A simulation model for educational process planning in an institution of higher education

Alexander A. Gorbunov
Lecturer, Department of Information Systems and Digital Infrastructure Management
National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow, 101000, Russian Federation
E-mail: agorbunov@hse.ru

Eugeni A. Isaev
Professor, Head of Department of Information Systems and Digital Infrastructure Management
National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow, 101000, Russian Federation
E-mail: eisaev@hse.ru

Alexander F. Morgunov
Associate Professor, Head of Department of Information Systems and Digital Infrastructure Management
National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow, 101000, Russian Federation
E-mail: amorgunov@hse.ru

Abstract

In universities and technical colleges with relevant IT qualifications in one semester multiple streams, courses and specializations can use software products for training purposes. IT services of universities should deal with the challenge of creating the infrastructure of educational applications that can support the educational process. We note that the number of specializations which study information technology are growing every year (for example, in HSE there are disciplines-minors, which can enroll students coming from any field). Also in the recent years, online courses have started to become popular. If the load is not planned ahead taking into account future trends, the power of even the most high-tech infrastructure will be insufficient. Calculation of the corresponding load on the infrastructure must be made in the planning process of the disciplines, so that we can reserve appropriate facilities, and thus organize an effective learning process.

Software developers use a variety of benchmarking tools that are complex and do not provide the necessary information for the participants of educational process planning.

This article discusses the construction of a simulation model that supports the educational process planning. The simulation is carried out using the capabilities of the tool AnyLogic 7. The aim of this work is to develop a simulation model designed to estimate the load on the information system used in the educational process. In addition, besides the description of the model, the article presents the results of calculations used for various options of the information system (private cloud or on a server at the university). The simulation results were confirmed by data obtained during practical classes at the university. This model gives us the opportunity to plan the educational process in order to achieve uniformity of the load on the services. If necessary, the model allows us to make a decision about the location of the educational information system: on servers of the university or in a private cloud.

Key words: educational process, information system, ERP system, simulation model, private cloud, IT infrastructure.

Citation: Gorbunov A.A., Isaev E.A., Morgunov A.F. (2017) A simulation model for educational process planning in an institution of higher education. *Business Informatics*, no. 2 (40), pp. 57–67.
DOI: 10.17323/1998-0663.2017.2.57.67.

Introduction

Simulation [1–3] is widely used in various fields of science and industry. In scientific publications we see the results of the simulation of complex economic [4], socio-economic [5] and technological systems [6].

Models of technical systems, in particular, allow us to estimate the load on the server hardware. Similar problems arise, for example, in the implementation and operation of corporate information systems. In higher educational institutions, for the training of students to work with corporate information systems one can use the resources of private cloud or cloud server infrastructure. For effective practical training, staff involved in the planning of the educational process must have information about the load on its infrastructure. Such assessment is particularly large at the stage of educational planning and curriculum development. Uniform distribution of load on the equipment during the school year helps to avoid disruptions when working and, as a result, affect the efficiency of study time.

1. Formulation of the problem

In *Figure 1*, we see the planning process of disciplines for which teaching requires equipped computer classes and access to specialized software types – ERP, CPM and CRM.

For educational institutions, leading Russian and foreign vendors of software provide facilities for the use of software products from a private cloud that eliminates the need to install, purchase licenses, create infrastructure, access, etc. The capability is limited and, as practice shows (*Table 1*), does not always provide stable, productive work while training a large number of students in one period.

In universities and technical colleges with relevant IT qualifications in one semester multiple streams, courses and specializations can be adapted to work with software products. IT services of universities should deal with the challenge of creating the infrastructure of educational applications that can support the educational process. We note that the number of specializations which study information technology are growing every year (for example, in HSE there are disciplines-minors, which can

