MONOLITHIC AND MICROSERVICE ARCHITECTURES FOR THE ORGANIZATION AND IMPLEMENTATION OF AN AUTOMATED ALGORITHMIC TECHNOLOGICAL INFORMATION SYSTEM

Bakhriddin Bozorov [0009-0003-8129-7570]

Deputy Head of the Information and Communication Technologies Department at JSC "Navoi Mining and Metallurgical Combine"

Abstract. The evolution of software architecture plays a critical role in the development and implementation of automated algorithmic technological information systems (AATIS). This study explores two primary architectural paradigms—monolithic and microservice architectures—and their applicability to the design and operation of AATIS. Monolithic architecture represents a traditional approach where all components of the system are integrated into a single, unified application. It is characterized by simplicity in design and deployment but often faces scalability and flexibility challenges in dynamic environments. In contrast, microservice architecture divides the system into independent, loosely coupled services, each responsible for specific functions. This approach enhances scalability, fault isolation, and maintainability, making it particularly suitable for complex, high-demand technological systems. This research analyzes the benefits, limitations, and practical applications of both architectural models in the context of AATIS. Key considerations include system performance, scalability, modularity, and ease of deployment. The annotation highlights how microservices, with their ability to facilitate continuous integration and deployment, offer distinct advantages in environments requiring frequent updates and adaptability. The findings aim to guide decision-makers and developers in selecting the most appropriate architectural paradigm for building robust, efficient, and future-proof automated algorithmic technological information systems.

Key words: automated system, algorithmic model, information part, operational part, software packages, algorithm bank, feature bank, application programs, monolithic, microservice

Annotatsiya. Dasturiy ta'minot arxitekturasining taraqqiyoti avtomatlashtirilgan algoritmik texnologik axborot tizimlarini (AATIS) ishlab chiqish va joriy etishda hal qiluvchi ahamiyatga ega. Ushbu tadqiqot ikkita asosiy arxitektura paradigmasini - monolit va mikroservis arxitekturalarini hamda ularning AATISni loyihalash va ishlatishga qo'llanilishini o'rganadi. Monolit arxitektura an'anaviy yondashuv bo'lib, unda tizimning barcha tarkibiy qismlari yagona, yaxlit ilovaga birlashtiriladi. U loyihalash va joylashtirishning soddaligi bilan ajralib tursa-da, ko'pincha o'zgaruvchan muhitlarda kengayish va moslashuvchanlik muammolariga duch keladi. Bunga qarama-qarshi o'laroq, mikroservis arxitekturasi tizimni mustaqil, bo'sh bog'langan xizmatlarga ajratadi, ularning har biri ma'lum vazifalarni bajarish uchun javobgar bo'ladi. Bu yondashuv murakkab, yuqori talabga ega texnologik tizimlar uchun ayniqsa mos kelib, kengayish imkoniyati, xatolarni izolyatsiyalash va texnik xizmat ko'rsatish qulayligini oshiradi. Ushbu tadqiqot AATIS doirasida har ikkala arxitektura modelining afzalliklari, cheklovlari va amaliy qo'llanilishini tahlil qiladi. Asosiy e'tibor tizim unumdorligi, kengayishi, modullilik va joylashtirishning qulayligiga qaratiladi. Annotatsiyada mikroservislarning uzluksiz integratsiya va joylashtirishni osonlashtirish qobiliyati, tez-tez yangilanish va moslashuvchanlikni talab qiladigan muhitlarda aniq afzalliklar taqdim etishi ta'kidlanadi. Tadqiqot natijalari qaror qabul qiluvchilar va dasturchilarni mustahkam, samarali va kelajakka yo'naltirilgan avtomatlashtirilgan algoritmik texnologik axborot tizimlarini qurish uchun eng mos arxitekturaviy paradigmani tanlashda yo'naltirish maqsadini ko'zlaydi.

