ЕКОЛОГІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

UDK 004. 413 (045)

Sidorov N.A., Grinenko O.O.
National Aviation University

SOFTWARE ECOSYSTEM MODELING

The definition of the software ecosystem as object of software ecology is presented; models of software ecosystem, the of main types elements of software ecosystems are considered. The example of Ukrainian software industrial ecosystem is shown.

Наводиться означення екосистеми програмного забезпечення, як об'єкту екології програмного забезпечення; розглядаються засоби моделювання екосистем програмного забезпечення, типи основних елементів екосистем програмного забезпечення; наводиться приклад екосистеми індустрії програмного забезпечення України.

Приводится определение экосистемы программного обеспечения, как обьекта экологии программного обеспечения; рассмтривая средства моделирования розвития экосистем программного обеспечения, типы основных элементов экосистем программного обеспечения; приводится пример экосистемы индустрии программного обеспечения Украины.

Keywords: *sofware, software engineering, software ecology, software ecosystem, modeling.*

Introduction

The application of ecological approach to the software researches is considered and the basic concepts of software engineering branch – software ecology – are formed. Ecological approach to the software research means methodology of the object's research (in this case, software) as the integer part of the environment in question, usually in the form of ecosystem based sustainable development. Ecological approach, complementing the system approach, allows to explore the software and the environment, in which it is developed, used and evolved in new perspectives – as an ecosystem, in evolutionary (historical) view and sustainable development.

There are three reasons to apply the ecological approach to software studying. The first two are related to the concept of sustainable development and are identified by the impact of software products and their production on the environment. Third basic reason is the need of software monitoring, as organized system in the context of the real world. The ecological approach is a ecocentric systematic approach. In a broad sense, the object of the study is the interaction between software and nature. In a narrow sense, the object of study – the interaction of software with the environment. The principles, methods, organizations, objects, subjects and processes of interaction are the subject of the research [1].

Main goal

Software ecology is said to be a new branch of software engineering today. In general, ecology is a science of the relationship between living organisms and their habitats. But the term is now used for software because there is a similarity between living organisms and software: the software has lifecycle, developing programs consume resources, interacting with each other and the environment, habitat (residence) programs – operating environment, runtime system, a community of users and developers and others. Thus, the "software ecology" is a term that affects the field of researches of the properties, behavior and laws of software systems and their impact on the environment and human activities. The objectives of ecology program may include:

- Optimization of technological and design solutions based on minimal damage to the environment;

- Prediction and assessment of possible adverse effects of existing and planned tools on the environment;

- Early detection and correction processes that harm the environment;

- Creation of recycling activity programs (Windows: bin, registry cleaning programs, etc.);

- The fight against software viruses and create systems that are resistant to them;

— - The creation of "environmental" development tools "environmental" programs [8]. Environmental objectives of the software are as follows

- Optimization of design , technology and engineering solutions based on minimal damage to the environment;
- Monitoring , prediction and evaluation of the possible effects of existing and planned software on the environment;
- Early detection and correction processes that harm the environment;
- Creation of recycling of creating and operating software;
- The creation of " ecological " methods , tools and technologies to develop "greener" software.

In studies of software ecology the following three areas are selected: "green" software (software, the use of which has the least impact on nature); ecological software engineering (software development with the least harmful impact on the environment) and software ecosystems. On fig. 1 shows the functional contextual model of software ecology, which is built on the methodology of SADT.

