

Informatics: Formation of computer software and technologies of software systems¹

^{1,3} E.M. Lavrisheva <lavr@ispras.ru>

^{1,2,4} A.K. Petrenko <petrenko@ispras.ru>

¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*

² *Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia*

³ *Moscow Institute of Physics and Technology (State University) 9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia*

⁴ *National Research University Higher School of Economics (HSE) 11 Myasnitskaya Ulitsa, Moscow, 101000, Russia*

Abstract. Formation of Informatics and aspects of computer software development, in particular, operating systems and information systems since the period of appearance of the first computers of 1948-1990 is considered. The program of informatization of Russia in 1992 and the development of the fundamentals of modern computer science or informatics and the intellectualization of the development of various types of software systems.

Keywords: computer science; computer; system and software engineering; technology; science; information; intelligent systems; digital economy.

DOI: 10.15514/ISPRAS-2018-30(5)-1

For citation: Lavrishcheva E. M., Petrenko A. K. Informatics: Formation of computer software and technologies of software systems. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 5, 2018, pp. 7-30. DOI: 10.15514/ISPRAS-2018-30(5)-1

1. Introduction

The emergence of informatics as an independent scientific and engineering discipline was preceded by a period of a large number of theoretical and practical works, the most important of which include the works by C. Babbage, G. Boole, K. Zuse, J. von Neumann, N. Wiener, A. Turing, and others.

The foundation for the development of informatics became research on cybernetics. N. Wiener in his book «Cybernetics: Or Control and Communication in the Animal

and the Machine» (1948) defined the term Cybernetics as the science of information, control and communication in animal and inanimate systems. Computer is a tool of this science, control and information processing processes should be based on information about the living nature, be performed using arithmetic operations on data, reflect human mental activity by means of a natural language.

The groundwork for informatics in the USSR was laid in the works by scientists and engineers of the first half of the last century. Historians of science and technology note the significant contribution of such scientists as A.A. Lyapunov, A.P. Ershov, V.M. Glushkov and many others. In the 1950s, the English terms «computer science» appeared. The «software engineering» term appeared in 1968.

Actually, the «informatics» term became widely known in European countries after the appearance in 1973 of the book by F. Bauer and G. Gooz (in Russian translation «Informatics. Introductory Course», Mir, M., 1976).

In the aforementioned book, F. Bauer [1] for the first line of the preface called computer science as informatics, which includes communication technology to determine physiology, psychology, and neurobiology. Informatics touches coding theory, information theory, logical calculus, automata theory, algorithms, algorithmic and intelligent systems. Many of these theories and concepts were formed during the creation of the first computers and promoted the emergence of information systems and technologies, neural systems, nanosystems, etc. In the period when first computers appeared, the ideas of intelligent, smart machines were also formed, solving mathematical problems of algebraic formulation in a special language called «Analyst» on the Mir 1-3 computer (the 1960s) [2], neurocomputers of N.M. Amosov and robots.

A lot of research in the USSR contributed the formation of informatics and cybernetics. The work by D.A. Pospelov «The Formation of Informatics in Russia» (<http://www.raai.org/about/persons/pospelov/pages.htm>), as well as the USSR conferences on the subject held by him for many years, can be named as particularly publications and in the book of D. A. Pospelov and Y. I. Fet «Essays on the history of Informatics in Russia». IIS SB RAS, 1998, etc. Cybernetics and Informatics, software engineering technologies for newly created computers occupies an important place. Brief description of the types of domestic programming technologies and scientific foundations of informatics are given below.

2. Programming technologies in the USSR in the 1950s and early 1960s

2.1. Programming systems and libraries of programs

Initially, programs for solving various tasks for computers were coded using simple languages such as AutoCode close to a machine language. The first programming program (PP) for A.A. Lyapunov language of operator calculus was developed in 1953-1954 at Moscow State University (E.Z. Lyubimsky, A.P. Ershov).

¹ This work is supported by RFBR grant 16-01-00352

The library method (V.M. Glushkov «On a Method of Programming Automation», Cybernetics Problems, Issue 2, 1957) is created for solving systems of differential equations for computers (MESM, Ural 1, UMSHn, Dnepr, etc.).

Reusable subprograms were collected in subprogram libraries (V.M. Kurochkin, Computing Center of AS of USSR). M.R. Shura-Bura and E.A. Zhogolev developed the technological approach to the management of program libraries in the works. One of the outstanding results in this direction was the so-called «Interpretive System» IS-2 (M.R. Shura-Bura) for the computer Strela.

By the early 1960s, programming languages Algol-60, Fortran, PL/1, Cobol, etc. appeared. Operating systems and translators from the Algol-60 (TA) language were developed for all domestic computers to work with programs in programming languages [5–9].

TA1– S.S. Lavrov (1962).

- TA2 – M.R. Shura-Bura and E.Z. Lyubimsky (IAM, 1963).
- TA3 – (Alpha-system) Russian version of the Algol-60 language, A.P. Ershov (Siberian Branch of the USSR Academy of Sciences, 1964).
- TA4 – E.L. Yushchenko, E.M. Lavrisheva for Dnepr-2 controlled computing complex (Institute of Cybernetics of the Academy of Sciences of the Ukrainian SSR).
- Translator with ALMO (E.Z. Lyubimsky, 1965).

The development of translators built a strong basis for research in the compiler technologies, which are actively developing in our country at the present time.

2.2. Operating systems for BESM-6

The first operating systems in the USSR were created for the BESM computer under the supervision of S.A. Lebedev and M.V. Keldysh as well as at the Joint Institute for Nuclear Research (N.N. Govorun) in Dubna, starting from the 1960s [3, 4].

1) OS-68 or Dispatcher 68 (L.N. Korolev, V.P. Ivannikov, A.N. Tomilin, and M.G. Tchaikovsky). OS provided with:

- Multi-program problem solving;
- Management of all communication channels and devices;
- Parallel execution of tasks and device operation;
- Dynamic allocation of resources in RAM and external memory;
- Distribution and buffering of input-output;
- Operator and interactive mode control. Dispatcher 68 (ND-70) became the basis of the BESM-6 OS, which has been used in the flight control centers of spacecraft, ballistic and telemetric software systems for more than 20 years.

2) IAM OS for BESM-6 began to operate in 1970 (S.S. Kamynin, E.Z. Lyubimsky, and I.B. Zadykhaylo). In this OS, resource management is implemented with the

means of process synchronization and emergency management, and a system of programming and debugging programs in programming languages is included in it.

3) Monitor – Dubna (N.N. Govorun and V.P. Shirikov) ensured management of tasks and library programs creation, including the European programs of CERN and the broadcasting of programs in Fortran, Algol, Pascal, and other programming languages.

The operating systems for BESM-6 supported smart interrupt handling systems, memory protection, real-time and time-sharing modes.

It should be additionally noted that in the mid-1960s (a little bit earlier than OS-68) M.R. Shura-Bura and V.S. Shtarkman (Institute of Applied Mathematics - IAM AS USSR) developed the operating system for the Vesna computer, but due to the security reason first and very brief information about this work was published much later.

3. Programming technologies in the early 1970s and 1980s

3.1. Operating systems in 1980-s

A further development of the research in operating systems became OS for multi-machine complex AS-6, in which various types of computers were combined. AS-6 OS design used experience gained in ND-70 development and exploitation. Work on the AS-6 OS has determined the direction of development of the OS and programming systems with OOP for the coming years.

