

SOFTWARE SET OF INTELLECTUAL SUPPORT AND SECURITY OF LMS MAI CLASS.NET

A. V. Naumov, Moscow Aviation Institute, Moscow, Russian Federation,
naumovav@mail.ru,

G. A. Mkhitaryan, Moscow Aviation Institute, Moscow, Russian
Federation, Russian Federation, grgmkn@mail.ru,

A. A. Rybalko, Moscow Aviation Institute, Moscow, Russian Federation, Russian
Federation, aar@mail.ru

The article discusses integration of mathematical methods and security components in a learning management system (LMS). The discussed methods provide a statistical analysis of user data in process of online training in mathematical disciplines and adapting the content of the system for different users. The software package allows automatic calculation of questions complexity and user ratings by using statistical data. This helps the system administrator to detect users which use illegal hints or help from others. A procedure is designed for content selection for a variety of tests and control activities, with restrictions and without a time limit for the test. Two probabilistic models were used during development of mathematical methods: the Rasch model to describe the probability of users' answer correctness and Van der Linden model to describe the time it took for a user to respond to the question. The software package contains special optimization procedures that estimate the parameters of these models based on the accumulated statistics across all users. Apart from discussing efficiency of LMS usage with the above mentioned methods, the article also discusses general security architecture of the LMS, a set of technologies used for developing system security with specific implementation examples.

Keywords: learning management system; LMS; e-learning; statistical analysis; adaptive characteristics; system security container technologies.

Introduction

Computer technologies play an important part in modern educational process. Specifically, these technologies are very important in distance education for different categories of students. A LMS (Learning Management System) is used to conduct online training. Moodle, Blackboard are examples of the more common LMS among other systems. In general, LMS is a complex software solution which allows one to create educational courses with theoretical (video, audio, electronic textbooks) and practical (questions, tests, lab work in simulated conditions) material. The management shell of a LMS allows collaboration of different users (teachers, students, system administrators).

MAI CLASS.NET [1] is a LMS used for teaching mathematical disciplines at Moscow Aviation Institute (National Research University). Currently, students of MAI have access to courses on probability theory, mathematical statistics, mathematical analysis, linear algebra and analytic geometry, differential equations, and several other courses are in development. The structure of electronic courses and functionality of LMS MAI CLASS.NET are discussed in Section 1.

Several mathematical problems were solved during LMS development, which allow automatic change in operation depending on user actions, and algorithms have been proposed to analyze the statistical data of real users. Problems include formation of

integral rating by statistical processing of test results [2], automatic assessment of questions complexity based on maximum likelihood estimation [3], the formation of individual test questions with and without restrictions at runtime [4]. The main purpose of considering these problems is the introduction of adaptive features in LMS, based on the developed algorithms. These features form a powerful competitive advantage of the LMS MAI CLASS.NET compared to the existing analogues. Another advantage of the LMS MAI CLASS.NET are special materials for creation of a course on one of the mathematical disciplines: different question types, including a formula-type question, questions for self-study and branched script execution, etc. The article describes the technology and toolset used by the authors to solve the problem of software implementation and integration of a set of mathematical methods in the management shell, which provides statistical analysis of users data and adaptation of the solution content to the changing user quantity. Along with the above methods, other methods were proposed for improving the efficiency of the LMS implementation to ensure safety of the mentioned complex techniques and LMS as a whole.

1. Specification of LMS

The complete specification of LMS MAI CLASS.NET is provided in [1]. We will briefly examine the outer shell of the system and several other software components, along with the technology implementation.

The LMS content sections with theory, practice and results are available to all types of users. In the "Theory" section the material is based on the lectures made by the department of "Probability Theory" staff, faculty No 8 "Applied Mathematics and Physics" of MAI. Theoretical material is organized as hypertext and is supplied with a glossary that allows usage as a reference manual and gives the possibility to begin material study practically at any place. The section «Practice» allows to get access to practical questions and laboratory simulation. Review questions of the practical part are divided into 4 types on the execution purposes. Types list and their description are provided in Table 1.