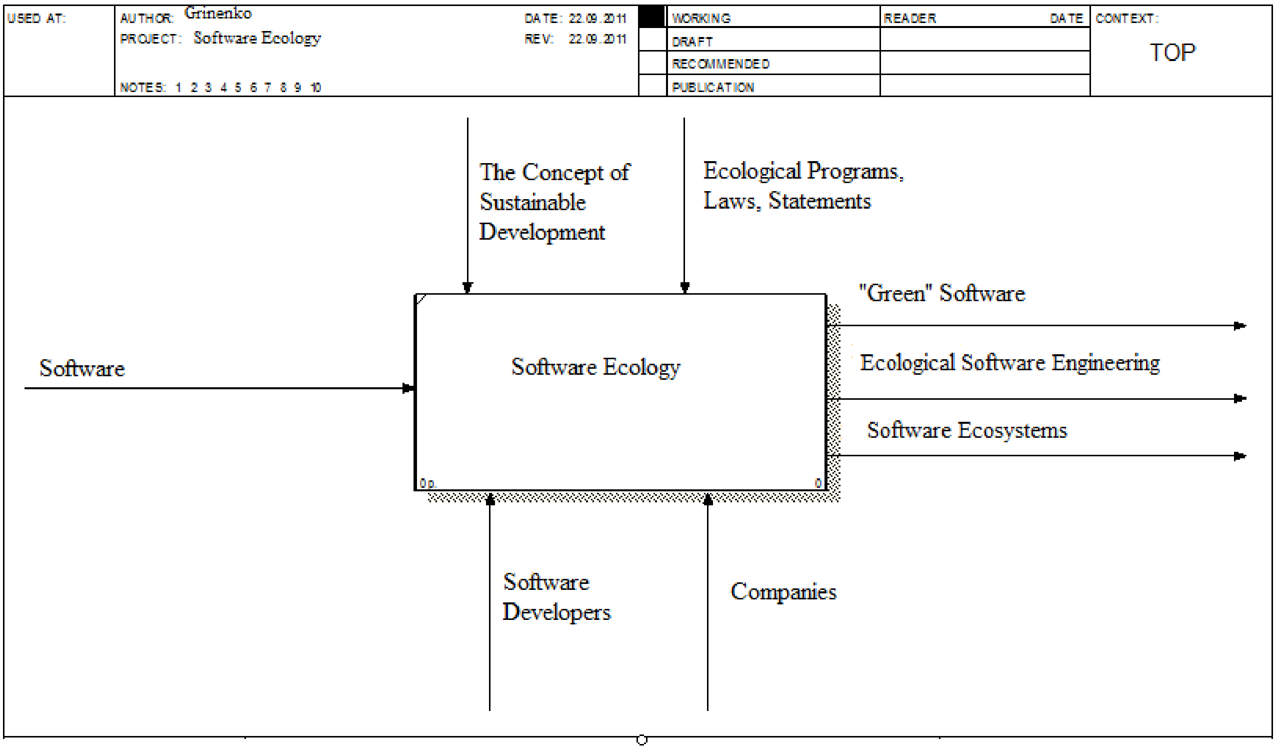


Fig.1 Contectual diagram of functional model of Software Ecology

Software Ecosystem

The notion of ecosystems originates from ecology and means a community of living organisms (plants, animals and microbes) in conjunction with the nonliving components of their environment (things like air, water and mineral soil), interacting as a system. Scientific works that uses the notion of software ecosystem are presented by the following [12-17]. The concept of "software ecosystem" is widely used by companies and researchers of the software. Researches into software ecosystems are represented by several works. Throughout this thesis we will consider a few definitions of software ecosystems. In [12] software ecosystem is a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts. The transaction between two software vendors is centered around software components and software services. In this thesis a definition of software components is used: software component a bundle of software functions accessible through a single interface or carrying a single name which is or can be used as an element in other software packages but of which the core functionality is developed separate from these packages. A software service is a software component accessible via communications outside the users native environment. User can be both human and non-human. Native environment changes depending on the user. If the user is human, the native environment is most probably his or her own computer. If the user is a software package, the native environment consists of the compiled or interpreted code

Another definition of software ecosystem is given in [6]. The discussion of software ecosystems is started from the notion of human ecosystem. A human ecosystem consists of actors, the connections between the actors, the activities by these actors and the transactions along these connections concerning physical or non-physical factors. For the discussion in this paper, we further distinguish between commercial and social ecosystems. In a commercial ecosystem the actors are businesses, suppliers and customers, the factors are goods and services and the transactions include financial transactions, but also information and knowledge sharing, inquiries, pre- and post-sales contacts, etc. Social ecosystems consist of users, their social connections and the exchanges of various forms of information. A software ecosystem consists of the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions. Of course, a software ecosystem is also an ecosystem, specifically a commercial ecosystem, and hence the goods and services are the software solutions and software services that enable, provide support for or automate activities and transactions.

Corporation "Microsoft" defines software ecosystem as a set of interactions and mutual influences of organizations (public, educational and commercial) and individuals that work with software [6].

In [13] software ecosystem is considered as a abstraction level of software products and projects that can be described by analyzing the lower levels. The author proposes a method and means for reverse engineering of ecosystems, which essence is to analyze the information components of projects for high-levels ideas that characterize the organization of software components and determined by the social structure.

In [17] the authors describe software ecosystems and the typical elements of ecosystems and their context, namely:

- software and its role in IT;
- users of software;
- software creating processes;
- management of the establishment and maintenance of software;
- supply and support of software;
- communication network of state agencies;
- software economy.

In the report of the Software Engineering Institute [16], devoted to Ultra-Large-Scale Systems, the authors consider that the industry tends to use concepts of ecosystems to describe the socio-technical systems of software. The authors explain this image of modern software Ultra-Large-Scale Systems scale dynamic communities of independent and competing organisms in a complex changing environment, where people, computing devices and organizations are organisms.

Software ecosystem is an artificial complex that includes the software, environment of its development, operation, maintenance and utilization associated with each exchange of software and intelligence. A software products and services, producers of products and services, customers, contacts are the main elements of ecosystem. For example, managers of software ecosystems have control over the software ecosystem and can develop a strategy to support the vibration of software ecosystems and yield to other organizations in software ecosystem.

Examples of software ecosystem is MySQL /PHP software ecosystem, and iPhone software ecosystem. These examples can be used to establish the typical characteristics of software ecosystems. Software ecosystem may be contained in other software ecosystem such as an software ecosystem Microsoft CRM, which is contained in the full software ecosystem

Microsoft. Another example of such an ecosystem can be a software ecosystem with its iPhone AppStore, which is a closed ecosystem, while software ecosystem MySQL/PHP is open, as organizations have access to the code and a common knowledge base.

Considering the software ecosystem as necessary to investigate internal and external interactions . External interactions are characterized by the presence of other ecosystems that are part of the software or are contiguous with other software . The main objective is to create models of ecosystems and their evolution models . Internal interactions are characterized by the presence of software programs clones application agents , etc. [3]. The aim of this study is its software operation, principles of cooperation programs and more.

Software Ecosystem Modeling

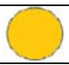
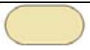
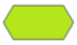
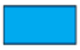
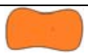
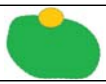
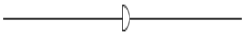
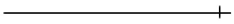
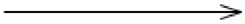
A software ecosystem is an artificial complex that includes software environment of its development, operation, maintenance and disposal that are linked exchange software and intelligence.

The main elements of the ecosystem of software are products and services, manufacturing and services , consumers links. We study the software ecosystem by creating their models.

To describe the software ecosystem simulation tools use the following: i *, UML, Petri nets .

