



## USING THE TECHNICAL EXPERIMENT IN THE COMPUTER SIMULATION TRAINING FOR PROSPECTING SOFTWARE ENGINEERS

Maryna Litvinova, Oleg Dudchenko, Oleksandr Shtanko, Svitlana Karpova

Department of Software Engineering, Physics and Mathematics, Kherson branch, Admiral Makarov National University of Shipbuilding, Ushakov Avenue, 44, Kherson, 73022, Ukraine

lmb965@gmail.com, kbnuos@gmail.com, sadmar954@gmail.com, sokarpova@gmail.com, <http://www.kb.nuos.edu.ua/>

### Paper history:

Received 11 October 2019

Received in revised form 18 December 2019

Accepted 17 February 2020

Available online 14 June 2020

### Keywords:

computer simulation package;  
training;

technical experiment;

models debugging;

software engineer.

**Abstract:** In present work, a new technology of the prospective software engineers training in computer simulation is described. The technology provides carrying out comparative analysis of opportunities, productivity, and the accuracy of the reproduction of different computer simulation packages (CSP) on the basis of direct performance of the technical experiment results. Training process includes the principal stages: carrying out of the independent technical experiment; its simulation using of various CSP; comparison of the result of the tested CSP to the results of the experiment; models of debugging; detection of advantages and shortcomings of each involved CSP. As an example, in Open Modelica and Mathcad packages analysis of simulation opportunities of a problem of the motion of the body thrown at an angle to the horizon is carried out. As a result, assessment of the efficiency of each CSP used for the solution of an objective is made. When training prospective software developers the offered technology is the basis for further development of the modern standard in the field of computer simulation.

Copyright © Research Institute for Intelligent Computer Systems, 2020.

All rights reserved.

## 1. INTRODUCTION

Currently there are more than ten popular computer simulation packages (CSP), each of them is developed in its own way, increasing the functionality demanded in practice [1, 2]. During their development the certain algorithms for solving scientific and technical tasks are implemented and can optimally satisfy the needs of users [3]. However, computer simulation training for both developers and qualified users remains difficult (often heuristic) long process.

The lack of CSP standards [4] leads to the fact that each development model environment is formed according to its own library of standard models. These are always unique collections which are of great interest for training in simulation, for research of packages' opportunities, productivity and accuracy of offered solution. At the same time comparison and converting of models from one environment to other often shows the intensive manual process which is difficult to formalize [5]. Moreover,

having even the same accurate written mathematical model in different CSP it is difficult to guarantee (in case of model is not trivial) that equivalent program implementation of this model is loaded into a package. Impossibility to read the source code CSP does not allow us to compare algorithms, to make analysis of different implementations efficiency. All this stuff is possible for indirect analyzing – by the overall performance of the software. As a result, the use of standardized models which are not completely corresponding to real installations can lead to omission of essential results of an experiment or to false results of its processing [6].

High cost of simulation packages including closed program source code created by developers of the simulation environment in terms of training does not give a chance to familiarize with the CSP inner pattern and those problems that the simulation tool must cope with. It is not always possible to see which representation the package works during

experiment with, which settings trigger particular solution method. Therefore, the choice of the optimal simulation environment during considering the specific objective by software (SW) engineer is defined in general according to the description of package opportunities from the developer.

Under production conditions this problem, undoubtedly, is fixed by comparison of results based on simulation to the data received from experimental installation. But more often the production experiment is conducted by one person and its simulation – by another. SW engineer deals only with an ideal object with a limited set of the parameters provided by the experimenter. He cannot evaluate a contribution of all essential factors which have an impact on model's work. At the same time the specialist who is responsible for the experiment is not always competent in providing data for assuring optimal process of simulation (is not familiar with simulation environment opportunities). As a result, the model of the user can lead to an incorrect mathematical system which cannot obviously be solved [7]. Therefore, the SW developer must be competent in order to provide proper technological experimental set up and to assess its results for comparison opportunity analysis, performance and accuracy of solution's reproduction in different simulation packages.

