

DOI: 10.15514/ISPRAS-2019-31(3)-3

Development of a software framework for real-time management of intelligent devices

T. Naumović, ORCID: 0000-0001-9849-7665 <tamara@elab.rs>

L. Baljak, ORCID: 0000-0003-3779-7335 <lukabaljak@elab.rs>

L. Živojinović, ORCID: 0000-0003-3536-3146 <lazar@elab.rs>

F. Filipović, ORCID: 0000-0001-7113-3802 <filipfilipovic@elab.rs>

*Faculty of organizational sciences, University of Belgrade,
Jove Ilića 154, 11000 Belgrade, Serbia*

Abstract. The subject of this paper is development of software framework for real-time management of intelligent devices. The framework enables intelligent management of IoT devices in cyber-physical systems using models based on recurrence relations and differential equations. The platform was developed using Python programming language, Django framework and wide corpus of modules and libraries that supports continuous simulation. The software framework incorporates application programming interface as well, for specification of system behaviour, transmission of input parameters and output results, sending control actions via web services to the IoT system.

Keywords: software framework; IoT; continuous simulation; python

For citation: Naumović T., Baljak L., Živojinović L., Filipović F. Development of a software framework for real-time management of intelligent devices. Trudy ISP RAN/Proc. ISP RAS, vol. 31, issue 3, 2019, pp. 35-46. DOI: 10.15514/ISPRAS-2019-31(3)-3

Acknowledgments. Authors would like to thank prof. dr. Božidar Radenkovic, prof. dr. Marijana Despotovic-Zrakić, prof. dr. Zorica Bogdanovic, prof. dr. Dusan Barac and prof. dr. Aleksandra Labus for guidance and mentoring throughout this research and software development.

Разработка программной среды для управления интеллектуальными устройствами в реальном времени

Т. Наумович, ORCID: 0000-0001-9849-7665 <tamara@elab.rs>

Л. Баляк, ORCID: 0000-0003-3779-7335 <lukabaljak@elab.rs>

Л. Живоинович, ORCID: 0000-0003-3536-3146 <lazar@elab.rs>

Ф. Филипович, ORCID: 0000-0001-7113-3802 <filipfilipovic@elab.rs>

*Факультет организационных наук, Белградский университет,
11000, Сербия, г. Белград, Йове Илич 154*

Аннотация. Предметом данной статьи является разработка программной среды для управления интеллектуальными устройствами в режиме реального времени. Платформа обеспечивает интеллектуальное управление устройствами Интернета вещей в кибер-физических системах с использованием моделей, основанных на рекуррентных соотношениях и дифференциальных уравнениях. Платформа была разработана с использованием языка программирования Python, инфраструктуры Django и широкого набора модулей и библиотек, поддерживающих непрерывное моделирование. Программная среда также включает интерфейс прикладного программирования для спецификации поведения системы, передачи входных параметров и выходных результатов, отправки управляющих действий через веб-сервисы для системы Интернета вещей.

Ключевые слова: программная среда; Интернет вещей; непрерывное моделирование; Python

Для цитирования: Наумович Т., Баляк Л., Живоинович Л., Филипович Ф. Разработка программной среды для управления интеллектуальными устройствами в реальном времени. Труды ИСП РАН, том 31, вып. 3, 2019 г., стр. 35-46 (на английском языке). DOI: 10.15514/ISPRAS-2019-31(3)-3

Благодарности. Авторы хотели бы поблагодарить проф., др. Божицара Раденковича, проф., др. Мариану Деспотович-Зракич, проф., др. Зорицу Богданович, проф., др. Душана Барака и проф., др. Александру Лабус за руководство и наставничество на протяжении всего исследования и разработки программного обеспечения.

1. Introduction

Cyber-physical systems (CPS for short) integrate devices, networks, interfaces, computer systems, and others with physical world. The fact that those elements are heterogeneous, hybrid, distributed and numerous, makes their analysis, design and implementation quite challenging and complex. In addition, CPSs are real time by their nature. Wide corpus of services, applications and interactions within CPS as well as huge growth of Internet of things further fuelled the need to change and improve existing approaches to managing those systems [1][2]. One of the most significant issues is to explore and model properties of CPS' elements, their connections and behaviour [3][4]. CPS immersed from the integration of devices with embedded systems, smart objects, people and physical environment typically connected via communication structure. Thus, it is no surprise that smart environments and systems are among the fields of CPS application.

