

УДК 004.588

DOI: [10.26102/2310-6018/2020.30.3.033](https://doi.org/10.26102/2310-6018/2020.30.3.033)

Один из аспектов разработки универсального конструктора задач для тренажерных систем

В.В. Сальников^{1,2}, М.Н. Кравцов²

¹Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»,
Тула, Российская Федерация

²Публичное акционерное общество Акционерная компания «Центральный научно-исследовательский институт систем управления»
Тула, Российская Федерация

Резюме: В работе рассмотрен вопрос построения блочной иерархии программно-информационных моделей, имитирующих предметную область в тренажерных системах, по принципу от простого к сложному, от элементарных функций к более комплексным. Это позволяет реализовать принцип адаптивности в тренажерных системах – возможность управления сложностью тренировочного процесса в зависимости от компетентности обучаемого. Предложен подход к построению универсального конструктора задач для тренажерных систем, основанного на представлении тренировочного процесса в виде изменения конечного числа параметров тренировочного блока путем воздействия на него обучаемого. Тренировочный блок в такой постановке рассматривается как совокупность параметров различного типа. Предложен метод оценки тренировочного процесса, который подразумевает проверку состояния блока по завершению тренировки, то есть проверки соответствия значений параметров заданным. Он позволяет создавать гибкие системы оценивания знаний. Корреляция определенных параметров с умениями, навыками обучаемого позволяет автоматизировать процесс контроля уровня его предметной компетентности и дает возможность реализовать принцип адаптивности тренажерных систем, заключающийся в постепенном росте сложности тренировочного процесса. Представлен прототип универсального конструктора задач для тренажерных систем. Его использование значительно упрощает редактирование задач, пренебрегая использованием сред программирования.

Ключевые слова: обучение, тренировка, тренировочный процесс, тренажерные системы, тренажер, моделирование задач.

Для цитирования: Сальников В.В., Кравцов М.Н. Один из аспектов разработки универсального конструктора задач для тренажерных систем. *Моделирование, оптимизация и информационные технологии*. 2020;8(3). Доступно по: https://moit.vivt.ru/wp-content/uploads/2020/08/SalnikovKravzov_3_20_1.pdf DOI: 10.26102/2310-6018/2020.30.3.033

One of the aspects of developing a universal task designer for training systems

V.V. Salnikov^{1,2}, M.N. Kravtsov²

¹Federal state budgetary educational institution of higher education «Tula state university»,
Tula, Russian Federation

²Public company Stock company «Central research Institute of control systems»,
Tula, Russian Federation

Abstract: The paper considers the issue of building a block hierarchy of software and information models that simulate the subject area in training systems, based on the principle from simple to complex,

from elementary functions to more complex ones. This allows us to implement the principle of adaptability in training systems – the ability to manage the complexity of the training process depending on the competence of the trainee. An approach to the construction of a universal task constructor for training systems based on the representation of the training process in the form of changing a finite number of parameters of the training block by affecting it by the trainee is proposed. The training block in this setting is considered as a set of parameters of various types. A method for evaluating the training process is proposed, which involves checking the state of the block at the end of training, that is, checking whether the parameter values match the specified ones. It allows you to create flexible systems for evaluating knowledge. Correlation of certain parameters with the skills of the trainee allows automating the process of controlling the level of his subject competence and makes it possible to implement the principle of adaptability of training systems, which consists in a gradual increase in the complexity of the training process. A prototype of a universal task designer for training systems is presented. Its use significantly simplifies task editing, neglecting the use of programming environments.

Keywords: training, training, training process, training systems, simulator, task modeling.

For citation: Salnikov V.V., Kravtsov M.N. One of the aspects of developing a universal task designer for training systems. *Modeling, Optimization and Information Technology*. 2020;8(3). Available from: https://moit.vivt.ru/wp-content/uploads/2020/08/SalnikovKravzov_3_20_1.pdf DOI: 10.26102/2310-6018/2020.30.3.033 (In Russ).

