Model of the “Department” Ecosystem

Sergii Kryvyi[1], Olena Grinenko[2]

[1] Taras Shevchenko National University of Kyiv, Volodymyrska Street, 58, Kyiv, 01601, Ukraine
[2] National Aviation University, Liubomyra Huzara ave., 1, Kyiv, 03058, Ukraine

Abstract
The article deals with the formalized definition of the "Department" ecosystem of a higher educational school, the components of this ecosystem and their characteristics. Approaches to the study of the "Department" ecosystem and the application of appropriate models for its effective functioning, capable of adapting to changing working conditions, are considered. Mathematical formalization of presentation and processing of knowledge within the "Department" ecosystem was developed using these approaches.
In particular, an ecosystem model is proposed, which describes the main objects and their functions in such medium of a higher educational institution as a department. The main object of research in such models are conditions of the successful development and functioning of the department. In addition, the methods of planning an equal educational load for teachers, together with the employees of the university department and the management of the faculty are considered. The main actors in the ecosystem are teachers (called servers), who teach the courses. The proposed model makes it possible to simplify the drawing up of the schedule, to quickly respond to force majeure circumstances, necessary exchange of the teacher, etc. The properties of ecosystem models are verified by automata-network methods. Such ecosystem can be generalized and expanded by adding models of the faculty, university, and Ministry in order to control the work of the faculty, university, and ministry, the effectiveness of their functioning, etc.

Keywords
Model, ecosystem, properties.

1. Introduction

In [1], a model of the software ecosystem (SE) was proposed, which in this paper is adapted to the construction of the ecosystem model, limited to the domain of the "Department" as the main link of the functioning of the higher education institution (HEI). Unfortunately, limitations on the volume of the paper do not allow us to describe in details all the properties of the model, so only the main stages of its functioning during one working day are described.

2. General ecosystem model

Despite the significant contribution of scientists to the development of theoretical provisions and practical recommendations in the researched area, it can be concluded that the results of their research are reduced to the description of the essence of individual components of technical ecosystems, individual factors and causes of their occurrence, qualitative and quantitative analysis of some elements of the ecosystem formation process and the issue of the life cycle of ecosystems. At the same time, until now there is no single formalized representation of ecosystems, clearly defined approaches to their research, as well as methods and algorithms for researching the properties of ecosystems are missing.

Thus, the relevance of this work is due to the need to:

- formalized definition of the ecosystems;

14th International Scientific and Practical Conference from Programming UkrPROG’2024, May 14-15, 2024, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
sl.krivoi@gmail.com (S. Kryvyi); olena.hrinenko@npp.nau.edu.ua (O. Grinenko)
0000-0003-4231-0691 (S. Kryvyi); 0000-0001-9673-6626 (O. Grinenko)

- research for approaches to the study of the ecosystems;
- research and develop models for building the ecosystems;
- investigate and develop effective methods for research the properties of the ecosystems;
- build the tools for researching the properties of the ecosystems.

The object of research in the ecosystem, which is related to education in higher education institutions, are methods, models and tools that make it possible to study the interaction of objects with the surrounding environment, and the interaction of ecosystem objects with each other. At the same time, the main aspect of such interaction is the sustainable development of the ecosystem as a whole.

The subject of research is the processes that take place at such an object of higher education as a department and its surroundings (methods, interaction, teachers, students, developers, customers, regulatory authorities, etc.).

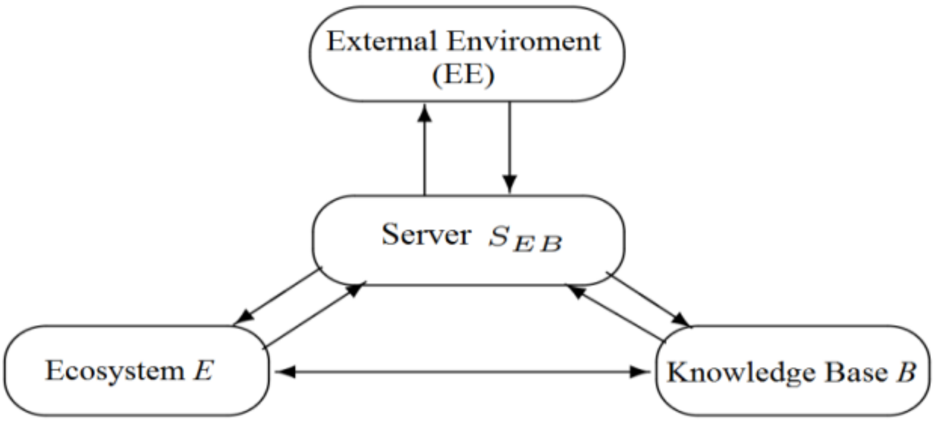
We propose the following definition of the ecosystem.