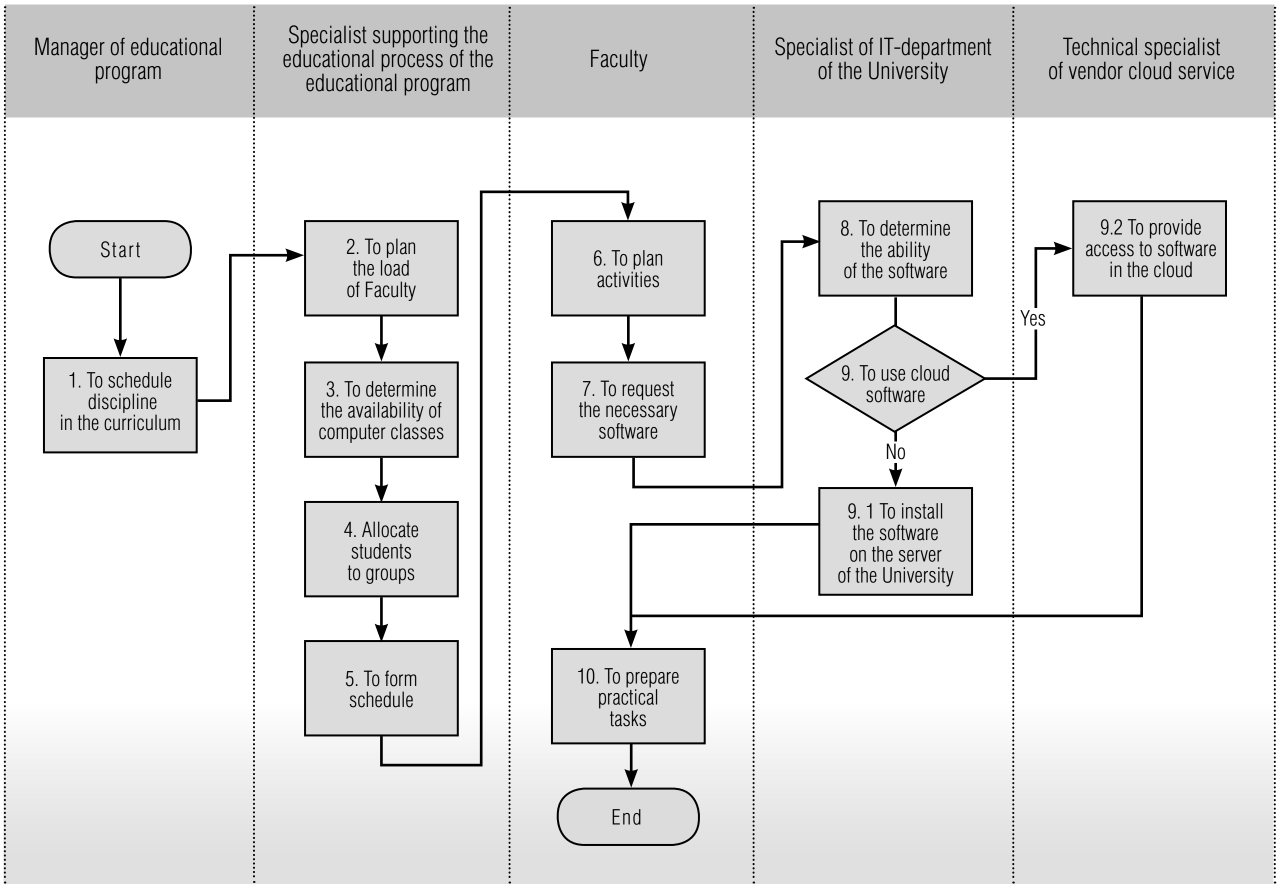

Fig. 1. The planning process of disciplines

Table 1.

The observed performance of cloud services

Service provider (vendor)	Software	The number of trainees in the semester	Time of the service unavailability (days)	Time of slow service (days)
SAP	SAP ERP	400	5	10
1C	1C:Manufacturing Enterprise Management (ERP Solution)		15	36

enroll students coming from any field). Also in the recent years, online courses have started to become popular. If the load is not planned to take into account future trends, the power of even the most high-tech infrastructure will be insufficient. Calculation of the corresponding load on the infrastructure is necessary to perform the planning process of the disciplines. This allows us to reserve appropriate facilities, and thus to organize an effective learning process.

Software developers use a variety of benchmarking tools that are complex and do not provide the necessary information to participants in planning the educational process.

The aim of this work is to develop a simulation model designed to estimate the load on the information system used in the educational process.

To achieve this objective, the following tasks are solved:

- ✦ the definition of simulation parameters for simulations which will provide the information necessary for participants of the process of planning the educational process;
- ✦ the development of a mathematical model and its implementation in AnyLogic 7;
- ✦ experimental design and comparison of results with real observations.

2. Assumptions of the model

The simulation model relies on several assumptions which are proved practically throughout the period of use of the ERP system.

Assumption 1. Practical classes attended by students who pass the completed tasks at a different time. By the duration of tasks, students can be divided into groups as follows:

- ✧ executes the exercises in time:
- ✧ performs tasks beyond the classroom (or earlier);
- ✧ performs tasks in time (if the student has not completed the practice during the lessons, the task runs in extracurricular time up to the next class);
- ✧ performs the task with a delay:
- ✧ performs tasks one week after the deadline;
- ✧ performs tasks two weeks after the deadline;
- ✧ performs tasks for a few days before the exam at the end of the semester.

Assumption 2. We assume that students are ready for practice and the execution time of the task affects only the performance and availability of the service. The following service parameters:

- ✧ slower application performance (the majority of users of the system are waiting while performing operations):
- ✧ during class;
- ✧ in extracurricular time;
- ✧ application unavailability:
- ✧ during class;
- ✧ in extracurricular time.

Assumption 3. Executing time varies for practical tasks in the various service states. For example, the estimated execution time of the user under the following conditions: the service is available and running slow – 2 hours, service is available and running fast – 1.5 hours.

Assumption 4. The performance and availability of the service provided by the resources of the server.

Practical classes on “Corporate information systems” and “Information systems management” are taught at the Higher School of Economics (HSE) between September to December 2016. At this period of training with the use of a cloud service, the tasks were carried out by 400 undergraduate students of senior courses.