Введение

Одной из важных областей деятельности современного общества является образование. Это связано с необходимостью перехода на инновационный путь развития. Классические подходы к образованию – обучение на всю жизнь – не позволяют готовить современных специалистов из-за большой динамики процессов, происходящих в обществе. В таких условиях работники всех уровней должны обновлять свои знания. Для успешного получения новых знаний необходимы электронные обучающие системы на основе передовых технологий [1]. Решению этой задачи посвящены работы многих специалистов в области информационных технологий [2-7]

Умение человека выполнять ту или иную работу зависит от индивидуальных особенностей интеллекта, способности переносить перезагрузки. Для определения способностей отдельного человека требованиям, предъявленным тем или иным видом деятельности, служит профессиональный отбор. Существует много различных методик отбора, позволяющих довольно точно определить будущих специалистов. Однако еще более важная роль в процессе становления специалиста отводится процессу обучения, а в особенности его практической части [8]. Организация практических занятий в определенных предметных областях не всегда бывает выгодна ввиду некоторых причин: неоправданно высокая цена их проведения; невозможность их организации по причине отсутствия средств обучения; сложность адекватной оценки результатов занятия и др. Тренажерные системы позволяют несколько сгладить эти недостатки. Исходя из опыта внедрения систем такого типа актуальными являются задачи построения гибкой системы оценки знаний и умений обучаемых и создание универсального конструктора задач.

Модель тренировочного процесса

Тренировка – это процесс, результатом которого является совершенствование каких-либо навыков и умений, то есть формирование предметной компетентности у обучаемых в результате выполнения теоретических и практических заданий. Тренажерные системы позволяют автоматизировать этот процесс путем реализации программно-информационных моделей, имитирующих конкретную предметную область. Как правило, такие модели являются многопараметрическими и одним из

главных их свойств является адаптивность – возможность управления сложностью тренировочного процесса в зависимости от компетентности обучаемого. Это достигается благодаря построению блочной иерархии таких моделей по принципу от простого к сложному, от элементарных функций к более комплексным. Можно говорить об организации тренировочного процесса путем выполнения задач на таких функциональных блоках. Описанная выше структура представлена на Рисунке 1. Блок может быть организован путем композиции более простых блоков, что показано на Рисунке 1 в правой части.

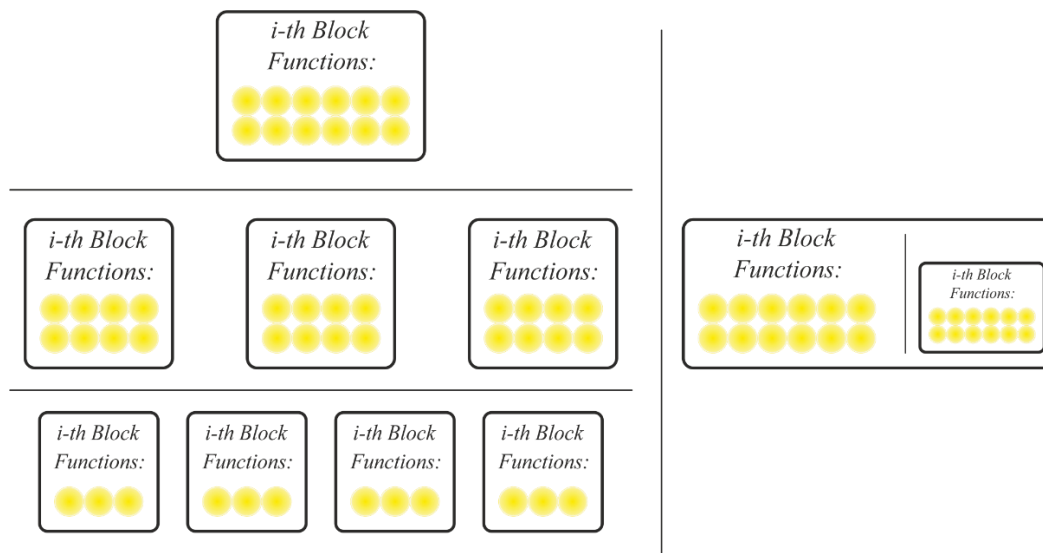


Рисунок 1 – Блочная иерархия тренажерной системы
 Figure – 1 Block hierarchy of the training system