Kalit so'zlar: avtomatlashtirilgan tizim, algoritmik model, axborot qismi, operatsion qism, dasturiy ta'minot paketlari, algoritmlar banki, funksiyalar banki, amaliy dasturlar, monolitik, mikroservis

Абстракт. Эволюция архитектуры программного обеспечения играет важную роль в разработке и внедрении автоматизированных алгоритмических технологических информационных систем (ААТИС). В данном исследовании рассматриваются две основные архитектурные парадигмы - монолитная и микросервисная архитектура - и их применимость к проектированию и эксплуатации ААТИС. Монолитная архитектура представляет собой традиционный подход, при котором все компоненты системы интегрируются в единое, унифицированное применение. Он характеризуется простотой в проектировании и развертывании, но часто сталкивается с проблемами масштабируемости и гибкости в динамических средах. В отличие от этого, микросервисная архитектура делит систему на независимые, слабо связанные услуги, каждая из которых отвечает за определенные функции. Такой подход повышает масштабируемость, изоляцию неисправностей и техническое обслуживание, что делает его особенно подходящим для сложных, востребованных технологических систем. В данном исследовании анализируются преимущества, ограничения и практические применения обеих архитектурных моделей в контексте ААТИС. К основным соображениям относятся производительность системы, масштабируемость, модульность и простота внедрения. В аннотации подчеркивается, как микросервисы, способные содействовать непрерывной интеграции и развертыванию, предлагают определенные преимущества в средах, требующих частых обновлений и адаптации. Результаты исследования направлены на ориентирование лиц, принимающих решения и разработчиков в выборе наиболее подходящей архитектурной парадигмы для построения надежных, эффективных и перспективных автоматизированных алгоритмических технологических информационных систем.

Ключевые слова: автоматизированная система, алгоритмическая модель, информационная часть, операционная часть, пакеты программного обеспечения, банк алгоритмов, банк функций, прикладные программы, монолитный, микросервис

Introduction

The rapid advancement of digital technologies has transformed the way organizations develop and deploy software systems, particularly in the realm of automated algorithmic technological information systems (AATIS). These systems play a crucial role in managing, analyzing, and optimizing complex technological processes across various industries. To ensure their efficiency and adaptability, selecting the right software architecture is a critical decision that directly impacts system performance, scalability, and maintainability.

Monolithic and microservice architectures are two contrasting approaches to system design, each with its own strengths and limitations. Monolithic architecture, as a traditional method, consolidates all system components into a single application. This approach simplifies development and testing but can become increasingly rigid and challenging to scale as systems grow in complexity. On the other hand, microservice architecture divides the system into smaller, independent services that work cohesively. This modern approach offers greater flexibility, fault tolerance, and the ability to scale specific components independently, making it a favored choice for dynamic and high-demand systems.

This study focuses on exploring and comparing the monolithic and microservice architectural paradigms in the context of AATIS. By analyzing their structural differences, operational efficiencies, and real-world applications, this research aims to provide insights into their suitability for various technological and organizational needs. The ultimate goal is to identify best practices for leveraging these architectures to design resilient, scalable, and adaptable information systems capable of meeting the challenges of modern industrial and technological environments.

Metodology

Applying the substitution rule, we can detail the banks of the system and present them in the following form: the bank of attributes has only an informational part represented as a hierarchy of attributes; the bank of models consists of an informational part representing the relation of a group of task attributes to a model and an operational part performing operations of model selection, synthesis of the conceptual model and construction of new models based on the composition of basic models in the form of functioning tables; The bank of algorithms consists of the information part, which represents the relation of a group of features of effective computability to the algorithm, and the operational part, which performs the operations of algorithm selection, construction of the computational scheme, optimization and construction of new algorithms in the form of functioning tables; The bank of application software packages consists of an information part representing algorithm/module relations and an operational part performing operations of video gram setting, software generation, software testing, software verification and software documentation; The operation bank consists of the information part, which is a hierarchical set of software modules and the operational part, which performs the operations of the dialogue with the user, management of the operational parts of the other banks, system initialization and system recovery after failures; A data bank consists of an information part, which is a network structure databases, and an operational part, which is a database management system that performs standard operations on the data [1-6].

The use of ideas and methods of algorithmicisation directly in the control of the object is one of the aspects of the functioning of this system. The software-controlled control system resides in this system in the application package bank; therefore, the operational bank, application package bank, and data bank are used during its operation. The algorithmic scheme proposed in [7,8] for formalising the control process of manufacturing systems includes both design and control issues.

The purpose of this paper is to address control issues, so the proposed algorithmic scheme uses only four banks: feature bank, application package bank, data bank and operational bank. The operational part of each component (bank) defines operations and rules of their execution over the information part of the bank. The information part of the bank in this case has a complex logical structure and appropriate means of access to it from the operational part of the bank.

