

THE SIMULATION MODEL OF IT-PRODUCT (SERVICE) DEVELOPMENT BY A "START-UP" COMPANY GROWING INSIDE AN ACADEMIC INSTITUTION

*Chernihiv National University of Technology University, Chernihiv, Ukraine

**The Institute of Mathematical Machines and Systems Problems of the NAS of Ukraine, Kyiv, Ukraine

Анотація. Розглядається імітаційна модель розробки IT-продукту або послуги, призначена для оцінки термінів досягнення проектних цілей "start-up" компанії, яка опікається з боку "бізнес-центру" при академічній установі. Відображено застосування мереж Петрі для проектування такої моделі. Наведено приклад реалізації моделі засобами агентного моделювання.

Ключові слова: імітаційне моделювання, життєвий цикл розробки програмного забезпечення, мережі Петрі, агентне моделювання, NetLogo.

Аннотация. Рассматривается имитационная модель разработки IT-продукта или услуги, предназначенная для оценки сроков достижения проектных целей "start-up" компанией, опекаемой со стороны "бизнес-центра" при академическом учреждении. Показано применение сетей Петри для проектирования такой модели. Приведен пример реализации модели средствами агентного моделирования.

Ключевые слова: имитационное моделирование, жизненный цикл разработки программного обеспечения, сети Петри, агентное моделирование, NetLogo.

Abstract. The simulation model of IT-product or service development by a "start-up" company growing inside an academic institution's "business center" is reviewed. The purpose of the model is to estimate final dates of reaching project goals of the "start-up" company. The use of Petri nets for design of such model is shown. An example of model implementation by means of agent simulation is given.

Keywords: simulation modeling, software life circle, Petri nets, agent simulation, NetLogo.

1. Introduction

Nowadays in developed countries devolved from industrial society to information-oriented society there is a tendency for arranging business centers and incubators inside Universities to support and guide young companies which create new technologies, products and services. First of all these are the companies associated with various information technologies – software, computer hardware, computer aided design, Internet and mobile applications [1].

Being guided by a business center, such companies are going through several stages, i.e. from an informal group of like-minded people to a start-up and a spin-off company, and then they become an independent business already with no participation of the business center. At all these stages, the supervisory board of the business center faces a task to make decisions as to support formation of such companies, transfer them from a start-up state to a spin-off, as well as close them down. These solutions are quite complex and require both methodologies and tools to evaluate quality of the companies supported by the business center, for example, based on SW CMM maturity model [2].

This publication considers modeling of an IT-project carried out on the basis of the most popular among small businesses life cycles: an evolutionary model of rapid prototyping [3]. Conceptual modeling will be performed using Petri networks, while the model itself will be implemented by means of NetLogo agent simulation [4].

2. Work statement

Activity of a start-up/spin-off company growing inside the business center under the University can be seen as a time-limited project. And from the point of view of the business center this will be a project on floating a successful independent company, but from the standpoint of the company, this will be a project on launching a newly created product (service) or project on the company mastering specific skills required for its business. Obviously, success or failure of the project undertaken by the start-up/spin-off company being guided will be the same result of this project from the view point of the business center under University.

In project management the classical strategy for a project modeling is the method of PERT (Program (Project) Evaluation and Review Technique – an assessment and analysis programs (projects) technology), where such modeling is performed using graphs of relationship of activities (arrows) and events (nodes). And the management of such projects is carried out by the critical path method, where the graph mathematical analysis result reveals a set of such activities that a change of duration of any of them leads to a change in the entire project implementation time.

At the same time, assessing certain activity duration is not a trivial task for the IT projects, since such assessments are very subjective and depend on the performers' background, especially for immature organizations, which are the start-up/spin-off companies. In addition, the models developed by the classical method tend to a cascade form of the life cycle, which does not reflect the iterative nature of the information technology development. Thus, the task is to build an adequate model of the project feasibility assessment.

Purpose of the suggested simulation model is to assess the feasibility of production goals supported by the company (product release, development of a new service, acquisition of specific competencies, improving the efficiency of labor, etc.). Such model should include:

- The life cycle of the project implementation through achieving the company's goals, such as creation of the product (service).
- Sets of typical activities which make up the above-mentioned life cycle phases.
- Lists of competencies required for activity and standardized duration of these activities, if implemented by a competent person.

The input data of this model will be as follows: a real set of experts available at the company; the planned network model of the project implementation; the company's strategy to allocate work among specific specialists; probability estimate of work performance and other information specific to a particular project.

