JOURNAL OF UNIVERSAL SCIENCE RESEARCH

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

A BUSINESS LOGIC MODEL DEVELOPMENT FOR SOFTWARE OPERATIONS

Vladyslav Yevsieiev[1], Svitlana Maksymova[1], Ahmad Alkhalaileh[2]

[1]Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2]Senior Developer Electronic Health Solution, Amman, Jordan

ABSTRACT

This article presents a business logic model development for software operation based on an event-driven approach. The proposed model ensures flexible interaction between users, events, and message generation processes, optimizing software performance. The study focuses on the mathematical representation of business logic and its practical application in automating email notifications. The results demonstrate the efficiency of the proposed model in enhancing adaptability and scalability.

Keywords: Business Logic, Software Development, Event-Driven Model, Automation, Adaptability, Scalability, Email Notifications, Mathematical Modeling.

INTRODUCTION

Developing a software business logic model is an important stage in the modern process of creating effective and scalable software solutions. In the context of rapid digital transformation and increasing requirements for software quality [1]-[10], business logic is a key component that ensures that the functionality of the software product meets the needs of users and the business processes organization.

Such models can be used for various areas of research and production organization [11]-[23]. In this case, special and classical approaches are used for their implementation [24]-[44].

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

Traditional approaches to software development often focus on technical aspects, while business logic design allows you to create systems that easily adapt to market changes, provide effective integration with other services, and simplify further development.

Using business logic models in the development process allows you to clearly define roles, relationships between objects, and rules for their interaction, which reduces the risk of errors and increases the reliability of the software. In modern methodologies, such as Domain-Driven Design (DDD) and Model-Driven Architecture (MDA), business logic takes a central place, allowing developers to focus on the essence of business processes, and not only on their technical implementation.

In addition, automation of business logic modeling using modern tools and frameworks helps to optimize costs and reduce development time. Thus, the study of business logic models in the context of software development is not only relevant, but also necessary for creating adaptive, flexible and competitive IT solutions.

RELATED WORKS

Analysis of software operation is an integral part of the work and implementation of this software. Accordingly, scientists consider this task from different sides. Including the creation of messages for users in accordance with the business logic of this software, we will consider several works on this topic

In the traditional application model, services are tightly coupled with the processes they support [45]. Authors in [45] clarify the relationship between currently developing standards such as UDDI, WSDL, and WSCL, and propose a conversation controller mechanism that leverages such standards to direct services in their conversations.

Ćatović, A., & et al. [46] describe RabbitMQ that acts as an intermediary between the various services. They demonstrated that the style of microservice architecture is an approach to the development of an application as a set of small services, each in charge of its own process and communication with other services.

Scientists in [47] consider the problem of log message template identification. It is aims to convert raw logs containing free-formed log messages into structured logs to be processed by automated log-based analysis, such as anomaly detection and model inference.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

The paper [48] introduces MontiThings, a C&C language offering automatic error handling capabilities and a clear separation between business logic and implementation details. The error-handling methods presented in this paper can make C&C-based IoT applications more reliable without cluttering the business logic with error-handling code that is time-consuming to develop and makes the models hard to understand, especially for non-experts.

Li, X., and co-authors [49] analyze a blockchain-based flight operation data sharing scheme, named BFOD is designed to achieve the privacy protection and secure sharing of flight operation data. In BFOD, physical entities of airlines, airports and air traffic control are divided into data owners, data requesters and authorization institutes according to the business logic.

Thus, we see that the problem of business logic analysis and management is multifactorial. Later in this article, we will present our business logic model development for software operations.