Description of banks of operations and attributes. The bank of operations in the algorithmic system is the core of the system. Its main functions are: conducting a dialogue with the user, controlling the operational parts of the banks of the algorithmic system, system initialization (preparation for operation) and system recovery after failures. The bank operation consists of five stages: system initialization, feature selection, selection of task models, software configuration and counting [9-12].

During system operation, the operating bank determines the required sequence of instructions and controls the operating parts of the system banks. In the operational bank there is a monitor, which controls the operation of the entire system, the processor of the input language, which carries out a dialogue with the user, and the bank of signs. The monitor, in turn, consists of the kernel (resident), monitors of operational parts of banks, scheduler and calculator [13-15]. The input language processor consists of a syntactic and lexical parser of the input language, a dialogue monitor and an output instruction generator. The structure of the operating part of the bank is shown in Fig. 1.

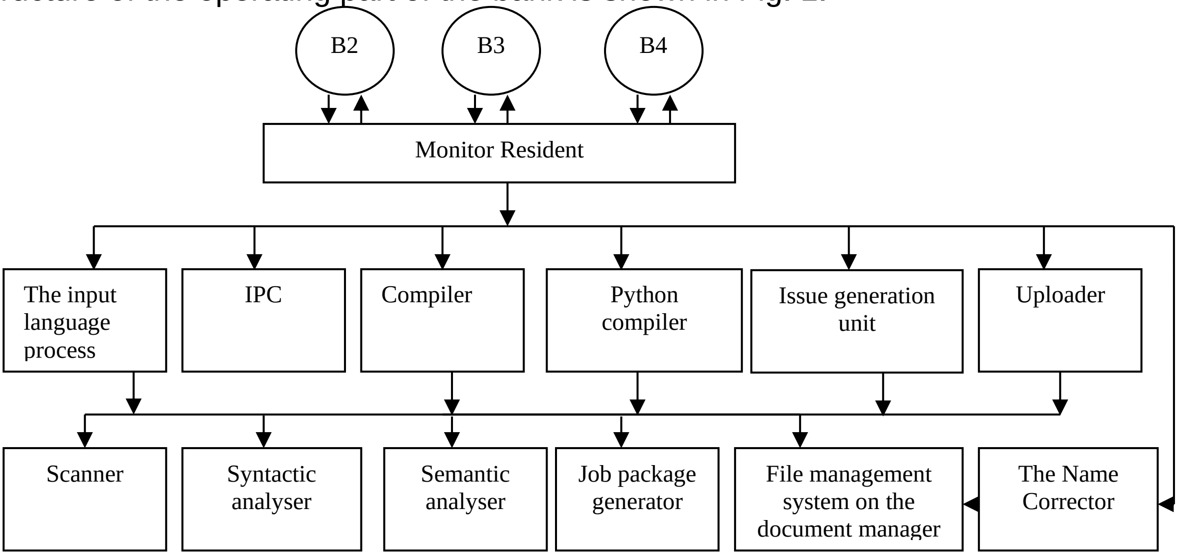

Fig.1. Structure of the operational part of the bank.

The bank of attributes contains the attributes of tasks, which allow to determine the necessary models and algorithms of the task on the basis of the group of attributes acceptable for the given system and to select the corresponding programs. The structural and information part of the feature bank is a hierarchical system consisting of five levels (Fig. 2).