Table 1

Types of questions

Test	questions designed to assess the degree of users knowledge of the base course of theoretical material and user readiness to handle practical questions
Self-study	questions with a multilevel system of hints for organization of self-study and preparation for exam procedures
Control	questions, designed for examination
Labs	questions with branched scenarios depending on step by step user action

There are various opportunities to design questions using LMS MAI CLASS.NET. Users can import different objects to design new questions, for example, images and various question types: text, checkbox, radiobutton, formula-type questions by using a calculator

with user-friendly interface. The calculator saves entered formulas as a LaTeX formulas. The LMS uses an algorithm based on parsing of the LaTeX formulas, which allows to check the correctness of an entered formula. One of MAI CLASS.NET advantages is extensive usage of random values in the parameters of all question types. This reduces the chances of using illegal ways to answer the question, for example, when the user knows the correct answer or when somebody else helps the user, and as consequence brings increase in the uniqueness of the questions, which also contributes to change of content in the LMS. There are also special question types developed for the course "Linear Algebra and Analytic Geometry": a matrix question type, another type for vector decomposition in the basis, for solving systems of linear equations, etc.

The section "Results" provides access to statistical data collected by the LMS for various types of users. Personal results of users with detailed information about learning progress for some courses are especially important. This information includes correctness of the answer to one question and full test mark, the time that the user spends to answer one question and to pass the full test, user ratings, access to comparative data of student groups or faculties, etc. This approach helps teachers to detect students that deserve high rating.

Software tools used in LMS MAI CLASS.NET form a set of implementations for mathematical algorithms processing statistical data of real users that work in LMS. LMS uses computer-adaptive testing methods (CAT), that adapt to the examinees' level. In order to advance the solution adaptivity, it is necessary to solve some applied mathematical problems based on special models: Rasch model and one-parameter logistic model (1PL) is used for analyzing categorical data, such as answers to questions on a reading assessment or questionnaire responses, as a function of the trade-off between the respondent's abilities and the item difficulty [5]; lognormal model of Van der Linden [6] is used for response-time (RT) modelling. All of the used models are stochastic, and that creates some problems in developing algorithms for their implementation in LMS. It is further proposed to consider mathematical algorithms based on Rasch and van der Linden models and software tools for their implementation in LMS solutions.

2. Mathematical Algorithms and Their Implementation

2.1. Brief Overview of Technologies Used for LMS

LMS MAI CLASS.NET uses a client-server model of computing. This means that, for example, display of course content is a client-side function, but other functions are server-side, for example, data storage and data processing in the database, evaluation of statistical data of users. LMS MAI CLASS.NET, as well as most of other modern solutions, separately stores static and dynamic content. Static elements of a web page (CSS styles, JavaScript sources, images, etc.) are stored on the server. This reduces system general workload, when it generates dynamic content: questions, theory and user statistical data.

A MySQL relational database management system is used as storage for statistical and other data. PHP (a server-side scripting language designed for web development) scripts are used for actions with the database, but server software allows us to also use other languages for the backend, Python being an example. The advantage of using different programming languages is that algorithms implementation at runtime may be different because of a selected language, and some languages may have special data types which

make implementation of difficult algorithms easier. Mathematical problems of computer-adaptive testing will be described below.

2.2. Mathematical Problems of Computer-Adaptive Testing

The problem of integral rating formation by statistical processing of test results was discussed in [2, 3] with **problem of the automatic assessment of question difficulty** using the maximum likelihood estimation.

Difficulty of the question is one of the parameters in Rasch model [5]. The probability of success in the model upon interaction between the relevant person j and the assessment question i looks like:

$$\mathcal{P}(\Theta_j, w_i) \triangleq \frac{\exp(\Theta_j - w_i)}{1 + \exp(\Theta_j - w_i)},$$

where w_i is the difficulty of question i , Θ_j is the ability of user j .

The model assumes that with changing of users ability from $-\infty$ to $+\infty$ the probability of success will change from 0 to 1. Initial question difficulty and users ability parameters are estimated by using maximum likelihood estimation, and for maximization of maximum likelihood function we used a quasi-Newton method algorithm of Broyden – Fletcher – Goldfarb – Shanno. Evaluated assessments w_i and Θ_j have been brought to standard for usage in the LMS default scale. For example, users ability have been converted to a five-point scale from 0 to 5.

The problem of content forming for a variety of tests and control activities was discussed in [4]. Modern LMS needs a variety of different functions for computer-adaptive testing, so we updated problems conditions in [7] based on problems conditions from [4]. The main addition is that the user has limited time to answer the question so the system or the teacher can give more accurate assessment of the student's potential. Moreover, face-to-face testing, which we can conduct using LMS MAI CLASS.NET, needs to select the right value of time, in the span of which all users can be considered sufficient for testing with selected probability. Below we will describe the **problem of content generation for a variety of tests with a time limit**.