Тренажерные системы, как правило, имеют архитектуру «учитель-ученик», «инструктор-оператор» или, с программной точки зрения, «клиент-сервер». В соответствии с предложенной структурой рабочее место инструктора выступает в роли хранилища эталонных функциональных блоков, а рабочее место обучаемого является отображением этого хранилища (см. Рисунок 2). На Рисунке также отражены основные компоненты, входящие в состав рабочего места инструктора: КЗ – конструктор задач; СОЗУ – система оценки знаний и умений; СЗТП – система запуска тренировочного процесса; БД – база данных для ведения статистики.

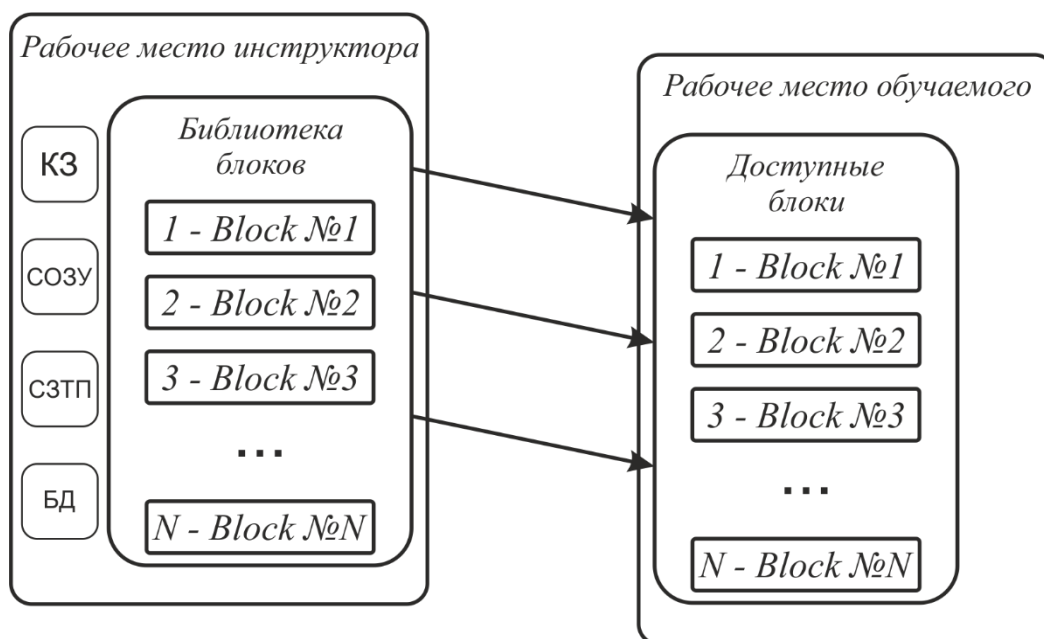


Рисунок 2 – Структура тренажерной системы
 Figure 2 – Structure of the training system

Функциональный блок задается совокупностью параметров, характеризующих его состояние в определенный момент времени:

$$p = (p_1, p_2, \dots, p_N),$$

где N – количество параметров функционального блока.

С точки зрения тренажерных систем по значениям параметров можно судить о выполнении или невыполнении обучаемым поставленной перед ним задачи. Параметры должны удовлетворять следующими требованиям:

- обладать средствами идентификации;
- обладать простыми средствами оценки и сравнения с заданными значениями;
- иметь вложенную структуру для описания составных параметров;
- характеризовать определенный навык или умение обучаемого.