Definition 1. Ecosystem (ES) is called a quadruple

$$ES = (EE, E, B, S_{EB}) \tag{1}$$

where

- EE is an external system;
- E is an autonomous family (set) of objects, which interact with each other and with environment (kernel or core of ES);
- B is a (ontological) knowledge base, where the evolution of family's development is saved as population (macro-knowledge of ES);



- S_{EB} is a server to provide the interaction between E, B and environment.

Figure 1: Model of high-level ecosystem

The external environment of *EE* is the external environment in which the ecosystem (department) functions. This object includes everything that has an external influence on this functioning.

The family of objects *E* in *ES* can be interpreted in different ways in general, but in this consideration these objects receive a very specific interpretation, which will be discussed later.

Knowledge base *B* in *ES* plays the role of a repository of knowledge about the stages of development of the department, about developed courses, laboratory work and practical tasks, completed projects, information about existing and new software, information about mistakes made when choosing courses and tasks, about mistakes made by management, positive experience, decisions made, about the software used by the department, about the experience and seniority of teachers currently working and about those who have left the department, about the structure of the department, educational programs, training programs, programs of individual courses, about tasks , which have been performed and are being performed by the department's employees, etc.

The S_{EB} server serves to ensure operational interaction between *E, B* and the external environment of *EE*. The *E* system can work directly with the DB in case there is no need to use the S_{EB} server. And when you need to get access to some private or confidential information, you need to get permission to access such information. This function, in particular, is performed by the S_{EB} server.

In addition, in this ecosystem, the main actors of such a server are the head of the department, professors, guarantors of educational programs, system administrator. Their functions include communication with the external environment (analysis of the market, scientific achievements, interaction with the dean's office, the rector's office, and with colleagues from other departments, faculties, and universities).

3. ES components and their characteristics

ES core *E* can be given as an ordered tuple

$$E = (S, \Omega, o, f), \tag{2}$$

where:

– $S = (S_1, \dots, S_n)$ is finite set, elements of which are called servers (executors), in our case - these are teachers of the department;

– $Q = \{w_1, \dots, w_n\}$ is finite set of operations, which can be performed by different servers, in our case, these are courses that can be taught by teachers of the department;

– $o: B \to \Omega$ is a set of operations, which can be performed by any single server, in our case, these are courses that can be taught and are taught by a specific teacher of the department;

– $f: S \times \Omega \to D \times Q^+$ is a function whose values are the year, month, day (*D*) and the time of execution of a separate operation on a separate server, in our case – this is a specific course that a specific teacher teaches or have taught in a particular period of time.

Knowledge Base (*KB*). Creation of Knowledge Base (*KB*) is based on usage of languages of descriptive logics. This creation considers definition of the set of atomic concepts and atomic roles, development of terminology (*TBox*), development of the set of facts (*ABox*) and the choice of logical language and algorithms for this language, which are used to generate responses on queries to *KB*. Evolution of *ES* in time, development stages, crash are described in terms of *TBox* and *ABox*. Logic *ALC*, for algorithms of which there exist effective implementations, is a base core of descriptive logics.

In addition, the *KB* also could be also as a repository of various types of information: thesis topics, thesis defense dates, software products developed in theses, information on employment of graduates of the department, personal files and data of teachers, information on seniority, terms of internship, changes in position of the department employee, change of address, phone number, etc. This information is available only to authorized persons, for example, the *KB* administrator. The presence of such a *KB* makes it possible to promptly respond to requests for information about this or that person.

Interaction server S_{EB}. The S_{EB} interaction server serves to ensure operational interaction between *E, B* and the surrounding environment. This server contains queries that have not yet been completed (in a high load situation), queries that have already been processed, but have not been requested, and current state of information about the *KB* and system *E*. Moreover, the main function of this server is interaction with environment *ES*. This interaction is about obtaining information about changes that occur in the external environment, the state and needs of the market, the achievement of competitors, new trends in software development, its cost, etc.