I* is a modeling language used for understanding and modeling of software ecosystems [10]. I * enables developers to get the right information at an early stage of the software development process . Means I * modeling are two basic types of models. The first type is called a model of strategic dependency (SD). It describes the dependency relationships between network participants (actors) of the system. The second type is called a sound strategic model (SR). It provides a representation of how each actor achieves its goal. Model SR includes SD model. The elements of the SR model is the aims, objectives and resources. Thus , the elements of SR model can be presented as in table 1.

Table 1
Elements of SR model

№ п/п	Graphical presentation	Meaning
1		actor
2		goal
3		task
4		resourse
5		aim (satisfaction of needs)
6		actor with boundary
7		dependency link
8		task decomposition link
9		means-ends link

Thus, using the notation i *, SR model example of software ecosystem is shown on fig. 2

In UML a software ecosystem model can be represented by Use Case diagram (Fig. 3). Charts precedents (Use Case Diagram) is a graphical means of specifying requirements that are used to determine the following:

- General scope and context of the subject area;
- General requirements for the functional behavior of the system, which is projected;
- The interaction of the system with an external світом.

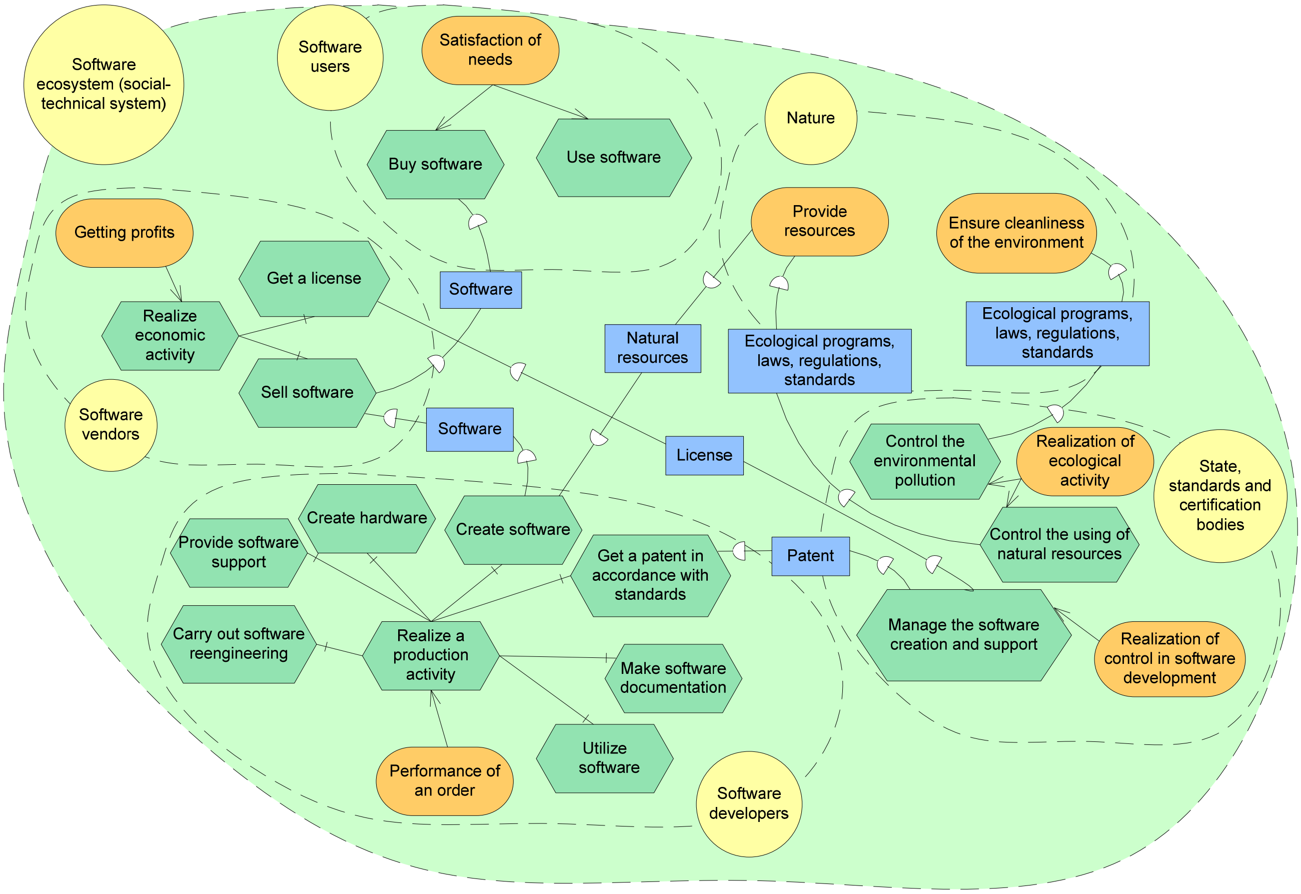

Software ecosystem (social-technical system)
Software users
Satisfaction of needs
Buy software
Use software
Nature
Provide resources
Ensure cleanliness of the environment
Getting profits
Get a license
Software
Realize economic activity
Sell software
Natural resources
Ecological programs, laws, regulations, standards
Ecological programs, laws, regulations, standards
Software vendors
Software
License
Control the environmental pollution
Realization of ecological activity
State, standards and certification bodies
Create hardware
Create software
Provide software support
Get a patent in accordance with standards
Patent
Control the using of natural resources
Carry out software reengineering
Realize a production activity
Manage the software creation and support
Realization of control in software development
Make software documentation
Utilize software
Performance of an order
Software developers