As a result of the run, the model shall answer the following questions:

- What are the project implementation terms considering the project realization strategy chosen by the company?
- Whether the allocated resources are sufficient for the project implementation?

3. Conceptual framework

According to [5], development of the conceptual model is the next step to create a simulation model after defining the problem and the general task setting. As a rule at the beginning such conceptual model is being described using the natural language in terms of the subject area, and only then the model is described using the simulation languages. Further on one can proceed either with the software implementation, or with the mathematical model and its formal solution. We will give here such description of our conceptual model.

The project life cycle (product creation) determines the generalized phases, in which the project exists from its beginning until the end. Typically, these are the phases of planning, designing, developing, testing and implementing. Such phases are not necessarily linear, but they may be repeated and held simultaneously. Each phase is characterized by the objectives to be achieved

at this stage, consists of a set of activities; and considered completed upon their successful implementation. Development of a set of design and technical documentation, test cycle, and implementation of a solution at an Employer's site may serve as an example of such objectives.

As noted above, each phase consists of activities that result in the creation of project artifacts – usually separate documents, its chapters and parts, as well as software and technical modules, as well as their integration. Activities are interconnected by a priority relation, for example, certain activity cannot be started before the previous one is completed, and together they form a network model being described by the transition graph from one activity to another.

A performer is connected to each activity – this is usually a specific company's representative who has a relevant competence to do this activity. A set of areas of knowledge from SWE-BoK [6] can be considered as such competencies for an IT-project. At the same time, people are a limited resource of the company both in quantitative and qualitative terms. For modeling purposes, we can range [0..1] an estimate of competences per each performer. Then some standardized activity will be done by a specific performer in a time inversely proportional to this assessment.

Standardized conditional time of work execution will be a modeling step for this model. In other words, in the framework of the model one activity is carried out in one unit of time, if the performer of this activity is competent to do it with a value of 1. The activities, an objective duration of which takes more time, shall be divided into several smaller activities. The model can include some specific activities – “training” which changes the competence level of the performer who “performs” it, thus the model can simulate the cost of specialists training.

The model input data will be the IT-project implementation life cycle chosen by the company, a list of planned activities, competency of the company's specialists and the strategy for their allocation per activities. Time required for the simulated IT-project implementation will be a result of modeling.

4. Model of the IT-project life cycle-prototyping

It is believed that the idea of prototyping when creating software was offered by Frederick Brooks in the famous work “The Mythical Man-Month”, although F. Brooks himself thought that this idea was antedated by Vinton Royce, creator of the cascade model [7]. In this article, a description of the structural evolution model of rapid prototyping will be given here in accordance to R. Futrell et al [3]. Below is its graphical representation.

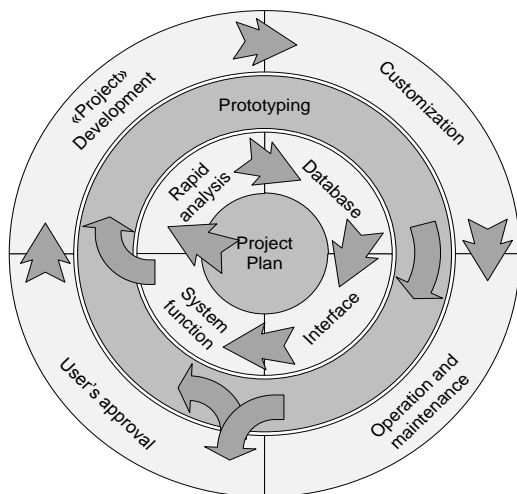


Fig. 1. Structural evolution model of rapid prototyping

The first stage of this model life cycle is the creation of a project plan based on the preliminary requirements. Then, the rapid analysis is done, the database, user interface, and system functions are designed. In this case, one is intentionally restricted by creation of a high-level partial model at the documentation level. Next, one proceeds with the development of a system prototype that is shown to an Employer; after that comments and suggestions are determined and the cycle of already improved prototype occurs again.

The process stops when the Employer officially approves a working model of the system. After that the design and technical documentation is being updated upon performed system and the working model is adjusted up to the level of work use. The life cycle is completed when the system is transferred to the operation and maintenance stage.

According to the teaching experience of the authors in the universities one can note that this model is very popular among students because from their point of view it allows you quickly

starting the system implementation and does not burden the "paperwork". Based on the above, this very model of the software life cycle will form the basis of our simulation model.