Server part *ES*. Teachers of the department (servers) who are involved in E are included in the model of the server part. Indeed, in this case, server part is a tuple, which includes

– S is a set of teachers (full-time teachers, part-time teachers),

– Ω are courses that can and are taught by faculty,

– $o(S_i) = \Omega_i$ is a set of courses (subjects) that can be taught by a specific teacher S_i, $i = 1, \dots, |\Omega|$, $f(S_i, w_j) = (y, m, d, t_i)$, where $S_i \in S$, $w_j \in \Omega$, $(y, m, d) \in D$, $t_i \in Q^+$is time of performance of w_j-th operation by S_i-th executor.

The values of the function f play an important role in *ES*, which is used to calculate the load on teachers and monitor the performance of tasks. These values are quantitative (time) estimates of the complexity of operations in the worst case, or taken from some empirical observations. In this case, both the amount and time of the workload, as well as the time spent on preparatory activities (creation of a course of lectures, development of practical, seminar classes and laboratory works), are taken into account, which determines the choice of teachers for the implementation of relevant operations. Having at your disposal the value of the function f, you can predict risks in *ES* taking into account seniority, experience and qualifications of teachers (this information is taken from *KB B*). Forecasting delays in work or the risks of disrupting the execution of planned classes provided for in the schedule makes it possible to quickly respond to challenges in *ES*. This can be achieved through timely replacements, informing students about changes in the schedule and rescheduling classes (upon agreement with the dean's office).

In addition, the values of the function f provide a basis for finding the optimal way of performing tasks in *ES*, since in model (2) it is assumed that the same task can be performed by different servers. And this makes it possible to use optimization algorithms to build such a plan for conducting classes that evenly distributes the load on teachers, provided for by the regulations, and departmental equipment.

4. Research on model properties

Let us consider some approaches to the study of the model (2) properties and an illustration of the proposed formalism on the example of system *E*. In general, such a system can have a large set of objects, and then there is a need for operational planning, namely optimal planning and management of the execution of tasks and access to its shared resources. Several methods of building optimal planning and management can be proposed. The difficulty of solving these problems lies in the fact that, in general, these problems contain contradictions of an objective nature. This is the reason for the existence of several methods, since a certain level of abstraction is chosen for the successful solution of this type of problem, for which appropriate methods are developed. This type of problem is most often formulated in the form of some optimization problem, and then a method for solving such a problem is developed.

Next, the automata-network model of solving the problems of planning the distribution of basic resources in the *E* system is considered [5, 6].

Let in system (2)

$$\Omega_i = \{w_1^i, w_2^i, \dots, w_{|\Omega_i|}^i\} \tag{3}$$

defines a set of courses (operations) that can be taught (executed) by the *i*-th teacher (server): $i = 1, 2, \dots, |S|$, $w_1^i, w_2^i, \dots, w_{|\Omega_i|}^i \in \Omega$. Hence,

$$o(S_i) = \Omega_i \tag{4}$$

From equations (3) and (4), we get that

$$f(S_i, w_k^i) = (y, m, d, t_k^i),\ i = 1,2, \dots, |S|,\ k = 1,2, \dots, |\Omega_i| \tag{5}$$

Since the system allows the possibility of teaching the same subject by different teachers, it makes it possible to evenly distribute the workload and time for different teachers. Thus, $\Omega_i \cap \Omega_j \neq \emptyset$, $(i \neq j)$.

It follows from this that in such a formulation, the task of planning the schedule can have several ways of implementation in the system *E*. Indeed, an arbitrary *p*-th task in *E* is determined by the schedule of classes and is a sequence of classes (operations)

$$\overline{O_p} = o_{p_1} o_{p_2} \dots o_{p_k}, \tag{6}$$

where $o_{p_1}, o_{p_2}, \dots, o_{p_k} \in \Omega$ are courses provided by the task. Since each such task is determined by its own sequence of the type (6) and there are several ways of its execution by teachers, there are also different implementation options for $\overline{O_p}$. Due to this situation, we impose the following restriction:

Condition 1: each teacher cannot start the next task (operation) until he\she finishes the previous task (operation).