We have a set $Z = (z_1, \dots, z_I)$ of I questions in LMS and M different types of those questions Z_m , $m = 1, \dots, M$. We will use a matrix A with the dimension $M \times I$ to define a question and it's type:

$$A = \|a_{mi}\|, a_{mi} = \begin{cases} 1, & z_i \in Z_m, \\ 0, & z_i \notin Z_m. \end{cases}$$

Consider a vector $u \in \mathbb{R}^I$ (below all vectors are row vectors), whose elements show whether the question was selected for test or not:

$$u_i = \begin{cases} 1, & \text{if question } i \text{ is in test,} \\ 0, & \text{otherwise.} \end{cases}$$

By $w \in \mathbb{R}^I$ denote the difficulty vector. Let c be the total difficulty of the test, that is the sum of the selected test questions difficulties.

We want to get the set of tests consisting of k questions of different types considering that $k \geq M$. First, we need to select the value of total difficulty c that is calculated using a special algorithm or based on expert review. Secondly, we consider the small difference between the selected value of c and a possible total difficulty of tests after using the algorithm. We use ε as the difference that was shown above.

We use Van der Linden model [6] for user n response-time for solving question i .

Let T_{ni} be a logarithmic transformation of user n response-time for solving question i . We consider that T_{ni} consists of several components: the first one is associated with the difficulty of a question ($\beta_i, i = 1, \dots, I$), the second one is associated with users abilities ($\tau_n, n = 1, \dots, N$), the third one is associated with all users and all questions (μ), and the last one is a random deviation. So, logarithmic transformation of user n response-time for solving question i is:

$$\ln T_{ni} = \mu + \beta_i + \tau_n + \varepsilon_{ni}, \quad i = 1, \dots, I, \quad n = 1, \dots, N, \quad (1)$$

where $\varepsilon_{ni}, i = 1, \dots, I, n = 1, \dots, N$ are independent random variables, $\varepsilon_{ni} \sim \mathcal{N}(0, \sigma^2)$ is a normal random variable.

We consider that N users participate in testing. Under the above assumptions, T_{ni} is random time with lognormal distribution. Let t be the time in which users must pass the test, and let T be the matrix with dimension $N \times I$, whose elements are T_{ni} :

$$T = \|T_{ni}\|.$$

Consequently, we have a problem of formation or generation of tests within the time-limit:

$$U_* = \underset{u}{\text{Arg min}} |c - wu^T| \quad (2)$$

$$e_I u^T = k \quad (3)$$

$$c - wu^T \geq -\varepsilon \quad (4)$$

$$c - wu^T \leq \varepsilon \quad (5)$$

$$Au^T \geq e_M^T \quad (6)$$

$$P(Tu^T \leq t_f^T) \geq \alpha, \quad (7)$$

где $w = (w_1, \dots, w_I), e_I \in \mathbb{R}^I, e_I = (1, \dots, 1), e_M \in \mathbb{R}^M, e_M = (1, \dots, 1), t_f \in \mathbb{R}^N, t_f = (t, \dots, t)$.

The model of distribution of random response-time during which a user is required to answer the question in the LMS, helps an administrator or a teacher to detect users that use an unauthorized way to answer or help from other users. For example, if a users' response-time is less than $(1 - \alpha)$ -quantile (α – selected high value of probability), an administrator of the LMS must check the user for using illegal methods. If the response-time is not enough for most users in a group and more than α -quantiles value of response-time distribution, the administrator can think about replacement of the question by another one. These CAT methods were discussed in [8, 9].