Comparative analysis of various CSPs is often used in practice. Such an analysis was implemented in [8] to answer the questions that are stumbled upon when using any specific simulation package (more than 50 different packages) in the market or in the academic arena. In the work [9] the main results and conclusions of an international benchmark study on the performance of computer simulation codes are presented (based on the prediction of parametric rolling of ships in waves). The comparative studies of various packages for modeling discrete events is presented in the research [10] (a literature review was conducted, in which 45 papers were considered relevant for this research). All the investigations described have two main purposes: identify CSP user problems and get information to improve CSP. However, the development of SW engineers' competence to the implementation of a comparative analysis of modeling packages based on a real technical (production) experiment has not been considered anywhere.

In this work we are offering approach to develop the prospecting SW engineers' competence based on providing of a technical experiment during computer simulation classes. Nowadays, digital support of educational process in physics, mathematics, electrical engineering

and other engineering subjects is methodically provided and widespread in Technical Universities [11-13]. At the same time the approach to competence development in computer simulation using technical experiment is not practically applied either in universities in Ukraine or in other countries [14-16].

The relevance of the study is confirmed by the data [17] on motivation in software development. 53 relevant documents were selected for data abstraction and analysis. As a result, it was found that understanding the relationship of the modeling package with the real object is an essential factor in the motivation of SW engineers.

## 2. EXPERIMENT'S DESCRIPTION

Usually during computer simulation training the student receives different tasks which he has to solve within some environment of simulation. We offer the inverted approach – investigation of one task in several simulation environments and comparison of their results based on the conducted technical experiment. At the beginning of work it is presumed that the student is familiar with the internal device of the package with automatically solved problems according to CSP. The process consists of the following stages:

1. Independent executing of the technical experiment;
2. Simulation of this experiment with the use of several CSPs;
3. A comparison of the results obtained using CSP with the results of the technical experiment, search of the reason of their possible discrepancies, debugging models;
4. A comparison analysis of the simulation results, detection of advantages and disadvantages of each involved CSP.

As an example, modification of the task which is the most widespread computing experiment (available at most libraries of the CSP standard models) – flight of the body thrown at an angle to the horizon is considered.

### 2.1 SETTING UP THE TECHNICAL EXPERIMENT

The corresponding installation for providing experiment in laboratory can be viewed in Fig. 1. It consists of: the gun (1) installed under an adjustable corner  $\theta$  to the horizon which has the ball bullet located in; the disk shaped rotary pendulum (2) with the mirror (3) located on its axis; the permanently attached laser which has  $\lambda=650$  wavelength nanometer (4) and the beam falling on the mirror; the rulers (5) with the MD-3

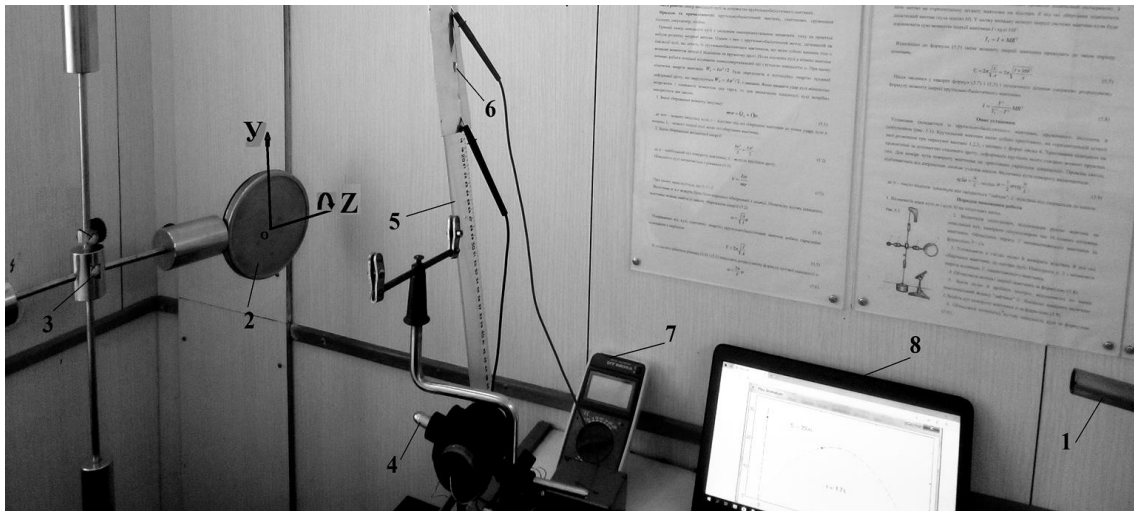


Figure 1 – The laboratory installation for providing the experiment: 1 – gun; 2 – rotary pendulum; 3 – mirror; 4 – laser; 5 – ruler; 6 – photodiodes; 7 – electric logger; 8 – computer