5. Modeling of Petri net

At present, the systems modeling using Petri nets is very popular in the academic environment, owing to its clearness and developed mathematical tools. A Petri net is a bipartite-oriented multi-graph consisting of vertices of two types – positions and transitions, interconnected by the arcs [8].

For modeling purposes, we will use the method of gradual detailing of Petri nets states and transitions, which in fact is a well-known method of information systems designing by the top-down method.

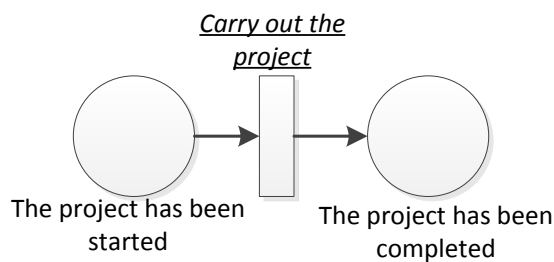


Fig. 2. Initial Petri net which modules the IT-product

“Project is being carried out” can be expanded to state “Project is being carried out”, preceded by transition “Start the Project” and with the subsequent transition to “Complete the project”. At the same time, state “Project has been started” can be detailed in the following states: “Goals<of the project> have been defined”, “Resources<for the project> have been allocated” and “Time<of the Project> has been set”.

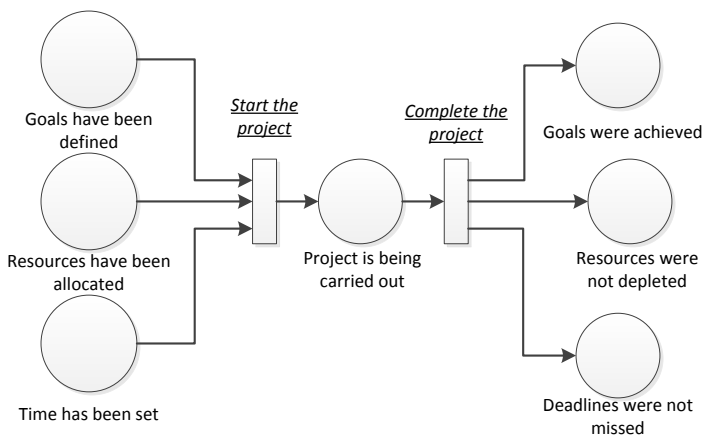


Fig. 3. Petri net which modules the IT-product creation project

the state "Project is being carried out" is detailed in accordance with its model.

5.2. Detailing the state of "Project is being carried out" on the model of the “rapid prototyping” life cycle

Fig. 4 presents the Petri net, detailing the state of “Project is being carried out” for the model of the “rapid prototyping” life cycle. In contrast to the above generalized model where prototyping cycle does not include a need to revise the project schedule, repeat the system analysis, review

5.1. Petri net of the top-level for a typical project

An initial Petri net for the project model can be presented in the form of two successive states “Project has been started” and “Project has been completed”, as well as one of transition “Start the project”, see Fig. 2. It is obvious that such a network is finite.

Let’s detail the initial network states and transition. Thus the transition “Carry out the Project” can be expanded to state “Project is being carried out”, preceded by transition “Start the Project” and with the subsequent transition to “Complete the project”. At the same time, state “Project has been started” can be detailed in the following states: “Goals<of the project> have been defined”, “Resources<for the project> have been allocated” and “Time<of the Project> has been set”. Similar actions can be performed for the state “Project has been completed”, see Fig. 2. Note that transition “Start the project” can fire only if all three previous states have network tokens. And firing of the “Complete the Project” transition makes its subsequent states “true”.

The Petri net presented in Fig. 3 can be a base for any model of the IT-products life cycle, where

the database, system functions, and human-machine interface (HMI), the proposed network considers this need, which makes it closer to real practical IT-projects.