In order to implement this limitation and operational communication with management and technical services in case of force majeure situations, we will allocate special servers *X* and *Y* in the system *E*, which we will call input and output, respectively, in contrast to other servers, which we will call workers. Input and output servers are subsystems that ensure the interaction of working servers between themselves and the management of the faculty or another governing body to which the department is subordinate. With the help of the input server, the management lays out the plan for the implementation of the task of the jth day (schedule-specification), and with the help of the output server, the execution of the task by the working servers is remotely monitored, announcements are made, orders are announced, danger is notified (for example, about an air alarm), etc. In addition, the working servers send information about the emergency completion of the task to the output server in order to coordinate the place and time of the interrupted task with the governing body (for example, the dean's office). It follows from this that the output server has a certain priority in relation to the input and working servers.

Therefore, the beginning of a certain sequence of operations in system (1) can be specified in terms of the input and working servers, and its end - in terms of the output server. Interactions between servers must be synchronized due to the fact that different operations can be performed by several servers. Such synchronization is provided for by the task specification schedule. But in order to quickly respond to unforeseen cases and ensure the fulfillment of condition 1, common Boolean variables $b_1, b_2, \dots, b_m = 0$ are introduced, the values of which can be read by each of the servers by communicating with the output server. Then, for example, the values $b_1 = 0, b_2 = 1, b_3 = 0$ mean that server S_2 is busy, and servers S_1 and S_3 are free and ready to execute task operations.

5. "Department" Ecosystem Model

In order to demonstrate the approach to the analysis of the properties of the given *ES* model, let's consider an example of the study of the "Department" ecosystem model properties, which has seven working servers for execution the tasks of one working day of the department, one input, one output servers and one management server. Such an ecosystem can be represented graphically (see Figure 2). Note that this specification does not limit the generality of the model, as this model can be adapted and detailed in accordance with the rules that regulate the work of a particular higher school. The designation S_m/S_n means that the implementation of the operation can be performed either on the S_m server or on the S_n server.

The schedule-specification of the task (the *j*-th working day of the department) is uploaded to the input server *X* - the sequence of operations (items) of the task. When the execution ends, its result is sent to the output server *Y*. For each task and each lesson, a set of servers from the set $S = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ and a set of operations from $\Omega = \{w_1^1 - w_1^4, w_2^1 - w_2^4, w_3^1 - w_3^4, w_4^1 - w_4^4\}$ are assigned. The same operation can be assigned to different tasks. The management server *D* implements the functions of the dean's office and its tasks include the functions of regulating the execution of operations or tasks in case of force majeure situations (impossibility of execution of operations, illness of executors, military circumstances, etc.), resolution of disputed or conflicting issues. In addition, one of the functions of this server is the coordination of the date and time of the transfer of tasks or individual task operations. This function is quite important, as it provides for the successful functioning of the dean's office as a body of management and communication with executors.