В связи с вышесказанным тренировочный процесс может быть представлен в виде изменения конечного числа параметров блока путем воздействия на него обучаемого. Результатом такого процесса является получение опыта, а при достижении заданных значений определенными параметрами – совершенствование навыков, умений и повышение предметной компетентности. При этом существенным является исходное состояние блока перед началом тренировочного процесса, т.е. состояние параметров перед стартом. Графически введенное описание представлено на Рисунке 3.



Рисунок 3 – Тренировочный процесс, реализуемый на функциональном блоке. p^{in} – совокупность параметров, характеризующих исходное состояние блока; p^{out} – совокупность параметров, характеризующих конечное состояние блока; z – воздействия обучаемого

Figure 3 – Training process implemented on the functional block. p^{in} – a set of parameters that characterize the initial state of the block; p^{out} – a set of parameters that characterize the final state of the block; z – impact of the trainee

В такой поставке оценивание тренировочного процесса подразумевает проверку состояния блока по завершению тренировки p^{out} , то есть проверка соответствия значений параметров блока заданным, а конструирование задач заключается в составлении набора значений этих параметров для получения определенной оценки.

Формализация универсального конструктора задач

Тренажерные системы должны обладать достаточной гибкостью в вопросах, связанных с моделированием тренировочных заданий. Это достигается путем создания универсального конструктора задач. Он позволит обеспечить удобный интерфейс добавления новых задач, редактирования и модификацию старых как для инструкторов на этапе использования, так и для разработчиков на этапе тестирования. В соответствии с предложенным подходом к описанию тренировочного процесса такой конструктор должен удовлетворять следующим требованиям:

- на вход принимать функциональный блок в параметризованном виде с целью отображения доступных оценочных параметров и добавления их в списки параметров;
- иметь инструменты для добавления новых задач, редактирования и модификации старых;
- при редактировании задачи обеспечить формирование списка значений параметров, характеризующих состояние блока перед стартом; при этом необходимо предоставить возможность объединять параметры в группы с признаком логических операций;
- обеспечить формирование перечня оценок выполнения задачи, так называемую оценочную ведомость или протокол оценки;
- обеспечить редактирование определенной оценки путем составления списка значений определенных параметров, характеризующих состояние блока на выходе;
- обеспечить сохранение сформированных задач в хранилище;
- обеспечить возможность провести запуск задачи на выбранном функциональном блоке для тестирования.

С программной точки зрения функциональный блок должен быть представлен в параметризованном виде, то есть описан массивом переменных разных типов данных, по значениям которых так или иначе можно охарактеризовать состояние блока в

определенный момент времени. Среди параметров могут быть, например, признаки состояния, значения, последовательность действий оператора и другие. По типу данных их следует классифицировать следующим образом: простые параметры (числовые, логические, строковые) и составные параметры (пользовательские структуры, массивы). Составные параметры в конечном итоге сводятся к набору простых параметров.

Параметризация блока происходит путем расширения его функциональности. Первым этапом разрабатывается и тестируется программно-информационная модель блока. После чего осуществляется ее параметризация, заключающаяся в создании списка доступных для конструктора параметров (см. Рисунок 4).

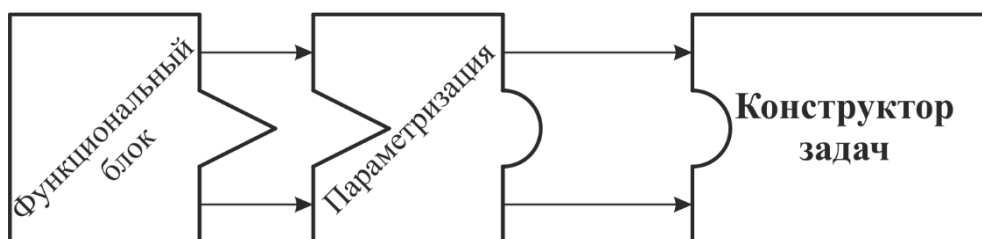


Рисунок 4 – Параметризация модели функциональных блоков тренажерной системы
Figure 4 – Parametrization of the model of functional blocks of the simulator system

