the easier it will be for a student to work with real systems and assemblies. An important advantage of software implementations is the ability to develop algorithms for recovering from emergency situations.

Consider the problem of developing a multi-user software simulator based on web-technologies, providing training in the implementation of TP operations, there are several operators in which. Primary requirements:

- interactive remote display to the operator;
- setting and monitoring the correct sequence of TP operations;
- role support: operator (student), supervisor (teacher) and editor;
- support for TP operation modes: training, testing, and self-testing;
- access to the training supervisor for operator test results;
- support for operator group work;
- support for emergency situations (branches of the correct sequence of operator actions associated with any third-party events);
- provide the operator with a text description of the actions that he must perform;
- availability of a web-based interface.

Existing solution

Consider the possibility of solving the problem using ready-made solutions:

- simulator Emerson DeltaV [3];
- computer simulators SIKE [4];
- simulator STDinamika of Systemotechnika 3D [STDynamics of Systematic Engineering 3D];
- computer simulators of LLC «Infrastructura TK» [Infrastructure TK LLC];
- Unisim training complex [5];
- a simulator for operators of technological installations of RPC Krug [RPC Circle] [6].

The Emerson DeltaV simulator is a specialized training environment that supports role sharing, student collaboration, emergency situations, and operator testing mode. However, there is no web version and textual description of the actions to be performed.

Computer simulators SIKE is simulators for personal computers in which the operator testing mode is available and there is a manual in text format, but it does not meet other requirements.

The simulator STDinamika of Systemotechnika 3D [STDynamics of Systematic Engineering 3D] is a software and computer complex that has the ability to edit, the ability to use emergency situations, the ability to go through testing mode and the text description presence of the action sequence.

Computer simulators of LLC “Infrastructura TK” [Infrastructure TK LLC] are a hardware-software complex for implementing the training environment for personnel of technological installations, which has instructions for operators, the ability to use the web version, the ability to implement emergency situations and the ability to control the learning process, but not suitable for students working together and don’t have a built-in editor.

Unisim training complex is a comprehensive training system for TP operators, which allows roles separation, the contingency use and has an editor, but does not satisfy the rest of the requirements.

A simulator for operators of technological installations of RPC Krug [RPC Circle] is a simulator that includes: role allocation, students’ team working, contingencies, editor and testing mode, but does not have a text description of the actions that the operator must perform, there is no web version.

None of the listed solutions fully meets the specified requirements. The most suitable simulators were Emerson and RPC Krug [RPC Circle], which have the necessary functionality for training personnel in specialized rooms with the teacher supervision. However, they are not suitable for self-training and operator training, as they do not have accompanying training material during work; there is also no web version. The remaining requirements are met by the simulator company “Infrastructure TK LLC”, however, this solution is on self-study without role allocation and the possibility of students working together.

The Simulator development

To solve this problem and fulfill all the requirements, there was a software simulator based on scenarios, its architecture is in Figure 1. In it, unlike solutions based on mathematical models, the correct sequence of actions is set by the scenario, deviations are unacceptable [7].

Roles in the system are with the authorization module using. For each role (operator, teacher, and editor), a set of features is the corresponding interface using.

There is the scenario display component to solve the problems of describing operator jobs and running the simulator scenario. Using the demonstration module, an object is sent from the server containing the resources (images and audio files) used in the simulator, and all components: the scenario, control panels and elements for the panels. The ER diagram of the implemented solution is in Figure 2.

To organize group lessons, the teacher uses the component of group lessons organization. At the same time, the action synchronization between trainers in the framework of a group lesson occurs using the synchronization module. After completing the step, one of the operators sends a message with the number of the completed step to the rest of the TP participants using web-sockets. At the moment of receiving a message, there is the automatic step completion.

The teacher can view the results of operators and change grades using the learning results control module.

There is an editing component to create and edit simulators. This module works in conjunction with a resource loading module for adding images and audio files, and with a module for working with JSON files to check the compliance of the simulator modifiable parts (scenario, panels, elements) with JSON schemes. The JSON-scheme structure is based on the developed solution structure, presented in Figure 2.

Optional fields have a mark with a «?» after the name.