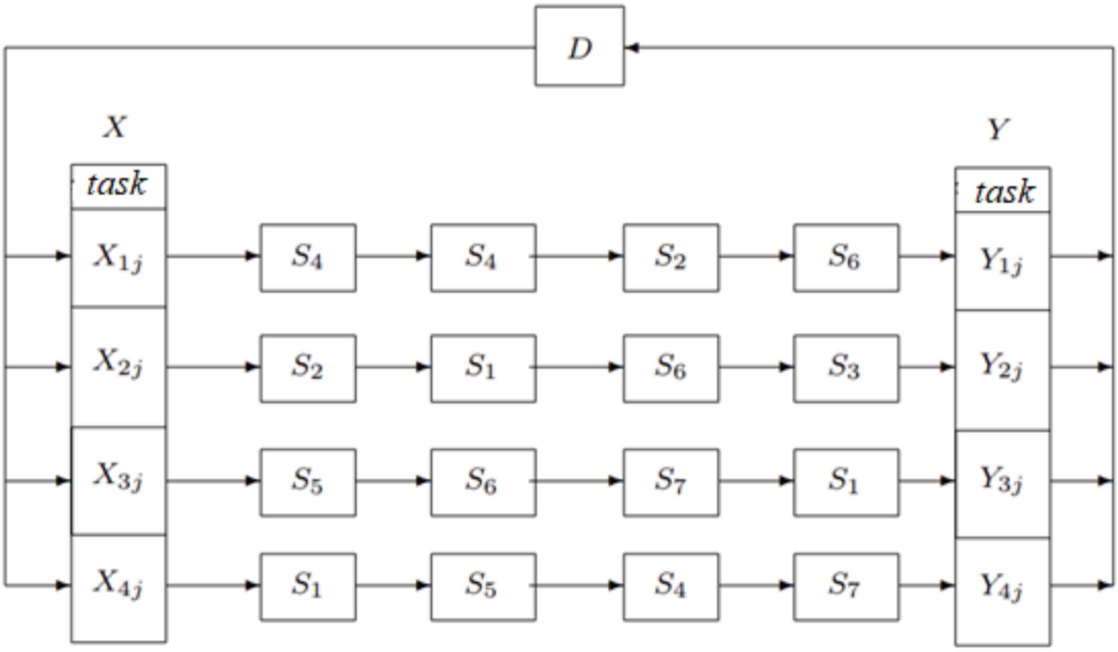


Figure 2: Graphic representation of the schedule of tasks for the *j*-th working day

6. Graphic representation of the schedule-specification

Let's assume that the schedule of tasks of the *j*-th day of the department work for four courses of the bachelor's degree is specified by the data from the table 1 and implemented on the described system.

Table 1
Schedule-specification of the j-th working day of the department for bachelor's degree

j-th day	Task 1		Task 2		Task 3		Task 4	
Lesson	Operation	Server	Operation	Server	Operation	Server	Operation	Server
1	w_1^1	S_1/S_4	w_1^2	S_4/S_2	w_1^3	S_5/S_3	w_1^4	S_7/S_1
2	w_2^1	S_2/S_3	w_2^2	S_5/S_1	w_2^3	S_4/S_6	w_2^4	S_5/S_3
3	w_3^1	S_2/S_4	w_3^2	S_3/S_6	w_3^3	S_7/S_2	w_3^4	S_4/S_2
4	w_4^1	S_3/S_6	w_4^2	S_3/S_5	w_4^3	S_1/S_4	w_4^4	$S_1/S_6/S_7$

Then, using to the notations from (3), we obtain:

$$\begin{aligned}
\Omega_1 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\
\Omega_2 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\
\Omega_3 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\
\Omega_4 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\
\Omega_5 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\
\Omega_6 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\} \\
\Omega_7 &= \{w_1^1, w_1^4, w_2^3, w_4^3, w_4^4\}
\end{aligned} \tag{7}$$

The procedures for the implementation of the plan are given by the following sequences:

$$\begin{aligned}
X_{1j} &= \{w_1^1, w_2^1, w_3^1, w_4^1\} = (S_1 \vee S_4)(S_2 \vee S_3)(S_2 \vee S_4)(S_6 \vee S_3) \\
X_{2j} &= \{w_1^2, w_2^2, w_3^2, w_4^2\} = (S_2 \vee S_4)(S_1 \vee S_5)(S_3 \vee S_6)(S_3 \vee S_5) \\
X_{3j} &= \{w_1^3, w_2^3, w_3^3, w_4^3\} = (S_5 \vee S_3)(S_4 \vee S_6)(S_2 \vee S_7)(S_1 \vee S_4) \\
X_{4j} &= \{w_1^4, w_2^4, w_3^4, w_4^4\} = (S_1 \vee S_7)(S_3 \vee S_5)(S_2 \vee S_4)(S_1 \vee S_6 \vee S_7)
\end{aligned} \tag{8}$$

where X_{ij} defines the i-th task of the j-th day of work, $i = 1,2, \dots ,6$.

Based on the schedule-specification of tasks, defined in table 1, it is possible to implement ways of performing tasks. At the same time, the results of the tasks are recorded on the output server Y, where, in the event of failure to execute any operation, information about this is transmitted to the output server via transit through all intermediate working servers (or in another possible way).

If the failure to complete a task or any operation of such a task is recorded on the output server, a sequence of actions is formed in agreement with the dean's office, which is transferred to the input server for implementation.

In the case of successful execution of the task, information about this is also transmitted to the output server, but such information does not involve a response.

7. Model properties

What can be usefully obtained from the given model, what properties can be followed with its help?