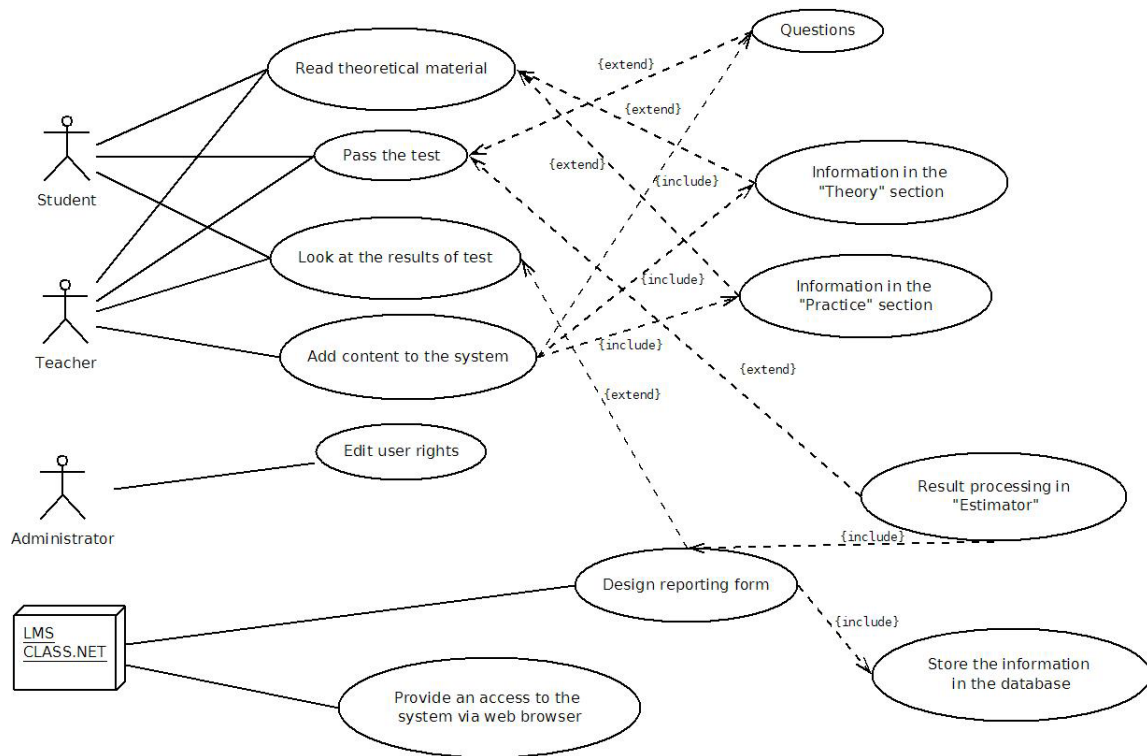


Fig. 1. Scheme of collaboration

2.3. Features of the Implementation of Algorithms in LMS Shell

The algorithm of solving the problem of test formation is based on a branch and bounds algorithm and checking probability using Monte Carlo method. However, this requires a lot of computing. LMS will need additional computing resources for implementation of algorithms like that. For example, it is better to use libraries with special data types, like data types from Python. In LMS MAI CLASS.NET we use PHP for algorithm implementation. In addition we use a random number generator for lognormal random variables. Also we changed our database and js-sources for fixing response-time of users.

We describe below the implementation of algorithms shown above in the terms used in [1].

LMS MAI CLASS.NET stores static and dynamic content separately: static elements of a web page (CSS (Cascading Style Sheets), JavaScript sources, images, etc.) stored on the server reduce overall system workload, when it generates dynamic content: questions, theory and user statistical data. The generation of dynamic content occurs as an interaction of the statistical database and the content database that consists of three components: "Checker", "Estimator", "Generator". Program implementation of all algorithms is in the "Estimator" component, which collaborates with the database to export and analyze statistical data. "Estimator" needs the result of analyzing to change value of difficulty of question or integral rating of users.

The scheme of collaboration of the main application, the database and "Estimator" components is on Fig. 1.

For the organization of computing processes, we use a time-based job scheduler to run implementations of algorithms periodically at fixed times, dates, or intervals. Typically,

it automates system maintenance and administration and helps to reduce the load on the server resources. Also, we can make a priority list for calculations and their order. This helps to avoid problems with the synchronization of algorithms, when they collaborate within one database.

The priority list is as follows:

1. Estimation of difficulty of questions.
2. Integral rating.
3. Formation of tests.

The problem of synchronization is solved by using blocking transactions that are available native in MySQL.

Most algorithms used in LMS MAI CLASS.NET require extensive calculations. So, we have to consider server specifications, software, system architecture, ability to upgrade the server or server software, web traffic and attendance hours. At the moment, more than 3000 users collaborate in LMS MAI CLASS.NET during the semester. Peak hours for one day are $t_{max} \approx 7\text{h}$ (usually evening hours from 18:30 to 1:30) and hours with less user activity are $t_{min} \approx 5\text{h}$ (morning hours from 5:00 to 10:00).