MATHEMATICAL REPRESENTATION OF SOFTWARE DEVELOPMENT BUSINESS LOGIC USING AN EVENT-DRIVEN MODEL

Business logic of software is a set of rules, algorithms and processes that determine how the software processes data, interacts with users, performs operations and ensures the achievement of set goals. It reflects the specific needs of the business or organization that the software solves, and is a central element of any program architecture. Business logic determines the order of functions, decision-making conditions and methods of processing information in accordance with the needs of end users. Its development allows you to make the program flexible, scalable and adaptable to changes in processes or requirements. Business logic is a necessary component of software development, as it establishes a connection between the technical capabilities of the system and its functional goals. It allows developers to create solutions that meet specific tasks, optimizing internal processes and ensuring effective interaction between users and the system. In the context of automation of e-mail messages, business logic determines which events require sending messages, how these messages are generated, personalized and delivered to users. Its development ensures that the system will respond to events in a timely, relevant and in

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

accordance with established business requirements. This reduces the risk of errors, increases the efficiency of the system and helps meet the needs of end users.

An event-driven business logic model was chosen for the software for automating the process of sending e-mail messages. This model is optimal because it ensures that the system responds to specific events stored in the database or received via API. The event-driven model allows you to build a clear and structured process for processing events that trigger appropriate actions, including message generation, content personalization and their distribution.

The event-driven model provides flexibility and scalability of the system, allowing you to easily add new types of events or functionality without significant changes to the basic architecture. In addition, such a model integrates well with modern analytics services and APIs for data collection, allowing you to respond to the results of event analysis in real time. This is important for achieving the main goals of the developed software, including personalization, responsiveness and communication efficiency.

An event-driven business logic model can be described as a mathematical system based on sets, functions and rules. To develop a business logic model based on an event-driven model, we introduce the following parameters, which are presented in Table 1.

In the context e-mail messages sending automation, the model consists of the following main components:

– sets and parameters of the software being developed:

$$U = \{u_1, u_2, ..., u_n\} \quad (1)$$

$$E = \{e_1, e_2, ..., e_m\} \quad (2)$$

$$M = \{m_1, m_2, ..., m_k\} \quad (3)$$

$$T = \{t_1, t_2, ..., t_n\} \quad (4)$$

U – users set;

u_1 – system user;

E – events set;

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

e_j – event that can trigger a mailing;

M – messages set;

m_k – system-generated message;

T – time set;

t_p – a time parameter that defines the moment when the event occurred or the message was sent.

– functions and dependencies event-driven business logic model for software development.

Table 1: Parameters

Parameter	Description
u_i	unique user ID
e_j	event ID.
m_k	unique message associated with the event
t_p	time of event activation or message sending
f_u	defines the logic of relationships between users and events
f_s	monitors message delivery status

Event handling function:

$$f_E : E \to M \tag{5}$$

$f_E : E$ – defines message m_k, that must be generated for the event e_j.

User subscriptions function:

$$f_U : U \to 2^E \tag{6}$$

$f_U : U$ – defines the set of events to which the user u_i is subscribed;

2^E – set of all subsets E.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

Messages generation function:

$$f_M : M \times U \rightarrow Content \quad (7)$$

$f_M(m_k, u_i)$ – forms the content of the message m_k for the user u_i taking into account personalization.

Time function:

$$f_T : T \rightarrow \{0,1\} \quad (8)$$

$f_T(t_p) = 1$, if message m_k is successfully delivered, and 0 in other case.

The system business logic is described as a functions sequence:

– the incoming event e_j is checked by the time function f_t to determine its relevance;

– for each user u_i who has a subscription to the event e_j, defined through f_U, the event processing function f_E is called, which creates a message m_k;

– the generated message is passed to the messages generation function f_M, which forms a personalized text;

– the message m_k is sent to the user u_i and checked by the delivery status function f_S.

