Automation Methodology for Complex Technical-Organizational Systems

V.V. Borisov
National Research University "MPEI"
Moscow, Russia
vbor67@mail.ru

A.E. Misnik
Inter-state educational institution of higher education “Belarusian-Russian university”
Mogilev, Republic of Belarus
anton@misnik.by

S.K. Krutalevich
Inter-state educational institution of higher education “Belarusian-Russian university”
Mogilev, Republic of Belarus
s_krutolevich@tut.by

S.A. Prokopenko
National Research University "MPEI"
Moscow, Russia
puss95@yandex.by

Abstract*— This article is devoted to methodology of automation of complex technical-organizational systems. Traditional approaches suffer from semantic gaps in the transfer of information from an expert to a knowledge engineer and from a knowledge engineer to developers, duplication of data in the system, complexity, and, often, the inability to organize relationships between data in different modules, necessity to involve developers to create new system applications, make changes to the logic and displaying data in the existing applications. All those factors lead to decrease of the flexibility and versatility of software lifecycle. The proposed approach designed to optimize the automation software lifecycle. It is proposed to decline the hard-coded systems, and embrace the software-instrumental environment, which allows to the Expert (advanced user of Excel, for example), set up the ontological data structure and implement business processes without including knowledge engineers and developers in its lifecycle. Using this approach, the semantic gap is being eliminated; the involvement of developers is only necessary in a situation when it is necessary to develop new or adjust existing tools of the environment. It is proposed to include in software structure ontological component, utilizing object-oriented approach to implement data structure and business processes; analytical component to process data and generate analytical information; user-interaction component to provide end-user with all needed functionality and background processes component. System implementation example industrial information system of collecting and processing data GIAP-DISTcenter has been presented

***Keywords*—complex systems, complex technical-organizational systems, ontology, automation**

I. Introduction

Modern enterprises, as technical-organizational systems, are becoming increasingly complex both in structure and in internal connections between structural components. The prerequisites for these processes are: expansion to foreign markets, high competition, tendency to reduce costs, ensuring the safety of production, the necessity for a dynamic response to market needs. Also, there are growing demands on product quality, environmental protection, as well as occupational safety and health.

Due to the development of technologies in the field of instrumentation, the amount of data available for analysis on the state of technical devices has grown significantly, besides, it is obvious, that automation of processes reduces the number of errors caused by the human factor. These aspects should be controlled using appropriate analytical data processing information systems, that ensure achievement of goals in conditions of significant uncertainties, especially in the context of long time horizons.

Currently, the growth rate of requirements for information systems and the necessity for their modification during operation is very high. Very often there is the necessity to modify the existing data structure, the way of data displaying, new data processing scenarios appear. Involving developers to solve such problems is usually costly both in terms of time and financial expenditures.

II. Traditional Approach to the Automation of Complex Technical-Organizational Systems

The lifecycle of traditional approach to the automation of complex technical-organizational systems is presented on Figure 1.

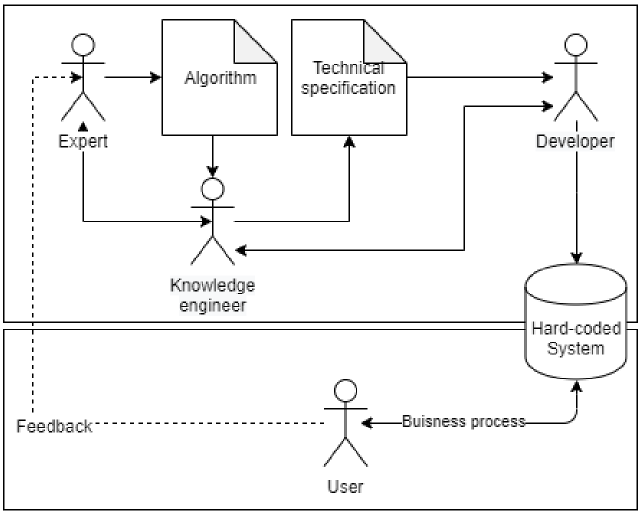

Fig. 1. Lifecycle of traditional approach to the automation of complex technical-organizational systems

In the framework of the traditional approach to implementation of automation software, each part of such a system is, in fact, a separate software module, with its own static data structure and internal logic that is rigidly defined at the stage of system design and implementation.

This approach has the following disadvantages:

- semantic gaps in the transfer of information from an expert to a knowledge engineer and from a knowledge engineer to developers;
- duplication of data in the system;

978-1-7281-4810-6/20/$31.00 ©2020 IEEE

- the complexity, and, often, the inability to organize relationships between data in different modules;
- the necessity to involve developers to create new system applications, make changes to the logic and displaying data in the existing applications.