Obviously, user activity increases at the end of the semester, which reduces the hours with less user activity to $t_{min} \approx 3\text{h}$.

3. Functional Security of LMS Software System

In process of security planning for LMS CLASS.NET potential attack channels were investigated. An expert model of potential threats was created. In order to create such a model, linkage analysis of the CLASS.NET software was conducted, and a model of extrinsic features and links was designed according to the IDEF-0 standard. The current implementation of the design uses container architecture of OpenVZ for components virtualization. The open source nature of the OpenVZ project enables the community, as well as commercial stakeholders to scan the source code of the control system (the hypervisor [10]) on a regular basis in search for undocumented backdoors, which positively counts towards the overall code security and stability (see Fig. 2).

LMS software is functioning in the institute intranet, as well as being available from the WAN (Wide Area Network). This means that the virtual NIC (Network Interface Controller) adapters of the application module are the points of demarcation in local security context. The first security measure taken is the isolation of control interfaces from user data channels and content configuration in a separate management network. The actual implementation of such a network can be made using VLAN (Virtual Local Area Network) technology, as well as the VPN (Virtual Private Network). The second approach is used in the security design of LMS CLASS.NET, because it provides flexibility of control and enables cascading of the VPN gateway with other gateway types, for example NAC (Network Access Control), NAT (Network Address Translation) or external user authentication via the RADIUS (Remote Authentication in Dial-In User Service) services.

Prevention of massive attacks (like DDoS), as well as exploitation of network protocol nuances using specially crafted network packets is enabled by controlling the access channel to the system with a network security scanner. The scanner works as an early alerting