photodiodes (6) which are linearly fixed along the rulers' length; the electric logger (voltmeter) (7) connected to computer (8). Position of the pendulum disk center on installation is leveled with the initial position of the bullet in the gun trunk. Radius of the pendulum disk is 0.2 m. Degree of accuracy of gun's slope angle is  $\pm 0.7^\circ$ . Speed of the bullet is 12 m/s. Y-axis and Z-axis are seen in Fig. 1.

After the shot the bullet that has gone off at an angle  $\theta$  to the horizon gets to the pendulum placed at a distance of  $L=1.2\text{ m}$  from the point of its departure. As a result, the pendulum turns on the corner  $\beta$  regarding axis Z. Value  $\beta$  is directly proportional to distance from the spin axis to the point of the bullet hit ( $h$ ), i.e. to the position of the bullet along to Y-axis, towards to departure point.

As the result of the pendulum turn under the angle  $\beta$  reflected from the mirror, laser beam deviates along the ruler on distance  $d$ . The signal from the tape photodiodes located on beam deflection border having discretization of 1 mm is registered by the voltmeter. The background flare of photodiodes on the registered wavelength has significantly lower value than intensity of the reflected laser radiation and does not influence on experiment's measure of inaccuracy.

Electric potential value received from photodiodes shows which quantity from them was lit as a result of beam deflection and allows us to set the value of  $d$ . The value of  $h$  is determined based on the graphical proportion between  $\beta$ ,  $d$  and  $h$ .

## 2.2 THE CHOICE OF THE COMPUTER SIMULATION PACKAGES

For training simulation basics on practice, it is possible to use different CSPs. The most popular

products are Model Vision Studium, Mathcad, Matlab, Anylogic, OpenModelica, etc. Capabilities of the most demanded visual simulation packages aggregated according to a number of works [6, 18, 19] are given in Table 1.

Table 1. Availability of the opportunities in the visual simulation packages

	Model Sorting	Event Description	Differential Algebraic Equations Solver	Index Reduction	Physical Simulation (text and graphics) Online visualization	Structural-dynamic analysis	Frequency analysis
Matlab	-	-	+	-	-	+	+
Simulink	+	+	+	-	-	+	+
Matlab/Simulink	+	+	+	-	-	+	+
Simulink/Stateflow	+	+	+	-	-	+	+
Simulink/Simscape	+	+	+	+	+	-	+
Simulink/Simscape/Stateflow	+	+	+	+	+	-	+
Matlab/SL/SS/Stateflow	+	+	+	+	+	+	+
ACSL	+	+	+	-	-	+	+
Dymola	+	+	+	+	+	-	-
MathModelica	+	+	+	+	+	-	-
Misilab	+	+	+	-	+	+	-
OpenModelica	+	+	+	+	+	-	-
SimulationX	+	+	+	+	+	-	+
AnyLogic	+	+	+	-	-	+	-
Model Vision Studium	+	+	+	+	+	+	+
Scilab/Scicos	+	+	+	+	+	+	+

Majority of environments provide the convenient graphic interface which allows us to quickly create models. In case of training the simulation developer, it looks like an efficient method (it is enough to show which buttons of the interface should be pressed, train to work with wizards setup). If the learning material is presented more profoundly at a higher level in order to enable the student to understand in details features that the simulation language can have, then it is useful to enable him to describe models in a text starting from the lowest level. In such format OpenModelica package is able to provide training. At the same time, one should not ignore capabilities of classical mathematical simulation packages such as Mathcad. These both packages (OpenModelica and Mathcad) that have different principles of the simulation process organization are selected for contrastive analysis of the CSP' opportunities based on the technical experiment.