3.2. Compilers in 1980-s

Translator from ALGOL-68 made at Leningrad State University (G.S. Tseitin, A.N. Terekhov, 1968-1991) for the first computers – CM1, CM2, ES EVM, and Sampson (HLL soviet computer). Programs in ALGOL-68 were translated into an intermediate language, and then into a specific machine code.

3.3. Modular and assembly programming

On first domestic computers, programs were initially implemented in machine codes or languages like AutoCode. Later, the programming languages appeared (Algol, Fortran, PL/1, Cobol, etc.). The above-mentioned translators from high-level programming languages had been used on domestic computers and had provided solutions to various kinds of problem domains (mathematical, physical, economic, etc.) before the ES EVM (clones of IBM's System/360 and System/370) appeared.

Within the defense industry specific computers for radars, spaceships, airborne equipment and software complexes PROTVA, YAUZA, RUZA, PROMETHEUS (Moscow Research Institute of Instrument Automatics, MNIIPA, V.V. Lipaev) were created. These complicated systems had a modular architecture, were written in different programming languages and were assembled into complexes using interfaces of the APROP system of the PROMETHEY project (1980-1990) [11-13].

The interface has become the main element of the program assembly and the foundation of assembly programming. Its development was influenced by V.M. Glushkov, V.V. Lipaev, A.P. Ershov, E.H. Tyugu, and others. The APROP system became a CASE-tool for automating the assembly process using primitive functions of the interface library for converting inconsistent data types in modules in the programming language (Lavrishcheva E.M., Grishchenko V.N. *Linking Multilingual Modules in the Unified Computers Operating System*. M.: Finance and Statistics, 1982. 127 p.). The development of this system and its use have been funded by the USSR Ministry of Radio Industry for more than 10 years, being an integral part of the PROMETHEUS, YAUZA, RUZA, and other systems. The APROP systems were transferred to Yerevan Research and Development Center

(1984) for using in other Software systems. The main developers of this complex were given the Award of the USSR Cabinet of Ministers (1985). This system was transferred in 52 organizations of the USSR and became an integral part of the ES EVM programming systems. Academician A.P. Ershov considered “assembly programming effective, since ready-made modules make it possible to quickly solve any tasks from a specific problem area for the ES EVM and mini-, micro- and macro computers” (A.P. Ershov. *Scientific basis of programming*. Report in the USSR Academy of Sciences, 1985, pp. 1-12.).

The results of the research and development in the assembly programming were defended in the Dissertations E.M. Lavrisheva (*Models, Methods and Means of Assembly Programming*, 1988), Grishchenko V.N. (*Methods and Means for Providing the Intermodule Interface in the Class of High Level Programming Languages*, 1990) and published in their monograph (*Assembling programming*, 1991, 136p.) and others articles [25-29]. The implemented interface is built into IBM OS 360 and other foreign operating environments. Later, the ISO/IEC FDIS 24765 – Configuration (assembling), 2009 standard was developed. There was no national standard for the assembly method, and only the domestic assembly concept was published in a number of articles in the *Programming and Compute Software Journal* (Academy of Sciences of USSR), in books “*Technology of assembling programming*” by V.V. Lipaev, A.A. Shtrik, B.A. Pozin, published in 1992 and presented in many papers for Scientific conferences in 1976, 1978, 1980, 1986, 1991.

3.4. Programming synthesis

Synthesizing programming (E.H. Tyugu) is the process of obtaining a program from the problem statement and the method of its solution. The synthesis of programs based on logical and analytical specifications and consists in proving the existence theorem and transforming these specifications into a problem-solving program. In case of a logical approach, the specification is interpreted as the formulation of a theorem stating the existence of a solution to the problem. In the case of the analytical approach, the specification is interpreted as an equation for the symbolic transformation of the program with validation. Program synthesis is a certain way of manipulating knowledge embodied in the domain problem specification and universal

knowledge that reflects general mathematical patterns and the essence of evidence-based reasoning [13, 17].

3.5. Compositional programming

The composition of programs is the operation of combining functions and data in one of the forms: “data – function – name” and “function-composition-description” on a set of named data, descriptions and denotations (values). Composition operations form a subclass of standard compositions and compositional functions at the programming language level (V.N. Red’ko). The theory of descriptive and declarative programmatic formalisms (N.S. Nikitchenko) provides adequate modeling of data structures, programs, and means of their design. The basis of this theory is semantics, a system of compositions and nominative data for describing functions and compositions. The theory is closely related to function and data algebra and differs from traditional program systems by a theoretical-functional approach, classes of single-valued functions of arity n with nominative data structures such as sets, multisets, relations, etc. An experimental system that displays functions and data compositions, which was used in teaching students of KNU [18-21] and France, had been created.

3.6. Functional programming – REFAL

Functional programming is used to solve problems related to pattern recognition, communication in the natural language, implementation of expert systems, proof of theorems, symbolic calculations, etc. The basis of this programming is the theory of lambda calculus (A. Church) and combinatorial logic (M. Schönfinkel and H. Curry). The functional program includes a set of definitions of functions, calls to other functions and control calls. Recursion is the fundamental basis for constructing semantics in functional programming. In domestic practice, the REFAL language (V. Turchin, 1966) is considered a functional one [22]. A function in it is an ordered set of sentences consisting of a sample and a pattern. Some expression is supplied to the input of the function. It is matched with the sample and if the matching is successful, then this expression will be the result of the function or an error will be recorded. An expression is a sequence of terms that can be letters, numbers, and label characters (identifiers), macro numbers – a character (digital) record of non-negative integers, floating-point numbers, variables from one of several predefined types, and an arbitrary expression in parentheses or in angle brackets to call functions, including recursive ones. This language is still being developed and is practically used in a number of industrial areas [23].

3.7. Layered (aspect oriented) programming

A. L. Fuksman [24] developed the technology of distributed actions or the vertical layering technology. According to this technology, the vertical layer (slice) contains a set of distributed actions, code fragments that implement a certain expanding function, while the process of developing and modifying a program is a sequence of

operations for adding or changing these functions. This concept forms the basis for a new strategy for incremental program development. The first stage has a “basis” created – a simplified version of the program. At subsequent stages, the program is implemented, and it is added to the vertical layer. The program may contain “stubs” – simulators of missing parts for inserting ready-made programs. The transaction is executed as the addition of a new component (vertical layer), which splits into modules designed for several horizontal layers. These modules add security, protection, and reliability. Any program has a series of extensions on the horizontal layer. This programming has received development abroad in aspect-oriented programming [25].

3.8. Algebraic and agent programming (AP)

The AP paradigm is based on the theory of rewriting terms using the equality system of the computation algorithm, the result of which is a term obtained using graph terms of data representation and knowledge of subject domains [26-28]. This device allows determining the behavior of systems and their equivalence in a transitive system. The main state of the transitive system is the system behavior, which is defined by expressions of the behavior algebra $F(A)$ on the set of algebra operations A . Two prefixing operations au define the behavior of u on the operation a and the non-deterministic choice $u + v$ of one of two u and v behaviors, which is associative and commutative. The final behavior is given by the constants: Δ , \perp , 0 denoting the state of successful, undefined and unsuccessful completion. The behavior algebra is partially defined by the \leq relation, for whom the \perp element is the smallest, and the operations of the behavior algebra are monotone. *The environment E* , where the object is located, is defined as an agent in the algebra of actions A and the insertion function of two arguments $Ins(e, u) = e[u]$. The first argument is the behavior of the environment; the second is the behavior of the agent, which is inserted into this environment in a given state. The agent algebra is the parameters of the environment, and the value of the insertion function is a new state of the same environment. AP integrates procedural, functional and logical programming. A term graph is used to represent data and knowledge using expressions of polybasic data (group, ring, and field) algebra. Insertion (insert) programming is formed on the basis of the AP. The AP was implemented on MIR 1-3 computer, and the agent theory was developed for management decisions making by E. A. Trakhtengerts [29].