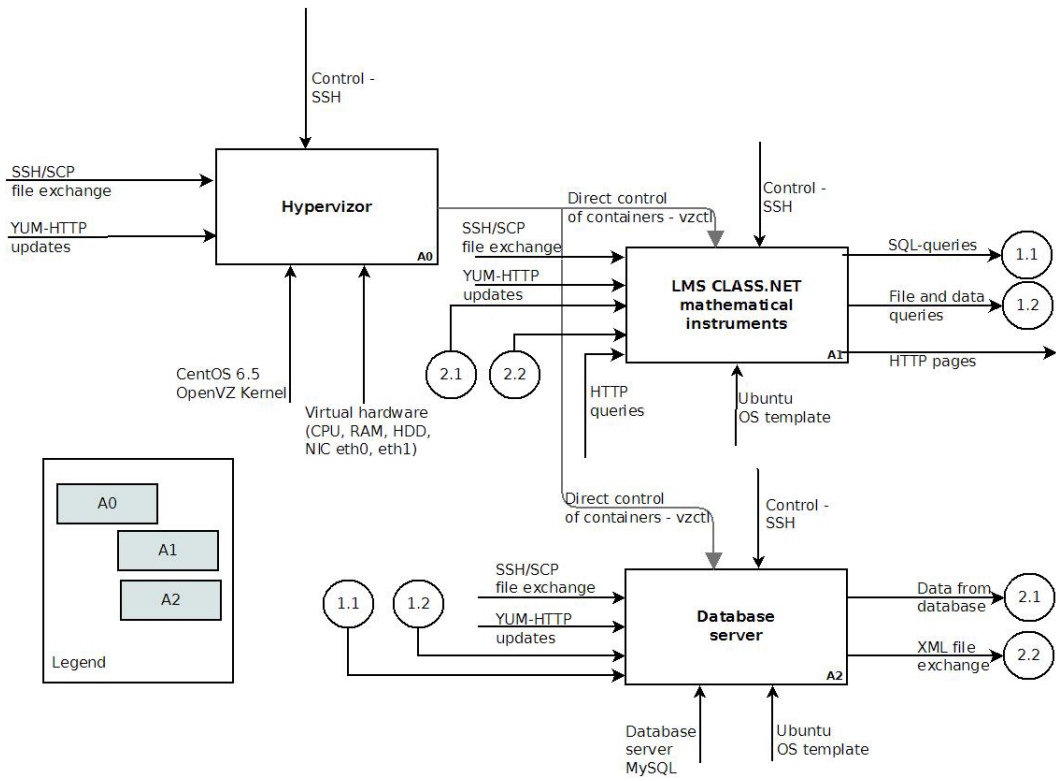


Fig. 2. Model of links for LMS software CLASS.NET

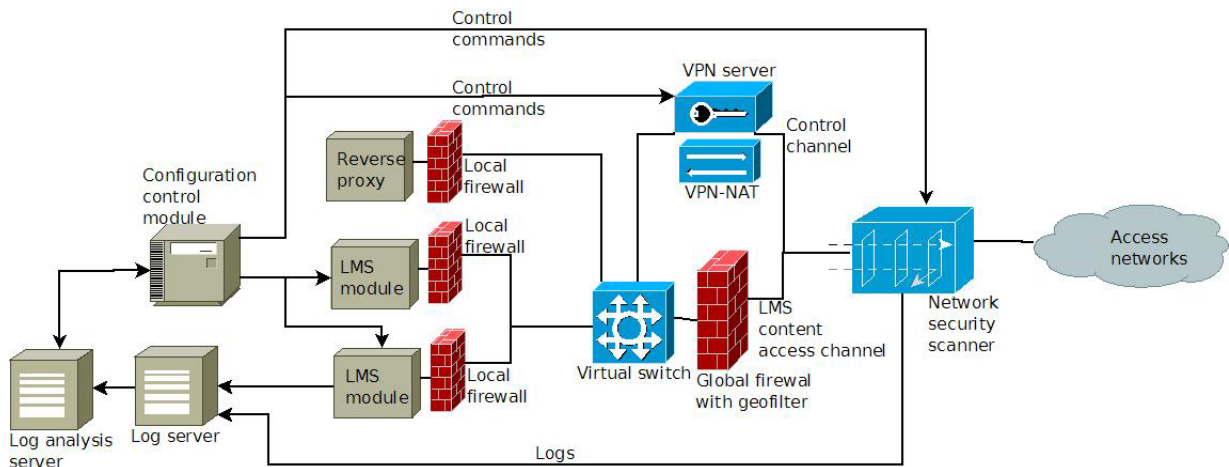


Fig. 3. Network security diagram of LMS CLASS.NET and service data exchange channels

instrument when the attack begins. The global firewall is configured to filter out packets from unrecognized sources. The global firewall can also be configured with a geofilter and blacklist to block large WAN segments, which is a simple but effective proactive measure against massive network attacks. The network security diagram (see Fig. 3) is also designed with the possibility of one of the virtual servers being overrun by hackers in mind. For example, an application worker module of LMS CLASS.NET.

The LMS software was built to work in an environment with load balancing features in mind [1] and an outage of N-1 worker modules of the clustered application (where N is the overall number of modules) does not impact the availability of the system itself. In order to prevent an attacker to use the captured modules to launch an attack in the LAN, an automatic micro-segmentation system was included in the design using the local module firewalls. The location of local firewall modules and their management must be external to the LMS software modules themselves to ensure additional security. The optimal placement of the local firewall filters is on the virtual switch level using the control module engine as a configuration controller for these firewalls.

Load-balancing between LMS software modules is conducted using the reverse-proxy instrument. Such a module can implement several functional security and availability tasks simultaneously. The reverse-proxy secures the data-exchange channel in LMS software. The reverse-proxy can consolidate traffic from LMS modules being sent via HTTP protocol, and publish the application as a whole outside using HTTPS secure protocol with prior key exchange. During a network attack, the reverse-proxy covers the web-servers by itself and is able to withstand more sophisticated attacks due to the lightweight nature and target specialization of the code. In case of a global outage of all N modules of the LMS instrument, for example as a result of an administrator error or during services hours of configuration or update of the whole system, the reverse-proxy can provide LMS users with cached static content. The Theory module static content can be cached and provided in such a way.

A central event log of the LMS system infrastructure must be collected and stored to empower decisions to execute attack prevention mechanisms, as well as for future security incident based forensics. Protection against potential attackers erasing or falsifying part of the logs after worker module capture is ensured by dedicating a separate server as a central point of log collection and storage from all infrastructure components (the schema on Fig. 2 defines only several logging and control channels for the sake of illustration simplicity). Manual analysis of the logs by an operator is a very time-consuming procedure with low speed of decision making, so an automatic search for correlations of log events and a prepared attack signature library should be considered. Log analysis is implemented as a separate server with the possibility to execute control commands on the configuration control module [11].