### 3. THE ANALYSIS OF SIMULATION OPPORTUNITIES OF THE TASK IN THE SELECTED CSP

#### 3.1. THE POSSIBILITIES OF THE MATHCAD PACKAGE

Bullet movement with the mass of  $m$  in the task illustrated in Fig. 1 is pulled apart two components – horizontal (the X-axis perpendicular to the disk plane) and vertical (Y-axis). As the result, we get and solve the system of differential equations (not including air resistance).

$$\frac{dx^2}{dt^2} \cdot m = 0; \quad \frac{dy^2}{dt^2} \cdot m = -mg;$$

$$v_{0x} = \left. \frac{dx}{dt} \right|_{x=0} = v_o \cos \theta; \quad (1)$$

$$v_{0y} = \left. \frac{dy}{dt} \right|_{y=0} = v_o \sin \theta.$$

In Fig. 2 the numerical solution of this task in the Mathcad package environment [20] through the using of the Solve block is shown:

1 – “Restrictions” area; 2 – “Solver” area.

In this block the user fills “Restriction” field with initial conditions (the bullet position and its speed at initial timepoint with values  $h_2$  and  $y_2$  for coordinates and  $m_2$  for its weight) and the differential equations system. “Solver” area should be filled with *odesolve* function that is built in Mathcad returning the user functions (coordinates of bodies in time) and describes the body movement

under gravity. Through simple conversions this problem can also be solved analytically (via symbols).

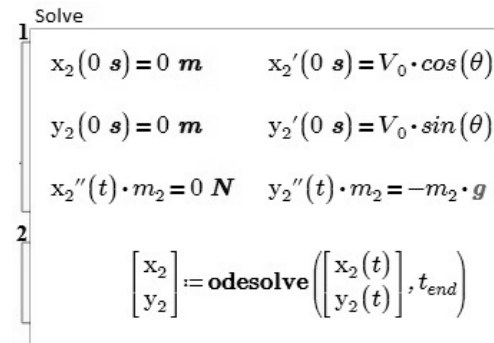


Figure 2 – The numerical solution to the task in the Mathcad package environment

The system described in Fig. 2 can be easily expanded by removing from the task a number of limitations. It is possible to consider the movement of a body taking into account air resistance, taking it, for example, proportional to the cross-sectional area of the bullet increased on environment specific density (air) and on squared speed. We can take into account the change in height of the gravity acceleration values, air density, etc. It is also possible to refuse a Cartesian coordinate system and switch to polar coordinates, i.e. to take into account the curvature of the Earth's surface but not the parallelity of gravity vectors. In Mathcad package there is also an opportunity to construct a trajectory of the bullet movement that is displayed in Fig. 3 in the diagram and to create animation (via the website).

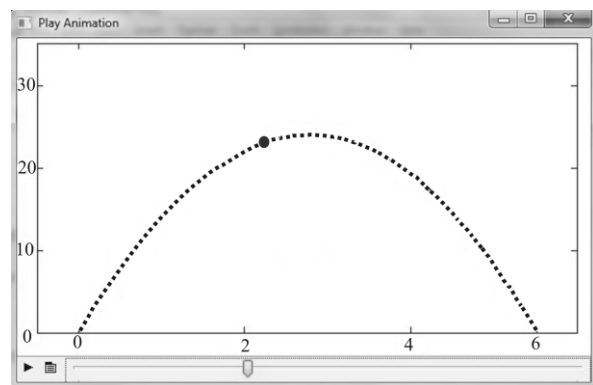


Figure 3 – Animation of the task in the Mathcad package environment ( $\theta=12^\circ$ ; units of measure on X-axis – meters, on Y-axis – centimeters ( $10^{-2} m$ ))

#### 3.2 CAPABILITIES OF THE OPENMODELICA PACKAGE

CSP OpenModelica uses the Modelica language [21] that currently is one of the most "advanced" languages for object-oriented

simulation. According to its capabilities it is coming closer to such computing environments as Matlab Simulink, Scilab-Xcos, having at the same time much more convenient concept of the studied block's system. The UML (Unified Modeling Language) is used for visual simulation. There is an opportunity of automatic generation of the program source code for the UML charts and inverse transformation (refactoring).

In the considered task it is possible to define dependence for value  $h$  at the bullet departure corner (in this system – bodies). The array of class copies is needed for conducting an experiment *MotionXY*:

```
Bodies: array [1..5] of MotionXY:= {for i in 1..5 | new
MotionXY(Teta0:=rad(5*i)}
```

In the structural scheme of the diagram this array will be represented as the multiobject described in Fig. 4.