The simulator is set in JSON format and consists of:

- JSON with elements (a elements’ collection, in the figure – orange with the index 3, the root element – Elements);
- JSONs with remotes (in the figure – green with index 2, the root element is Panel);
- JSON with scenarios (in the figure – red with index 1, the root element – Course).

Each simulator stage consists of a step set (step) and is considered passed when all steps are completed. An abnormal situation is a stage in which a branch from the startStep step of a stage called stageName occurs. An abnormal situation may return to the normal stream if the endStep attribute is specified.

Each step has a number assigned to it, which indicates the order in which the step should be performed within the stage to which it belongs. The step corresponds to a certain operator role (role), the active zone (active-ZoneId) of the element (itemId) on the panel (panelId) and the action (action), which includes the necessary checks, after passing through which the logic specified by the editor in the code of this action is executed. The action will have activation if all the steps specified in the attribute (previous) have an execution. Because the simulator allows several roles within one stage, and one operator can have only one role at a time, then for each

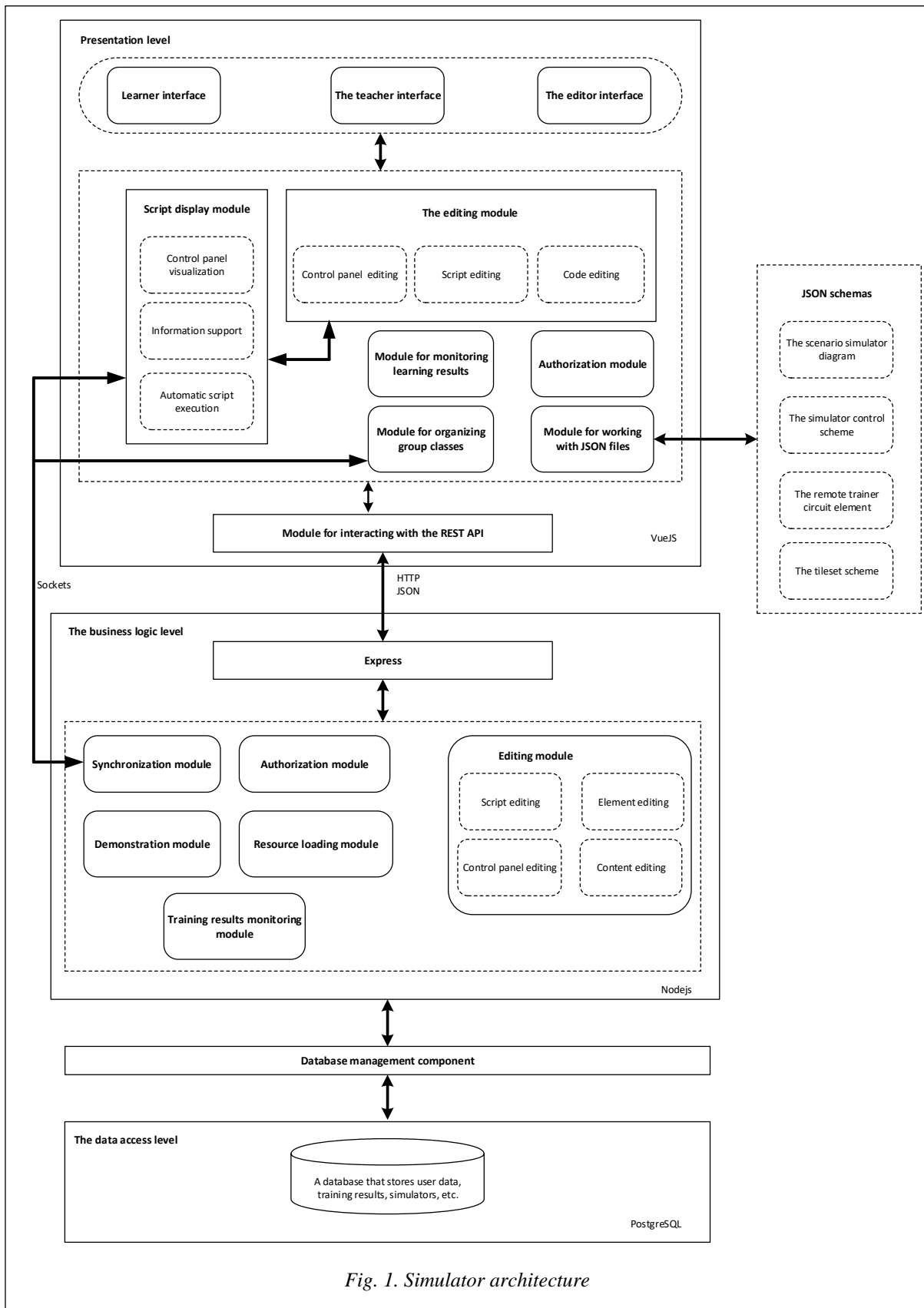
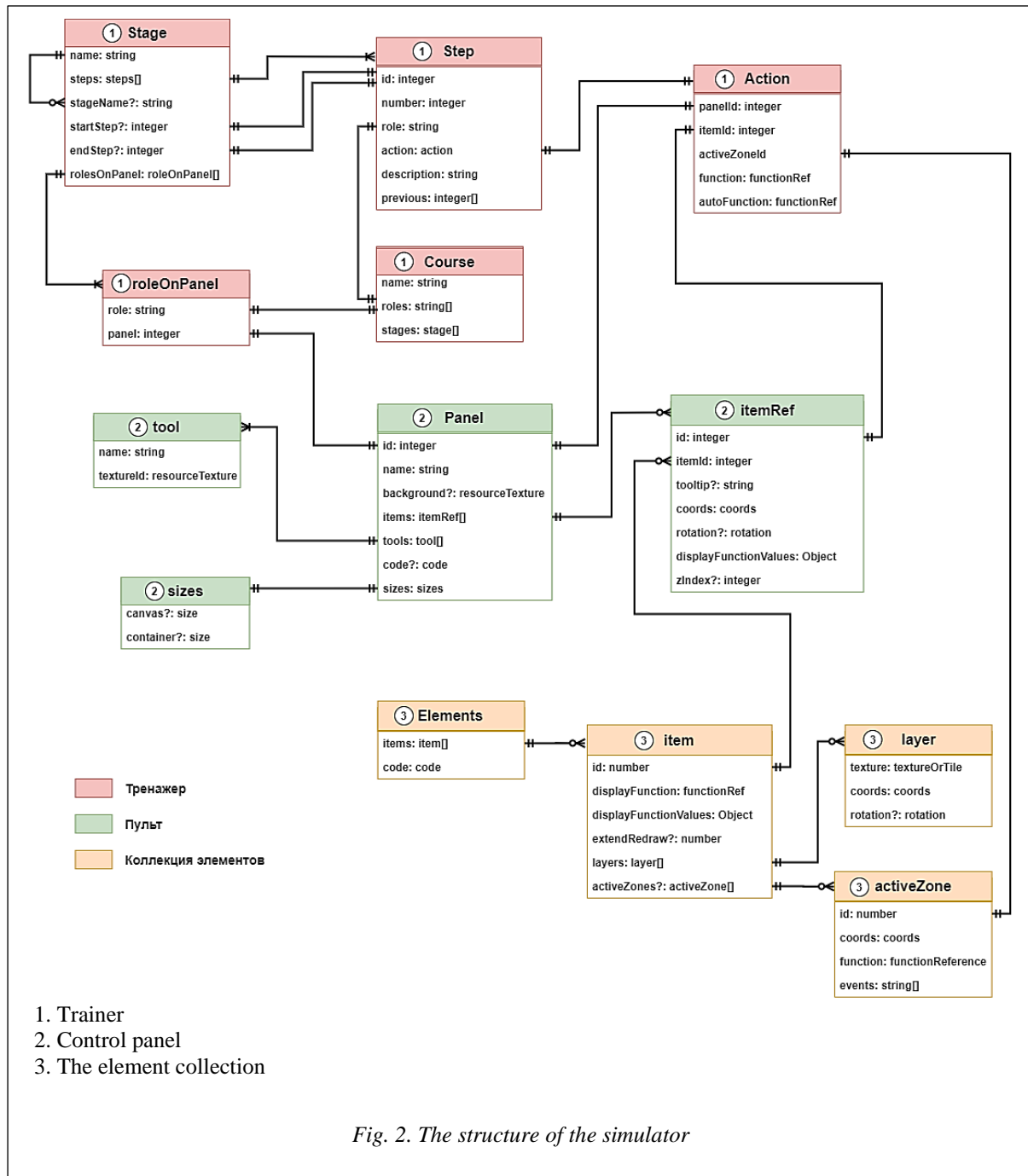