Smart systems are integral part of CPS. The key technology for developing cyber physical systems is Internet of Things, IoT [5][6][7]. According to [8], cyber-physical systems, Internet of things and big data are related concepts of cooperative solutions, where people, autonomous devices and the environment interact with one another to achieve a certain goal. IoT technologies enable the connection of a large number of users, devices, services and applications to the Internet [9]. Management of intelligent devices often needs to be done in real-time. Real-time Control System (RCS) is a reference model of architecture that defines the types of functions needed for intelligent real-time control [10]. RCS provides a comprehensive and basic methodology for design, engineering, integration and testing of control systems [11].

In RCS systems, the state of many variables changes continuously over time, so the management of these systems can be modelled using differential equations and recurrence relations. Hence, simulation enables investigation of behaviour of such dynamic systems by developing appropriate models and using these models in experiments designed to provide an insight into the future behaviour of the system under specific conditions [12][13][14]. Simulation of CPS is becoming extremely important for both academia and practice as results of simulations have huge potential to be applied in research, business and engineering. [15].

The main idea of the research is to develop a comprehensive platform that would enable modelling and simulation of different smart environments. To achieve this goal, it is vital to define a uniform formal model applicable to any smart environment whose mathematical representation can be mapped to its implementation as one-to-one correspondence. The software framework for real-time management of intelligent devices represents a cyber-physical system incorporating IoT devices as the physical component of the system and software framework accompanied with required network infrastructure as the cyber component. Having available information and input data from intelligent devices in real-time allows the simulation engine, as an integrated part of the solution, to calculate and create a plausible outcome. On the other hand, outcome created as the result of the simulation can be a trigger dispatching control actions towards the IoT system.

The formal model, implementation and example illustrating the applicability of the presented mathematical model will be explained further in the paper.

2. Formal Model

2.1 Continuous system simulation in IoT context

Continuous system simulation refers to experimenting with models whose states are changing continuously in time [11]. These types of simulation systems are often described by differential equations. Time is independent variable in most cases. Continuous simulation can be used in different contexts and covers numerous types of real-world problems [16]. Considering time as an independent variable, digital computer has constraints solving differential equations, which is why it was necessary to develop a specific language to resolve this issue.

Different specialized languages for continuous simulation were developed, such as: CSMP (Continuous Simulation Modelling Programme) ESL (European Simulation Language) ACSL (Advanced Continuous Simulation Language) CSSL4 (Continuous System Simulation Language, Simulink, Matlab, Modelica and others that have been developed to simplify modelling, and to minimize problems related to programming continuous systems [16]. However, a majority of simulation tools have limitations related to low level of flexibility and adaptability, high costs, platform dependence, maintenance difficulties, etc. [16]

CSMP/FON platform for continuous system simulation was developed following these principles [17]:

- minimize required hardware resources and improve speed of execution;
- suitable and easy to use for educational purpose;
- simple and rich user interface;
- support for scientific research;
- saving costs.

The CSMP/FON is an open source solution and can be downloaded from the web site <https://elab.fon.bg.ac.rs/softer/csmf>. It has been used for many years in research and teaching within simulation related courses at University of Belgrade.

Software framework for real – time management of intelligent devices and IoT systems in general is a time dependent system, which requires a tool that can overcome any time – related performance issues. Ergo, using CSMP simulation logic in the software development process was a way of introducing control mechanism in the system.

2.2 Formal model of a continuous simulation system

Formal model of a continuous simulation system can be given as a tuple [18]:

$$M = (U, Y, S, \delta, \lambda, S0) \quad (1)$$

with the following meanings:

- U – set of inputs;
- Y – set of outputs;
- S – set of state variables;
- δ – transfer function: $\delta: U \times S \rightarrow S$;
- λ – output function: $\lambda: U \times S \rightarrow Y$;
- $S0$ – set of initial states.