3.9. Graph programming

The theory of graphs began to develop relatively to programming in the school of A.P. Ershov (V.I. Kasyanov, V.E. Itkin, A.A. Evstigneev and others). The modular programming paradigm is based on the representation of modules and interfaces in graph nodes [11]. The graph is defined as an adjacency matrix and a characteristic vector. The adjacency matrix is used to prove the reachability of graph nodes and the correctness of the modular structure, as well as the formal conversion of the data types

transferred between inconsistent modules (see Chapter 3, 4 in the Assembly Programming Book, 1991). Last year Graph theory is being actively developed in ISP RAS for the schematic organization of memory for Linux and is used in medicine, aviation, biology, etc. The direction of this theory presented in the works [30-31].

3.10. Technology of programming and conferences in the USSR

The definition of programming technology in the USSR was given in the following works:

1. Glushkov V.M. Basic Research and Programming Technology. Programming. 1980, No. 2. p. 3-13.
2. Velbitsky I.V. Programming Technology. K.: Technika, 1984. 277 p.

After the death of V.M. Glushkov (1982), I.V. Velbitsky became responsible for holding the All-Union Conferences of the All-Union Communist Party I-III (1982, 1985, 1992) on behalf of the Executive Committee of the Academy of Sciences of the Ukrainian SSR and the State Committee on Science and Technology of the USSR [33-35]. These conferences gathered thousands of specialists from the country. The following scientists presented scientific reports on programming technologies: A.P. Ershov, V.V. Lipaev, S.S. Lavrov, M.R. Shura-Bura, E.L. Yushchenko and others scientists. The conference materials are stored in the computer science museums (<http://www.computer-muse.ru>).

4. Informatics and its role in society

N. Wiener (1948) his book on cybernetics in 1948, in discussion on nature of information said "the ideal computing machine must then have all its data inserted at the beginning, and must be as free as possible from human interference to the very end. This means that not only must the numerical data be inserted at the beginning, but also all the rules for combining them, in the form of instructions covering every situation which may arise in the course of the computation. Thus the computing machine must be a logical machine as well as an arithmetic machine and must combine contingencies in accordance with a systematic algorithm».

According to A.P. Ershov, “information is a body of knowledge about actual data, dependencies between them for the cognition and reproduction of human thinking in an artificial intelligence system. Informatics is a science that studies the structure and general properties of information, the laws of the processes of exchange, processing, storage, search and distribution of scientific, financial, economic and other information. Informatics provides mechanisms for reforming documents in natural languages and creating information retrieval, information logical, intelligent systems using system analysis and artificial intelligence. In other words, informatics is considered as a complex scientific discipline, which includes the theory of designing and operation of complex computer, information and intelligent systems (knowledge and data) as well as programming technologies. The term “computer science” began to be used in Russia in the late 1970s, particularly due to the theory of artificial intelligence, which allows simulating human mental activity in the works of domestic

scientists – A.Dorodnitsyn, E.Velikhov, G.Pospelov and D.Pospelov, V.Glushkov, and others [39-44].

4.1. Engineering sciences of computer science teaching

A computer science education program was published in 2005 [R. Shackelford, J.H. Cross, G. Davies, et al. Series "training programs in Informatics", Commission of training programs in Informatics]. It presented the disciplines of Computer Science (Computer Engineering, Electronic Engineering, Software Engineering, System Engineering, IT – Information Technology, Information Systems). A computer science-training program for each individual discipline is provided at <https://curriculum.code.org>. Below, their brief definitions are provided (Fig. 1).

Information technologies are technologies for managing solutions, business, commerce, economics, social sphere, etc.).

Information Systems – information processing systems (exchange of information, documents, data, etc.).

Software Engineering – methods, means and tools for the development, operation, maintenance and termination of programs usage. It occupies a central place in computer science.

System Engineering – the theory, methods, and principles of software design for computers (OS, translators, interpreters, analyzers, schedulers, etc.), information and software systems maintenance, data management, etc.

Computer Engineering – the theory and principles of building computers, supercomputers, multiprocessor and macroconveyor machines, clusters, etc. The basis of Computer Engineering includes Turing machine, von Neumann, automata, algorithms, mathematics, logics, and other theories.

4.2. Intellectual systems

The intellectual system performs creative tasks, the knowledge of which is stored in its memory. Such a system includes a knowledge base, a decision output mechanism, and an intelligent interface. Intellectual systems are studied by means and methods of artificial intelligence [49].

The main tasks of artificial intelligence:

- symbolic modeling of thinking processes,
- working with natural languages,
- presenting and using knowledge
- machine learning,
- biological modeling of artificial intelligence,
- robotics.

These systems were on the agenda as soon as computers appeared. A number of scientists have been studying these problems to the present day [50]. During the period of 1985-1990, specialists from Japan were engaged in the issues of the

intellectualization of knowledge and the construction of smart machines. First Japanese robots became the result of this project [49].

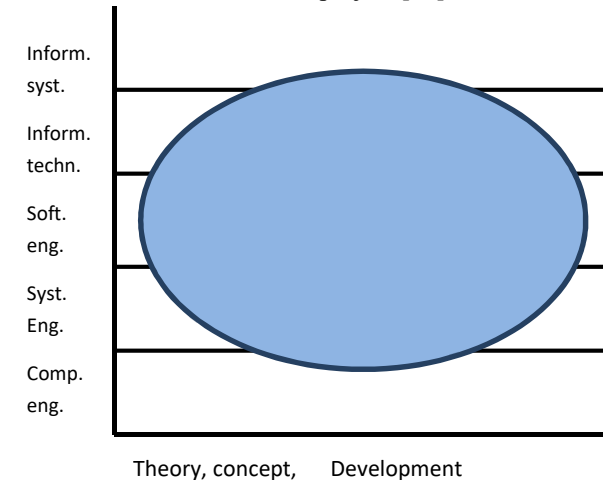


Fig. 1. Informatics space (Software Engineering in the middle)

Information systems and information technologies in computer science have become the main tool for the everyday life and interactions of modern society and the basis for computerization and informational support of the global world.

Today, in a time of the digital economy, the solution of machine learning problems, the use of accumulated knowledge, machine learning and robotization are on the agenda of the government and the Ministry of Education of the Russian Federation. A great contribution to the development of artificial intelligence was made by G.S. Pospelov, who has been holding thematic conferences on Artificial Intelligence in Russia for 10 years.

4.3. Knowledge engineering in engineering domains

Knowledge of a certain subject area (biology, chemistry, physics, etc.) is presented in the Knowledge Base (KB), which contains inference rules and information on human experience and knowledge with consideration of the ISO/IEC 2382-1: 19931-BR standard. Not only is the actual information of the region placed in the KB, but also the rules of inference for performing automatic deduction about the existing or newly entered facts and thereby is semantic (meaningful) processing of information carried out. Modern KBs work in conjunction with the search and retrieval systems for information represented by ontology as a set of concepts and their relationships. Ontologies are described using special ontology languages (OWL, FODA, ODM, etc.) [51, 52]. One of the authors, together with the students of MIPT, developed the ontology of the life cycle domain in OWL, which contains information about processes and actions, as well as artifacts and dependencies between them. The life cycle description in OWL was processed by Protégé 2.3 and the output code in XML

was obtained [53]. The life cycle automation approach was reported at the Science and Information-2015 conference in London. Today, the ontological approach is gaining momentum on the Internet, widely represented in the Semantics Web (<http://semanticweb.org>) for building ontologies, knowledge extraction (Data Mining), neural, quantum and intelligent technologies.