III. Proposed Methodology of the Automation of Complex Technical-Organizational Systems

In our opinion, it is optimal to develop a software-instrumental environment that allows a specialist in the subject area with the skills of an advanced Excel user (Expert) to create and modify data structure, develop applications for displaying and interacting with data, and implement data processing scenarios. The lifecycle of such automation process is presented on Figure 2.

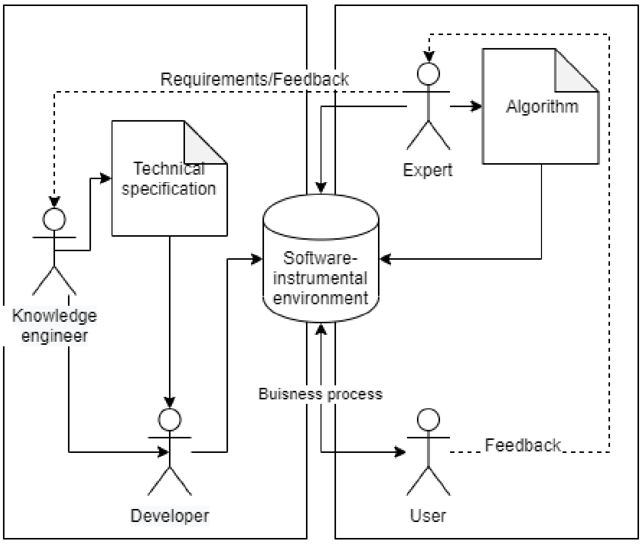

Fig. 2. Lifecycle of proposed approach to the automation of complex technical-organizational systems

Using this approach, the semantic gap is being eliminated; the involvement of developers is only necessary in a situation when it is necessary to develop new or adjust existing tools of the environment. The expert must have only basic programming skills. The efficiency of the implementation of system's information and analytical processes, within the framework of the existing tools, is high. The structure of software-instrumental environment is presented on Figure 3.

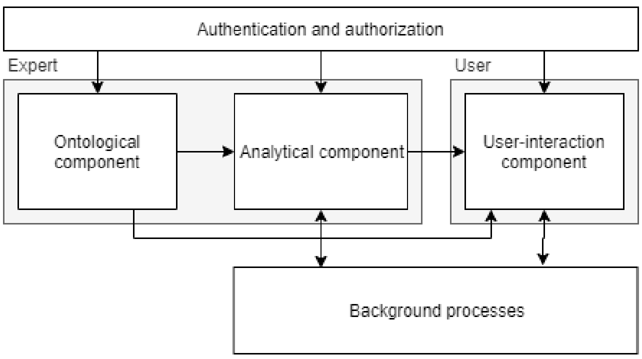

Fig. 3. Structure of software-instrumental environment

The ontological component's structure is presented of Figure 4.

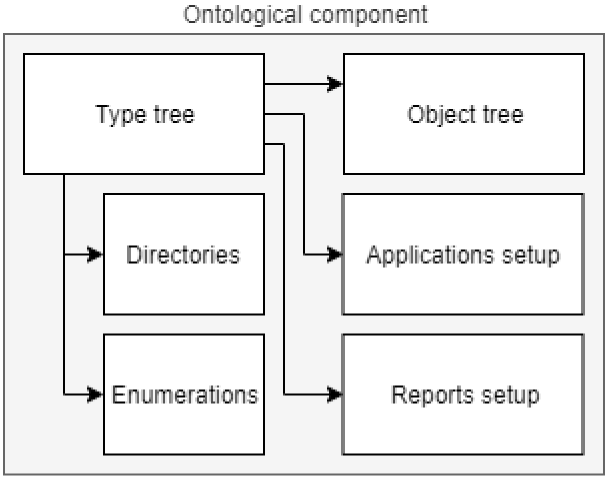

Fig. 4. Ontological component

It is proposed to use an object-oriented approach as the basis for organizing data structure - the subsystem "Type Tree".

Type, in full accordance with the classical approach, is a description of the objects through their common attributes. An attribute of a type is a named property or characteristic of an object of that type. Each attribute of a type is characterized by a name unique within this type and the datatype that this attribute will store. All instances (objects) of the same type have the same set of attributes. Type instances differ in attribute values. Attributes by the values of which each object can be identified from other objects of this type are called key attributes. If there are several key attributes, a composite key is generated. It is proposed to use the following main types of attributes: string, number, counter, date, date and time, text, file. For attributes of the form string, number, date, date and time, the file should be available the ability to set the property "array". Also, an attribute may be a link to another type of system's data structure [1].