The developed mathematical model provides a formal representation of event-driven business logic, on the basis of which software development is built. The model visualizes the key components of business logic, such as sets of users, events, messages, time and functions that describe their interaction. The model helps to understand how events are processed, messages are generated, their relevance is checked and delivery is ensured. A graphical representation of the mathematical model of event-driven business logic for software for automating the process of sending e-mail messages based on the results of event analysis is presented in Figure 1.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

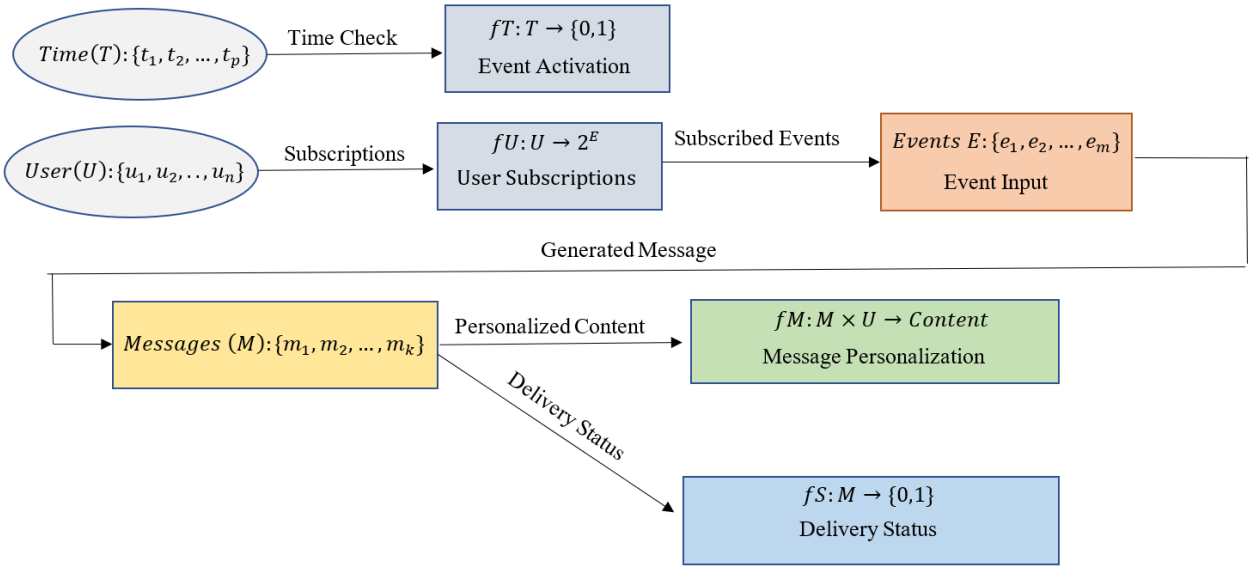

Figure 1: Graphical representation of the mathematical model of event-driven business logic for software to automate the process of sending e-mail messages based on the event analysis results

Based on the developed database structure, the following solution is proposed, that the software is built on the principle of an event-oriented model, where the key components are events (Events), users (Users), user subscriptions (UserSettings) and work log (WorkProtocol). This provides flexibility in creating new analytical events, managing users and their subscriptions, as well as maintaining a log of sendings. The logic of the work involves integration with APIs for data collection, their analysis in the form of events and automatic sending of results to e-mail or messengers. The general view of the algorithm is presented in Figure 2.

Let us describe the purpose of each block of the general algorithm presented in Figure 2.

System initialization:

– loading basic event data from the Events table;

– connecting to the API for collecting information.

User management:

– adding/editing users via the management interface;

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

– setting up event subscriptions from the UserSettings table.

Event launch:

– events can be triggered manually by the user or automatically, if regular triggering is provided;

– collecting data from external sources (for example, external APIs).

Data analysis – performing calculations and generating results in a format ready for sending (text report or table).