5. The era of informatization and market oriented economy (1992-2018)

After the collapse of the USSR, the «Informational Support of Russia» Scientific and Technical Council (1992) was established to determine the focus area on the development of information technology for the upcoming years. The council included 30 leading specialists from institutes of the Russian Academy of Sciences and other research organizations of Russia. This council has developed an informational support strategy for Russia, consisting in the development of concepts and principles for the development of promising and typical informational support projects, unification and standardization of design solutions based on international standards and information and communication technologies (ICTs) that improve the efficiency of informational support, security, quality of information technologies and economic mechanisms of informational support [62].

The basics of informational support of Russia included the following areas:

1. Economic and social tasks of informational support.
2. Technologic and economic development of the informational support market economy.
3. Scientific and technical informational support of Russia:
 - 3.1. Fundamental development of IT and systems.
 - 3.2. Technology and software engineering systems development.
 - 3.3. Ensuring information protection.
 - 3.4. Standardization of system creation tools.
 - 3.5. Support for informational support in education.
 - 3.6. Development of informational support infrastructure in Russia.
4. Development of informational support infrastructure in Russia.
5. System projects for informational support in Russia.

The main focus of the informational support project was on the justification of a market economy for various informational support projects.

In area 3.1, the following are related to the scientific areas of informational support (V.P. Ivannikov):

- algebraic and logical theories of system formalization;
- analysis of parallel and distributed systems;
- formal specifications and verification of complex systems;
- programming paradigms;

- knowledge presentation formalisms;
- models of human-machine interfaces and visualization.

The area 3.2 defines the main tasks of software engineering (V.V. Lipaev): development methods, safety, reliability, protection, quality and standardization of the life cycle of information systems, as well as a strategy for developing the market economy of software and hardware and organization of education.

The infrastructure and informational support program of Russia included: a communication system for computing tools and networks to provide interaction of information facilities and technologies; programs for the hardware, media and software complexes; a personnel education system for the effective development of the country's informational support. The market economy is defined as the main tool of informational support.

5.1. Directions of technology development after 1992

The presented domestic programming technologies have come a long way of development and have played an important role in the development of programming of software and system complexes for various purposes on a computer. Practically developed domestic technologies were not brought to the standard in this country, but many standards appeared in software and computer engineering abroad after 1992 [14].

These include, for example: UML (Unified Modeling Language), OWL (Ontology Web Language), WSDL (Web Service Definition Language), XML, XSL, RDF, and others. Since 2002, the Model (Model Variability) has been formed on programs production Product Lines/Product Family [32] and in Grid systems. It is used in AppFab software factories [33].

In domestic practice, a theory of object-component modeling of software systems [36] and a data type conversion system have been developed, taking the ISO/IEC 11404 GDT standard and Big Data non-structural data types into account [Lavrishcheva E.M., Ryzhov A.G. Application of the theory of general data types of ISO / IEC 12207 GDT standard to Big Data].

5.2. Industrial technologies after 2005

A few words are to be said on one important direction in the development of computer science, which gained momentum in the early 2000s, related to such concepts as program families and program factories. A program factory is an integrated architecture of pipeline assembly and fabrication of Programming Products (PP) from ready-made resources, which are standardized in programming languages, and their interfaces – standardized in the WSDL language [56-59, 64].

The program factory includes the following elements:

- ready-made resources (artifacts, modules, programs, systems, reuses, assets, objects, components, services, etc.);
- interfaces in one of the languages IDL, SIDL, WSDL, etc.;

- technological and product lines;
- assembly line;
- production and planning techniques;
- configuration of custom product versions;
- development of an environment for individual elements of ready-made resources and functions.

Creation of a product from ready-made elements depends on their volume or quantity. The level of programming programs production using the following functions:

$v = F(z, u)$, where $v = (v_i)$ is the output vector, $z = (z_i)$ is the resource expenditure vector, $u = (u_i)$ is the expenditure function dependency matrix $z = F(w_j, u)$, $j = 1, 2, \dots, n$. The indicators of the w_j function set the volume of production, the structure of the production assets and the level of ready-made resources specialization of the factory.

The overall Cobb-Douglas production function of output has the form $v = ne^{-t} L^\alpha K^\beta$, where v is the generalized output, n is the normative factor, e is the basis of the algorithm, t is an indicator of the level of scientific and technical progress, L is the cost of human labor, K – the amount of capital, α, β – elasticity coefficients.

The calculations of these functions determine the estimates of the revenue part of the production at the program factory. The efficiency, consistency, and proportionality of PP output depend on resource management by factory services (control, testing, quality assessment of PP, etc.). In a market economy, the quality and cost of PP evaluations correspond to the demand for a product.

The following factories are known for PP production at the moment:

- Product Line/Product Family (K. Pohl) [32].
- Conveyor multigeneration (K. Czarnecki) [48].
- AppFab in VS.Net ([https://msdn.microsoft.com/ru-ru/library/ee677312\(v=azure.10\).aspx](https://msdn.microsoft.com/ru-ru/library/ee677312(v=azure.10).aspx)).
- IBM Sphere (<https://www.ibm.com/developerworks/ru/websphere/newto/>).
- Streaming assembly by J. Greenfield [58].
- Chart assembly use case by G. Lenz and M. Fowler's continuous integration.
- Programs assembling according to V. Glushkov [47, 64].
- BEA WebLogic Oracle, SAP NetWeaver, and other factory systems.

For planning and making economic calculations at the factories programs was development special disciplines [46-48]. The production of PP is based on the technological processes of manufacture of certain product types using the theory of the design and usage of tool environments. These disciplines are used in the program CS-Curricula-2013 and was demanded many organizations of China, India and other countries [39, 45, 54].

Brief description of Software Engineering (SE) domains is presented below.

Scientific disciplines of SE include classic Sciences (theory of algorithms, set theory, proof theory, mathematical logic, discrete mathematics); theory of programming of the theory of abstract data, management science, etc. This discipline defines the basic

concepts of the objects and the formalism of the description of the system components and data description [15-20] etc.

Engineering disciplines of SE include methods of using technology rules and procedures, processes, life cycle, methods of measuring and assessing the quality of development PP. This discipline defines the set of engineering methods, techniques, tools and standards focused on the production of the target PP. Basic concepts of engineering SE include: core knowledge SWEBOK; the basic processes of SE; infrastructure environment [45].

Project management is based on the theory of management of projects and the IEEE Std.1490 PMBOK (Project Management Body of Knowledge); method CRM (Critical Path Method) for the graphic representation of works, operations and their execution time; method of network planning PERT (Program Evaluation and Review Technique), etc. In the PMBOK defined processes lifecycle of the project and the main areas of knowledge and processes of planning, monitoring, management and completion.

Economic disciplines. This disciplines are used for the analysis of the different parties activities of developers in the implementation of the project and identify the costs. The time and economic indicators are according to the requirements of PP. Used methods: predicting the size of PP (FPA – Function Points Analyses, Feature Points, Mark II Function Points, 3D Function Points, etc.); the evaluation effort for the development of PP by using models COCOMO and systems (Angel, Slim, Seer-SEM, etc.), as well as the quality of PP [35].

Production disciplines determine the production of PP. In software industry mass software products are produced by such worldwide leaders as Microsoft, IBM, Intel, Oracle, and other factory programs owners [13, 58, 59, 64].