The first thing that emerges from the given model is the possibility to build a convenient schedule for the implementation of tasks for teachers, taking into account the evenness of the distribution of the load on teachers. Indeed, based on what has been said, we get the following ways of performing the first task on the basis of (8):

$$S_1, S_2, S_2, S_6,\ S_4, S_2, S_2, S_6,\ S_1, S_3, S_2, S_6,\ S_4, S_3, S_2, S_6;$$
$$S_1, S_2, S_2, S_3,\ S_4, S_2, S_2, S_3,\ S_1, S_3, S_2, S_3,\ S_4, S_3, S_2, S_3;$$
$$S_1, S_2, S_4, S_6,\ S_4, S_2, S_4, S_6,\ S_1, S_3, S_4, S_6,\ S_4, S_3, S_4, S_6;$$
$$S_1, S_2, S_4, S_3,\ S_4, S_2, S_2, S_3,\ S_1, S_3, S_4, S_3,\ S_4, S_3, S_4, S_3.$$

Among these ways, the one that is most convenient for teachers and to monitor their workload is chosen. Since there are a finite number of these options, their analysis can be performed manually. At the same time, based on information about the qualifications of teachers, their conduct of classes, the level of training of students and other characteristics, it is possible to choose the best way to implement tasks in the selected system model. For example, the implementation of the first task can be done by the sequence S_2, S_4, S_4, S_6 or by the sequence S_4, S_4, S_2, S_6, and also by the sequence S_4, S_2, S_4, S_6, which is obviously not convenient for the teacher S_4.

Second, the opportunity to facilitate the work of methodologists of both the department and the deanery by adding additional information to the graphic image. This information is entered in the event of a disruption in the performance of some operation (for example, the impossibility of the performer's physical presence at the workplace, delay of vehicles in traffic jams, illness of the performer of the operation, absence of students in the classroom, etc.).

Such a modification of the graphic representation of task execution takes the form (see Figure 3).

In this way modified schedule indicates the possible replacement of the operation by other executors who are able to fully execute the operation and possibly the task as a whole.

Thirdly, the presence of such a graphic image makes it easier for the employees of the dean's office to develop the work schedule of both the department and the faculty as a whole. The latter requires graphic images from all departments of the faculty. At the same time, methodologists get the opportunity to develop a schedule with the minimization of the number of teachers during the working day in the event of a shortage of the classroom fund and technical equipment.