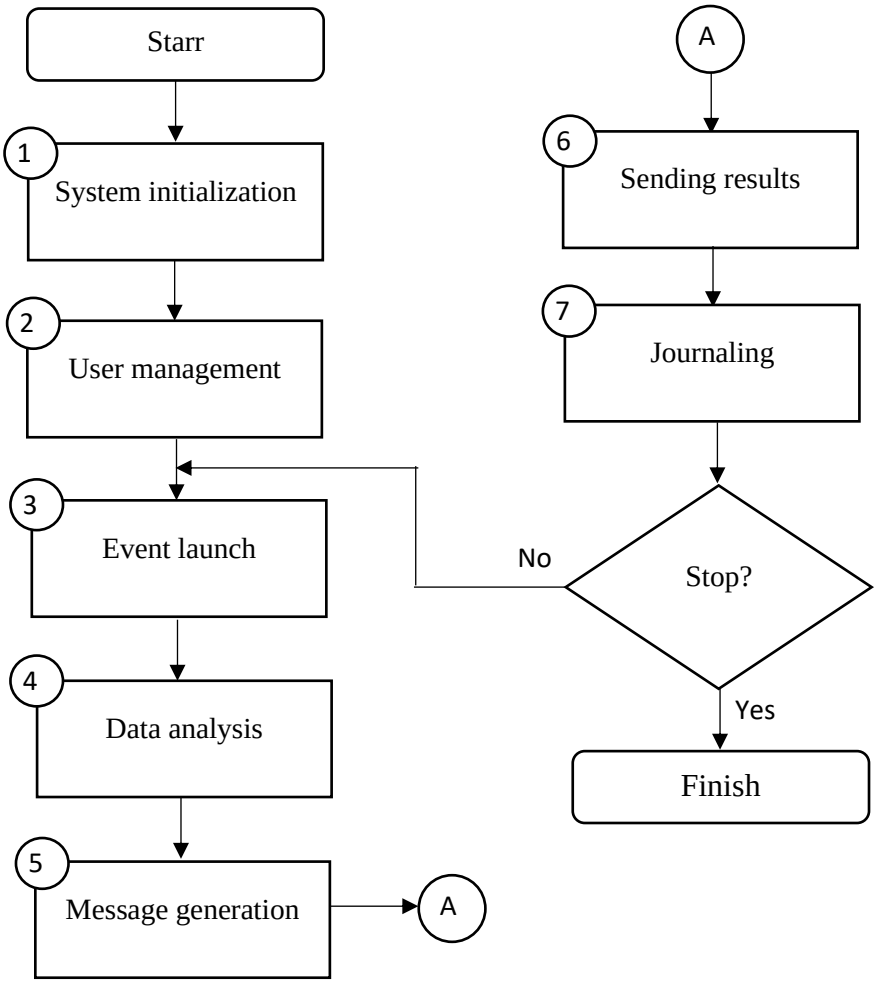

Figure 2: Software operation general algorithm for the sending e-mail messages process automation

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

Message generation – personalizing content for each user based on their subscriptions and specific settings.

Sending results:

– integration with email or messenger services;

– sending messages to users who are subscribed to the corresponding events.

Logging:

– recording data about sending messages in the WorkProtocol table;

– saving information about the date, user, event, and sending status.

The algorithm begins with the launch of an event that initiates the data collection process from the API. The system analyzes the received data, comparing it with the specified event parameters, and generates an analytical report. Based on user subscriptions, those to whom the results need to be sent are determined. Next, content personalization occurs, taking into account the user name, his subscriptions, and the message format. Generated messages are added to the queue for sending via integration with e-mail services or messenger APIs. All operations are logged in the log for further analysis.

The developed algorithm for automating the sending of e-mail messages and messages to messengers has a number of advantages that ensure the efficiency, flexibility, and reliability of the system. It allows you to process events in both manual and automatic modes, which increases the usability for various scenarios. Thanks to integration with the API for data collection, the algorithm can dynamically analyze information and quickly respond to changes. Message personalization ensures that content is precisely tailored to user needs, which contributes to better interaction with customers. The presence of logging allows you to track the history of sending, analyze results, and ensure transparency of work. The use of queues for sending messages minimizes system overload and ensures stable operation even with a large number of requests. This algorithm structure supports scalability and easily adapts to new requirements or additional functions.