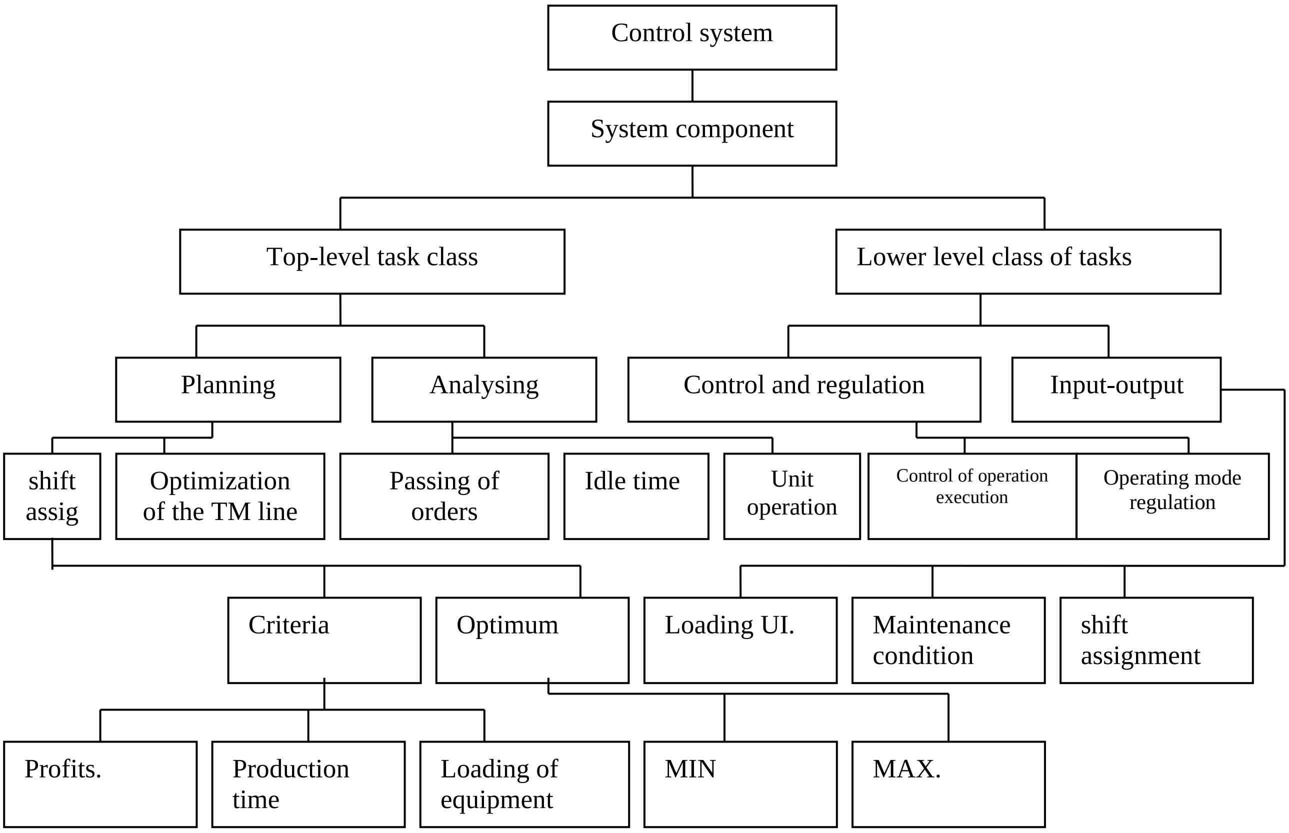

Fig.2. Structural and information part of the feature bank.

At the first level, attributes define the system to be developed, which is characterized by production type, type and system components. At the second level, the system components are categorized by level into task classes. The upper-level task classes include planning, analyzing and input-output of information from the system. The lower-level task classes at an existing point in time include the control and regulation task class. At the third hierarchical level the task classes are divided into separate tasks. The class of tasks "planning" includes compilation of shift and daily assignment and optimization of technological routes of the automatic line. The class of tasks "analysis" includes analyses of order flow, downtime and machine operation. The class of "input-output" tasks includes input and output of information on the state of the process equipment and the daily shift assignment (SHA). The class of tasks of the lower level "control and regulation" includes the loading of the PP, control of the performance of technological operations and regulation of the work rhythm. At the fourth level the planning tasks are detailed (supplemented) by auxiliary attributes defining the optimization criteria. The fifth level contains detailed values for optimization and planning criteria: production time, equipment loads and profit [16].

System operation starts with system initialization and identification of the configuration of technical means. For this purpose, the kernel of the control monitor is used, which loads the necessary components of the operational bank and, first of all, the scheduler and dialogue monitor. Next, the user identification and his access rights in the system are carried out in the dialogue mode. Then the control is transferred to the dialogue monitor, which interacts with the user, determines the nature of the work to be performed in the system and transfers the control to the scheduler of the computational process. The scenario of the dialogue between the user and the system in the operation mode (account) is designed in the form of "Menu Selection". Semantics. A predicate is a co-occurring formulation expressed by Russian inductive sentences. In such a formulation, the predicate, expressed by the verb form of the imperative mood, is placed after the beginning mark of the sentence. The action defines a specific pointer on which either finding, calculating, or achieving the goal depends. The action is followed by a list of goals in the sentence. The goal is the unknown that needs to be found. The main members of the sentence, the action and the list of goals, are placed at the beginning of the sentence. The rest of the sentence is included in the action-object construction [19].