5.3. The way of automation systems modeling

This paper does not assume a detailed analysis of the current state of computer science neither in Russia nor abroad. Nevertheless, it can be stated that intellectual systems are nowadays the main direction of computer science development. An important area of research in this direction is the development of robots in the manufacturing industry, medicine and other areas of their application. Robot automation is understood as the automation of the production of individual objects based on the usage of industrial unmanned technologies. The purpose of robot automation is to improve working conditions, increase technical and economic indicators of enterprises, and ensure the highest and most efficient production. This area is the most developed in Japan. They were one of the first to develop robots, starting from the late 1960s – early 1970s. Today, this country plans to launch the first unmanned taxi. In 2015, a prototype was presented bearing the name of Robot Taxi. It was shown in 2016 at the top level – during the G7 summit. It is expected that the full implementation of such machines in the market will occur in 2020 and be prepared by the time of the Olympics in Tokyo.

In terms of the evolution of design technologies, the development and analysis programs, the problems of ensuring information security, reliability and resiliency of software and hardware of the critical infrastructure come to the fore. One of the important aspects of the reliability and information security of software systems is designing and verification of so-called “variable” software systems or software product families.

One of the important aspects of reliability and information security of computer systems is modeling and verification of «variable» systems or software product families. The project of RFBR 16-01-00352 «Theory and methods of development of variable software systems» (head E. M. Lavrisheva) is devoted to the solution of a number of problems of modeling of variable systems. These include the implementation of the following tasks:

- modeling of software, operating and Web systems from ready-made software resources (GOR) by OCM method and creation of *MF* (Feature Model), *Msys* and *Mconfig* models [58-63];

- verification of data models, GOP, information and Web systems from using of services and services component, which founding in Internet libraries reuses and CPI [67-74];

- testing of GOR and configured systems on models *Mconfig*, *MF*, *Msys* with data collection about errors, defects, failures, etc. [54];

- integrated testing of evaluation of reliability and quality of the created the information and Web systems [74].

A description of the methods to implement the above processes, modeling of complex software, information and web-systems are given in the works of the participants of the project RFBR 16-00-00352. The main results of this project are published in [65-74].

6. Conclusions

The ways of establishing informatics and programming technologies for automating the creation of software and application systems since the advent of the first Electronic computers, as well as the new discipline for processing information on both the first computers and modern ones during the period of the informational support, are considered. Computer and information systems, as well as technologies for the automation and intellectualization of development of software and application systems of various types, are provided (the program of informational support of Russia after 1992 is described. According to the tasks developed and described new domestic technologies and tools of systems analysis, modeling of system and site with the help of ready GOR (objects, components, services, etc.) on new variable models and method of Configuration (assembling) are presented. There are new ways of development of intelligent systems in the direction of robotics devices, machines, etc. for different areas, especially for medicine. Innovative developments in the field of intellectualization, ontologization and ensuring the reliability and information security of software systems are noted.

References

- [1] Bauer F.E., Gooz G. Informatics. M: 1976, 484 p. (in Russian)
- [2] V. M. Glushkov "Fundamentals of paperless Informatics" // M.: Nauka, 1982, 552 p.
- [3] Ivannikov V.P., Korolev L.N., Lyubimsky E.Z. et al. Development of the Moscow school of computer OS. International Symposium Computers in Europe: Past, present and future, Kyiv, Ukraine, 5-9 October 1998, pp. 265-270 (in Russian)
- [4] Ivannikov V.P., Gontarenko S.V., Govorun N.N. Architecture of operating systems for supercomputers. Problems of Cybernetics. Problems of creating high-performance computers, 1984. pp. 117-126 (in Russian).
- [5] Shura-Bura M.R., Lyubimsky E.Z. Translator from ALGOL-60. Computational Mathematics and Mathematical Physics, vol. 4, № 1, 1964 (in Russian).
- [6] Babetskii G.I., Bezhanova M.M., Ershov A.P. et al. Programming System ALPHA.- Computational Mathematics and Mathematical Physics, №2, 1965, pp. 317-325 (in Russian).
- [7] Lavrisheva E.M., Borisenko, L.G., Grishkevich E.I. et al. The translator from the language of D-ALGAMS for UVK Dnepr-2. K: IK Ukrainian Academy of Sciences, 1970. - 186 p. (in Russian)
- [8] Terekhov A.N. Identification and structure of a compiler language ALGO-68. Programirovanie, N2, 1975 (in Russian)
- [9] A.P. Ershov and M.R. Shura-Bura. The Early Development of Programming in the USSR. In History of Computing in the Twentieth Century, Academic Press; First Edition edition, October 12, 1980, pp. 125-136. (in Russian)
- [10] Safonov V.O. Languages and methods of computer programming Elbrus. M.: Nauka, 1989
- [11] E.M. Lavrisheva, V.N. Grishchenko. Assembly programming. K.: 1991, 213 p. (in Russian)
- [12] V.V. Lipaev, B.A. Posin, A.A. Shtrik. Technology of Assembly programming. M.: 1992, 271 p. (in Russian)
- [13] E.M. Lavrisheva, V. N. Grishchenko. Assembly programming. The basics of the industry systems", Kiev, Naukova dumka, 2009.-371 p. (in Russian)
- [14] Lipaev V.V. Processes and standards of life cycle of complex software tools. Reference book. M.: SYNTEG, 2006, 260 p. (in Russian)
- [15] Lavrisheva E.M. Development of the theory of programs and systems in the USSR: History and modern theories. In Proc. of the Fourth International Conference on Computer Technology in Russia and in the Former Soviet Union (SORUCOM). Moscow, Zelenograd, 2017, pp. 31-43.
- [16] Kahro M.I., Kalia, A.P., Tyugu E.X. Instrumental programming system UCS (PRIZ). M.: Finance and statistics, 1981 (in Russian).
- [17] Tyugu E.X. Conceptual design. M., Nauka: 1984, 287 p. (in Russian)
- [18] Red'ko V.N. Program compositions and composition programming // Programirovanie, № 5, 1978, pp. 17-26. (in Russian)
- [19] Nikitchenko N.S., Shkilnyak S. S. Mathematical logic and theory of algorithms. K.: Kiev University: 2008, 528 p. (in Ukrainian)
- [20] Lavrisheva E.M., Nikitchenko N. S. Omelchuk L.P. Programming technology of information systems: methods, tools, tools. K., Kiev University: 2017, 457p.
- [21] Systems of computer algebra of the ANALYTIC family, group of authors. K.: NASU, IPMI, 2010, 762 p. (in Russian)