CONCLUSION

During the research, an event-oriented model of software business logic was developed, which allows for effective organization of interaction between users, events

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

and data processing mechanisms. The proposed approach provides high flexibility and scalability of the system, which is an important factor in modern software. The mathematical representation of the model allowed for the formalization of key processes; including event generation, message personalization and delivery management. The algorithmic approach to the implementation of business logic ensures stability and reliability of the system even under heavy load. The proposed model effectively integrates with external services and APIs, which allows for improved interaction between components of the software complex. The developed system for automating the process of sending e-mail messages demonstrates high efficiency in the context of timely response to events and content personalization. The use of an event-oriented approach allows for minimizing data processing time and increasing the speed of the system's response to changes in business processes. The implementation of such business logic contributes to an increase in the level of automation, which reduces the need for manual intervention and optimizes software support costs. Thus, the results of the study confirm the feasibility of using an event-driven model for building adaptive and productive software solutions.

REFERENCES

1. Demska, N., & et al. (2025). Analysis of Methods, Models and Algorithms for a Collaborative Robots Group Decentralized Control. ACUMEN: International journal of multidisciplinary research, 2(2), 235-249.
2. Yevsieiev, V., & et al. (2024). Calculation of the Distance to Objects in Collaborative Robots Workspace Using Computer Vision. Journal of universal science research, 2(11), 240-255.
3. Basiuk, V., & et al. (2024). Command System for Movement Control Development. Multidisciplinary Journal of Science and Technology, 4(6), 248-255.
4. Chala, O., & et al. (2024). Analysis of Systems for Coordination of Enterprise Subsystems Control. Journal of universal science research, 2(10), 127-137.
5. Yevsieiev, V., & et al. (2024). A Program for Analyzing the Structure of a Web site Development Using the Parsing Method Based on the Python. Journal of Universal Science Research, 2(4), 172-183.
6. Gurin, D., & et al. (2024). Web site reliability analysis using the python parsing method. Journal of Universal Science Research, 2(5), 113-126.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

7. Yevsieiev, V., & et al. (2024). Data Fusion Research for Collaborative Robots-Manipulators within Industry 5.0. ACUMEN: International journal of multidisciplinary research, 1(4), 125-137.
8. Moiseev, M., & et al. (2024). Program Algorithm for Monitoring System Development. Journal of universal science research, 2(7), 33-43.
9. Chala, O., & et al. (2024). Switching Module Basic Concept. Multidisciplinary Journal of Science and Technology, 4(7), 87-94.
10. Yevsieiev, V., & et al. (2024). Human Operator Identification in a Collaborative Robot Workspace within the Industry 5.0 Concept. Multidisciplinary Journal of Science and Technology, 4(9), 95-105.
11. Attar, H., Abu-Jassar, A. T., Amer, A., Lyashenko, V., Yevsieiev, V., & Khosravi, M. R. (2022). Control system development and implementation of a CNC laser engraver for environmental use with remote imaging. Computational intelligence and neuroscience, 2022(1), 9140156.
12. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural networks as a tool for pattern recognition of fasteners. International Journal of Engineering Trends and Technology, 69(10), 151-160.
13. Abu-Jassar, A. T., Al-Sharo, Y. M., Lyashenko, V., & Sotnik, S. (2021). Some Features of Classifiers Implementation for Object Recognition in Specialized Computer systems. TEM Journal: Technology, Education, Management, Informatics, 10(4), 1645-1654.
14. Abu-Jassar, A. T., Attar, H., Yevsieiev, V., Amer, A., Demska, N., Luhach, A. K., & Lyashenko, V. (2022). Electronic user authentication key for access to HMI/SCADA via unsecured internet networks. Computational intelligence and neuroscience, 2022(1), 5866922.
15. Nevliudov, I., Yevsieiev, V., Baker, J. H., Ahmad, M. A., & Lyashenko, V. (2020). Development of a cyber design modeling declarative Language for cyber physical production systems. J. Math. Comput. Sci., 11(1), 520-542.
16. Lyashenko, V., Abu-Jassar, A. T., Yevsieiev, V., & Maksymova, S. (2023). Automated Monitoring and Visualization System in Production. International Research Journal of Multidisciplinary Technovation, 5(6), 9-18.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