Fig. 1. Simulator architecture

function (function), which receives the action parameters and checks whether it is executed correctly, the auto-Function function should be implemented, which will automatically lead the console to a state that satisfies the checks in function. In Figure 3 presents an activity diagram demonstrating the operation of this mechanism. Initial checks for performing the necessary actions and for an element that had an interaction with the user, has

an association with the number, previous, and itemId and panelId fields, respectively. Then there is a condition check in function and the role field of the next step. If the operator does not have an assignment for the received role in the current lesson, then there is an autoFunction for the next step, which guarantees successful completion of the function checks for the next step and allows you to complete training without assigning students to some of the stage roles.



To reuse items, they have a division into items in the collection (item) and on the remote (itemRef). The function *displayFunction* displays an item in the collection on canvas [8] using the parameters specified in *displayFunctionValues*. The list of textures used in displaying the element is in *layers*. When interacting with the active zone (activeZone) of an element, there is an *activeZone.function*, which allows you to update the *displayFunctionValues* of the element and thereby change its appearance. The item on the remote contains a pointer to the item in the collection corresponding to it, as well as its coordinates on the remote. At the same time, the *displayFunctionValues* in the element on the remote control overrides the values of the *display* parameters in the collection, and the element in the collection does not change when the scenario is executed, all changes are in the element on the remote.