Function of a variable ϕ is a mapping of a non-empty set X , of variables x , signed as domain, in non-empty set Y , of variables y , signed as scope (or codomain, set of function values) [18]:

$$\phi: X \rightarrow Y;$$

a function of many variables is presented through mapping:

$$\phi: X \times X \times X \times \dots \times X \rightarrow Y;$$

a block is presented as ordered set of three elements

$$b = (\phi, X, Y),$$

each $x \in X$ is input, while $y \in Y$ is output from the block.

The process of continuous simulation is based on solving differential equations and recurrence relations [17][18]. CSMP simulation language is block-oriented languages designed for solving systems of differential equations. Each block is specified by a set of inputs and parameters and a graphic symbol [12]. The graphic display of elements in the general form is presented in fig. 1

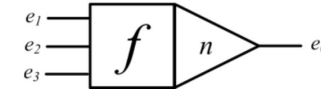


Fig 1. Graphic display of an element [17]

2.3 Formal model of a hybrid IoT system for real-time simulation

The current simulation model describes a system that allows solving differential equation systems in the given time with predefined variables and inputs [17][18]. The software framework for real-time management of intelligent devices requires a broaden model that will be suitable for use in real-time IoT systems [19].

Fig. 2 presents the concept of a hybrid IoT system for real-time simulation. This model enables having values measured in the environment in real-time included as variables of the simulation systems. In addition, the model enables managing the IoT system using variables obtained through the simulation.

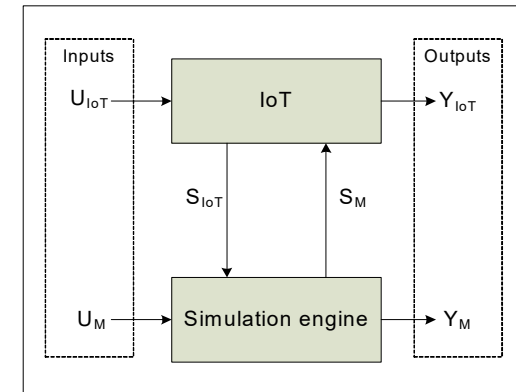


Fig 2. Hybrid IoT system for real-time simulation

For mathematical modelling of the hybrid IoT system for real-time simulation, the presented formal model needs to be extended with a set of state variables, inputs and outputs from the IoT system:

$$S = S_M \cup S_{IoT} \quad (2)$$

$$U = U_M \cup U_{IoT} \quad (3)$$

$$Y = Y_M \cup Y_{IoT} \quad (4)$$

In order to have the set of state variables S in the simulation model, it is necessary to get the values of state variables from the IoT system (S_{IoT}). This is done by developing and providing API of the IoT system. This API needs to implement the following functions:

$$\rho: S_{IoT}(t) \rightarrow S(t) \quad (5)$$

$$\omega: S(t) \rightarrow S_{IoT}(t) \quad (6)$$

$$\gamma: Y_{IoT}(t) \rightarrow Y(t) \quad (7)$$

The operation ρ is the operation of reading the values of variables from IoT system. These values then can be used in the simulation system for calculations. The operation ω enables writing the values of variables into the IoT system. These values are calculated in the simulation engine and then sent to the IoT system. These values can also be used for triggering specific actions of the IoT system. The operation γ enables reading the outputs of the IoT systems.

Having in mind that IoT systems are distributed, all these operations for interaction between the simulation and IoT systems need to be realized via web, using web services. Depending on the scenario, both PUSH and POP methods can be used.

After extending the formal model of continuous simulation system with the IoT elements, the process of continuous system simulation can be described with the finite automata equations [18]:

$$S(t) = I \times A_1 \cdot \{U(t), S(t)\} \quad (8)$$

$$Y(t) = A_2 \cdot \{U(t), S(t)\} \quad (9)$$

where A_1 and A_2 are matrix representation of algebraic functions, and I is the matrix representation of the integration operator (fig 3).