The students were divided as follows:

- ✧ 240 4th year students of “Management” carried out practical classes in groups with a maximum number of 15 people – 16 subgroups;
- ✧ 160 3rd year students of various disciplines in the framework of university discipline (minor) performed work in groups, with a maximum of 32 people – 5 subgroups.

It should also be noted that the students had no previous knowledge of corporate systems and had no practical skills to perform practical tasks based on experience. However, they had received the necessary information from the teacher at the beginning of the class. This clarification eliminated the connection between previous skills of work and completing practical tasks on time.

Assumption 5. If the service is available and the application works well or slowly, the student can perform the task during class time or perform it in extracurricular time.

Assumption 6. In the organization of ERP systems in the university infrastructure, the resources of a single server will be used to support the performance and availability of the two applications.

For modelling purposes, there are several practical scenarios to be performed by groups of students on ERP systems of different vendors (or one vendor); the systems use the resources of a single server. Thus, it is possible to imagine a model as a queueing system [7, 8] in which there are three sources (students performing scenarios), the request queue from two information systems and the one server that will handle the requests.

As instrumentation for modelling using the functionality of AnyLogic 7 [4].

3. Features and key design parameters of the model

3.1. Calculation of the values of various time parameters by individual characteristics of students

Students attend classes on a schedule; the number of those who come to classes is displayed at blocks *Main_Dis, Sub_Dis, Second_Dis.* Those students who arrive enter in the queue which simulates students waiting while the lecturer explains the instructions for executing tasks, gives the theoretical basis and so forth. (block *wait1*, *wait2* or *wait3*) (*Figure 2*).

The length of time the teacher explains the material τ_0 is a random value distributed in the range of 10–30 minutes, depending on the rank $r > 0$ (complexity of challenge):

$$\tau_0 = U(10{,}30)\cdot\frac{r}{2}, \qquad (1)$$

where $U(10,30)$ is a function of the uniform distribution with parameters 10, 30.

After the teacher's explanation, students begin practical training with a certain time delay τ_i'', $i \in \{1,\ldots,n\}$, which depends on the speed of task execution, which is individual for each student (Formula 3).

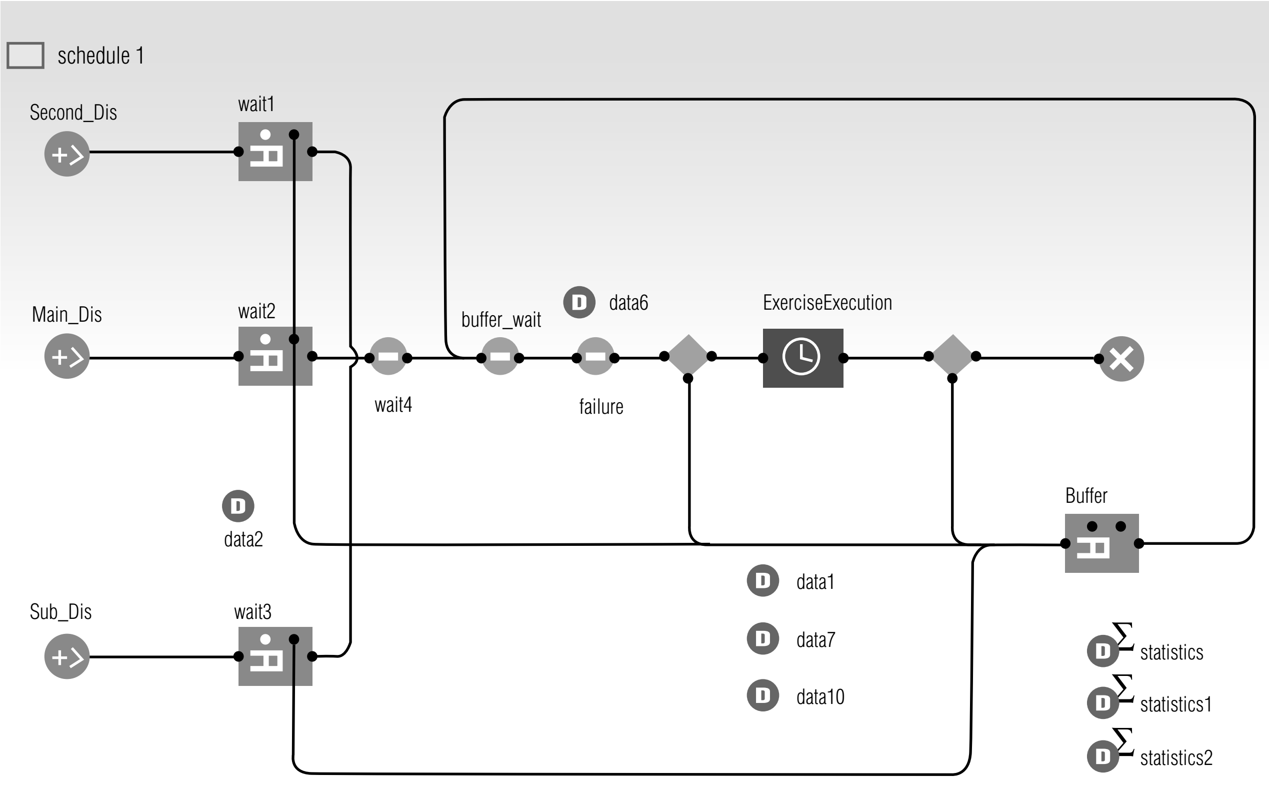

Fig. 2. The logical scheme of the simulation model

The execution speed of the task (k_i) indicates the percentage of lesson time that the i-th student is busy executing it. Let the variable d time of class, then the following formula is fair:

✦ to compute the time during which the i-th student busy completing tasks in class:

$$\tau_i' = k_i d - \tau_0 \tag{2}$$

✦ to calculate time delay before starting the execution of the task by the i-th student:

$$\tau_i'' = (1 - k_i) \cdot d. \tag{3}$$

Time of class can be calculated as the sum of the time parameters:

$$d = \tau_0 + \tau_i' + \tau_i''. \tag{4}$$

When a student enrolls in one of the blocks (*Main_Dis*, *Sub_Dis*, *Second_Dis*), a value of execution speed is assigned which is calculated in the following way: choose a random uniformly distributed number from the interval [0, 1), which is multiplied by 100.

The smaller the value k_i, the less time the student is busy performing practical tasks τ_i' and the more time τ_i'' is delayed before starting the task.

Example 1. Let students solve the task, the complexity (rank) $r = 2$. According to Formula 1, calculate the time explaining the task by the teacher $\tau_0 = 30$ min. Calculate parameters τ_i', τ_i'' of the model for two students ($i = 1, 2$). Assume that the speed of execution of tasks of the first and second students $k_1 = 0.85$, $k_2 = 0.4$. Then, after the class time d (2 hours 40 minutes) the first student may be busy performing tasks within the time

$\tau_1' = 0{,}85 \cdot$ (2 h. 40 min.) – 30 min = 1 h. 46 min.,

and the second student during the time

$\tau_2' = 0{,}4 \cdot$ (2 h. 40 min.) – 30 min. = 34 min. (Formula 2). Thus the time delay before starting the job for the first student is τ_1''= 24 minutes, and for the second one –

τ_1'' = 1 hour 36 minutes (Formula 3).

The result of simulation experiments to establish the number of students able to perform the practical task of rank r for the class and those who require out-of-class time for self-study work, the following notation is introduced:

t_{j1}, t_{j2} – start and end time of the j-th class, $j \in \{1,..., m\}$, where m is the number of classes;

t_{ij} – the time of the completion of the j-th job by the i-th student ($t_{ij} \geq 0$);

z – calculated (reference) execution time of the task (constant, determined by the teacher);

$g_i\left(t_{ij}, \tau_i'\right)$ – the probability that the i-th student is potentially able to perform the j-th task:

$$g_i\left(t_{ij}, \tau_i'\right) = \begin{cases} 1, \text{if } z \leq \tau_i' \text{and } \ t_{ij} \leq t_{j2} \\ 0, \text{if } z > \tau_i' \text{ and } \left(z \leq \tau_i' \text{ and } t_i > t_{j2}\right). \end{cases} \tag{5}$$

Example 2. Based on the calculations given in example 1, determine which of the two students is able to perform the first ($j = 1$) task whose rank is $r = 2$. Let the class begin at the time t_{j1} = 12 hours 10 minutes, end in a moment t_{j2} = 15 hours 00 minutes, while the estimated execution time of the task defined by the teacher, is z = 1 hour 40 minutes. The first student completes the task at the moment t_{11} = 14 hours 50 minutes. Applying Formula 5, we get: $g_1(t_{11}, \tau_1')$ =1 (1 h. 40 min. ≤1 h. 46 min.; 14 h. 50 min. ≤ 15 h. 00 min.). For the second student, the first parameter does not matter, since z = 1 h. 40 min. > τ_2' = 34 min., $g_2(t_{21}, \tau_2') = 0$. Thus, of the two students, only the first will be able to perform the task in class, and the second will require extracurricular time.

Students who require time for extracurricular activities ($g_i(t_i, \tau_i') = 0$, the block *Buffer*), can enter the system to finish exercises not completed in the normal time of classes d. Login (block *wait_buffer*) extracurricular time occurs through certain time intervals that are distributed exponentially. The number of logons (number of hits in the buffer) of the i-th student to complete practical tasks is determined by the formula (square brackets mean that only integral part is taken):

$$c = \left[\frac{z}{\tau_i'} - 1\right] \cdot 100. \tag{6}$$

Student i completes a practical assignment and is removed from the buffer after he enters the system c times.