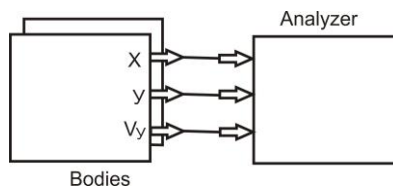


Figure 4 – The structural scheme of the diagram based on the task

A set of throwing angles can be defined in the *DTetas* parameter. Model of the bullet can be taken as condition *S1* as local activity with present value of the "*Teta\_0=rad (Dtetas [i])*" throwing angle. Unconditional transition from *S1* condition to the point of branching is performed after completion of the shell model behavior, i.e. after hit of a bullet. Flying range in *i-m* testing is remembered in *X[i]*. After enumerating of all values of a vector *DTetas*, required dependence (chart *X (DTetas)*) plus the test trajectories are received (presented in Fig. 5).

In the package different optional behavior cards are put that give an objective solution. The enumerating logic of search of throwing angles' set can also be implemented by means of looping statement in output discrete actions. Besides, two operation modes of the plot command are defined:

a) it is possible to build temporary diagrams (required unknown values are specified in comma separated form as *plot* command parameter, further they need to be reflected in the diagram);

b) it is possible to build phase charts (required unknown values are specified in comma separated form as *plot* command parameter that will be displayed along the vertical axis and as a separate operator – required variable; further phase charts will be built according to this variable).

Modular (automatic) and functional (user) testing are used for check classes operability in package.

Then requirements to a system are specified and new iteration is provided. There is an opportunity to perform a parametric optimization for calculation of

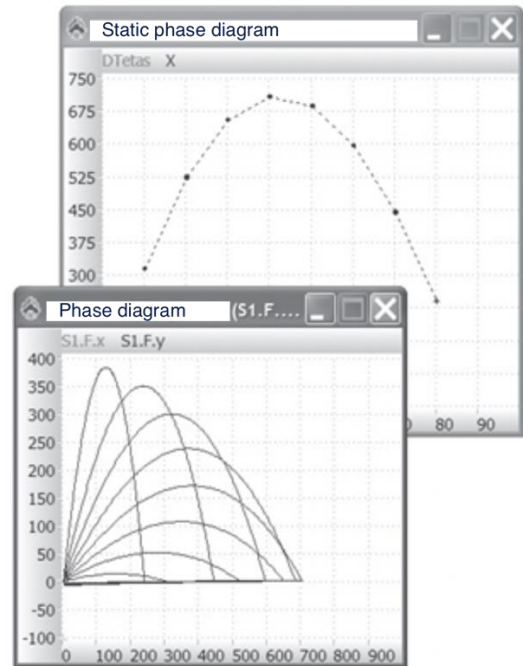


Figure 5 – The result of the simulation in the CSP OpenModelica:

**Static diagram – the dependence of the bullet flying range from departure angle (units of measure: on X-axis – degrees, on Y-axis – millimeters ( $10^{-3} m$ ));**  
**Phase diagram – the bullet movement trajectory for different corners of a departure (unit of measure on X and Y axis – millimeters ( $10^{-3} m$ ))**

a target function's value in the set space point and it also should include optimization of additional parameters (such as air resistance, etc.).

#### 4. THE EXPERIMENTAL RESULTS AND ANALYSIS

In Fig. 6 the points indicate the values  $h$  found by results of a physical experiment for different angles of a bullet departure  $\theta$  ranging from 3 up to 11 degrees with a discrete step  $2^\circ$ . The shot is made for each value of a corner five times. Value  $h$  is as mean value on five shots. The error is defined in each point as root mean square deviation. Its maximum value is designated for  $\theta=11^\circ$  and is  $\pm 2.3 \cdot 10^{-3} m$ .

The dependence of  $h(\theta)$  received as a result of simulation of the bullet movement taking into account air resistance and does not differ for CSP Mathcad and OpenModelica. This dependence is represented in Fig. 6 by a solid line.