Для программирования параметров, отличающихся как по типам данных, так и по внутренней структуре, необходим единый интерфейс для их описания. Он должен как минимум включать в себя следующие функции:

- предоставлять доступ к имени (Name) и описанию параметра (Description);
- предоставлять доступ к уникальному идентификатору (ID), по которому можно идентифицировать как простой параметр, так и простой в составе сложного параметра;
- предоставлять доступ к значению (Value);
- иметь вложенную структуру для описания сложных параметров;
- иметь инструменты редактирования списка навыков и умений (SkillImpact);
- обеспечить редактирование значения (EditValue) для формирования исходного и конечного состояния блока.

Результаты

На Рисунке 5 представлен прототип универсального конструктора задач, состоящий из следующих основных частей: список доступных параметров модели; панель формирования оценочной ведомости и панель формирования состояния блока перед стартом. Как видно из Рисунка в списке доступных параметров представлен сложный тип параметров – пользовательские структуры.

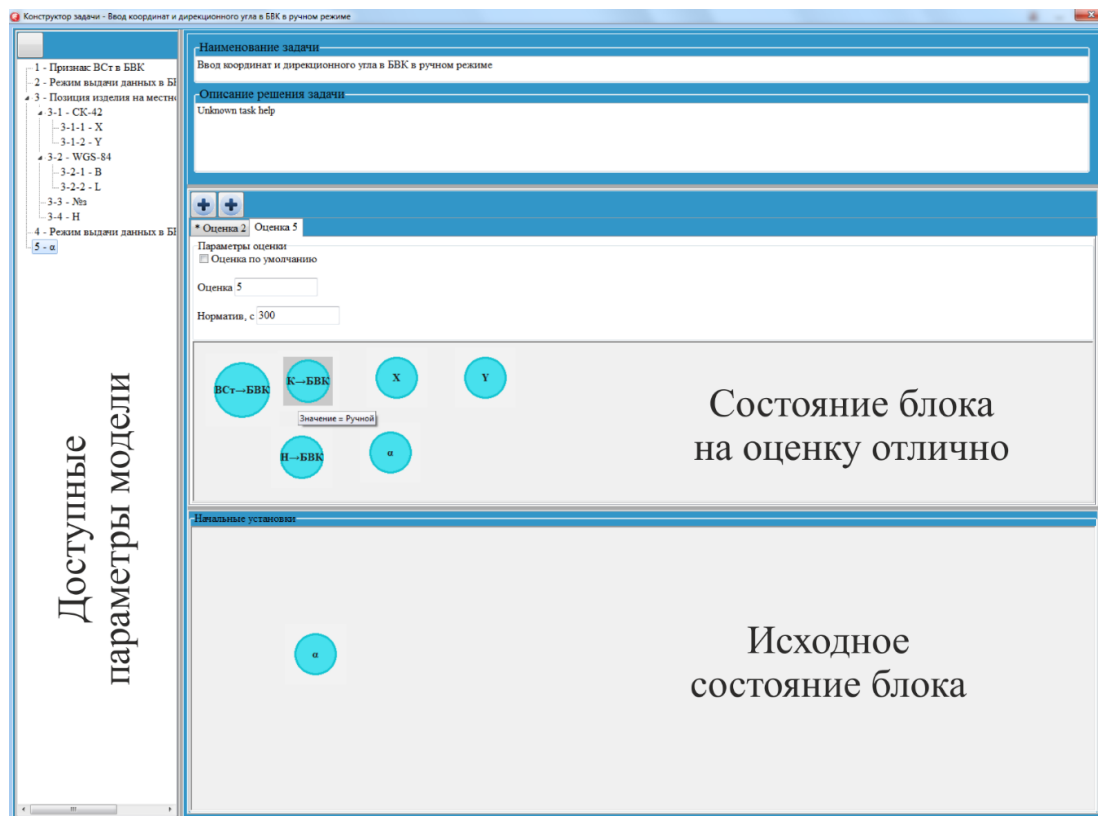


Рисунок 5 – Прототип конструктора задач
Figure 5 – Prototype of the task designer