Example 3. In example 2, it is shown that one of the students will require extracurricular time to complete job 1. In order to determine how many times the student will be able to define the extracurricular time, calculate the c parameter. As the value of the parameter used, we take the value calculated in example 1. Perform the calculation:

$$c = \left[\frac{1 \text{ h.}40 \text{ min.}}{34 \text{ min.}} - 1\right] \cdot 100 = 100.$$

3.2. Calculation of influence of indicators of reliability of the computing infrastructure of the cloud service educational ERP system at various time parameters

The real computational infrastructure that supports the educational ERP system cloud service is opaque, and this presents difficulties for developing a simulation model. Among the scientific works dedicated to the topic of computing system architectures we highlighted the publications of V.G. Khoroshevsky [6], which presents the method of calculation of indicators of reliability and feasibility of the solution of tasks using computer systems. Assume that the computing infrastructure of the cloud service corresponds to the architecture of the computing system with structural redundancy [6].

This computer system looks to the user as a virtual system having the number of elementary machines [6], which power allows us to carry out the implementation tasks, appropriate ranks (tasks of appropriate difficulty). Use the techniques of [6], assuming that the rank of a task is the number of elementary machines that balancing the load while practice is executing with the maximum number of users.

The essence of this assumption is the following: if there is a possibility that practical exercises (tasks) are executed simultaneously by multiple users (students), it means that some elementary machines are required for the implementation of the solution of the task. The number of elementary machines (EM) equals the rank of the task. Theoretically, it should always avoid a situation of significant load (for hours, days and so forth) on computing infrastructure as follows from observations of the delivery of practical tasks within the discipline providing for the execution of works with the use of the software.

Let R(t), U(t) – is a function respectively of reliability and recovery of a system with structural redundancy. The mean uptime and mean time to repair computing system (CS) can be calculated by the formula [6]:

$$\theta = \int_0^\infty \mathrm{R}(t)\,dt \qquad (7)$$

$$T = \int_0^\infty t\,d\mathrm{U}(t). \qquad (8)$$

If average uptime decreases, then τ_i'' (average delay time before the start) increases, and τ_i'(time of occupation of the student in class) is reduced. Also, for a given it is true that the average number of students who will perform tasks during the class will decrease, while the average number of students who require out-of-class time will increase. Thus, when the average uptime is reduced, the buffer will be on average higher than when operating in failover mode.

3.3. Determination of the influence of the intensity of the task flow at different time parameters

Suppose that the computational system of cloud service is running in the maintenance mode task flow. Then the simplification of the General case can be considered [6]. In the general case, there are tasks of different ranks $1 \le r \le N$, where N – the number of elementary machines (EM) in the computational network (CN). One or more subsystems are allocated for each rank to load balancing of EM, the number of which is equal to the corresponding rank. To determine the feasibility of the task solution, we must calculate the following parameters: mean of the number of tasks in the system and the number of occupated EM [6, 9].

Let's consider two cases:

✦ case 1. The task flow has a weak intensity. The case corresponds to a situation in which there are a few groups of students, and the stream of parallel sessions is not present. This case is simple, so it does not require careful consideration and conducting simulation experiments;

✦ case 2. Task flow received on CN has a high intensity. For example, groups of students have greater numbers, while classes are held in several concurrent streams, and there is a probability of the maximum load on a computer system.

Case 2 is the subject of experiments in the simulation model.

4. Experimental design

For the solution of practical problems in the assessment of the load on equipment and software when carrying out the practice, you must calculate and compare the indicators, carrying out the discipline under different conditions.

There has been designed a series of experiments which allow outputting under the following options of the organization of the discipline:

✧ experiment 1: These are a cloud-based service for one discipline with one stream of students;

✧ experiment 2: it is infrastructure for a single discipline and two streams of students;

✧ experiment 3: it is infrastructure for two disciplines (advanced experiment 2).

4.1. Experiment 1. Practical exercises are performed using the ERP system in the cloud

Conditions:

- the first week – 2 EM works that allow loading balance while simultaneously running 30 users on the lesson, the students perform the task with a rank of 2;
- the second week – the service has been work slowly, the number of EM decreases to 1, students perform tasks with a rank of 1;
- third and fourth weeks – the service is unavailable the first half of the total time, the second half – one EM runs, students perform the task with rank 1.

4.2. Experiment 2. The practical exercises are performed using the ERP system supported by the server infrastructure of the university

Conditions:

- server infrastructure supports two EPR systems of different manufacturers used to carry out practical tasks in two parallel groups of students (discipline 1) at the same time; the number of EM in infrastructure is equal to 2; it supports 30 users.
- the maximum rank of the tasks that can be performed by students – 1; the maximum number of students per group should not exceed 15.

4.3. Experiment 3. An extended version of Experiment 2

Conditions:

- conditions of the 2nd experiment;
- discipline 2 is taught at the university and students perform practical exercises using the ERP system supported by the server infrastructure of the university;
- days of discipline's classes does not coincide with the days of classes 1;
- students were learning discipline 2 and had not fulfilled the task during the basic training; they can come in to finish the job in extracurricular time;
- time of entrance of students into the system of extracurricular time is random.