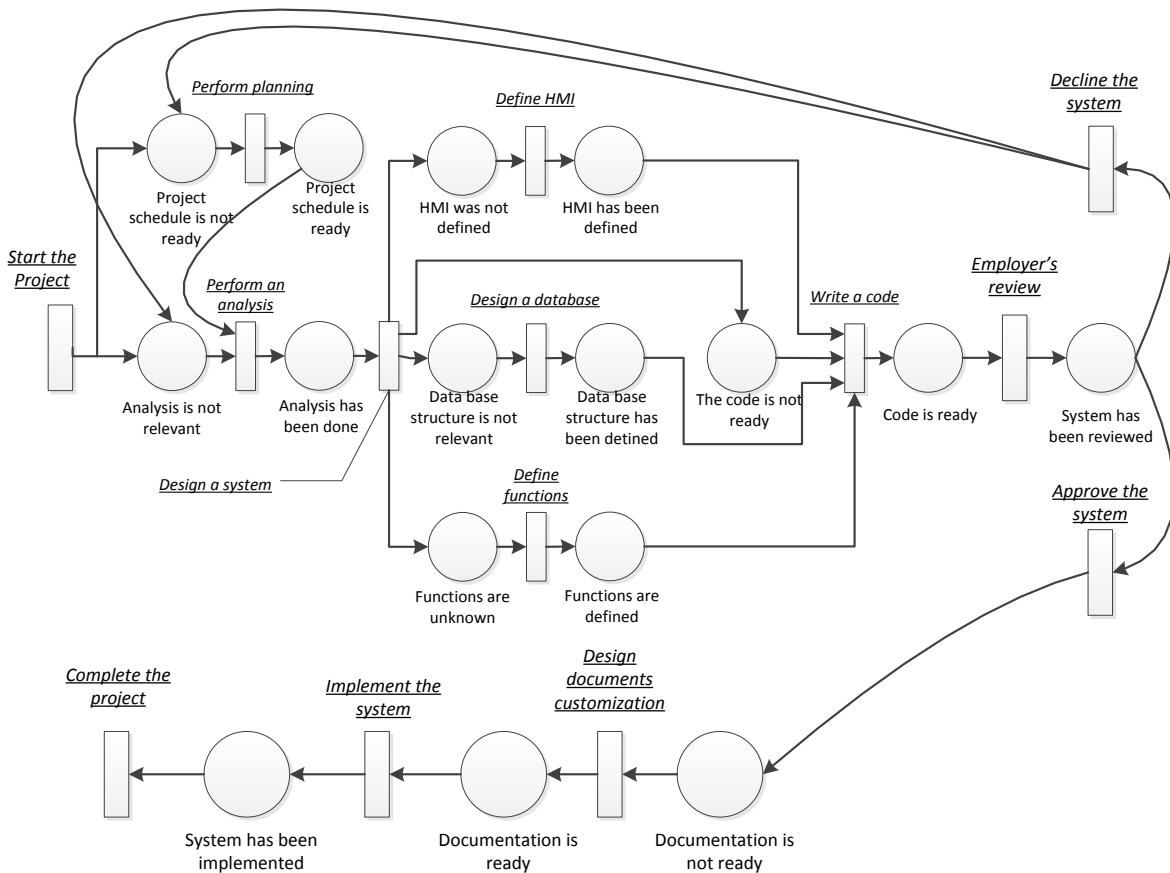


Fig. 4. Petri net for state “Project is being carried out” of the “rapid prototyping” life cycle model

As can be seen from Fig. 4 transition “Start the Project” initializes the state of “Project schedule is not ready” and “Analysis is not relevant”. Transition “Run Analysis” takes into account the fact that the project analysis cannot be started without having the project schedule available, i.e. time and resources to perform such analysis, as well as subsequent activities are not taken into account. Transition “Design the system” allows actions to develop HMI, database, system functions, and code that in real projects are usually carried out in parallel. State “System has been considered” allows either of the two transitions “Decline ...” and “Approve the system”. Approval of the system leads to the subsequent transition to complete the project, while the deviation of the system repeats the cycle of project planning and system analysis (in this case for the upcoming changes).

In the present form, Petri net does not yet take into account a need for resources and time, and thus cannot be used for the stated purposes of simulation modeling.

5.3. A Petri net which models activity work on the project tasks

As noted above in the description of conceptual model, the project is characterized by a variety of works, planned completion due dates and a set of performers, at that all three of these varieties are finite. In Petri nets this can be simulated by a set of tokens in the states and inputs/outputs on transitions, see [9]. Let us consider the following Petri net, see Fig. 5.