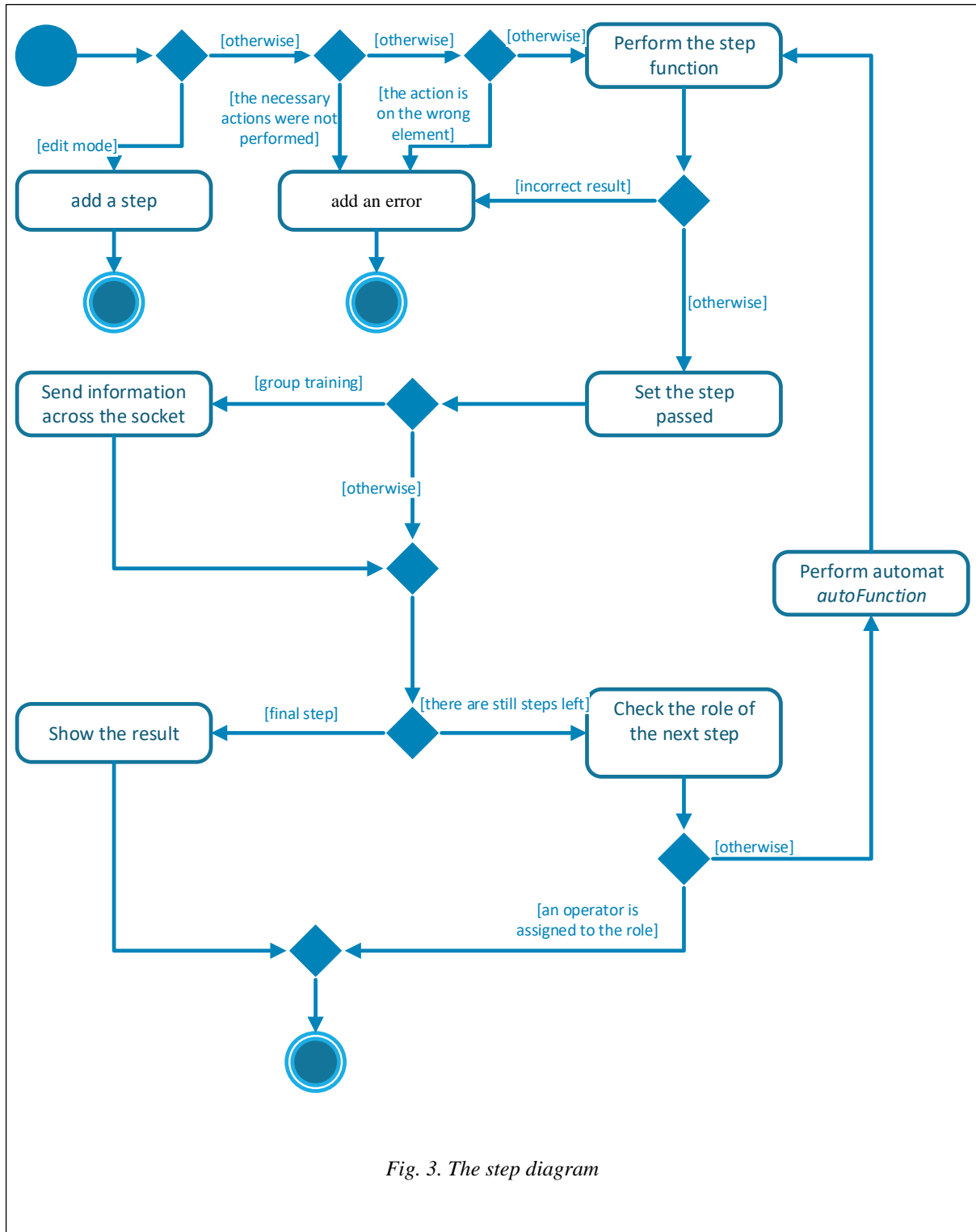


Fig. 3. The step diagram

Scenarios, control panels and remote-control elements are in a database in jsonb format [9]. This solution allows you to abandon the data integrity restriction in query performance favor and the stored structure simplification.

Using the Simulator

In the proposed solution, the operator can use the scenario execution modes:

- individual training;
- individual self-test;

- group training;
- group testing.

The training mode differs from testing in the ability to view the instruction, which contains a textual description of all steps for each role, and at the same time, there is a selection of already completed steps.

In the individual mode, the student himself chooses the stage and role for the scenario, the remaining role actions of the selected stage are automatic. In the training mode in the group, the teacher assigns students to roles and selects the stages for passing. Operators need to follow the action sequence for each role and wait for the action completion of their colleagues in the team. If a team does not have any roles, the role actions will perform automatically.

The student can use different tools to perform actions. In those cases when the operator performs an action not according to the scenario (the wrong element, the wrong element state, the wrong tool, too early), then he receives a text notification about the error and the error counter increases. Upon the work completion, the student receives an assessment, depending on the number of mistakes.

The teacher and editor are available to individually execute scenarios, as well as the operator.

The teacher has the opportunity to organize group classes. He selects the steps that the operators will perform, distributes them into groups and roles within each group. The group training uniqueness lies in the real-time collaboration possibility and joint algorithm development for recovering from emergency situations. Abnormal situations are unavailable for individual training and are hidden from students. This allows you to check not only the learned material but also the analysis skills of emergency situations using various approaches, for example, fuzzy logic [10].

For the editor implemented interfaces:

- resource loading;
- changes in operator roles;
- adding and changing control panels;
- control panel editing elements;
- the simulator scenario formation.

Adding and removing stages that have an attachment bundle console and role form the simulator scenario. After adding a stage and assigning roles to the panels, the editor, either using the provided visualization (the mode of execution of a script by recording step-by-step), sets the step sequence and then refines them in the provided scenario editor, including setting preconditions, postconditions, delays, etc. If the changed simulator does not correspond to JSON-schemes or if there are other errors (for example, there is a nonexistent resource), the editor sees the error counter display with a detailed description. The editor has the opportunity to perform the step sequence of the stage himself using the scenario display module.

The developed simulator tested like in [11] and integrated into the «Gazprom Classes» system. The work demonstration on the control panels is in Figures 4 and 5.