The object of action - so we have conditionally named the information, necessary and sufficient conditions with which the search for the unknown is connected. The main element of the action is the object itself. It must be specified in the problem statement. Other elements of the object supplement or limit it, they can be omitted in the sentence. The keywords of syntactic rules are specified by the user from the subject area [20]. For each particular subject area, the meaning of the task is set differently, but the structure must satisfy the syntactic requirements of this formulation. The task formulation must satisfy the grammar of the Russian language.

Description of the bank of application software packages. Models of tasks and description of technological processes are stored in the bank of modules in the form of operation tables. Based on the results of interaction of the operating personnel, the control system of the system module bank is analyzed in dialogue mode and according to the data of the bank of attributes and a general model of the control system is created. In this case, first of all, the types of tasks are determined. For the algorithmic technological information system three groups of tasks are distinguished:

1) operational planning;
2) control;
3) analysis and regulation of production situations.

For each type of tasks, the models corresponding to them are defined according to the given attributes. For the first type, planning models and optimization criteria are defined for the given nature and type of production. Based on the attributes defined in the dialogue, input and output forms are analyzed and selected for operational personnel, as well as for adjacent and upper management levels. This approach to setting up input and output forms allows the use of a single database to display non-overlapping sets of details.

According to the specific nature and type of production and selected planning methods, a selection is made on the basis of the attributes of the production situation analysis model.

The conceptual model of the control system built as a result of model composition in the form of a sequence of operations on the tables of functioning is placed in the information part of the bank of application software packages. Then it is analyzed, i.e., the type of computational scheme is established and the scheme of access to the initial data is developed. In the general case for the same private model of the problem there are a number of algorithms effective in the sense of some given criterion, for example, time of the problem solution, minimally used initial information, efficiency of placement in the memory of the machine, accuracy of calculation or reliability of the control system software, etc. According to the calculated pseudo-criterion of the computational scheme efficiency, using the substitution operation, we can transform the conceptual model of the control system into a computational algorithmic scheme, since for each model we have in the information part of the algorithm bank some number of algorithms written in the form of a technological task. Thus, the computational scheme is constructed; it remains to define the schemes of access to the initial information, which are constructed for the whole scheme of the control system. In this case, the input and output of information, carried out in the dialogue mode, can be defined in a convenient form for each specific user.

Then the software is generated in the bank of application software packages according to the given computational scheme and its data access schemes.

The finished software is transferred to an external medium for further operation. The information part of the application package bank consists of library sets, which store source and object images of programs, program data sheets, sets, which store video gram and tabulagram output forms and text data sets.

Conclusion

In conclusion, the choice between monolithic and microservice architectures for the organization and implementation of automated algorithmic technological information systems (AATIS) significantly impacts their performance, scalability, and adaptability. Each architectural paradigm has its own strengths and weaknesses, making their selection dependent on the specific requirements and operational demands of the system being developed. Monolithic architecture, with its simplicity and ease of development, remains a viable option for smaller, less complex systems with stable requirements. Its unified structure allows for streamlined deployment and maintenance, but it becomes increasingly cumbersome as system complexity and scalability needs grow. This can lead to challenges in flexibility, fault isolation, and the ability to adapt to evolving technological and business needs. In contrast, microservice architecture provides a modern, modular approach to system design, enabling independent development, deployment, and scaling of individual components. This architecture is particularly well-suited for dynamic environments where flexibility, fault tolerance, and scalability are paramount. However, it also introduces additional complexity in terms of system integration, communication, and management, which must be carefully addressed through robust strategies and tools. Ultimately, the decision to adopt a monolithic or microservice architecture should be guided by a thorough evaluation of the system's goals, operational context, and long-term scalability requirements. By leveraging the strengths of the chosen architecture and mitigating its limitations, organizations can build AATIS that are robust, efficient, and capable of driving innovation in technological processes. This analysis underscores the importance of architectural planning as a foundational step in the successful deployment of automated systems in the modern digital era.