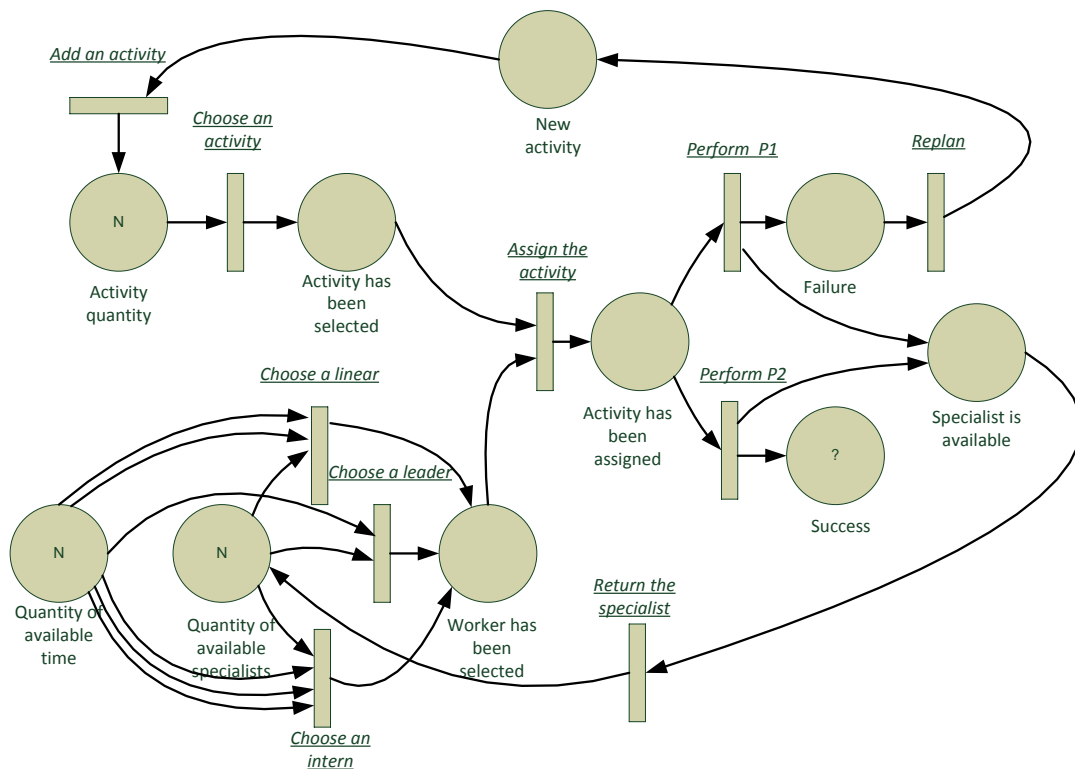


Fig. 5. Petri net, which simulates activity of the project group on the tasks

Initial Petri net marking is defined by arrangement of tokens as follows:

- Number of the project tasks is set by a number of tokens in the “Number of activities” state;
- Duration of the project (in terms of conditional time planning) is set by a number of tokens in the “Availability of time” state;
- Number of available specialists is set by a number of tokens in the “Number of available experts” state;
- All other states remain “empty”.

Transitions “Select ... a leader, a linear, and an intern” define “cost” of work performance by a specialist with diverse degree of training in unit time. In this example, the lead specialist reduces time available to complete the project per 1 unit, the linear – per 2 units and the intern by 3 conditional units of the project time.

Transition “Assign activity” connects the single work with a specific performer. Transitions “Run” are stochastic and fire with a certain probability of success (P_2) or failure (P_1) when performing work. After the work is completed the expert returns to the pool of available specialists. “Failed” work recurs in a variety of outstanding works for a new run.

Network stops when the available time or activities are exhausted. Herewith a number of tokens in the “Success” state will show a number of successfully completed activities, in the “Number of activities” state – a number of outstanding activities, and in the “Availability of time” state – remaining time in case of exhaustion of all outstanding work or zero, when the available time is exhausted.

5.4. Petri net simulating work on a specific task

The Petri net models considered above simulates execution of a project task through a direct link between a specialist’s qualification and time needed for him to do the work. That may be enough for the initial planning tasks and the project life assessments. However, another model may be

required to obtain more accurate estimates of time spent for work. An example of such model is shown in Fig. 6.

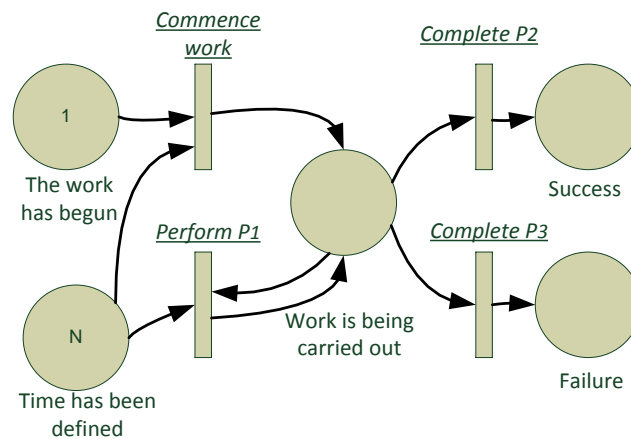


Fig. 6. Stochastic Petri net modeling the time spent at work on a specific task

The initial marking of this Petri net is defined by tokens arrangement as follows:

- Token in "Work has been started" defines a possibility to start work.
- Terms of work performance (in units of conditional planning time) are set by the number of tokens in the "time has been defined" state.
- All other states remain "empty".