In the process of conducting each experiment was performed ten runs, the averages presented in *Table 2.*

4.4. Analysis of experimental results

The indicators "average delay before starting the task", "residual in buffer", "the number of failed exercises" in the experiments 2 and 3 (*Table 2*) are considerably lower compared to experiment 1. In addition, you should also note that experiments 2 and 3 conducted for two and three simultaneous streams, and the number of assignments at the end of period per stream on average were almost 1/3 greater than the value of the same indicator for experiment 1 (*Table 2*).

Thus, the discipline carried out with the use of a cloud service ERP system is inferior by most measures to the options used the infrastructure to support the educational applications.

5. Comparing modelling results and the actual observations

Accumulated values of the indicators of *Table 1* were obtained as a consequence of the simulation experiment 1. *Figure 3* presents the results of one run, where NumStud1 – number of tasks executed in the class; NumStud2 – number of tasks executed before the deadline; NumStud3 – number of tasks executed with 1 week delay; NumStud4 – number of tasks executed with 2 weeks delays; NumStud5 – number of tasks that were not executed.

The conditions of the experiment 1 correspond to conditions in which there were practical lessons on the discipline "Corporate information systems" in the first semester. To determine where the data were obtained in the simulation result was reliable. Experiments were carried out to verify via actual observations. We made ten runs in experiment 1; the obtained average values for the totality runs for one of the simulated indicators (NumStud1), and the actual observations are shown in *Figure 4.* The comparison shows that the discrepancy between the results of numerical experiments and the actual data does not exceed 6.5%.

Conclusion

This article has demonstrated an approach which is based on the use of simulation models for evaluating the load of the software which is being used during practical classes at the university. The problem of estimating the burden on the software is relevant for specialists involved in planning various aspects of academic disciplines using "heavy" software, for example, teachers, managers, curriculum specialists. Existing approaches are designed to measure the load at the beginning of the introduction or optimization of the already existing information systems. These are complex and at the same time do not provide the necessary information.

Table 2.

The experiment results

Indicator	Number of experiment		
	1	2	3
1. Average indicators for the modelling period from 22.09.16 12:10 to 15.12.16 15:00			
Speed of task execution (general)	0.605	0.597	0.592
Estimated time of student's occupation in class (hours)	1.36	0.96	0.95
Delay before starting the task (hours)	3.21	0.48	0.52
Task's explanation by the teacher (hours)	0.5	0.5	0.5
Number of tasks executing in extracurricular time (buffer)	32	12	30
Length of time between re-logins (minutes)	49	19	18
Number of system calls in extracurricular time	3	2	2
Uptime of the system (mean time to failure) (days)	12	30	30
Time during which the task is done in class (hours)	1.4	1.4	1.4
Duration of the execution of the task beyond class hours, in time before the deadline (days)	0.98	1.37	1.47
Length of time from the end of classes before the deadline (days)	2	2	2
Duration of the task's execution in extracurricular time with a delay of 1 week (days)	4.25	3.02	3.79
Duration of the task's execution in extracurricular time with a delay of 2 weeks (days)	11.5	0	0
2. Accumulated indicators for the modelling period from 22.09.16 12:10 to 15.12.16 15:00			
Number of tasks carried out in class	133	321 (160 tasks per stream)	467 (155 tasks per stream)
Number of tasks completed before deadline	48	291 (145 tasks per stream)	210 (70 tasks per stream)
Number of tasks that were completed 1 week late	57	172 (86 tasks per stream)	480 (160 tasks per stream)
Number of tasks that were completed 2 weeks late	63	0	0
Number of failed tasks	23	35 (17 tasks per stream)	68 (22 tasks per stream)
3. Residual indicators at the end of modelling period 15.12.16 15:00			
Tasks to be done in extracurricular time (buffer)	141	111 (55 tasks per stream)	170 (56 tasks per stream)
Completed tasks	301	784 (392 tasks per stream)	1157 (385 tasks per stream)

The model was developed based on an analysis of the results of using the cloud services of ERP applications in the educational process. It relies on various assumptions related to the influence of the infrastructure parameters (the server) on the duration of practical scenarios in ERP application. In fact, it can be used when planning preparation of lesson plans, including the calculation of the volume of practical tasks. In addition to this, the model can be useful while defining the term in which it is possible to plan the teaching discipline along with calculating duration, the load of the teacher, the maximum number of students in the group, defining the hosting infrastructure (private server, cloud service) and so forth. ■