Для редактирования значений параметров вызывается окно редактирования параметра, представленное на Рисунке 6.

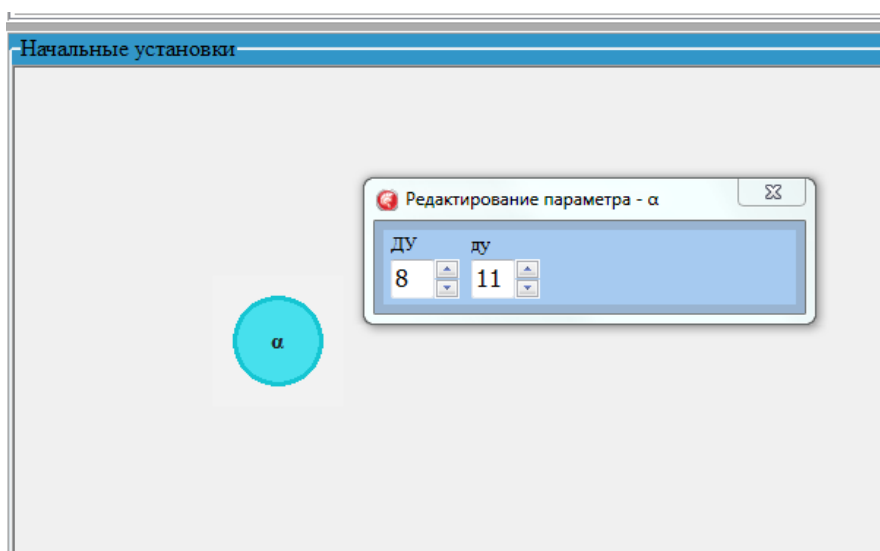


Рисунок 6 – Редактирование значения параметра
Figure 6 – Editing a parameter value

Ниже приведен протокол оценки результата выполнения составленной в конструкторе задачи:

Ваша оценка: 2
Время выполнения задачи: 123 с
Оценка по умолчанию: 2

Проверка оценки: 5 Норматив: 300
Проверка параметра ВСт→БВК
SUCCESS: Значение установленного параметра |Признак: ВСт в БВК| соответствует требуемому (true = true)
Проверка параметра X
FAIL: Значение установленного параметра |X| не соответствует требуемому (5001068 <> 5657182)
Проверка параметра Y
FAIL: Значение установленного параметра |Y| не соответствует требуемому (5001068 <> 5258857)
Проверка параметра α
FAIL: Значение установленного параметра |α| не соответствует требуемому (00 - 67 ду <> 12 - 67 ду)
Проверка параметра К→БВК
FAIL: Значение установленного параметра |Режим выдачи данных в БВК| не соответствует требуемому (Автоматический <> Ручной)
Проверка параметра Н→БВК
FAIL: Значение установленного параметра |Режим выдачи данных в БВК| не соответствует требуемому (Автоматический <> Ручной)

Заключение

Предложенный подход позволяет построить универсальный конструктор задач для тренажерных систем. Его использование значительно упростит редактирование библиотеки задач, пренебрегая использованием сред программирования. Подход к оценке тренировочного процесса путем сравнения значений параметров блока с заданными позволит создавать гибкие системы оценивания знаний. Корреляция определенных параметров с умениями, навыками обучаемого позволит автоматизировать процесс контроля уровня его предметной компетентности и даст возможность реализовать принцип адаптивности тренажерных систем, заключающийся в постепенном росте сложности тренировочного процесса.