Transition "Start the work" fires provided that time is available ($N > 0$ in the "Time has been defined") and the token in the "Work has been started" state.

Transitions with marks P_1 , P_2 , P_3 are stochastic and fire with a certain probability. In this case, $P_1(n)$ is the probability of further work, $P_2(n)$ – of successful completion of work, and $P_3(n)$ – of a failure of work performance on the n th time step.

"Success" or "Failure" states show status upon completion. Herewith the network will stop in one of these two states when the available time is exhausted, since the transition "Perform P_1 " can no longer be made and one of the transitions P_2 or P_3 will fire.

6. Software implementation

The model is implemented in the NetLogo modeling environment, developed in 1999 by Uri Wilensky in the Center for Connected Learning and Computer-Based Modeling, United States [6]. This tool which is not an environment for Petri nets modeling was chosen in order to check the hypothesis of commensurability of a programmer's efforts to transfer into a program code of models made in the Petri nets form and those models made in the UML diagrams form. Another hypothesis under check was a possibility of an agent-based simulation modeling, where the agent's behavior sets the Petri net, similar to the example shown in Figure 6. It is worth noting that both hypotheses were practically proved out. Working example of the software implementation is shown in Fig. 7.

The result model allows evaluating:

- timing of the project having a given configuration of staff: a number of performers, their qualitative composition (graduate students, masters, bachelors, students);
- amount of work performed by a single performer.

Having this information one can estimate an adequacy of allocated resources to implement project within a given period of time.



Fig. 7. An example of the simulation process

7. Conclusions and suggestions.

As a result of conducted investigation the simulation model of implementation process per the life cycle model of IT-project structural prototyping by a small company or a university team was developed, which allows us to estimate time required to implement such project depending on the strategy of the human resources allocation for the activities being performed.

Using Petri nets to formalize the conceptual model enables rapid transition to a software implementation, and does not yield to UML means which are traditionally used to build models before writing a code. Petri nets have shown their clearness when describing an agent's behavior in an agent-based simulation.

REFERENCES

1. Литвинов В.В. Опыт взаимодействия Университетов и промышленности в сфере трансфера ИТ-технологий в Западной Европе / В.В. Литвинов, В.С. Харченко // Математичні машини і системи. – 2015. – № 1. – С. 111 – 123.
2. Інструментальні засоби оцінки зрілості іт-компаній, які підтримуються бізнес-центрами при університетах / В.В. Литвинов, М.В. Савельєв, І.С. Скітер [та ін.] // Вісник Чернігівського державного технологічного університету. – 2015 (у друці).
3. Фатрелл Р.Т. Управление программными проектами: достижение оптимального качества при минимуме затрат / Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И.; пер. с англ. – М.: Издательский дом "Вильямс", 2003. – 1136 с.
4. Wilensky U. NetLogo itself [Електронний ресурс] / U. Wilensky. – NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999. – Evanston IL. – Режим доступу: <http://ccl.northwestern.edu/netlogo>.
5. Томашевський В.М. Моделювання систем / Томашевський В.М. – К.: Видавничка група ВНУ, 2005. – 352 с.
6. Software Engineering Body of Knowledge [Електронний ресурс]. – SWEBOK V3, IEEE computer society. – Режим доступу: <http://www.computer.org/web/swebok/v3>.
7. Брукс Ф. Мифический человек-месяц или как создаются программные системы / Брукс Ф.; пер. с англ. – СПб.: Символ – Плюс, 2001. – 394 с.
8. Murata T. Petri Nets: Properties, Analysis and Applications / T. Murata // Proceedings of the IEEE. – 1989. – Vol. 77, N 4. – P. 541 – 558.
9. Башкин В.А. Некоторые методы ресурсного анализа сетей Петри: дис. ... доктора физ.-мат. наук: 05.13.17 / Башкин Владимир Анатольевич. – Ярославль, 2014. – 268 с.

Стаття надійшла до редакції 30.10.2015