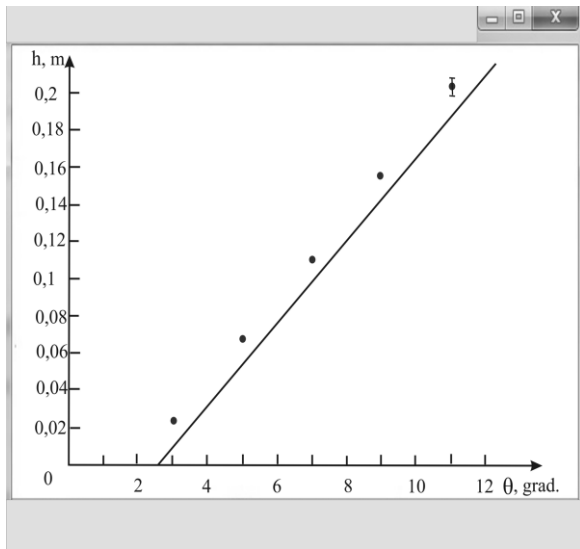
The analysis provided by students consists of the following stages.

1. *Analysis of graphic dependences.* It is possible to see that distance of the solid line deviation started



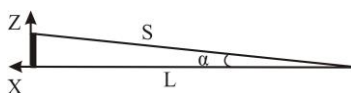
from the experimental point is bigger than interval of given measure of inaccuracy. Therefore, there is statistically significant discrepancy of results of the technical experiment and computer simulation.

2. *Forming of a hypothesis regarding the discrepancy reasons.* All experimental points are placed above the line of computer simulation. Trajectories of the bullet movement represented in Fig. 3 and Fig. 5 show that all points for  $\theta \leq 11$  of its hit in the pendulum correspond to the area of



**Figure 6 – The dependence of value  $h$  for different corners of a bullet departure. The points are result of a technical experiment, a solid line – the result of simulation in CSP Mathcad and OpenModelica**

increase in a parabolic function. Therefore, a hypothesis is made according to which during laboratory experiment the bullet flies in the horizontal direction distance of  $S$ , which is bigger than  $L=1.2\text{ m}$ . It is possible if the sight is deflected and the bullet moves, as shown in Fig. 7, under some corner  $\alpha$  to X-axis (X-axis is perpendicular to Z-axis, as it is shown in Fig. 1). It should be noted that the sight deviation on a corner  $\alpha$ , without the knowing of the student, is exposed by the teacher before the experiment.



**Figure 7 – The bullet movement at an angle to X-axis**

3. *Refinement of the solution using of the CSP.* Refinement of solution in the CSP Mathcad is provided by using the equations system (1). Stepwise correction of flying range  $S$  [22] is carried out by the formula:

$$S = L / \cos \alpha, \quad (2)$$

and the value of a corner  $\alpha$  is defined which has minimal difference between the laboratory result and the simulation result.

Refinement of the solution in the CSP OpenModelica is possible based on built-in debugging tools of a visual model with the subsequent run of model by means of *do* operator. At the same time for the plan of model run it is used own behavior of the "Model" class [23]. In particular, it is necessary to write directly in a system of equations of the class "Model" the equation (2) as "additional impact". As a result of a Stepwise Refinement, the model passes into end state.

The specified dependences of  $h(\theta)$  received in the CSP Mathcad and OpenModelica are combined with results of a technical experiment and compared among themselves. If the hypothesis is not confirmed, then new one is moved forward new with the subsequent check (return to stage 2). If discrepancy of results of the technical experiment and computer simulation are not statistically significant, then the student passes to the fourth stage.

4. *A contrastive analysis of efficiency of the CSP' using for the task solution.* At contrastive analysis of efficiency of using the CSP for a solution of the task such indicators are used: usability of the CSP; efficiency of automatic creation of the model; efficiency of creation of diagrams/charts; capability of establishing reasons for rejection of the received results based on real and virtual test boards; simplicity and efficiency of model's debugging. In addition, the indicator "general impression from the CSP' using" is represented. This indicator is based on the analysis provided in Sections 2 and 3 and also includes the aspects which are not included into other indicators. The student evaluates all listed indicators in 1 to 5 marks range and fills in Table 2. The last "Total points" line of the table shows which CSP is more preferable to apply for computer simulation of the considered type of the tasks.