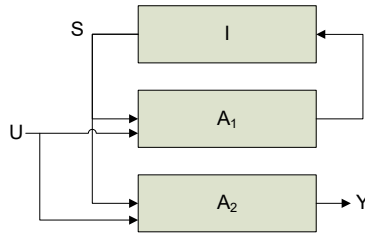


Fig 3. The structure of finite automata for simulation of continuous systems [18]

A more granular structure of continuous system simulation is presented in the fig. 4.

Fig. 4 depicts the decomposition of the operator A_1 to its elementary and primitive functions, represented as algebraic blocks. As explained later in section D, input of every block is an element that can come from either a set of inputs, a set of state variables or a set of the associated variables that represent inputs of the preceding algebraic blocks.

2.4 Orderliness

The essential feature of any non-trivial mathematical model of the continuous simulation is the feedback. The feedback occurs in the model as a result of a chain of cause-and-effect that generates a loop [20]. Considering the case of continuous simulation the model develops the feedback loop if it is impossible to mark all blocks from the set that satisfy a condition $i < j$, where block's b_i output is connected to block's b_j input [17]. The feedback loop imposes a compulsory requirement for computability of mathematical model called orderliness, defined as:

The set A of all countable algebraic blocks (blocks that correspond to algebraic functions) of M models is called orderly if all distinct objects $a_j \in A$ can be ordered (sorted) in such linear list where inputs of every distinct object a_j are elements of some of the following sets [18]:

- 1) U – set of inputs;
- 2) S – set of state variables;
- 3) Subset $C' \subset C$ defined as: $C' = \{c \in C \mid \forall i < j, \forall a_j = (\varphi_i, X_i, Y_i): c \in X_i\}$.

3. Mapping mathematical model to implementation

Mapping the mathematical method given in equations 1-9 is represented through series of UML sequence diagrams, where each method has its corresponding diagram.

The implementation of the software framework described through this research will be based on the concurrent computing and NoSQL concepts, such as threads and use of the MongoDB document-oriented database program.

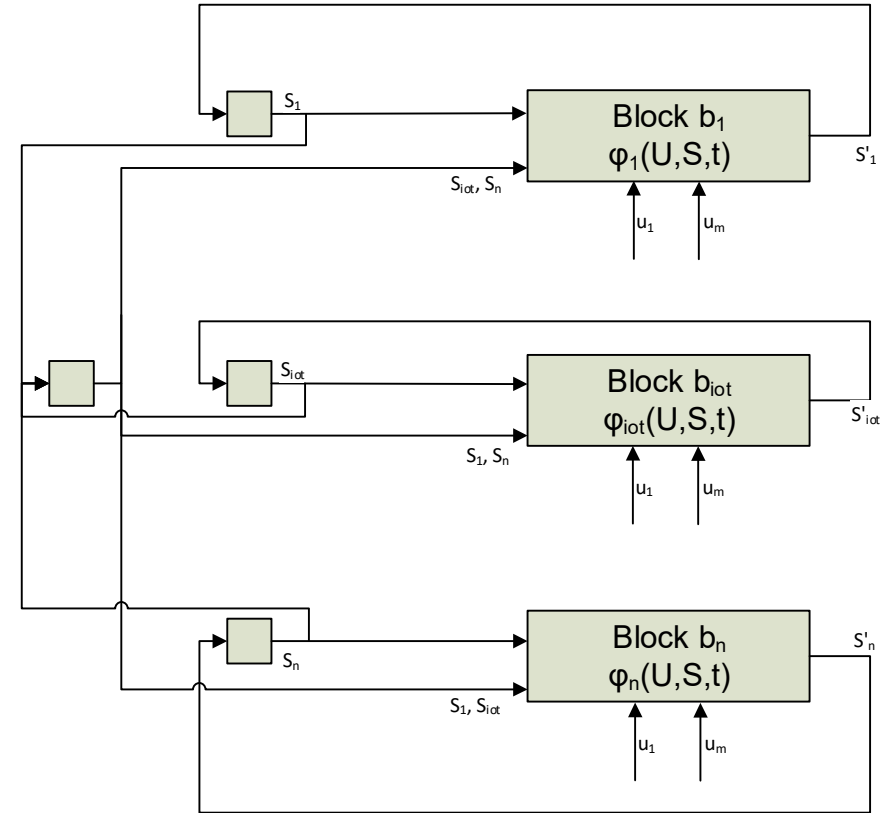


Fig 4. Block diagram of granular structure of continuous systems

3.1 Simulation engine

Fig. 5 illustrates the core simulation process depicted in equations (8) and (9). The diagram represents a typical continuous simulation process: begins with loading the simulation object from the MongoDB database in the engine, sorting the blocks, setting the primary conditions and starting the calculation process.