Fig. 2. SR model of software ecosystem

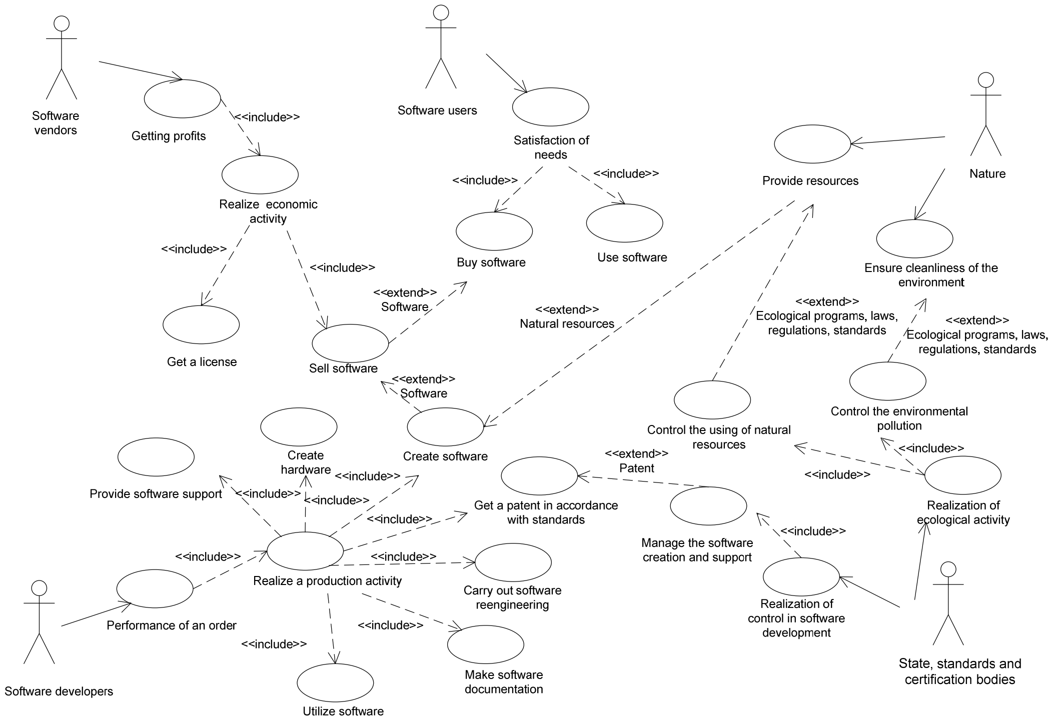


Fig. 3. Use case diagram of software ecosystem

Petri nets

In practice, modeling of systems often have to solve problems related to the formalized description and analysis of causal relationships in complex systems that simultaneously runs multiple processes in parallel. Petri net is the most common nowadays formalism that describes the structure and interaction of parallel systems and processes [5].

Formally, Petri net is defined by four types:

$G(V, E, I, O)$, where

V – set of events,

E – set of transitions, $E \neq \emptyset$, $V \cap E \neq \emptyset$,

I – input function (direct function of incidence)

$I: V \times E \rightarrow \{0,1\}$,

O – output function (the inverse function of incidence)

$O: E \times V \rightarrow \{0,1\}$.

$I(e_j) = \{v_i \in V \mid I(v_i, e_j) = 1\}$,

$O(e_j) = \{v_i \in V \mid O(e_j, v_i) = 1\}$,

$I(v_i) = \{e_j \in E \mid I(e_j, v_i) = 1\}$,

$O(v_i) = \{e_j \in E \mid O(v_i, e_j) = 1\}$,

$i = 1, \dots, n, \quad j = 1, \dots, m, \quad n = |V|, m = |E|$

.

For this system, its goals and objectives will be presented a plurality of positions and relations between objects are set of transitions.

Graphically model system is depicted as a bipartite oriented multyhrafiv (since it assumes the existence of multiple arcs from one vertex to another), which is a set of positions and transitions (Fig. 4).