The indicators values in Table 2 show the opinion of an individual student and are subjective. Besides, the analysis of one technical experiment results is not enough to conclude about the utility of the CSP. The training process for software engineers should consist of a series of laboratory experiments that are carried out according to the described technology. It should be noted that during training there can be used not only tasks of different branches of physics, theoretical mechanics, electrical engineering, but also experimental problems of other subjects which aware execution of laboratory works and also experiments made within course and graduate

**Table 2. Assessment of CSP use efficiency**

The name of the option	Mathcad	OpenModelica
Usability		
Efficiency of automatic model creation		
Efficiency of diagrams creation		
Capability to establish reasons for rejection of the results received based on real and virtual test boards		
Simplicity and efficiency of model debugging		
General impression of the CSP' using		
<i>Total points</i>		

works (corresponding example from the graduate work is given in the herein [24]). The cumulative experience of comparative analysis of CSP based on real technical experiments forms an important competency, which is the basis for the development and improvement of computer simulation packages.

## 5. SUMMARY AND CONCLUSION

The offered technology of applying the technical experiment during classes in computer simulation training has the following advantages. First, the competence of the choice and assessment of CSP capabilities is formed in order to solve set task. Secondly, new skills of analytical comparison of conditions for providing production experiments and input parameters for computer simulation and ability to select the factors essential to the studied process appear. Thirdly, experience of the CSP analysis based on real tasks is a basis for further development of the modern standard in the simulation sphere.

The specialist who is able to understand both internal specifics of simulation packages work and specifics of their applying in real engineering tasks will be much more qualified, than the specialist who is able to work only with formalized data. Thus, using the offered technology, in general, increases the level of SW engineer's professional skills.

## 6. REFERENCES

- [1] R.V. Mayer, "Computer modeling: modeling as a method of scientific knowledge. Computer models and their types," *Scientific electronic archives*, 2016 [Online]. Available: <http://econf.rae.ru/article/6722>. (in Russian).
- [2] S. Winkler, M. Bicher, F. Breitenecker, "A comparison of different modelling and simulation approaches for hybrid dynamical systems," *Proceedings of the UKSim-AMSS 19th International Conference on Computer Modelling & Simulation*, UKSim, 2017, pp. 97-102 [Online]. Available t: <https://ieeexplore.ieee.org/xpl/conhome/8358641/proceeding>.
- [3] P. Mohagheghi, "Evaluating software development methodologies based on their practices and promises," *Proc. of the Somet'08: New Trends in Software Methodologies, Tools and Techniques*, IOS Press, 2008, pp. 14-35.
- [4] E. Hull, K. Jackson, J. Dick, *Requirements Engineering*, Springer; 2th ed., Printed and bound in the USA, 2005, 198 p.
- [5] R. France, B. Rumpe, "Model-driven development of complex software: A research roadmap," *Proc. of the FOSE'07: Future of Software Engineering*, Washington, DC, USA, IEEE Computer Society, 2007, pp. 37-54.
- [6] Yu. B. Kolesov, Yu. B. Senichenkov, *Mathematical Modeling of Hybrid Dynamical Systems: study guide*, St. Petersburg: Publishing house of Polytechnic university, 2014, 236 p. (in Russian).
- [7] C. A. Chung, *Simulation Modeling Handbook: A Practical Approach*, Taylor & Francis, 2003, 608 p.
- [8] E. Abu-taieh, A. Rahman, A. El Sheikh, "Commercial simulation packages: A comparative study," *International Journal of Simulation: Systems, Science and Technology*, vol. 8, issue 2, pp. 66-76, 2007.
- [9] A. Papanikolaou, D. Spanos, "Comparative study of simulation methods on the prediction of parametric roll of ships in waves," in book: *CENTEK Book on Marine Technology and Engineering*, Taylor & Francis, 2011 [Online]. Available at: <https://www.researchgate.net/publication/259283716>.
- [10] A. Vieira, L. Dias, M. Santos, G. Pereira, J. Oliveira, "Setting an Industry 4.0 research and development agenda for simulation – a literature review," *International Journal of Simulation Modelling*, vol. 17. pp. 377-390, 2018.
- [11] H. Shen, J. Zhang, B. Yang, "Development of an educational virtual reality training system for marine engineers," *Computer Applications in Engineering Education*, vol. 27, issue 3, pp. 527-579, 2019 [Online]. Available at: <https://doi.org/10.1002/cae.22099>.
- [12] A. J. Magana, T. de Jon, "Modeling and simulation practices in engineering education *Computer Applications in Engineering Education*," vol. 26, issue 4, pp. 731-738, 2018.