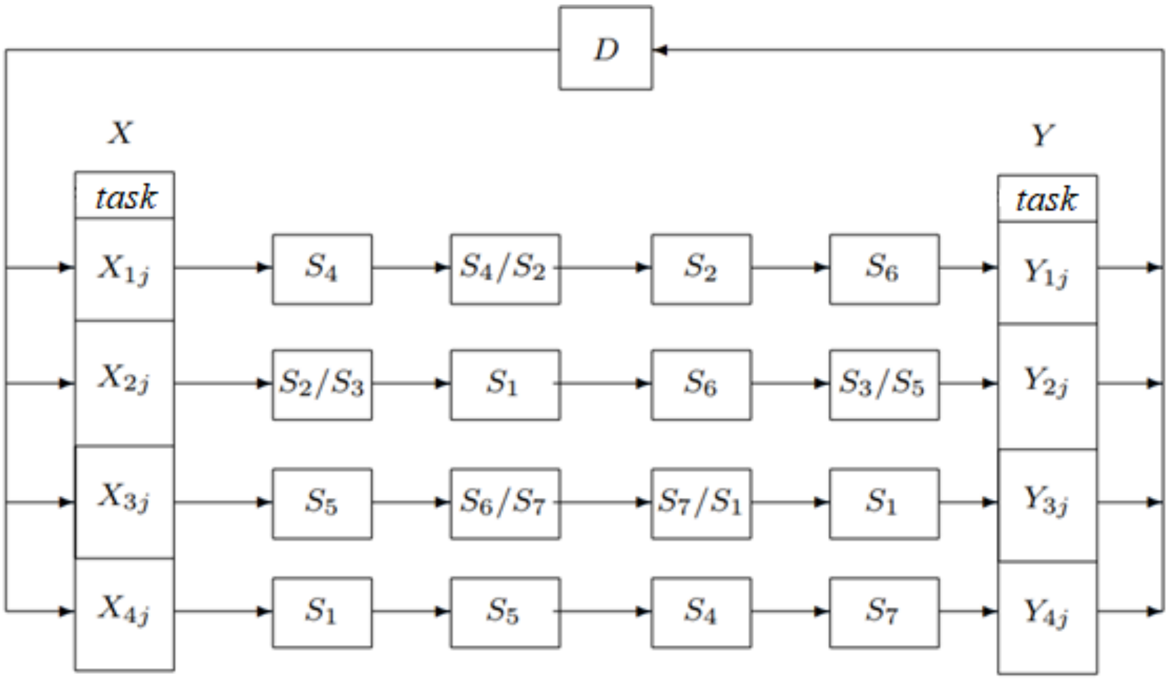


Figure 3: A modified graphic representation of the *j*-day task schedule

8. Automata-network model of the ecosystem

One of the factors determining the efficiency of ecosystems is their structure. To build a qualitative model, it is necessary to use modeling methods that reliably reflect the properties of ecosystems. An important role in the modeling of ecosystems is played by approaches characterizing them as discrete asynchronous models, since these models have all the main properties:

– discreteness (a finite number of components, each of which has a finite number of states);
– existence of the processes and their components, clearly expressed in phases, during which a change of state occurs;
– informing about the completion of each phase of the process;
– consistency (the beginning of each subsequent phase of the process requires the completion of all previous phases necessary for the start of its implementation);

parallelism (possibility of simultaneous transitions in several subprocesses and simultaneous execution);

asynchrony (absence of restrictions on the duration of each subprocess and the transition from one phase to another).

It is necessary to develop opportunities for modeling real ecosystems using these approaches, which will create conditions for improving the quality of their design and functioning.

Despite the considerable duration of theoretical research in the development of mathematical methods for the analysis of discrete asynchronous systems, especially when they are used in computer science for the design of electronic computing machines, this field still needs to be developed. Designing production systems requires in-depth details and adaptation of existing mathematical methods. This direction is still not sufficiently covered in scientific publications. The approach can be used both for the macro-design of production systems as a whole, and for the design of their individual nodes and elements, which will contribute to their quality improvement and create conditions for the direct application of analysis and optimization methods.

Since the execution of the tasks in the system occurs in parallel, we will simulate the process of execution of these tasks by a network of automata, which is a suitable mathematical model for such processes [2]. First, consider an automata representation of the execution of one task in the system.

The automata that simulates the execution of the *k*-th task takes the form:

$$A_{kj} = \left(A = \{D, X_{kj}, W_{kj}, Y_{kj}, Sh_{kj}, w_{kj}^1, w_{kj}^2, w_{kj}^3, w_{kj}^4\}, Z_{kj} = \right. \\ \left. = \{b, b_{kj}, s_{kj}, t_{kj}, a_{kj}^1, a_{kj}^2, a_{kj}^3, a_{kj}^4, b_{kj}^1, b_{kj}^2, b_{kj}^3, b_{kj}^4\}, D, f_{kj}, D\right), \quad (9)$$

where

- state W_{kj} represents a chain of states of the operations execution $w_{kj}^1, w_{kj}^1, w_{kj}^1, w_{kj}^1$;
- states X_{kj}, Y_{kj} represent the input and output servers of the *k*-th task;
- Z_{kj} is an input alphabet of the automata;
- state w_{kj}^r represents the states of the operation w_{kj}^1, $r = 1, 2, 3, 4$;
- state Sh_{kj} represents the execution planning of a task that ended in an failure w_{kj}^r, $r = 1, 2, 3, 4$;
- symbols b, b_{kj}, s_{kj} define the start of the execution and the successful completion of the task, respectively;
- symbols $a_{kj}^1, a_{kj}^2, a_{kj}^3, a_{kj}^4, b_{kj}^1, b_{kj}^2, b_{kj}^3, b_{kj}^4$ define the failure termination of the *r*-th operation and the repeated execution of this operation, respectively.

The initial and final state of the automata is state *D*, and the transition function f_{kj} is depicted by the transition graph in figure 4.