ЛИТЕРАТУРА

1. Трембач В.М. Электронные обучающие системы с использованием интеллектуальных технологий. *Открытое образование*. 2013;(4):52-62.
2. Голенков В.В., Емельянов В.В., Тарасов В.Б. Виртуальные кафедры и интеллектуальные обучающие системы. *Новости искусственного интеллекта*. – 2001;(4).
3. Рыбина Г.В. Экспертные системы и инструментальные средства для их разработки: некоторые итоги. *Информационно-измерительные и управляющие системы*. 2013;11(5):35-48.
4. Стефанюк В.Л. Поведение квазистатической оболочки в изменяющейся нечеткой среде. *Сборник научных трудов национальной конференции с международным участием «Искусственный интеллект 94»*. 1994;1:199-203
5. Тарасов В.Б. *От многоагентных систем к интеллектуальным организациям: философия, психология, информатика*. 2002.
6. Трембач В.М. Системы управления базами эволюционирующих знаний для решения задач непрерывного образования. 2013.

7. Telnov Y. The Model of Continuous Profession-oriented Learning in the E-environment Based on a Competence Approach and Academic Knowledge Management. *11th European Conferences of Knowledge Management Systems*. 2010, Porto
8. Гузик В.Ф., Золотовский В.Е., Переверзев В.А. Тренажерные системы на базе локальных вычислительных сетей. *Известия ЮФУ. Технические науки*. 2002;(2):175-179

REFERENCES

1. Trembach V.M. Electronic training systems using intelligent technologies. *Open education*. 2013;(4):52-62.
2. Golenkov V.V., Emelyanov V. V., Tarasov V. B. Virtual departments and intellectual training systems. *Artificial intelligence news*. 2001;(4).
3. Rybina G.V. Expert systems and tools for their development: some results. *Information-measuring and control systems*. 2013;11(5):35-48.
4. Stefanyuk V.L. Behavior of a quasi-static shell in a changing fuzzy environment. *National conference with international participation "Artificial intelligence 94": collection of scientific papers*. 1994;(1):199-203
5. Tarasov V.B. From multi-agent systems to intellectual organizations: philosophy, psychology, Informatics. 2002.
6. Trembach V.M. Systems of management of bases of evolving knowledge for solving problems of continuous education: monografiya. 2013.
7. Telnov Y. The Model of Continuous Profession-oriented Learning in the E-environment Based on a Competence Approach and Academic Knowledge Management. *11th European Conferences of Knowledge Management Systems*, 2010, Porto
8. Guzik V.F., zolotovskiy V. E., Pereverzev V. A. Training systems based on local computer networks. *SFU news. Technical science*. 2002;(2):175-179

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Сальников Владимир Владимирович, аспирант, кафедры вычислительной техники, ФГБОУ ВО «Тульский государственный университет» Институт прикладной математики и компьютерных наук; ведущий инженер, ПАО «АК» Центральный научно-исследовательский институт систем управления», отдел по разработке программного обеспечения специального и общего назначения, Тула, Российская Федерация.
e-mail: vladimirsalnikov95@yandex.ru

Vladimir V. Salnikov, Phd Student, Computer Engineering Department, Tula State University;
Lead Engineer, Central Research Institute Of Control Systems, Special And General Purpose Software Development Department, Tula, Russian Federation

Кравцов Михаил Николаевич, начальник отдела, ПАО «АК» Центральный научно-исследовательский институт систем управления», отдел по разработке программного обеспечения специального и общего назначения, Тула, Российская Федерация.
e-mail: kravcov@cniisu.ru

Mikhail N. Kravtsov, Head Of The Department, Central Research Institute Of Control Systems, Special And General Purpose Software Development Department, Tula, Russian Federation.