- [22] Turchin V. F. the Algorithmic language of recursive functions (REFAL). Moscow: Preprint of IAM AN SSSR, 1968 (in Russian)
- [23] Smirnov V. K., the Hardware realization of the language Refal in IPM im.M. V. Keldysh. Preprint KIAM RAS, №99, 2003, pp. 1-21 (in Russian).
- [24] Fuksman, A. L. Technological aspects of building software systems. M.: Statistika, 1979, 183 p. (in Russian).
- [25] Aspect-oriented programming. AspectJ (<http://aspect.org>).
- [26] Kapitonova Y.V., Letichevsky A.A. Methods and Means of Algebraic Programming. *Kibernetika/Cybernetics*, 1993, issue 3, pp. 7-12 (in Russian).
- [27] Letichevsky A., Gilbert D. A general theory of action languages. *Cybernetics and Systems Analysis*, vol. 34, no. 1, 1998, pp. 12-30. DOI: 10.1007/BF02911258.
- [28] Traktengerts E. A. The Interaction of agents in multi-agent environments. *Automation and telemekhanics*, № 8, 1998, pp. 3-52 (in Russian).
- [29] Traktengerts E. A. Computer methods of realization of economic and information management decisions. Volume 1. Methods and means. Volume 2 the Implementation of solutions. Moscow, 2009 (in Russian).
- [30] A.A.Evstigneev. Application of graph theory in programming. Moscow, Nauka, edited by A.P. Ershov, 1985, 351 p. (in Russian).
- [31] Christofides N. Graph theory. Algorithmic approach. M.: Mir. 1978 (in Russian)
- [32] Pohl K., Böckle G., van der Linden F. J. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag, 2005. DOI: 10.1007/3-540-28901-1.
- [33] Lavrischeva E.M. Theory and Practice of Software Factories. *Cybernetic and Systems Analyses*, vol.47, No.6, 2011, pp.961-972.
- [34] J.Hebeler, M.Fisher, R.Blace, A.Perez-Lopez. Semantic Web Programming. Wiley Publishing, Inc., 2008, 565 p.
- [35] Lavrishcheva E.M. Software engineering and programming technology of complex systems. Moscow, Yurayt, 2018, 432 p. (in Russian).
- [36] Lavrishcheva E.M. The theory of object-component modeling of modified software systems. Preprint of ISP RAS, 2016. ISBN 998-5-91474-025-9 (in Russian),
- [37] E.M. Lavrischeva, A.K. Petrenko. Software Product Lines Modeling. *Trudy ISP RAN/Proc. ISP RAS*, vol, 28, issue 6, 2016, pp. 49-65 (in Russian). DOI: 10.15514/ISPRAS-2016-28(6)-4
- [38] Catalogue of Technologies. ISP RAS. Moscow, 2017, 34 p. <http://www.ispras.ru/downloads/ISPRAS-Catalogue-En.pdf>
- [39] E.P. Velikhov. Computer science is an important direction of Soviet science. In *Cybernetics. The formation of Informatics*, 1986 (in Russian).
- [40] A.A. Dorodnitsin. Informatics. Subject and tasks In *Cybernetics. The formation of Informatics*, 1986 (in Russian).
- [41] V.S. Mikhalevich et al. Informatics – new field of science and practice. In *Cybernetics. The formation of Informatics*, 1986 (in Russian).
- [42] Shileiko A., Shileiko T. Conversations about informatics. Moscow, Molodaya Gvardia, 1989, 287 p. (in Russian)
- [43] V.M. Glushkov. *Cybernetics, computers, informatics*. K.: Naukova Dumka, 1990 (in Russian).
- [44] V.M. Glushkov. *The Basics of Paperless Informatics*. M.: Nauka, 1982, 552 p. (in Russian).

- [45] Curricula Recommendations. Available at: <https://www.acm.org/education/curricula-recommendations>
- [46] E.M. Lavrischeva. Classification of software engineering disciplines. *Cybernetics and Systems Analysis*, vol. 44, Issue 6, 2008, pp. 791–796.
- [47] Lavrishcheva E.M. Software engineering. Training manual in 3 parts. Moscow: MIPT, 2016 (in Russian).
- [48] Czarnecki K., Eisenecker U. Generative programming. Methods, tools, application. SPb.: Peter, 2005, 730 p. (in Russian).
- [49] H. Ueno, V. Isudzuka. Representation and use of knowledge. M.: Mir, 1987, 220 p. (in Russian).
- [50] G.S. Pospelov. Artificial intelligence – new information technology. In *Cybernetics. The formation of Informatics*, 1986 (in Russian).
- [51] Standard ISO / IEC 2382-1:1993, Information technology – Vocabulary – Part 1: Fundamental terms.
- [52] T.A. Gavrilova et al. Knowledge bases of intelligent systems. Textbook for technical universities. SPb.: Peter, 2000 (in Russian)
- [53] Lavrischeva E.M. Ontology of Domains. *Ontological Description Software Engineering Domain – The Standard Life Cycle*. *Journal of Software Engineering and Applications*, vol. 8 No. 7, 2015.
- [54] Lipaev V. V. Software engineering of complex custom software products. Tutorial. M.: MAKS-press, 2014 (in Russian)
- [55] E.M. Lavrishcheva, L.E. Karpov, A. N. Tomilin. Approaches to the representation of scientific knowledge in the Internet science. In Proc. of the XIX all-Russian scientific conference "Scientific service on the Internet", 2017, pp. 310-326 (in Russian).
- [56] Lavrischeva Ekaterina. (2015). Ontological approach to the formal specification of the standard life cycle. In Proc. of the 2015 Science and Information Conference (SAI), 2015, pp. 965-972.
- [57] Lavrishcheva E.M. Fundamentals of software engineering. In Proc. of the 5th International conference on Actual problems of system and software engineering. CEUR Workshop Proceedings, vol. 1989, 2017, pp. 163-177 (in Russian).
- [58] Lavrischeva K.M. Theory and Practice of Software Factories. *Cybernetic and Systems Analyses*, vol. 47, no. 6, 2011, pp. 961-972.
- [59] Lavrischeva K., Aronov A., Dzyubenko A. Programs Factory – A conception of Knowledge Representation of Scientific Artifacts From Standpoint of Software Engineering. *Computer and Information Science*, Vol. 6, No. 3, 2013, pp. 21–28.
- [60] Gorodnyaya L.V. Programming Paradigms: Analysis and comparison. Novosibirsk, SB RAS, 2017, 223 p. (in Russian).
- [61] Lavrischeva E.M. Software Engineering. Programming technology. Moscow, MIPT, 2016, 52 p. (in Russian).
- [62] Scientific and technical bases of informatization of Russia. Ministry of Science and Technical Policy of the Russian Federation. Moscow, 1992, 151 p. (in Russian).
- [63] E.M. Lavrischeva, I.B. Petrov, Ways of Development of Computer Technologies to Perspective Nano. In Proc. of the Future Technologies Conference (FTC-2017), 2017, pp.539-547.
- [64] Lavrischeva E. M. Assembly line of program factories – the idea of academician V. M. Glushkov. In V. M. Glushkov: The past is looking to the future. K.: *Academperiodica*, 2013, p. 130-143 (in Ukrainian)

- [65] E.M. Lavrisheva, L.E. Karpov, A.N. Tomilin. Semantic resources for the development of ontology of scientific and engineering subject areas, In Proc. of the XVIII all-Russian scientific conference "Scientific service on the Internet", 2016, pp. 223-239 (in Russian).
- [66] Lavrishcheva E.M. Theoretical foundations of modeling software systems from objects and components. In Proc. of the International scientific-practical conference on Theory of active systems (TAS-2016), 2016, pp. 314-325 (in Russian).
- [67] Kuliamin. V.V., Lavrishcheva E.M., Mutilin V.S., Petrenko A.K. Verification and analysis of variable operating systems. Trudy ISP RAN/Proc. of ISP RAS, vol. 28, issue 6, pp. 48-59 (in Russian). DOI: 10.15514/ISPRAS-2016-28(3)-12
- [68] Lavrisheva E. Assembling Paradigms of Programming in Software Engineering. Journal of Software Engineering and Applications, vol. 9, no. 6, 2016, pp. 296-317.
- [69] Lavrisheva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle. In Proc. of the Science and Information Conference, 2015, pp. 965-972.
- [70] Lavrisheva E. M., V. S. Mutilin, A. G. Ryzhov. Aspects of modeling of variable software and operating systems. In Proc. of the XIX all-Russian scientific conference "Scientific service on the Internet", 2017, pp 327-341 (in Russian).
- [71] Lavrisheva E.M., Mutilin V.S., Ryzhov A.G. Designing variability models for software, operating systems and their families. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 5, 2017, pp. 93-110. DOI: 10.15514/ISPRAS- 2017(5).
- [72] Lavrisheva E. Scientific Basis of System Programming. Journal of Software Engineering and Applications, No. 11, pp. 408-434.
- [73] E.M. Lavrisheva. The Scientific basis of Software Engineering. International Journal of Applied and Natural Sciences, vol. 7, issue 5, 2018, pp. 15-32.
- [74] Lavrisheva E.M., Pakulin N.V., Ryzhov A.G., Zelenov S.V. Analysis methods for assessing the reliability of equipment and systems. Application practice methods'. Trudy ISP RAN/Proc. ISP RAS, vol.30, issue 3, 2018, pp.99-120 (in Russian). DOI: 10.15514/ISPRAS-2018-30(3)-8