The calculation process itself is a looped process where series of computations are performed on every block in the simulation model: block type analysis, output generation through block function, output appending and call for next computation. The block type analysis determines if the current block is an IoT block, if it is engine provides a call to the IoT service, which performs specific operations based on the type of the call. Call types can be divided into two groups: a) reading and b) writing.

The call is a software representation of functions described through equations (5), (6) and (7). Depending on the call type, the simulation system will process data sent from the IoT system and

embed them as a part of the continuous simulation process, it will send a control action to the IoT system as a result of the continuous simulation process or it will read the output from the IoT system (fig 6).

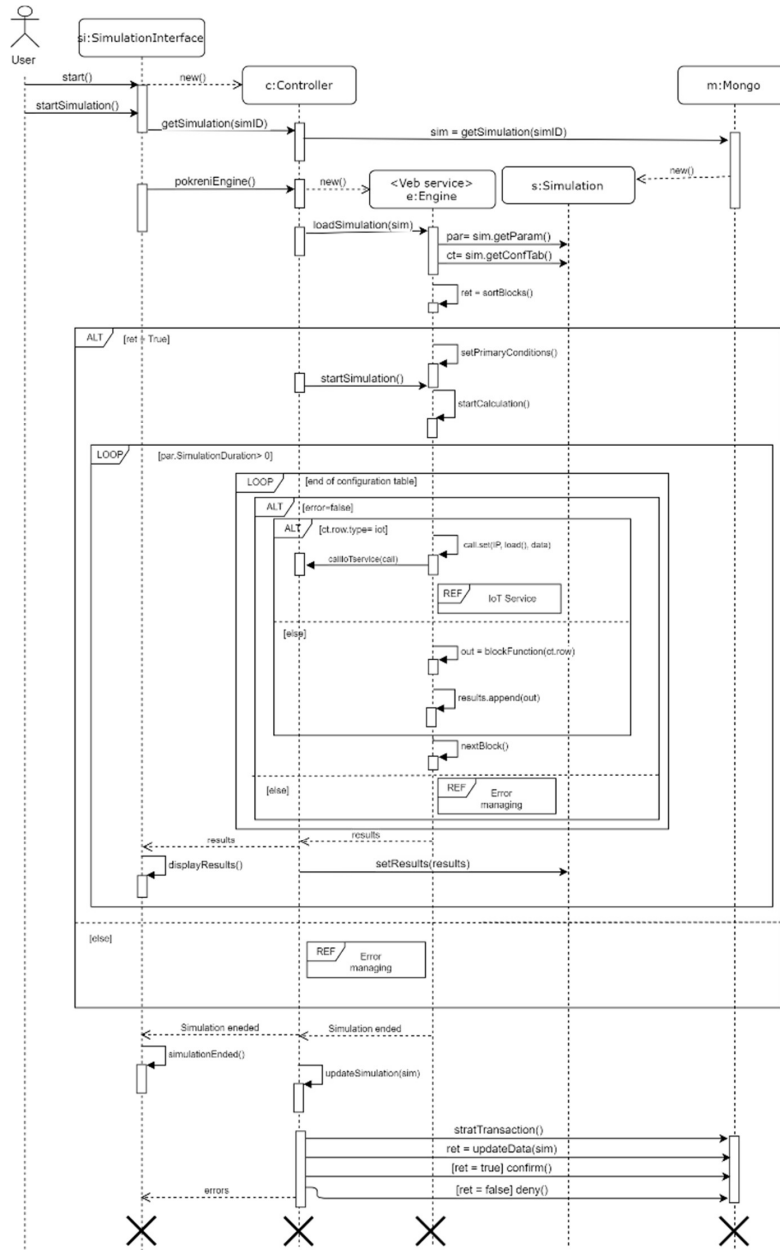


Fig 5. UML sequence diagram of the core simulation process

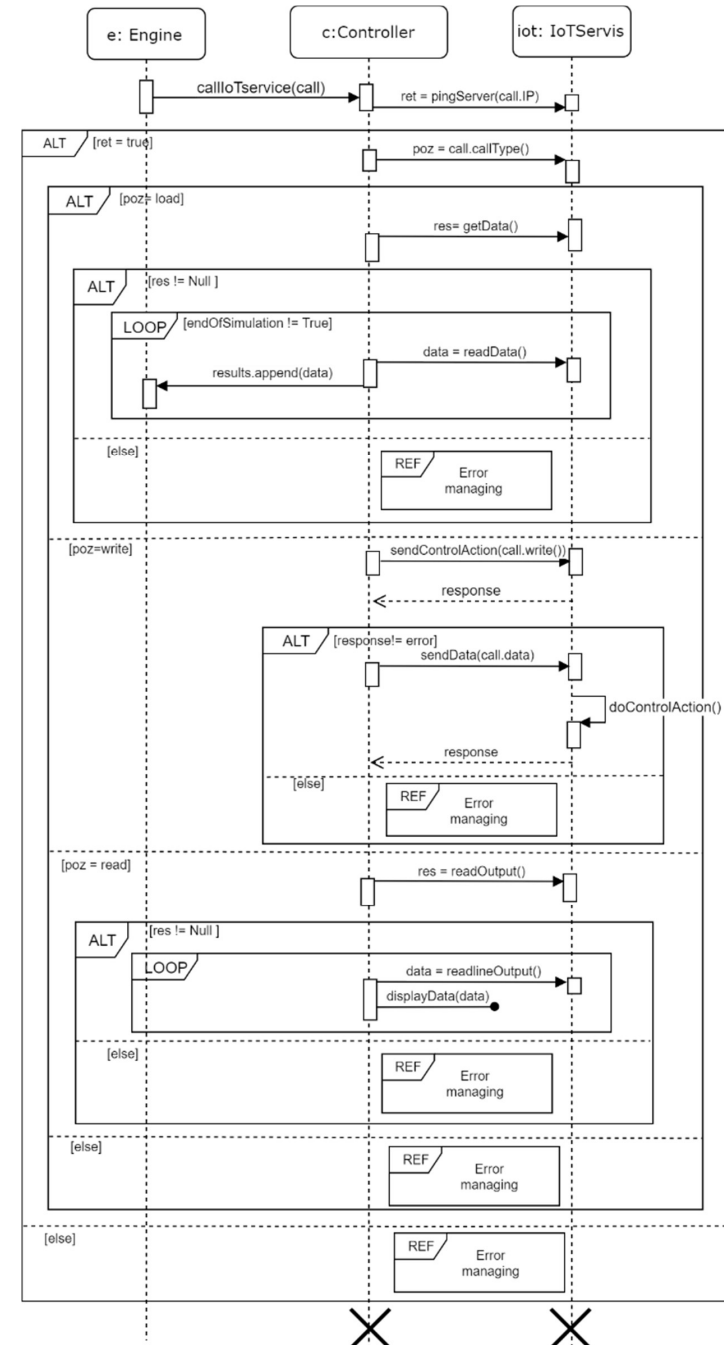


Fig 6. UML sequence diagram of processing calls to IoT system

- [8] S. F.Ochoaa, G. Di Fatta. Cyber-physical systems, internet of things and big data. Future Generation Computer Systems, vol. 75, 2017, pp. 82-84.
- [9] B. Radenković, M. Despotović-Zrakić, Z. Bogdanović, D. Barać, A. Labus, Ž. Bojović. Internet inteligentnih uređaja [Internet of intelligent devices]. Beograd: Fakultet organizacionih nauka, 2017 (in Serbian).
- [10] J. S. Albus. A Reference Model Architecture for Intelligent Systems Design. Springer, 1993. Available at: https://web.archive.org/web/20080916153507/http://www.isd.mel.nist.gov/documents/albus/Ref_Model_Arch345.pdf
- [11] F. E. Cellier, E. Kofman. Continuous System Simulation. Springer-Verlag, First Edition, 2006.