References

[1]. Schek H. J., Pistor P. Data Structures for an Integrated Data Base Management and Information Retrieval System //VLDB. – 1982. – Т. 82. – С. 197-207.

[2]. Ботиров Т. В. и др. Синтез интервальных управляющих устройств в адаптивные системы управления с эталонной моделью //Инновационные геотехнологии при разработке рудных и нерудных месторождений. – 2020. – С. 231-234.

[3]. Kalandarov I. I., Namozov N. N. Yer osti konlarida xodimlarni xavfsizligini ta’minlash individual qurilmalaridagi signallar kechikish vaqtini aniqlash modeli //Journal of Advances in Engineering Technology. – 2024. – №. 1. – С. 33-36.

[4]. Dayal U., Smith J. M. Probe: A knowledge-oriented database management system //On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies. – New York, NY: Springer New York, 1986. – С. 227-257.

[5]. Kalandarov I., Namozov N., Bozorov B. Analyses and algorithms of personnel safety in mines using event tree and Bayesian network method //E3S Web of Conferences. – EDP Sciences, 2024. – Т. 531. – С. 03018.

[6]. Kalandarov I., Namozov N. N. LoRa signallari yordamida yer osti kon ishlarida xodimlar harakatlanish traektoriyasini aniqlash modeli //Digital Transformation and Artificial Intelligence. – 2023. – Т. 1. – №. 4. – С. 146-148.

[7]. Hoffer J. A., Ramesh V., Topi H. Modern database management. – Pearson, 2016.

[8]. Vakhromeev A. G. et al. Geoecological monitoring of the “Mustaqillikning 25 yilligi” gas field //E3S Web of Conferences. – EDP Sciences, 2023. – T. 417. – C. 04007.

[9]. Chen X. et al. The Binding Database: data management and interface design //Bioinformatics. – 2002. – T. 18. – №. 1. – C. 130-139.

[10]. Kalandarov I., Namozov N., Bozorov B. Yer osti kon ishlarida xodimlar xavfsizligini ta’minlash tizimlari tahlili //Innovatsion texnologiyalar. – 2023. – T. 52. – №. 04.

[11]. Kabulov A. et al. Control System and Algorithm for Construction of Optimal Technological Routes for Machining Parts in the Machining Shop //International Scientific Conference on Agricultural Machinery Industry “Interagromash"”. – Cham: Springer International Publishing, 2022. – C. 2566-2574.

[12]. Frehner M., Brändli M. Virtual database: Spatial analysis in a Web-based data management system for distributed ecological data //Environmental Modelling & Software. – 2006. – T. 21. – №. 11. – C. 1544-1554.

[13]. Каландаров И. И. и др. Преобразователь передачи информации в информационную систему контроля горюче-смазочных материалов //Journal of Advances in Engineering Technology. – 2022. – №. 3. – C. 5-8.

[14]. McHugh J. et al. Lore: A database management system for semistructured data //ACM Sigmod Record. – 1997. – T. 26. – №. 3. – C. 54-66.

[15]. Namozov N. N. Tog ‘-kon sanoatida piyodalar va transport vositalari o ‘rtasida to ‘qnashuvlarini oldini olish usullari va sun’iy intellekt texnologiyasining o ‘rni //Indexing. – 2024. – T. 1. – №. 1.

[16]. Zolnai Z. et al. Project management system for structural and functional proteomics: Sesame //Journal of structural and functional genomics. – 2003. – T. 4. – C. 11-23.

[17]. Namozov N. N. Database and Structure Modeling of Personnel Safety Management Information System //Indexing. – 2024. – T. 1. – №. 1.

[18]. Hudson S. E., King R. Cactis: A self-adaptive, concurrent implementation of an object-oriented database management system //ACM Transactions on Database Systems (TODS). – 1989. – T. 14. – №. 3. – C. 291-321.

[19]. Намозов Н. Н. Анализ систем предотвращения столкновений транспортных средств на подземных месторождениях //Ответственный редактор. – 2024. – C. 60.

[20]. Namozov N. N. Kon sanoatida xodimlar xavfsizligini ta'minlashda xavf omillarini ekstrapolyatsion tahlil asosidagi matematik modeli //Journal of Advances in Engineering Technology. – 2024. – №. 3. – C. 126-130.