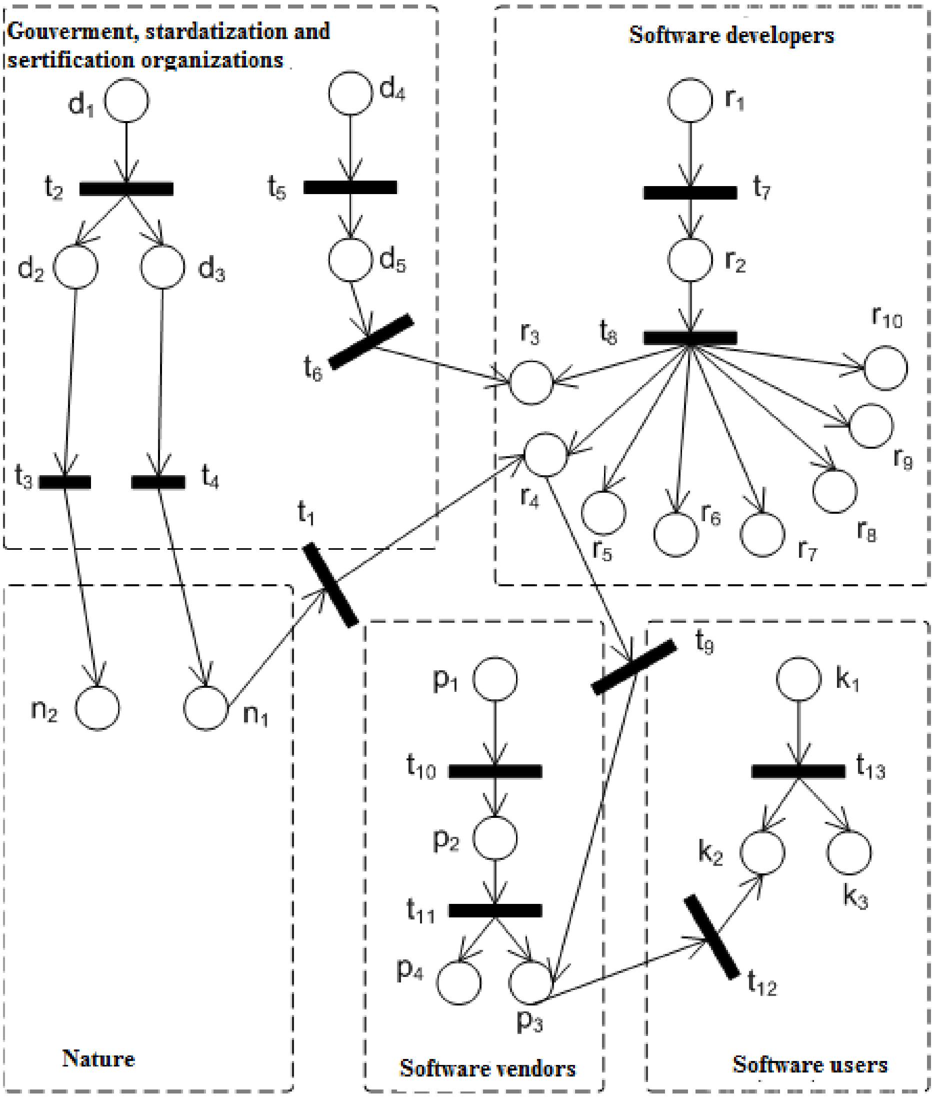


Fig. 4. Petri nets of software ecosystem.

Analytical representation system using Petri nets have the following form:

$$V = \{n_i, d_j, r_s, p_q, k_t \mid i = \overline{1,2}; j = \overline{1,5}; s = \overline{1,10}; q = \overline{1,4}; t = \overline{1,3}\};$$

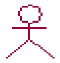
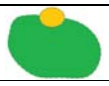
$$m = \overline{1,6} \quad \};$$

$$I(t_1) = \{n_1\} \qquad O(t_1) = \{r_4\}$$
$$I(t_2) = \{1\} \qquad O(t_2) = \{d_2, d_3\}$$
$$I(t_3) = \{d_2\} \qquad O(t_3) = \{n_2\}$$
$$I(t_4) = \{d_3\} \qquad O(t_4) = \{n_1\}$$
$$I(t_5) = \{d_4\} \qquad O(t_5) = \{d_5\}$$
$$I(t_6) = \{d_5\} \qquad O(t_6) = \{r_3\}$$
$$I(t_7) = \{r_1\} \qquad O(t_7) = \{r_2\}$$
$$I(t_8) = \{r_2\} \qquad O(t_8) = \{r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}\}$$
$$I(t_9) = \{r_4\} \qquad O(t_9) = \{p_3\}$$
$$I(t_{10}) = \{p_1\} \qquad O(t_{10}) = \{p_2\}$$
$$I(t_{11}) = \{p_2\} \qquad O(t_{11}) = \{p_3, p_4\}$$
$$I(t_{12}) = \{p_3\} \qquad O(t_{12}) = \{k_2\}$$
$$I(t_{13}) = \{k_1\} \qquad O(t_{13}) = \{k_2, k_3\}$$

Petri nets are used to describe the events of arbitrary duration. In this case, the model is built using Petri net reflects only the order of occurrence of events in the studied system. To display the temporal parameters of the system, which is modeled using Petri nets, using the expansion device Petri nets: net time, E - network, network, etc. Merlin.

The means of correspondence between the simulation are presented in table. 2.

Table2
Correspondence between ecosystem modeling tools

Elements of ecosystem	I*	UML	I
actor	●		——
goal	⬭	⬭	◉
task	⬡	⬭	○
resourse	🟦	——	——
actor with boundary		——	——
dependency link	——D——	<<extend>> - - - - - - - - ->	
task decomposition link	————+	<<include>> - - - - - - - - ->	
means-ends link	————>		

In [4] software ecosystem is considered in terms of the concept of systems (System of systems). System of Systems is any system composed of systems that are themselves autonomous. Under the system refers to any group of interacting or interdependent entities, which forms a purposeful unity. Under autonomy refers to the ability of each system to perform independent actions or decisions. Operational management and evolutionary independence and unexpected behavior arising from autonomy of the system components . All of the above makes it possible to consider the ecosystem of software as a system of systems, autonomous components which will be considered:

- Software (technical abstract system);

- Hardware (technical material system);

- Natural objects and phenomena (natural system);

- The state standards bodies , manufacturers and vendors , users of the software (the economic system);

- Developers and managers, social communities (social system).

Ecosystem software industry Ukraine

Let's consider the model of ecosystem of software industry of Ukraine (fig. 5) [1 , 2]. Main activities include various types of economic activities of individuals and organizations (vendors, software developers , service providers and network providers , consumers, competitors and other related industries and businesses, government agencies, industry). Together with mutual competition and cooperation, common development and evolution, as well as the specific features of the economy , they form a coherent whole. In an ecosystem of software industry , the major players are suppliers, developers staple , service providers , financial institutions (venture capital investor, bank, etc.) and government organizations (government, trade associations, organizations that set standards, etc.) and consumers. A simplified model of the ecosystem of software industry of Ukraine is given in Fig. 5