All of the mentioned components in the network security schema can be implemented as virtual machines (full virtualization [10]), and also all modules with no network level functions can be implemented as virtual containers. Actual implementation of the mentioned security and infrastructure control components may differ. As an example for the current research work, a set of ready instruments is proposed, focusing on open source and distinct features of the LMS system itself. The instruments were grouped in a table (see Table 2).

An example of security instruments implementation

Component	Implementation
Network security scanner	Suricata 1.4.4
VPN server	SoftEther 4.19
Reverse-proxy	Nginx 1.8.0
Global firewall	PfSense 2.2.6
Virtual switch	Open vSwitch 2.4.0
Central log server	Logstash 1.8 + Elasticsearch 2.1
Analytics server	OSSEC 2.8.3
Configuration control module	Ansible 1.9.4

Conclusion

The article considers the program implementation of intellectual support and security of LMS MAI CLASS.NET. We use special algorithms as computer-adaptive testing methods (CAT), which differs from similar learning management systems. Statistical information, that we can calculate using algorithms, can be imported into other LMS or university automation systems. Used LMS software security tools significantly improve the reliability of its work, which is confirmed by many years of experience using the system in the MAI.

References

1. Naumov A.V., Dzhumurat A.S., Inozemtsev A.O. [Distance Learning System for Mathematical Disciplines CLASS.NET]. *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Herald of Computer and Information Technologies], 2014, vol. 1, no. 10, pp. 36–40. (in Russian)
2. Kibzun A.I., Panarin S.I. Generation of Integral Rating by Statistical Processing of the Test Results. *Automation and Remote Control*, 2012, no. 6, pp. 1029–1045. DOI: 10.1134/S0005117912060082
3. Kibzun A.I., Inozemtsev A.O. [Using the Maximum Likelihood Method to Estimate Test Complexity Levels. *Automation and Remote Control*, 2014, no. 4, pp. 607–621. DOI: 10.1134/S000511791404002X
4. Naumov A.V., Inozemtsev A.O. [The Algorithm of Formation of Individual Tasks in Distance Learning Systems]. *Vestnik komp'yuternykh i informatsionnykh tekhnologii* [Herald of Computer and Information Technologies], 2013, vol. 1, no. 6, pp. 35–42. (in Russian)
5. Rasch G. *Probabilistic Models for Some Intelligence and Attainment Tests*. The University of Chicago Press, 1980.
6. Van der Linden W.J. *Conceptual Issues in Response-Time Modeling*. Law School Admission Council, 2008.
7. Naumov A.V., Mkhitaryan G.A. On the Problem of Probabilistic Optimization for Tests within the Time-Limit. *Automation and Remote Control*, 2016, vol. 77, no. 9, pp. 1612–1621. DOI: 10.1134/S0005117916090083

8. Naumov A.V., Say Khin Aung Tint. [On the Adaptation of Learning Systems for the Retraining of Young Specialists at the Enterprises of Aviation Complex]. *Elektronnyy zhurnal "Trudy MAI"*, 2011, issue 42, pp. 230–236. (in Russian)
9. Dzhumurat A.S. [Mathematical Methods of Adaptation of LMS CLASS.NET] *Nauka i obrazovanie v sovremennoy konkurentnoy srede: materialy Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Science and Education in Today's Competitive Environment: Materials of the International Scientific-Practical Conference]. Ufa, 2014, vol. 2, no. 6, pp. 211–215.
10. Rybalko A.A. [Virtualization as the Foundation of Computer Security Systems of New Generation]. *Vestnik Moskovskogo Aviatsionnogo Instituta*, 2009, vol. 16, no. 2, pp. 12–18. (in Russian)
11. Rybalko A.A. [Modelling Cloud Protection with Virtualization Mechanisms] *Vestnik Moskovskogo Aviatsionnogo Instituta*, 2010, vol. 16, no. 6, pp. 143–149. (in Russian)

Received April 1, 2016

УДК 004.416.6

DOI: 10.14529/mmp160412

ПРОГРАММНЫЙ КОМПЛЕКС ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ И ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ФУНКЦИОНИРОВАНИЯ СДО МАИ CLASS.NET