17. Nevliudov, I., & et al.. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis. International Journal of Emerging Trends in Engineering Research, 8(10), 7465-7473.
18. Mustafa, S. K., Yevsieiev, V., Nevliudov, I., & Lyashenko, V. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. SSRG International Journal of Engineering Trends and Technology, 70(1), 139-145.
19. Nevliudov, I., Yevsieiev, V., Lyashenko, V., & Ahmad, M. A. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive Cyber-Design CPPS Development. Advances in Dynamical Systems and Applications, 16(2), 441-455.
20. Kobylin, O., & Lyashenko, V. (2014). Comparison of standard image edge detection techniques and of method based on wavelet transform. International Journal, 2(8), 572-580.
21. Abu-Jassar, A. T., Attar, H., Lyashenko, V., Amer, A., Sotnik, S., & Solyman, A. (2023). Access control to robotic systems based on biometric: the generalized model and its practical implementation. International Journal of Intelligent Engineering and Systems, 16(5), 313-328.
22. Babker, A. M., Abd Elgadir, A. A., Tvoroshenko, I., & Lyashenko, V. (2019). Information technologies of the processing of the spaces of the states of a complex biophysical object in the intellectual medical system health. International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3221-3227.
23. Al-Sherrawi, M. H., Lyashenko, V., Edaan, E. M., & Sotnik, S. (2018). Corrosion as a source of destruction in construction. International Journal of Civil Engineering and Technology, 9(5), 306-314.
24. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized procedure for determining the collision-free trajectory for a robotic arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.
25. Tvoroshenko, I., Lyashenko, V., Ayaz, A. M., Mustafa, S. K., & Alharbi, A. R. (2020). Modification of models intensive development ontologies by fuzzy logic. International Journal of Emerging Trends in Engineering Research, 8(3), 939-944.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

26. Khan, A., Joshi, S., Ahmad, M. A., & Lyashenko, V. (2015). Some effect of Chemical treatment by Ferric Nitrate salts on the structure and morphology of Coir Fibre Composites. Advances in Materials Physics and Chemistry, 5(1), 39-45.
27. Sotnik, S. Overview: PHP and MySQL Features for Creating Modern Web Projects/ S Sotnik, V. Manakov, V. Lyashenko //International Journal of Academic Information Systems Research (IJAISR). – 2023. – Vol. 7, Issue 1. – P. 11-17.
28. Matarneh, R., Tvoroshenko, I., & Lyashenko, V. (2019). Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space. International Journal of Recent Technology and Engineering, 8(4), 1687-1693.
29. Lyashenko, V. V., Matarneh, R., Baranova, V., & Deineko, Z. V. (2016). Hurst Exponent as a Part of Wavelet Decomposition Coefficients to Measure Long-term Memory Time Series Based on Multiresolution Analysis. American Journal of Systems and Software, 4(2), 51-56.
30. Lyashenko, V. V., Deineko, Z. V., & Ahmad, M. A. Properties of wavelet coefficients of self-similar time series. In other words, 9, 16.
31. Kuzemin, A., Lyashenko, V., Bulavina, E., & Torojev, A. (2005). Analysis of movement of financial flows of economical agents as the basis for designing the system of economical security (general conception). In Third international conference «Information research, applications, and education (pp. 27-30).
32. Deineko, Zh., & et al.. (2021). Features of Database Types. International Journal of Engineering and Information Systems (IJEAIS), 5(10), 73-80.
33. Lyashenko, V., Laariedh, F., Ayaz, A. M., & Sotnik, S. (2021). Recognition of Voice Commands Based on Neural Network. TEM Journal: Technology, Education, Management, Informatics, 10(2), 583-591.
34. Ahmad, M. A., Baker, J. H., Tvoroshenko, I., & Lyashenko, V. (2019). Computational complexity of the accessory function setting mechanism in fuzzy intellectual systems. International Journal of Advanced Trends in Computer Science and Engineering, 8(5), 2370-2377.
35. Tahseen A. J. A., & et al.. (2023). Binarization Methods in Multimedia Systems when Recognizing License Plates of Cars. International Journal of Academic Engineering Research (IJAER), 7(2), 1-9.
36. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2025). Development and Investigation of Vision System for a Small-Sized