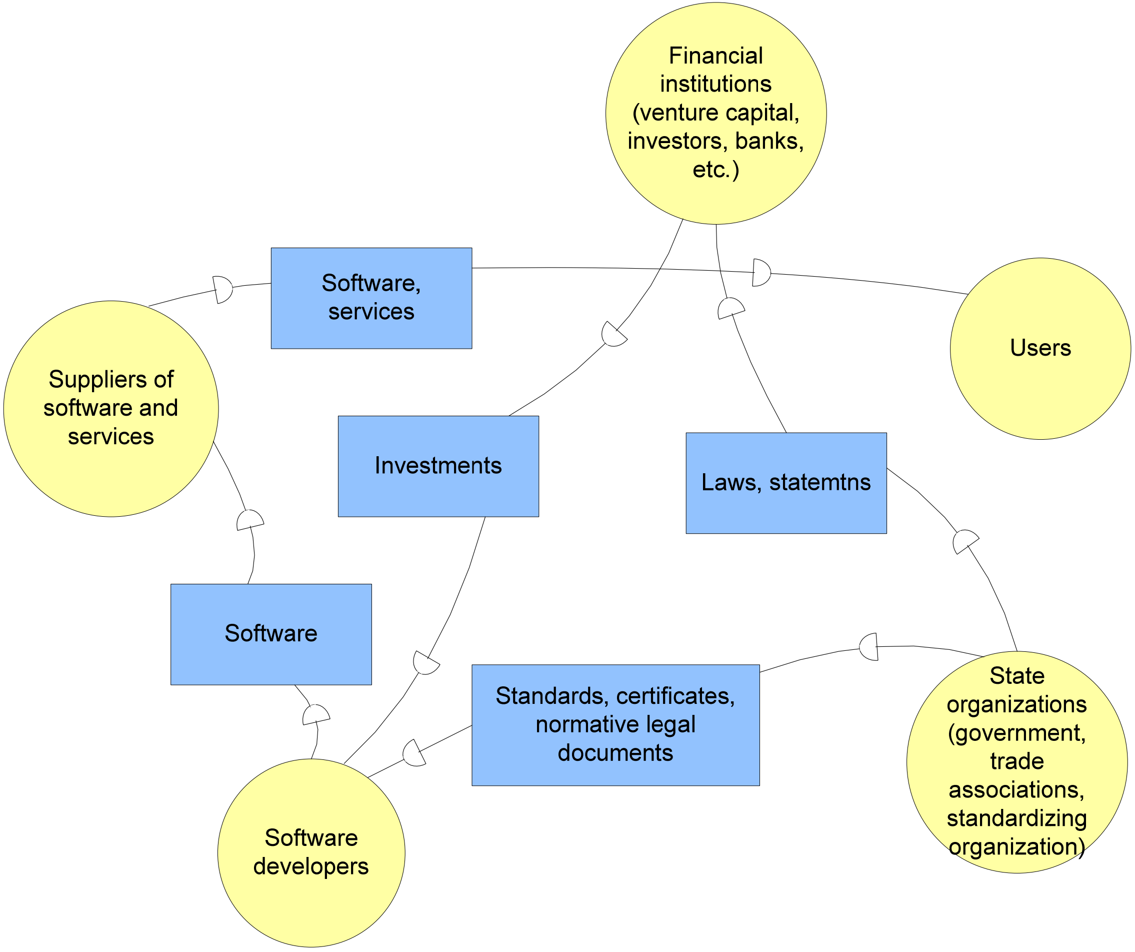

Financial institutions (venture capital, investors, banks, etc.)
Software, services
Users
Suppliers of software and services
Investments
Laws, statemtns
Software
Standards, certificates, normative legal documents
State organizations (government, trade associations, standardizing organization)
Software developers

Fig. 5. Ecosystem model of software industry of Ukraine

Consider some of the reviews for the software industry for Ukraine given model. So that, Marco Iansiti and Ray Levien offered important criteria for evaluating the "health" business ecosystem [11].

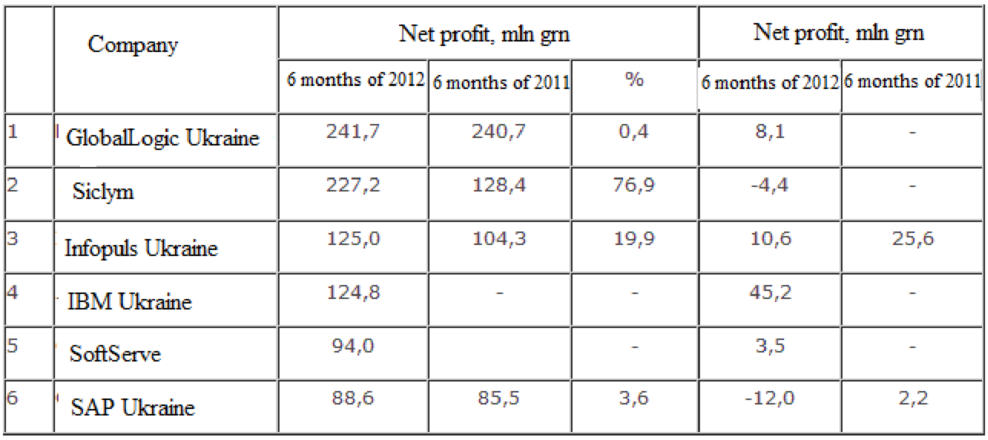
Capacity shows the results of statistical analysis on a national scale product development and software services in Ukraine during the period.

Variety criterion shows the ranking of IT companies according to the following indices: "Revenue from software", "Revenue from system integration" and "Income from software services." In 2012 according to data from IT-CATALOGUE IT market in Ukraine 2495 companies represented, of which 1284 - the company software development , 249 - software vendors and HW, 361 – Telecommunications, 177 - host providers, 168 - IT education, 292 - IT services, 57 - IT recruitment.