А.В. Наумов, Г.А. Мхитарян, А.А. Рыбалко

В работе рассматриваются особенности программной реализации и интеграции в управляющую оболочку системы дистанционного обучения (СДО) комплекса математических методов, обеспечивающих статистический анализ работы пользователей системы дистанционного обучения математическим дисциплинам и адаптацию контента этой системы под изменяющийся контингент пользователей. Программный комплекс позволяет автоматически вычислять сложности заданий, рейтинги пользователей СДО, используя статистические данные, выдавать сигналы администратору системы о возможной компрометации пользователями заданий, т.е. использования ими несанкционированных подсказок или помощи других пользователей. Автоматизируется также процедура формирования контента для проведения различных тестов и контрольных мероприятий с ограничением и без ограничения на время выполнения теста. При разработке указанного комплекса математических методов использованы две вероятностные модели: модель Раша для описания вероятности ответа пользователя на задание и модель Ван дер Линдена для описания времени ответа пользователя на задание. Программный комплекс содержит специальные оптимизационные процедуры для оценки параметров указанных моделей на основе накопленной статистики о работе пользователей. Кроме того, в работе предложены решения задач повышения эффективности работы программной реализации указанного комплекса методов и обеспечения безопасности функционирования СДО. Рассмотрена реализованная архитектура системы безопасности, используемые при ее разработке технологии и возможный вариант реализации приложений обеспечения контуров безопасности.

Ключевые слова: система дистанционного обучения; статистический анализ; адаптивные свойства, безопасность системы, контейнерные технологии.

Работа проводилась при финансовой поддержке гранта РФФИ №15-07-02914-а.

Литература

1. Наумов, А.В. Система дистанционного обучения математическим дисциплинам CLASS.NET / А.В. Наумов, А.С. Джумурат, А.О. Иноземцев // Вестн. компьют. и информ. технологий. – 2014. – Т. 1, № 10. – С. 36–40.
2. Кибзун, А.И. Формирование интегрального рейтинга с помощью статистической обработки результатов тестов / А.И. Кибзун, С.И. Панарин // Автоматика и телемеханика. – 2012. – Т. 1, № 6. – С. 119–139.
3. Кибзун, А.И. Оценивание уровней сложности тестов на основе метода максимального правдоподобия / А.И. Кибзун, А.О. Иноземцев // Автоматика и телемеханика. – 2014. – Т. 1, № 4. – С. 20–37.
4. Наумов, А.В. Алгоритм формирования индивидуальных заданий в системах дистанционного обучения / А.В. Наумов, А.О. Иноземцев // Вестник компьютерных и информационных технологий. – 2013. – Т. 1, № 6. – С. 35–42.
5. Rasch, G. Probabilistic Models for Some Intelligence and Attainment Tests / G. Rasch. – The University of Chicago Press, 1980.
6. Van der Linden, W.J. Conceptual Issues in Response-Time Modeling / Wim J. van der Linden. – Law School Admission Council, 2008.
7. Наумов, А.В. О задаче вероятностной оптимизации для ограниченного по времени тестирования / А.В. Наумов, Г.А. Мхитарян // Автоматика и телемеханика. – 2016. – № 9. – С. 124–135.
8. Наумов, А.В. Об адаптации обучающих систем переподготовки молодых специалистов на предприятиях авиационного комплекса / А.В. Наумов, Сай Кхин Аунг Тинт // Электронный журнал «Труды МАИ». – 2011. – Т. 1, № 42. – С. 230–236.
9. Джумурат, А.С. Математические методы адаптации СДО CLASS.NET / А.С. Джумурат // Наука и образование в современной конкурентной среде: материалы Междунар. науч.-практ. конференции. – 2014. – Т. 2, № 6. – С. 211–215.
10. Рыбалко, А.А. Виртуализация как основа систем компьютерной безопасности нового поколения / А.А. Рыбалко // Вестник московского авиационного института. – 2009. – Т. 16, № 2. – С. 12–18.
11. Рыбалко, А.А. Моделирование системы защиты облачных сервисов с использованием механизмов виртуализации / А.А. Рыбалко // Вестник московского авиационного института. – 2010. – Т. 16, № 6. – С. 143–149.

Андрей Викторович Наумов, доктор физико-математических наук, профессор, кафедра «Теория вероятностей», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), naumovav@mail.ru.

Георгий Араикович Мхитарян, студент магистратуры, кафедра «Теория вероятностей», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), grgmkn@mail.ru.

Алексей Алексеевич Рыбалко, аспирант, кафедра «Вычислительная математика и программирование», Московский авиационный институт (национальный исследовательский университет) (г. Москва, Российская Федерация), aagern@mail.ru.

Поступила в редакцию 1 апреля 2016 г.