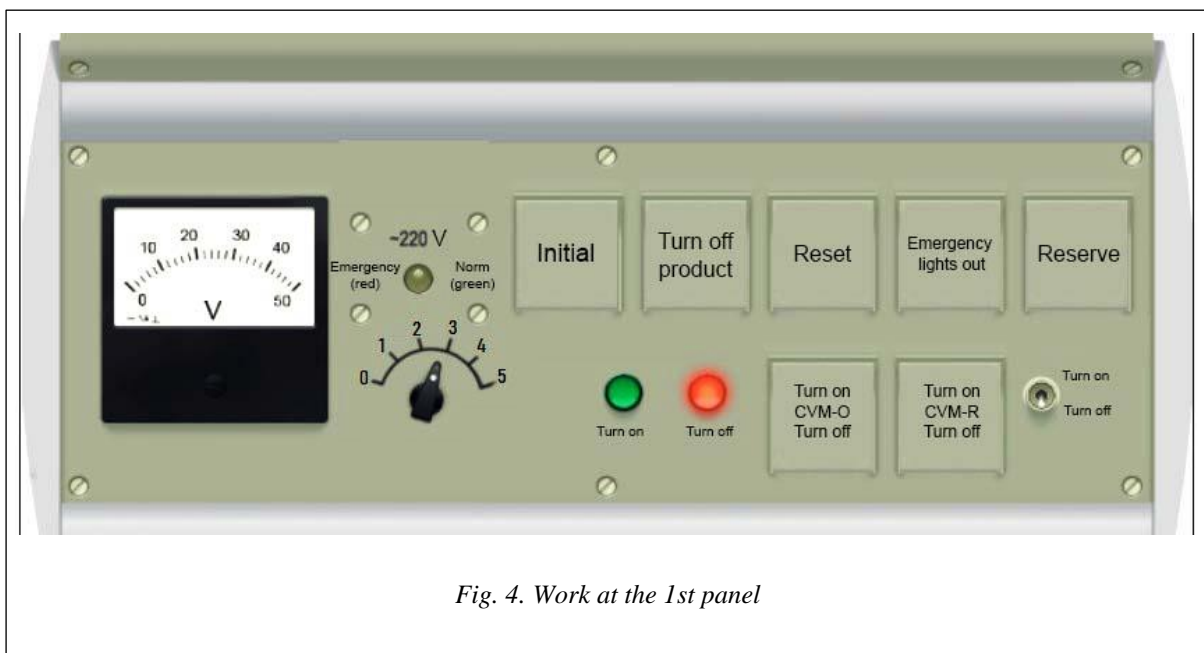


Fig. 4. Work at the 1st panel

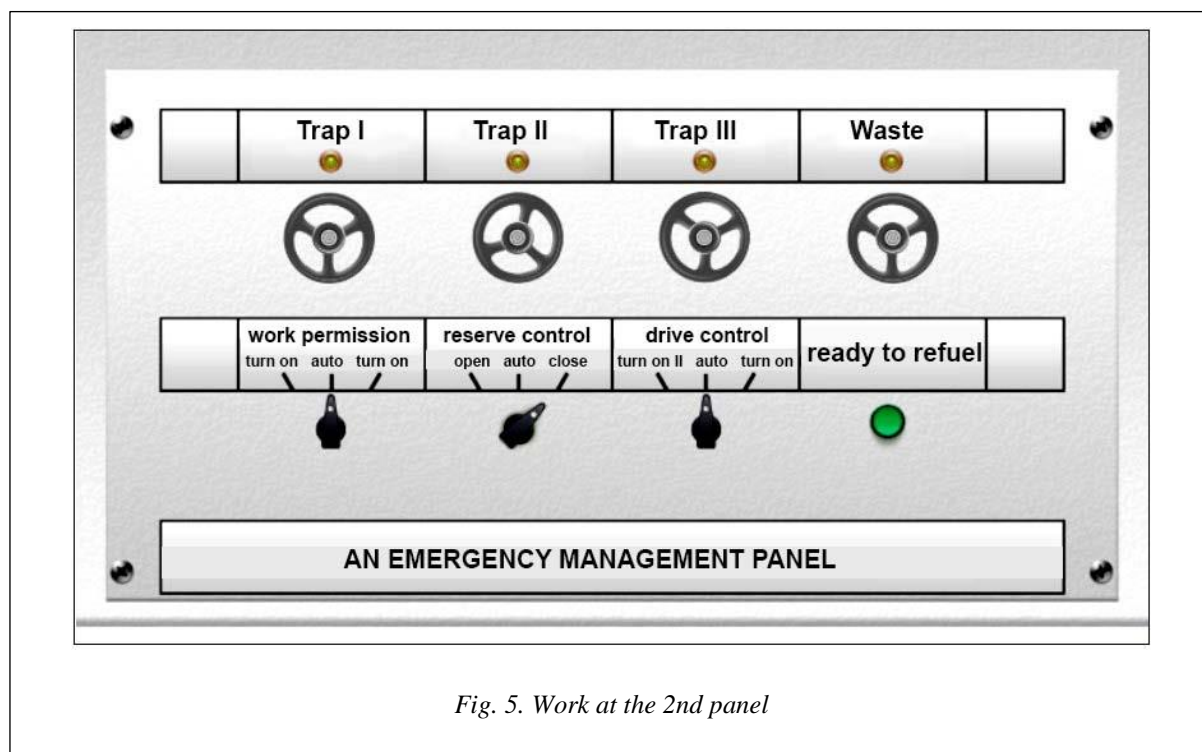


Fig. 5. Work at the 2nd panel

Conclusion

An important advantage of the developed solution is the relative simplicity of creating scenarios because it does not require the mathematical model construction and have a restriction by the minimal knowledge presence of the JavaScript language. However, behind the creation simplicity lies the main limitation – the need to strictly follow the scenario in the learning process, which makes it not universal from the full study viewpoint of TP [7], since there is no way to study the reaction to actions outside the scenario. The developed software simulator complies with all the requirements for such solutions and can be used to train operators of technical systems and complexes when performing operations on a technological schedule. The system can be applied in various fields of application since the development of execution sequence in various fields of activity differs only in the equipment over which actions are performed and editing of which is in the system.

The proposed solution has been tested as part of the Gazprom Classes project and is for training future TP operators. Further development of the developed system is possible with the mathematical model partial use, which will expand the simulator scope and will not lead to a significant complication of the scenario development process.

Referenses

1. Dozortsev V.M. New challenges to industrial safety – will computer simulators help? *Occupational Safety in Industry*, 2019, no. 9, pp. 31–38 (in Russ.)
2. Nagimov T.R., Nikulina N.O. Reengineering the process of training enterprise employees using simulators. *Proc. Conf. Technological Cooperation in Science and Production: New Ideas and Development Prospects*, 2018, pp. 77–85 (in Russ.)