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

Mobile Humanoid Robot in a Smart Environment. International Journal of Crowd Science, 9(1), 29-43.

37. Abu-Jassar AT, Attar H, Amer A, et al. Remote Monitoring System of Patient Status in Social IoT Environments Using Amazon Web Services (AWS) Technologies and Smart Health Care. International Journal of Crowd Science, 2024.
38. Attar, H., Abu-Jassar, A. T., Lyashenko, V., Al-qerem, A., Sotnik, S., Alharbi, N., & Solyman, A. A. (2023). Proposed synchronous electric motor simulation with built-in permanent magnets for robotic systems. SN Applied Sciences, 5(6), 160.
39. Abu-Jassar A. Building a Route for a Mobile Robot Based on the BRRT and A*(H-BRRT) Algorithms for the Effective Development of Technological Innovations / Amer Abu-Jassar, Hassan Al-Sukhni, Yasser Al-Sharo, S. Maksymova, V. Yevsieiev, V. Lyashenko // International Journal of Engineering Trends and Technology. – 2024. – V. 72(11). – P. 294-306.
40. Ababneh, J., Abu-Jassar, A., Abuowaida, S., Liubchenko, V., & Lyashenko, V. (2024, December). Evaluation of Three Different Operators for Object Highlighting in Medical RGB Images: Canny, Roberts, and LoG in Independent Color Spaces. In 2024 25th International Arab Conference on Information Technology (ACIT) (pp. 1-7). IEEE.
41. Kuzomin, O., Lyashenko, V., Tkachenko, M., Ahmad, M. A., & Kots, H. (2016). Preventing of technogenic risks in the functioning of an industrial enterprise. International Journal of Civil Engineering and Technology, 7(3), 262-270.
42. Serhiienko, O., Novikova, T., & Lyashenko, V. (2023). Comparative Analysis of the Dynamics of Futures for the Dow Jones, S&P 500 and Nasdaq. International Journal of Academic Accounting, Finance & Management Research (IJAAFMR), 7(9), 22-28.
43. Matarneh, R., Sotnik, S., Belova, N., & Lyashenko, V. (2018). Automated modeling of shaft leading elements in the rear axle gear. International Journal of Engineering and Technology (UAE), 7(3), 1468-1473.
44. Omarov, M., Tykha, T., & Lyashenko, V. (2019). Use of Wavelet Techniques in the Study of Internet Marketing Metrics. Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering, 20, 157-163.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2024 SJIF 2024 = 5.073/Volume-3, Issue-3

45. Kuno, H., & et al. (2001). Conversations+ interfaces= business logic. In Technologies for E-Services: Second International Workshop, TES 2001 Rome, Italy, September 14–15, 2001 Proceedings 2 (pp. 30-43). Springer Berlin Heidelberg.
46. Ćatović, A., & et al. (2022). Microservice development using RabbitMQ message broker. Science, engineering and technology, 2(1), 30-37.
47. Khan, Z. A., & et al. (2022). Guidelines for assessing the accuracy of log message template identification techniques. In Proceedings of the 44th International Conference on Software Engineering, 1095-1106.
48. Kirchhof, J. C., & et al. (2022). Montithings: Model-driven development and deployment of reliable iot applications. Journal of Systems and Software, 183, 111087.
49. Li, X., & et al. (2023). BFOD: Blockchain-based privacy protection and security sharing scheme of flight operation data. IEEE Internet of Things Journal, 11(2), 3392-3401.