According to the "Top - 100" leading Ukrainian software developers in 2012, the following companies are (table 3):

Table 3

Variety



	Company	Net profit, mln grn			Net profit, mln grn	
		6 months of 2012	6 months of 2011	%	6 months of 2012	6 months of 2011
1	GlobalLogic Ukraine	241,7	240,7	0,4	8,1	-
2	Siclym	227,2	128,4	76,9	-4,4	-
3	Infopuls Ukraine	125,0	104,3	19,9	10,6	25,6
4	IBM Ukraine	124,8	-	-	45,2	-
5	SoftServe	94,0		-	3,5	-
6	SAP Ukraine	88,6	85,5	3,6	-12,0	2,2

Flexibility determines what sector of market has the most faster growing (figure 6).

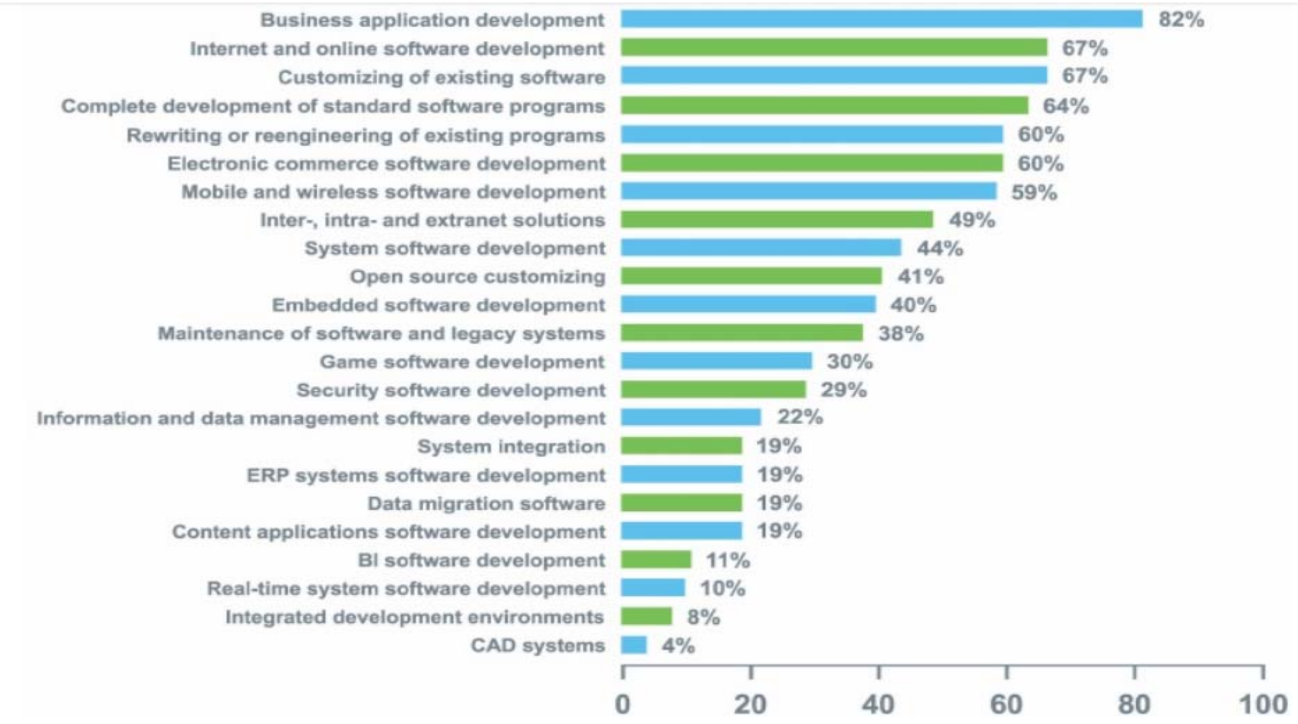


Fig. 6. Types of software development provided by Ukrainian companies (2012)

Performance determines the impact of investment in the sector. Since 2000, the traditional leader among Ukrainian consumers of services were United States. During the last 2-3 years, European consumers have become more active. Currently, the share of the EU consumption of services increases (Fig. 7). This is primarily dependent on the geographical and cultural proximity to Ukraine and other European countries, as well as the ability to more easily develop sales network in Europe.