- [12] J. Banks. Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. John Wiley & Sons, 1998.
- [13] J. S. Keranen, T. D. Raty. Model-based testing of embedded systems in hardware in the loop environment. IET Software, vol. 6, no. 4, 2012, pp. 364-376.
- [14] N. L. Celanovic, I. L. Celanovic, Z. R. Ivanovic. Cyber Physical Systems: A New Approach to Power Electronics Simulation, Control and Testing. Advances in Electrical and Computer Engineering, vol.12, no.1, 2012, pp.33-38.
- [15] P. Garrahan, D. McKee, X. Ouyang, D. Webster, J. Xu. SEED: A Scalable Approach for Cyber-Physical System Simulation. IEEE Transactions on Services Computing, vol. 9, no. 2, 2016, pp. 199-212.
- [16] M. Despotović-Zrakić, D. Barać, Z. Bogdanović, B. Jovanić, B. Radenković. Software Environment for Learning Continuous System Simulation. Acta Polytechnica Hungarica, vol. 11, no 2, 2014, pp. 187-202.
- [17] B. Radenković. Program za simulaciju kontinualnih i diskretnih Sistema CSMP/MICRO [Program CSMP/MICRO for simulation of continuous and discrete systems]. Automatika, vol. 25, 1984, pp. 235-238 (in Serbian).
- [18] B. Radenković, M. Stanojević, A. Marković. Chapter 6. Simulacija kontinualnih sistema. Racunarska Simulacija [Simulation of Continuous Systems. Computer Simulation], 2009, IV edition, pp. 89-110 (in Serbian).
- [19] T.Naumović, B. Radenković, M. Despotović-Zrakić, D. Barać, A.Labus. A framework for real-time management of intelligent devices: an educational perspective. International Conference on New Horizons in Education, Proceedings Book, Volume 1, 2018, pp. 33-34
- [20] Ford. Modeling the Environment: An Introduction To System Dynamics Modeling Of Environmental Systems, Second Edition, 2010.

Информация об авторах / Information about authors