3. Kreidlin E.Yu. DeltaV control system - from process control to operator training. *Automation in the Industry*, 2016, no. 3, pp. 13–16 (in Russ.)
4. Chudinova Yu.A., Tikhonov A. V. Trainer-simulator «SIKE. Metal rolling in a reversing rougher heavy bar mill stand». *Pass Design Bureau*, 2017, no. 11–17 (in Russ.)
5. Dozortsev V.M. *High-Precision Modeling of TP on Real Examples of Simulator Projects*. 2017. Available at: https://www.honeywellprocess.com/en-US/online_campaigns/russian-technology-summit/Pages/document/2017/day2/10.%20Высокоточные%20тренажерные%20модели.pdf (accessed February 14, 2020) (in Russ.)
6. Proshin A.I., Shekhtman M.B. CPC “Trail” - a universal tool for the development of simulators for operators of technological plants. *Informatization and Management Systems in the Industry*, 2019, no. 2, pp. 109–111 (in Russ.)

7. Belyaev S.A. Software simulators based on mathematical models of complex technical devices. *Modern Education: Content, Technologies, Quality. Proc. XXV Conf.*, 2019, pp. 288–290 (in Russ.)
8. Kochitov M.Ye. Consider the possibility of web drawing on HTML5 using the canvas. *Postulate*, 2018. Available at: <http://e-postulat.ru/index.php/Postulat/article/view/1752> (accessed February 14, 2020) (in Russ.).
9. Petković D. JSON integration in relational database systems. *I. J. Comp. Appl.*, 2017, vol. 168, no. 5, pp. 14–19.
10. Ekalo A.V., Kudryakov S.A., Shapovalov E.N., Ostapchenko Yu.B., Belyaev S.A. Algorithm for making informed decisions in emergency situations based on models of fuzzy sets. *Bull. ETU "LETI"*, 2016, iss. 9, pp. 16–21 (in Russ.).
11. Chernaya O.S., Fedorova Yu.Yu., Belyaev S.A. Applying of methods and tools to automated testing telemetry data processing software. *Bull. ETU "LETI"*, 2013, iss. 9, pp. 55–58 (in Russ.).