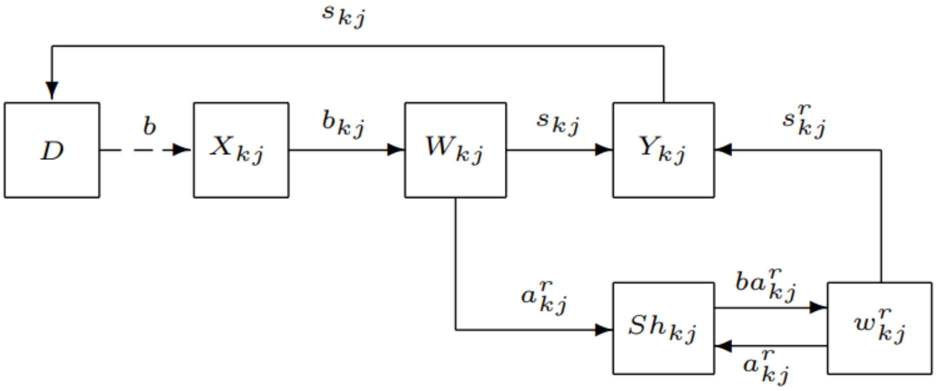


Figure 4: Automata A_{kj} executing the *k*-th task on the *j*-th day

Now it is possible to represent the execution of the tasks of the *j*-th day of the department's work by the following network of automata (see Figure 5).

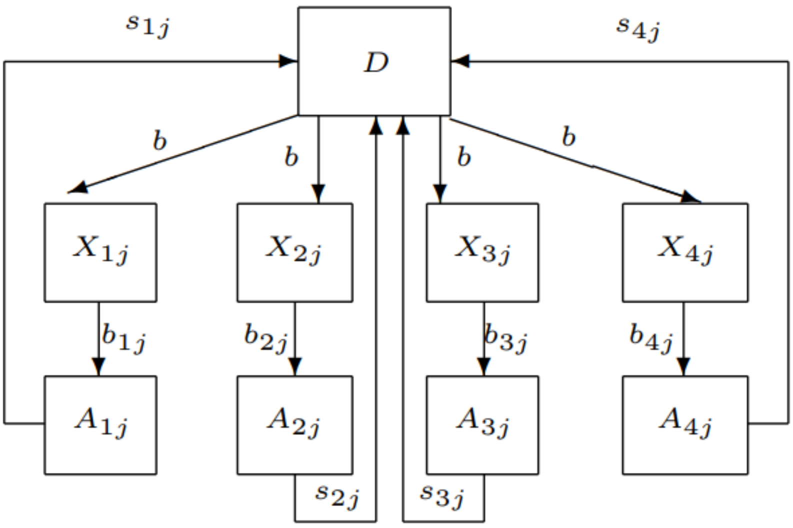


Figure 5: A network of automata that simulates the work of the *j*-th working day of the department

As follows from the above, all processes involved in the model can be automated. And this allows you to control such an object as the "Department" in both manual and remote mode. It is also clear that this model can be applied to modeling the work of the faculty and higher school as a whole.

9. Conclusions

The article provides a formalized definition of the "Department" ecosystem of a higher education school, the components of the ecosystem and their characteristics. The application of the automata-network approach to the study of the ecosystem is considered. A mathematical formalization of presentation and processing of knowledge in the ecosystem using the above approach has been developed. Moreover, the description of ecosystems as an asynchronous discrete process creates conditions for the analysis of their architecture, which can be used at the stages of designing new and reengineering existing systems to increase their efficiency.

10. Acknowledgements

The authors express their sincere gratitude to their respective universities, such as Taras Shevchenko National University of Kyiv and National Aviation University, for their invaluable support and resources that greatly contributed to the completion of this research.

References

[1] Kryvyi S., Opanasenko V., Grinenko E. Logical Approach to the Researcher of Properties of Software Engineering Ecosystem. 11th Int. IEEE Conf, on Dependable Systems, Services and Technologies. (IEEE DESSERT 2020), Kyiv, 2020.
[2] Kryvyi S. Skinchenni avtomaty: teoriya, algoritmi, skladnist, Chernivtsi-Kyiv, Bukrek, 2020.
[3] Baader F., Horrocks I., Carsten L., Sattler U. An Introduction to Description Logics, Cambridge University Press, 2017.
[4] Balmelli L.,Brown D., Cantor M., Mott M. Model-driven systems development. IBM Systems Journal, V. 45, 2006.
[5] Reisig W. Petri nets. An introduction, Springer Verlag: Berlin Heidelberg, 1985.
[6] Arnold A. Finite Transition Systems: Semantics of Communicating Systems, Paris: Prentice Hall, 1994.
[7] Ghosh S., Ramanujam R., Strategies in Games: A Logic-Automata Study. Lectures on Logic and Computation. Berlin: Springer Verlag, 2012.
[8] Kondratenko, Y.P., Gordienko, E., “Implementation of the neural networks for adaptive control system on FPGA,” in 23rd DAAAM International Symposium on Intelligent Manufacturing and Automation, 2012.