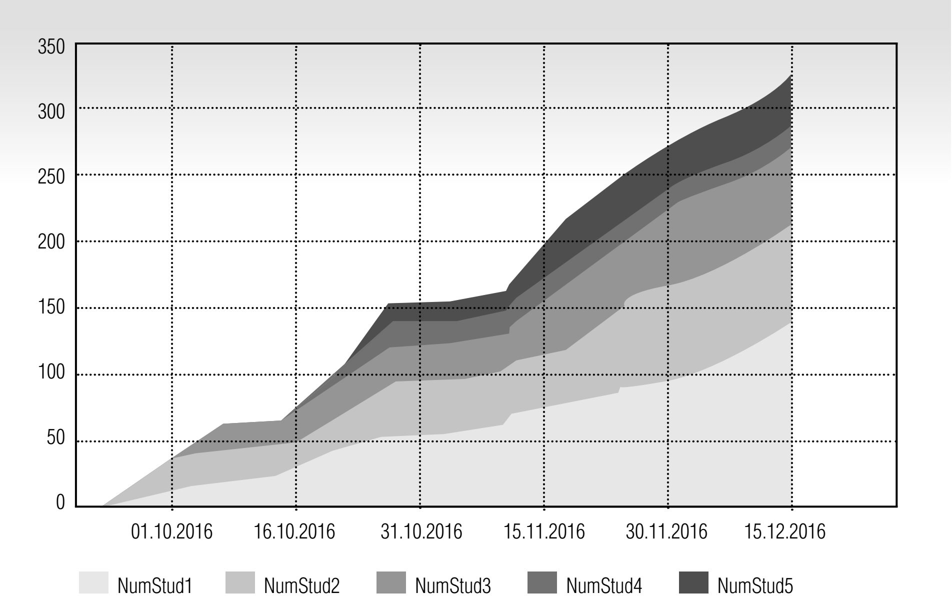

Fig. 3. The result of simulation experiment 1

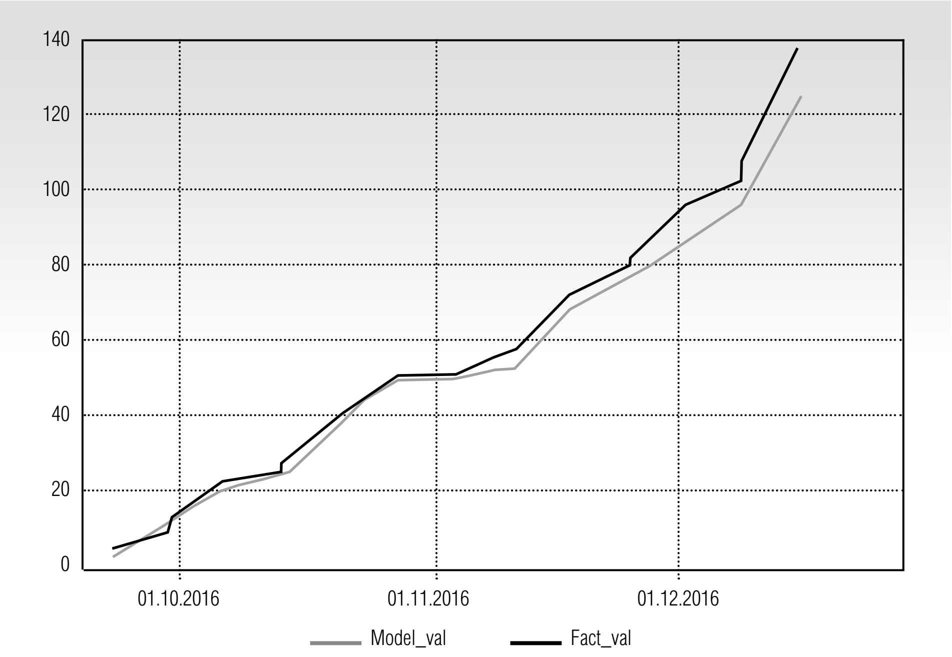

Fig. 4. Comparison of the results of experiment 1 and the actual observations

References

1. Akopov A.S. (2014) *Imitatsionnoe modelirovanie* [Simulation modeling]. Moscow: Urait (in Russian).
2. Lychkina N.N. (2014) *Imitatsionnoe modelirovanie ekonomicheskikh protsessov* [Simulation modeling of economic processes]. Moscow: INFRA-M (in Russian).
3. Khoroshevsky V.G. (2008) *Arkhitektura vychislitel'nykh sistem* [Architecture of computer systems]. Moscow: BMSTU (in Russian).
4. Aliev T.I. (2009) *Osnovy modelirovaniya diskretnykh sistem* [Basics of discrete systems modeling]. Saint Petersburg: ITMO (in Russian).
5. Pavsky V.A., Pavsky K.V. (2008) Otsenka pokazateley osushchestvimosti resheniya zadach na raspredelennykh vychislitel'nykh sistemakh [Evaluating feasibility of tasks solution on distributed computer systems]. *Tomsk State University Journal of Control and Computer Science*, no. 4 (5), pp. 61–68 (in Russian).

6. Saaty T.L. (2010) *Elementy teorii massovogo obsluzhivaniya i ee prilozheniya* [Elements of queueing theory with application]. Moscow: Librokom (in Russian).
7. Borshhev A.V. (2005) *Prakticheskoe agentnoe modelirovanie i ego mesto v arsenale analitika* [Practical agent based modeling and its place in an analyst's arsenal]. Available at: http://www.gpss.ru/immod05/p/borshev/print.html (accessed 18 March 2017) (in Russian).
8. Pavlovsky Y.N., Belotelov N.V., Brodsky Y.I. (2008) *Imitatsionnoe modelirovanie* [Simulation modeling]. Moscow: Academy (in Russian).
9. Sidorenko V.N. Krasnoselsky A.V. (2009) Imitatsionnoe modelirovanie v nauke i biznese: podkhody, instrumenty, primenenie [Simulation modeling in science and business: approaches, instruments, application]. *Business Informatics*, no. 2 (08), pp. 52–57 (in Russian).