УДК 004.588

DOI: 10.15827/2311-6749.20.2.1

Разработка многопользовательского программного тренажера с использованием web-технологий

В.С. Иванов¹, студент, mrcitron@mail.ru

С.А. Беляев¹, к.т.н., доцент, bserge@bk.ru

¹ Санкт-Петербургский государственный электротехнический университет «ЛЭТИ», г. С-Петербург, 197376, Россия

В статье обоснована актуальность использования программных тренажеров в обучении операторов технологических процессов. Рассмотрена задача разработки многопользовательского программного тренажера на основе web-технологий. Для этого был сформирован набор требований к разрабатываемому решению, которые необходимы для обучения выполнению операций технологического процесса и для отработки соответствующих последовательностей действий как в присутствии инструктора, так и в его отсутствие.

В статье выполнен анализ готовых решений и приведена архитектура разработанной системы, представлена ее ER-диаграмма с описанием ключевых моментов, влияющих на ее работу.

Описано применение тренажера преподавателем и обучаемым, показаны особенности разных режимов, в частности, режима для обучения алгоритмам выхода из нестандартных ситуаций. Представлены примеры пультов управления, которые реализованы в данной системе с целью использования для обучения выполнению операций технологического процесса.

Предложенное решение апробировано в рамках проекта «Газпром-классы». Основные преимущества предложенного решения – простота описания сценариев и возможность визуального формирования пультов, которыми пользуются операторы при выполнении операций технологических процессов.

Ключевые слова: программный тренажер, оператор технологического процесса, сценарий, нестандартная ситуация, web-технологии, совместная работа.

Литература

1. Дозорцев В.М. Новые вызовы промышленной безопасности - помогут ли компьютерные тренажеры? // Безопасность труда в промышленности. 2019. № 9. С. 31–38.
2. Нагимов Т.Р., Никулина Н. О. Реинжиниринг процесса обучения сотрудников предприятия с помощью тренажеров // Технологическая кооперация науки и производства: новые идеи и перспективы развития: сб. тр. конф. 2018. С. 77–85.
3. Крейдлин Е.Ю. Система управления DeltaV – от управления процессами к обучению операторов // Автоматизация в промышленности. 2016. № 3. С. 13–16.
4. Чудинова Ю.А., Тихонов А.В. Тренажер «SIKE. Прокатка металла в реверсивной клетки крупно-сортного стана» // Калибровочное бюро. 2017. №. 11-17.

5. Дозорцев В.М. Высокоточное моделирование ТП на реальных примерах тренажерных проектов // Передовые решения Honeywell для цифровой экономики. 2017. URL: https://www.honeywellprocess.com/en-US/online_campaigns/russian-technology-summit/Pages/document/2017/day2/10.%20Высокоточные%20тренажерные%20модели.pdf (дата обращения 14.02.2020).
6. Прошин А.И., Шехтман М.Б. КТК «Тропа» - универсальное средство разработки тренажеров для операторов технологических установок // Информатизация и системы управления в промышленности. 2019. № 2. С. 109–111.
7. Беляев С.А. Программные тренажеры на основе математических моделей сложных технических устройств // Современное образование: содержание, технологии, качество: матер. XXV Междунар. науч.-методич. конф. 2019. С.288–290.
8. Кочитов М.Е. Рассмотрение возможности веб-рисования на HTML5 с помощью canvas // Постулат. 2018. URL: <http://e-postulat.ru/index.php/Postulat/article/view/1752> (дата обращения 14.02.2020).
9. Petković D. JSON integration in relational database systems. I. J. Comp. Appl., 2017, no. 5, pp. 14–19.
10. Экало А.В., Кудряков С.А., Шаповалов Е.Н., Остапченко Ю.Б., Беляев С.А. Алгоритм принятия обоснованных решений в нештатных ситуациях на основе моделей нечетких множеств // Изв. СПбГЭТУ «ЛЭТИ»: Информатика и компьютерные технологии. 2016. № 9. С. 16–21.
11. Черная О.С., Федорова Ю.Ю., Беляев С.А. Применение методов и средств автоматизированного тестирования для проверки качества программных комплексов обработки измерительной информации // Изв. СПбГЭТУ «ЛЭТИ»: Компьютерные и информационные технологии в науке и образовании. 2013. № 9. С. 55–58.