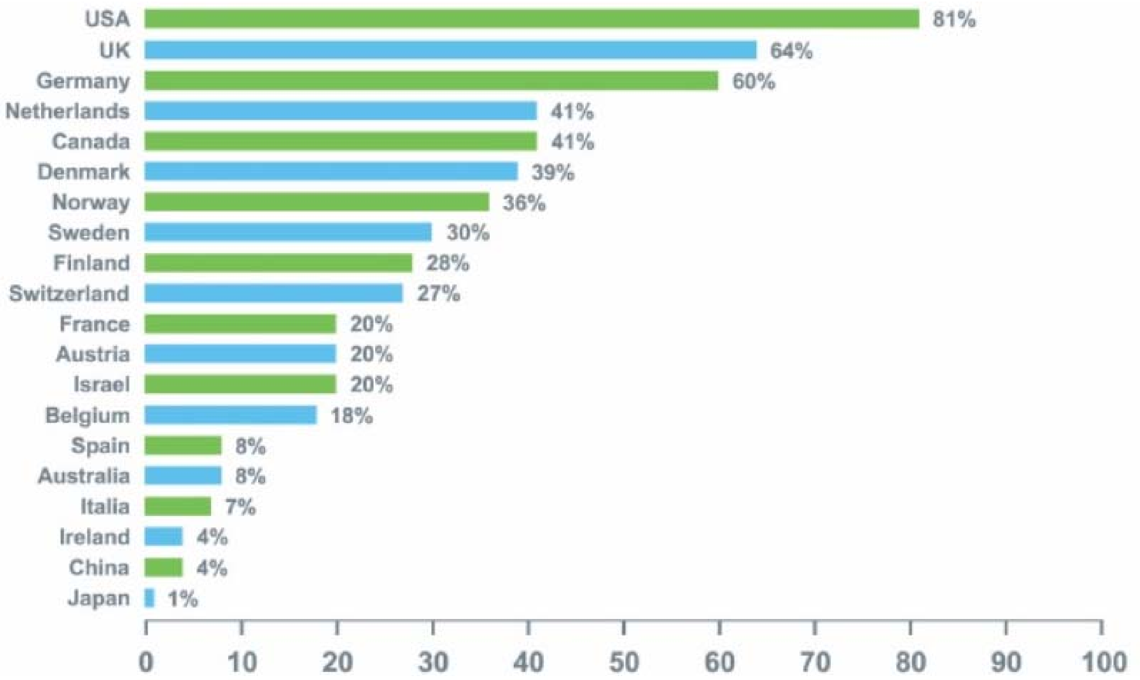


Fig. 7. The main consumer countries of Ukrainian Services

The majority of the customers of Ukrainian companies were small and medium-sized firms. With the adoption of several laws of Ukraine to support the IT services and software development , global companies have begun to explore the Ukrainian market and the establishment of R & D centers in Ukraine . Examples of global companies that work with the Ukrainian software developers are : ABBYY Ukraine, AMD, ASBIS Ukraine , ASUS, Canon, Datalux, Dell, DiaWest, ELKO, Epson, ERC, IBM, Intel, KM DISTI, Lenovo, Lexmark, LG, Microsoft, MTI, NIS, OKI, Panasonic, Prexim-D, Samsung, Schneider Electric Ukraine, SI BIS, VERNA, AMI , VCT , K -Trade Lab in Ukraine , Genesys, Autodesk, Beko, Bloomberg, BNP Paribas, Boeing , Bombardier, Bosch, Cisco, Comoto, Dell, Deloitte, Deutsche Bank, DHL, Ebay, HP, Ikea, Jaguar, Kodak, MTV, Nokia, Novell, Skype, Sony, Symantec, Volvo, VAB, USAID.

Viability (vividness) determines the total income in all sectors of the IT market.

Conclusions

The software creates organizational and technical systems that appear , evolve and degrade like natural biosystems . Ecology Software - a term that affects the industry study of the properties , behavior and laws of software systems and their impact on habitat and human activities.

A key role in the study of the ecology of software ecosystems play software. Further research ecosystem of software may be associated with the identification and formalization of ecosystem types of software, development of recommendations for creating and maintaining the ecosystem of software and so on. Given the importance of the well-known concept of sustainable development , it is necessary to consider and develop software as part of aviation engineering in the context of eco -efficiency , eco-justice and eco- efficiency. To achieve this goal are important ecosystem software.

References

1. Сидоров Н.А. Экология программного обеспечения //Інженерія програмного забезпечення. – Київ. – 2010. – № 2. – С. 1 – 9.

2. D.G. Messershmitt, C. Szyperski. Software Ecosystems: Understanding an Indispensable Technology and Industry. – London: MIT press, 2003. – 233 p.

3. Любимский Э.З. На пути к построению общества программ //Программирование. – 2009. - №1. – С. 4 – 10.

4. Луцький М.Г. Методологія інформатизації науково-практичного супроводження експлуатації авіаційної техніки// Інженерія програмного забезпечення. – Київ. – 2011. – № 1. – С. 25 – 35.

5. Советов Б.Я., Яковлев С.А. Моделирование систем. – М.: Высшая школа, 2001. – 344 с.

6. Хоменко В.А. Экосистемы програмного обеспечения// Вісник НТУ. – Київ, 2011. - № 23. – С. 114 – 118.

7. Чернышов Л.Н. Экология программирования. – М., 2008.

8. Bosch J. From Software Product Lines to Software Ecosystems// 13th International Software Product Line Conference. – San Francisco, August 24 – 28, 2009. – P. 1 – 10.

9. Boucharas V. Formalizing software ecosystem modeling. // Proceedings of the 1st international workshop on Open component ecosystems. – P. 41 – 50.

10. Duinkerken W. Transaction Cost Economics in Software Ecosystems: some empirical evidence. – April 20, 2009. – P. 22 – 37.

11. Jansen S, Bosch J. Understanding Software Ecosystems: A Strategic Modeling Approach. // Proceedings of the Workshop on Software Ecosystems 2011. – P. 65 – 76.

12. Jansen S., Finkelstein A., Brinkkemper. A sense of community: A research agenda for software ecosystems // 31th International Conference on Software Engineering, New and Emerging Research Track. – 2009. – P. 12 – 16.

13. M.F. Lungu. Reverse Engineering Software Ecosystems. – Doct. Diss. – USI. – 2009. – 208 p.

14. G. Schulz. The Green and Virtual Data Center. // Taylors Francis G. – 2009. – 218 p.

15. L. Webber, M. Wallance Green Tech: thow two and to plan implement sustain – able IT solutions. – FBACOM. – 2009. – 29 p.

Information about authors

Sidorov Nikolay Aleksandrovich – Doctor of Technical Sciences., Professor, Dean of Software Engineering Department of the National Aviation University. Scientific interests: software engineering.

E-mail: nikolay.sidorov@livenau.net

Grinenko Olena Oleksandrivna – post-graduate student of the National Aviation University. Scientific interests: software engineering, software ecology, domain analysis.

E-mail: Elena.Grinenko@livenau.net