Имитационная модель поддержки планирования учебного процесса в высшем учебном заведении

А.А. Горбунов

преподаватель кафедры управления информационными системами и цифровой инфраструктурой
Национальный исследовательский университет «Высшая школа экономики»
Адрес: 101000, г. Москва, ул. Мясницкая, д. 20
E-mail: agorbunov@hse.ru

Е.А. Исаев

кандидат технических наук
профессор, заведующий кафедрой управления информационными системами и цифровой инфраструктурой
Национальный исследовательский университет «Высшая школа экономики»
Адрес: 101000, г. Москва, ул. Мясницкая, д. 20
E-mail: eisaev@hse.ru

А.Ф. Моргунов

кандидат технических наук
доцент кафедры управления информационными системами и цифровой инфраструктурой
Национальный исследовательский университет «Высшая школа экономики»
Адрес: 101000, г. Москва, ул. Мясницкая, д. 20
E-mail: amorgunov@hse.ru

Аннотация

В вузах и втузах с профильными ИТ-специальностями в одном семестре могут обучаться работе с программными продуктами сразу несколько потоков, курсов, специальностей. Поэтому перед ИТ-службами учебных заведений возникает задача создания инфраструктуры учебных приложений, которая сможет обеспечить поддержку учебного процесса. Следует учитывать, что число специальностей, на которых изучаются информационные технологии, с каждым годом растет (например, в НИУ ВШЭ преподаются дисциплины-майноры, куда может записаться студент с любой специальности). Также в последнее время популярностью стали пользоваться дистанционные курсы. Если не планировать нагрузку, то с учетом будущих трендов, мощности даже самой высокотехнологичной инфраструктуры будут недостаточны. Расчет соответствующей нагрузки на инфраструктуру необходимо производить в процессе планирования учебных дисциплин, что позволит выполнять резервирование соответствующих мощностей и тем самым организовать эффективный учебный процесс.

Разработчики программного обеспечения используют различные бенчмаркинговые инструменты, которые сложны и не предоставляют необходимой информации для участников планирования учебного процесса.

В статье рассматривается построение имитационной модели поддержки планирования учебного процесса. Моделирование осуществляется с использованием возможностей инструмента AnyLogic 7. Целью данной работы является разработка имитационной модели, предназначенной для оценки нагрузки на информационные системы, используемые в ходе учебного процесса. Помимо описания модели, в статье приведены результаты расчетов с ее использованием для различных вариантов размещения информационной системы (в частном облаке или на сервере в университете). Результаты моделирования подтверждены данными, полученными в ходе проведения практических занятий в вузе. Данная модель дает возможность планировать учебный процесс с целью добиться равномерности нагрузки на сервисы. В случае необходимости модель позволяет принять решение о месте размещения учебной информационной системы: на серверах университета или в частном облаке.

Ключевые слова: учебный процесс, информационная система, ERP-система, имитационная модель, частное облако, ИТ-инфраструктура.

Цитирование: Gorbunov A.A., Isaev E.A., Morgunov A.F. A simulation model for educational process planning in an institution of higher education // Business Informatics. 2017. No. 2 (40). P. 57–67.
DOI: 10.17323/1998-0663.2017.2.57.67.

Литература

1. Акопов А.С. Имитационное моделирование: Учебник и практикум для академического бакалавриата. М.: Юрайт, 2014.
2. Лычкина Н.Н. Имитационное моделирование экономических процессов: Учебное пособие. М.: ИНФРА-М, 2014.
3. Хорошевский В.Г. Архитектура вычислительных систем: Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2008.
4. Алиев Т.И. Основы моделирования дискретных систем. СПб: СПбГУ ИТМО, 2009.
5. Павский В.А., Павский К.В. Оценка показателей осуществимости решения задач на распределенных вычислительных системах // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2008. № 4 (5). С. 61–68.
6. Саати Т.Л. Элементы теории массового обслуживания и ее приложения. М.: Либроком, 2010.
7. Борщев А.В. Практическое агентное моделирование и его место в арсенале аналитика. [Электронный ресурс]: http://www.gpss.ru/immod05/p/borshev/print.html (дата обращения 18.03.2017).
8. Павловский Ю.Н., Белотелов Н.В., Бродский Ю.И. Имитационное моделирование. Учебное пособие для студентов высших учебных заведений. М.: Академия, 2008.
9. Сидоренко В.Н., Красносельский А.В. Имитационное моделирование в науке и бизнесе: подходы, инструменты, применение // Бизнес-информатика. 2009. № 2 (08). С. 52–57.