Информатика: становление программного обеспечения и технологий программных систем

^{1,3} Е.М. Лаврищева <lavr@ispras.ru>

^{1,2,4} А.К.Петренко <petrenko@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25

² Московский Государственный университет имени М. В. Ломоносова Москва, 119991, ГСП-1, Ленинские горы, д. 1

³ Московский физико-технический институт, 141700, Московская область, г. Долгопрудный, Институтский пер., 9

⁴ НИУ "Высшая школа экономики", 101000, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. Рассматривается становление информатики и аспектов компьютерной поддержки создания программного обеспечения, в том числе системных и прикладных

программных систем, начиная от периода появления первых ЭВМ 1948-1990-х годов XX века. Рассмотрена программа информатизации России 1992 года и пути развития фундаментальных основ современных компьютерных наук или информатики и интеллектуализации разработки различных видов программных систем.

Ключевые слова: информатика; компьютерная, системная и программная инженерия; технология; наука; информационные системы; интеллектуальные системы; индустрия; цифровая экономика.

DOI: 10.15514/ISPRAS-2018-30(5)-1

Для цитирования: Лаврищева Е.М., Петренко А.К. Информатика. Становление программирования и технологий программных систем. Труды ИСП РАН, том 30, вып. 5, 2018 г., стр. 7-30. DOI: 10.15514/ISPRAS-2018-30(5)-1

Список литературы

- [1]. Ф. Бауэр, Г. Гооз. Информатика. М.: Мир, 1976, 484 с.
- [2]. Глушков В.М. Основы безбумажной информатики. М.: Наука, 1982, 552с.
- [3]. Иванников В.П., Королев Л.Н., Любимский Э.З. и др. Разработка Московской школы ОС ЭВМ. Международный симпозиум «Компьютеры в Европе. Прошлое, настоящее и будущее (Computers in Europe. Past, Present and Future) 5-9 октября 1998. Киев, Украина, 265-270 стр.
- [4]. Иванников В.П., Гонтаренко С.В., Говорун Н.Н. Архитектура ОС суперЭВМ.- Вопросы кибернетики. Проблемы создания высокопроизводительных ЭВМ.- Совет по проблеме «Кибернетики», 1984. -М., стр.117-126.
- [5]. Шура-Бура М.Р., Любимский Э.З. Транслятор с языка Алгол-60. Журнал вычислительной математики и математической физики, том 4, № 1, 1964.
- [6]. Бабеецкий Г.И., М.М.Бежанова, Ершов А.П. и др. Система программирования АЛЪФА. Журнал вычислительной математики и математической физики, №2, 1965, стр. 317-325.
- [7]. Лаврищева Е.М., Борисенко Л.Г., Гришкевич Е.И. и др. Транслятор с языка Д-АЛГАМС для УВК Днепр-2. К.: ИК АН УССР, 1970, 186 стр.
- [8]. Терехов А.Н. Проблемы идентификации и структура компилятора с языка АЛГОЛ 68. Программирование, No 2, 1975, стр. 61-67.
- [9]. А.П.Ершов, М.Р.Шура-Бура. Становление программирования в СССР (переход ко второму поколению языков и машин). Новосибирск: Изд. ВЦ СО АН СССР, Препринт №13, 1976.
- [10]. В.О. Сафонов. Языки и методы программирования для ЭВМ «Эльбрус», М.: Наука, 1989, 389 стр.
- [11]. Е.М. Лаврищева, В.Н. Грищенко. Сборочное программирование. Наукова думка, Киев: 1991, 213 стр.
- [12]. В.В. Липаев, Б.А. Позин, А.А. Штрик. Технология сборочного программирования. Москва:1992, 271 стр.
- [13]. Е.М. Лаврищева, В.Н. Грищенко. Сборочное программирование. Основы индустрии систем, Киев.: 2009, 371 стр.
- [14]. Липаев В.В. Процессы и стандарты жизненного цикла сложных программных средств. Справочник. Москва, Синтег. 2011, 260 стр.

- [15]. Лаврищева Е.М. Развитие теории программирования в СССР. История и современные теории. SoGusom-17. Развитие ВТ в России и странах бывшего СССР, 2017, стр. 162-177.
- [16]. Кахро М. И., Калья А. П., Тытугу Э. Х. Инструментальная система программирования ЕС ЭВМ (ПРИЗ). М.: Финансы и статистика, 1981.
- [17]. Тытугу Э. Х. Концептуальное проектирование. М.: Наука: 1984, 287 стр.
- [18]. Редько В.Н. Композиции программ и композиционное программирование. Программирование, № 5, 1978, стр. 17–26.
- [19]. Никитченко Н.С., Шкильняк С.С. Математическая логика и теория алгоритмов. К., ИПЦ Киевский университет, 2008, 528 стр.
- [20]. Лаврищева Е.М., Никитченко Н.С., Омельчук Л.П. Технология программирования информационных систем. К.: ВШН КНУ, 2010, 351 стр.
- [21]. Системы компьютерной алгебры семейства АНАЛИТИК, Коллектив авторов. К.: НАНУ ИПММИ, 2010, 762 стр.
- [22]. Турчин В. Ф. Алгоритмический язык рекурсивных функций (РЕФАЛ). М.: ИПМ АН СССР, 1968.
- [23]. Смирнов В.К. Аппаратная реализация языка Рефал в ИПМ им.М.В.Келдыша. Препринт ИПМ им. М.В.Келдыша, № 99, 2003, 21 стр.
- [24]. Фуксман А.Л. Технологические аспекты создания программных систем. М.: Статистика, 1979, 183 стр.
- [25]. Аспектно-ориентированное программирование. AspectJ (<http://aspect.org>).
- [26]. Капитонова Ю.В., Летичевский А.А. Методы и средства алгебраического программирования. Кибернетика, № 3, 1993, стр. 7–12.
- [27]. Letichevsky A., Gilbert D. A general theory of action languages. Cybernetics and Systems Analysis, vol. 34, no. 1, 1998, pp. 12-30 . DOI: 10.1007/BF02911258.
- [28]. Транхтенгерц Э.А. Взаимодействие агентов в многоагентных средах. Автоматика и телемеханика, № 8, 1998, стр.3-52.
- [29]. Трахтенгерц Э.А. Компьютерные методы реализации экономических и информационных управленческих решений. Том 1. Методы и средства. Том 2 Реализация решений. М.:2009.
- [30]. Евстигнеев А.А. Применение теории графов в программирование. Москва.-Наука, 1985, 351стр.
- [31]. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир. 1978.
- [32]. Pohl K., Böckle G., van der Linden F. J. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag, 2005. DOI: 10.1007/3-540-28901-1.
- [33]. Lavrischeva E.M. Theory and Practice of Software Factories. Cybernetic and Systems Analyses, vol.47, No.6, 2011, pp.961-972.
- [34]. J.Hebeler, M.Fisher, R.Blace, A.Perez-Lopez. Semantic Web Programming. Wiley Publishing.Inc., 2008, 565 p.
- [35]. Лаврищева Е.М. Программная инженерия и технология программирования сложных систем. Москва, Юрайт, 2018. 432 стр.
- [36]. Е.М. Лаврищева. Теория объектно-компонентного моделирования программных систем. Препринт ИСП РАН №29, 2016, 52 стр. ISBN 978-5-91474-025-9
- [37]. Лаврищева Е.М., Петренко А.К. Моделирование семейств программных систем. Труды ИСП РАН, том 28, вып. 6, 2016, стр. 49-64. DOI: 10.15514/ISPRAS-2016-28(6)-4
- [38]. Каталог технологий ИСП РАН. Москва, 2017, 34стр.