- [Online]. Available at: <https://doi.org/10.1002/cae.21980>.
- [13] P. Litwin, D. Stadnicka, "Computer modeling and simulation in engineering education: Intended learning outcomes development," *Proceedings of the International Conference "MANUFACTURING'2019: Advances in Manufacturing IP"*, Springer Link, 2019, pp. 169-184.
- [14] T. de Jong, M. C. Linn, Z. C. Zacharia, "Physical and virtual laboratories in science and engineering education," *Science*, vol. 340 (6130), pp. 305-308, 2013.
- [15] C. Pöll, M. Bicher, I. Hafner, S. Winkler, A. Körner, "Making modelling teachable MMT," *Proceedings of the ERK – International Electronical and Computer Science Conference*, Sept. 16-18, Portoroz, Slovenia, 2013, pp. 164-168.
- [16] C. Martin-Villalba, A. Urquia, Y. Senichenkov, Y. Kolesov, "Two approaches to facilitate virtual lab implementation," *Computing in Science and Engineering*, vol. 16, issue 1, pp. 78-86, 2014.
- [17] C. França, T. Gouveia, P. Santos, C. Santana, F. Silva, "Motivation in software engineering: A systematic review update," *Proceedings of the 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)*, 11-12 April, USA, 2011, pp. 154-163.
- [18] F. Breitenecker, N. Popper, "Extended and structural features of simulators – A comparative study," *Journal on Developments and Trends in Modelling and Simulation*, vol. 18, no 3-4, 2008, pp. 27-38.
- [19] F. Breitenecker, N. Popper, "Classification and evaluation of features in advanced simulators," *Proceedings of the MATHMOD'09*, Vienna, vol. 2, 2009, pp. 1445-1467.
- [20] V. K. Tolstykh, *Programming in the Programming Environment MathCAD: Tutorial for Bachelors of Engineering and Physical Specialties*, Donetsk: DonNU, 2010, 128 p. (in Russian).
- [21] Official site of the project OpenModelica [Online]. Available: <https://openmodelica.org>.
- [22] H. Benker, *Practical Use of Mathcad: Solving Mathematical Problems with a Computer Algebra System*, Springer London Ltd. England, United Kingdom, 2001, 519 p.
- [23] R. Franke "Formulation of dynamic optimization problems using Modelica and their efficient solution," *Proceedings of the 2nd International Modelica Conference*, March 18-19, 2002, pp. 315-323. [Online]. Available:

[https://www.modelica.org/events/Conference2002/index\\_html/papers/p39\\_Franke.pdf](https://www.modelica.org/events/Conference2002/index_html/papers/p39_Franke.pdf).

- [24] B. M. Politycin, O. D. Shtanko, M. B. Litvinova, S. O. Karpova, "Energy recovery device for the internal combustion engine," *Scientific Bulletin of the National Mining University*, no. 3, pp. 82-89, 2017.



**Maryna Litvinova**, Doctor of Pedagogics, Candidate of Physics and Mathematics Sciences, Master of Technology degree in System software, Associate Professor in Department of Software Engineering, Physics and Mathematics, Kherson branch of Admiral Makarov National University of Shipbuilding. Areas of scientific interests: physics, information technology in education.



**Oleg Dudchenko**, Candidate of Sciences (Engineering), Associate Professor, Head of Department of Information Technology and Physics and Mathematics, Kherson branch of Admiral Makarov National University of Shipbuilding. Areas of scientific interests: information technology, computer simulation in shipbuilding.



**Olexandr Shtanko**, Candidate of Sciences (Physics and Mathematics), Master of Technology degree in System software, Associate Professor in the Department of Software Engineering, Physics and Mathematics, Kherson branch of Admiral Makarov National University of Shipbuilding. Areas of scientific interests: electrical and mechanical engineering, information technology in education.



**Svitlana Karpova**, Master of Technology degree in System software, Researcher of the Department of Software Engineering, Physics and Mathematics, Kherson branch of Admiral Makarov National University of Shipbuilding. Areas of scientific interests: computer simulation in shipbuilding.