Тамара НАУМОВИЧ получила степень магистра в 2018 году и в настоящее время готовит диссертацию на кафедре электронного бизнеса факультета организационных наук Белградского университета, Сербия. Ее область интересов включает Интернет вещей, физическое и непрерывное моделирование систем, а также разработку программного обеспечения.

Tamara NAUMOVIĆ received the M.S. degree in 2018 and is currently pursuing the Ph.D. degree at the Department of e-business Faculty of Organizational Sciences, University of Belgrade, Serbia. Her research and field of interest include Internet of Things, physics and continuous system simulation, and software development.

Лука БАЛЈАК – студент факультета организационных наук. Его исследовательские интересы включают Интернет вещей, краудсенсинг и мобильную разработку.

Luka BALJAK is a graduate student at the Faculty of Organizational Sciences. His research interest include Internet of Things, crowdsensing and mobile development.

Лазар ЖИВОИНОВИЧ – аспирант кафедры электронного бизнеса факультета организационных наук Белградского университета, Сербия. Его исследовательские интересы включают в себя языки моделирования и симуляции, агентное моделирование и электронное обучение.

Lazar ŽIVOJINOVIĆ is a Ph.D. student at the Department of e-business Faculty of Organizational Sciences, University of Belgrade, Serbia. His research interests include simulation and simulation languages, agent-based simulation and e-learning.

Филип ФИЛИПОВИЧ – студент факультета организационных наук. Его исследовательские интересы включает 3D-моделирование, Интернет вещей и нейромаркетинг.

Filip FILIPOVIĆ is a graduate student at the Faculty of Organizational Sciences. His research interest include 3D modelling, Internet of Things and neuromarketing.