- [39]. Е.П. Велихов. Информатика – актуальное направление советской науки. В сб. "Кибернетика. Становление информатики", М.:Наука, 1986.
- [40]. Дородницын А.А. Информатика: предмет и задачи. В сб. "Кибернетика. Становление информатики", М.:Наука, 1986.
- [41]. В.С. Михалевич и др. Информатика – новая область науки и практики В сб. "Кибернетика. Становление информатики", М.:Наука, 1986.
- [42]. Шилейко А., Шилейко Т. Беседы об информатике.-Москва. Молодая Гвардия, 1989, 287 стр.
- [43]. В.М. Глушков. Кибернетика. Вычислительная техника. Информатика. Киев, Наукова думка, 1990..
- [44]. В.М. Глушков «Основы безбумажной информатики». М.: Наука, 1982, 552 стр.
- [45]. Curricula Recommendations. <https://www.acm.org/education/curricula-recommendations>.
- [46]. E.M. Lavrischeva. Classification of software engineering disciplines. Cybernetics and Systems Analysis, vol. 44, issue 6, 2008, pp. 791–796.
- [47]. Лаврищева Е.М. Программная инженерия. Учебно-методическое пособие в 3-х частях. М.: МФТИ, 2016.
- [48]. Чернецки К., Айзенкер У. Порождающее программирование. Методы, инструменты, применение. Издательский дом «Питер», 2005, 730 стр.
- [49]. Х. Уэно, Т. Кояма, Т. Окамото, Б. Мацуби, М. Исидзука. Представление и использование знаний. Москва, Мир,1987, 220 стр.
- [50]. Г.И. Поспелов. Искусственный интеллект – новая информационная технология В сб. "Кибернетика. Становление информатики", М.:Наука, 1986.
- [51]. ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.
- [52]. Т.А.Гаврилова и др. Базы знаний интеллектуальных систем. Учебник для вузов. СПб.: Питер, 2000.
- [53]. Lavrischeva E.M. Ontology of Domains. Ontological Description Software Engineering Domain – The Standard Life Cycle. Journal of Software Engineering and Applications, vol. 8 No. 7, 2015.
- [54]. Липаев В.В. Программная инженерия сложных заказных программных продуктов. Учебное пособие. М.: МАКС-ПРЕСС, 2014.
- [55]. Е.М. Лаврищева, Л.Е. Карпов, А.Н. Томилин. Подходы к представлению научных знаний в Интернет науке. Труды. XIX Всероссийский научной конференции «Научный сервис в сети Интернет», 2017, стр. 310-326.
- [56]. Lavrischeva Ekaterina. (2015). Ontological approach to the formal specification of the standard life cycle. In Proc. of the 2015 Science and Information Conference (SAI), 2015, pp. 965-972.
- [57]. Лаврищева Е.М. Фундаментальные основы программной инженерии. Труды 5- международной конференции «Актуальные проблемы системной и программной инженерии, 2017, стр.163-177.
- [58]. Lavrischeva K.M. Theory and Practice of Software Factories. Cybernetic and Systems Analyses, vol. 47, no. 6, 2011, pp. 961-972.
- [59]. Lavrischeva K., Aronov A., Dzyubenko A. Programs Factory – A conception of Knowledge Representation of Scientific Artifacts From Standpoint of Software Engineering. Computer and Information Science, Vol. 6, No. 3, 2013, pp. 21–28.
- [60]. Городняя Л.В. Парадигмы программирования: Анализ и сравнение. Новосибирск: Изд-во СО РАН, 2017 г., 232 с

- [61]. Лаврищева Е.М. Программная инженерия. Технология программирования. Москва, МФТИ, 2016.
- [62]. Научно-технические основы информатизации России. Мин. Науки, высшей школы и технической политики РФ. Москва, 1992, 151 стр.
- [63]. E.M. Lavrischeva, I.B. Petrov. Ways of Development of Computer Technologies to Perspective Nano. In Proc. of the Future Technologies Conference (FTC-2017), 2017, pp. 539-547.
- [64]. Лаврищева Е.М. Сборочный конвейер фабрик программ – идея академика В.М.Глушкова. В книге «В.М.Глушков: Прошлое устремленное в будущее». К.: Академперіодика, 2013, стр. 130—13.
- [65]. Лаврищева Е.М., Карпов Л.Е., Томилин А.Н. Семантические ресурсы для разработки онтологии научной и инженерной предметных областей. Труды. XVIII Всероссийский научной конференции «Научный сервис в сети Интернет», 2016, стр. 223-239.
- [66]. Лаврищева Е.М. Теоретические основы моделирования программных систем из объектов и компонентов. Труды Международной научно-практической конференции «Теория активных систем», 2016, стр. 314-325.
- [67]. Кулямин В.В., Лаврищева Е.М., Мутилин В.С., Петренко А.К. Верификация и анализ переменных операционных систем. Труды ИСП РАН, 2016, том 28, вып. 6, стр. 48-59. DOI: 10.15514/ISPRAS-2016-28(3)-12
- [68]. Lavrischeva E. Assembling Paradigms of Programming in Software Engineering. Journal of Software Engineering and Applications, vol. 9, no. 6, 2016, pp. 296-317.
- [69]. Lavrischeva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle. In Proc. of the Science and Information Conference, 2015, pp. 965-972.
- [70]. Е.М. Лаврищева, В.С. Мутилин, А.Г. Рыжов. Аспекты моделирования переменных программных и операционных систем. Труды. XIX Всероссийский научной конференции «Научный сервис в сети Интернет», 2017, стр. 327-341.
- [71]. Lavrischeva E.M., Mutilin V.S., Ryzhov A.G. Designing variability models for software, operating systems and their families. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 5, 2017, pp. 93-110, DOI: 10.15514/ISPRAS- 2017(5).
- [72]. Lavrischeva E. Scientific Basis of System Programming. Journal of Software Engineering and Applications, No. 11, pp. 408-434.
- [73]. E.M. Lavrischeva. The Scientific basis of Software Engineering. International Journal of Applied and Natural Sciences, vol. 7, issue 5, 2018, pp. 15-32.
- [74]. Лаврищева Е.М., Пакулин Н.В., Рыжов А.Г., Зеленев С.В. Анализ методов оценки надежности оборудования и систем. Практика применения методов. Труды ИСП РАН, том.30, вып. 3, 2018, стр. 99-120. DOI: 10.15514/ISPRAS-2018-30(3)-8.