Types should be in a hierarchical relationship and can be arbitrarily connected both with each other and with themselves if required by the data structure or business process. [3,9]

Such an approach allows, without the involvement of developers and database engineers, to form hierarchical data structures of arbitrary nesting by Expert with the necessary connections between hierarchy levels.

To support functionality of "Type tree" the subsystems "Directories" and "Enumerations" are proposed. They should be capable to create and store in unique form data sources for the attributes of the "Type Tree".

The "Object tree" is the subsystem for displaying the system data in full accordance with the "Type Tree" structure.

In practice, business processes rarely perfectly reflect the data structure. To create a way of data displaying for the end-user of the system, a subsystem is required that can interconnect data located at different levels of the hierarchy, and in different branches of the "Type Tree". [3]

As such a subsystem, "Applications setup" is proposed.

Within the framework of this subsystem, the capabilities to configure the displaying of data both in the form of modal forms and in the form of tables for viewing and editing information should be realized. The range of settings should allow the formation of a wide variety of business processes.

User interfaces should have a hierarchical structure that provides consistent output of related data.

This approach allows to abandon the pre-configuration of all possible options for using in the business process.

Complex technical-organizational systems often require the formation of “hard” copy of information. Usually, the form of such “hard copy” is different both from data structure and from the business process. It requires the implementation of subsystem capable to create templates for such documents and form a “hard” copy using such template. To solve this task the subsystem “Reports setup” is proposed.

In the complex technical-organizational systems it is not enough just to create means to input and output information. A key feature of data processing systems oriented to use by engineering and technical personnel is the necessity to provide the user with analytical data based on the information available in the system.

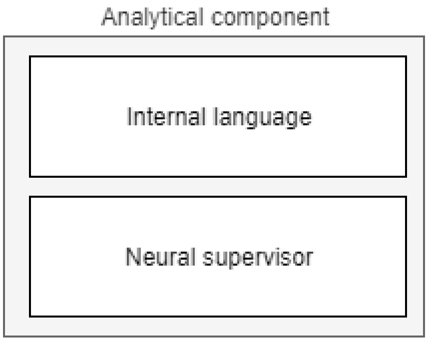

Fig. 5. Analytical component

To process data and generate analytical information, it is proposed to use the analytical component, which should have an intrasystem meta-programming language that allows, first of all, to seamlessly operate data in the system in the best possible user-friendly form, mainly using system dialogs - "masters".

In order, to reduce the user entry threshold for interacting with this subsystem and increase the level of security, it is not recommended to use traditional programming languages, but to develop unique internal meta-programming language. This meta-language should implement the principles of linear programming, including basic algorithmic constructs, have access to the data structure of the subsystem "Type Tree" in a user-friendly form.

Practice shows that the main necessity of users when working with industrial analytical systems for data collection and processing is to obtain the necessary data, simply process it and record the results. Any algorithmically complex operations for the user can be implemented by programmers within the framework of functions connected to the “Internal language” subsystem [2] .

The application of this approach will allow Expert to form business logic for data processing within the system without involving developers.

The big problem of analytical systems for data collection and processing is the quality of the data, both system inputs and results of analysis. For input data, the problem is exacerbated if they are fully or partially entered by operators. As for the analytical data obtained by calculations, in addition to the risks associated with the source data, the risks of imperfection of the calculation method affect their accuracy.

Existing approaches, such as, for example, double-entry of information or constant monitoring of its entry, are quite expensive both financially and in terms of time. Such solutions are not always reasonable and effective.

One of the possible solutions of the quality problem of source and analytical data in industrial systems is to use the supervisor neural network module, which able in real time to check the data changing in the system. The implementation of such a “Neural supervisor” in each case should be individual, therefore it is most convenient to implement it in the form of a certain designer tool, within which the Expert can choose the network architecture and topology, as well as input and output parameters. If there is a possibility of incorrect data entry, the supervisor informs the operator about the necessity to verify the entered data, if the probability of error is recognized in the analytical data, a message is sent to the expert or system Architect. For additional training, the neural network "Neural supervisor" monitors the response of users to messages sent to them about probable errors. Also, training can take place under the supervision of an Expert [4] .

Using the neural network "Neural supervisor" allows to improve the quality of the input data and strengthen control over the results of analytical calculations [5,6,9] .

User-interaction component structure is presented on Figure 6.

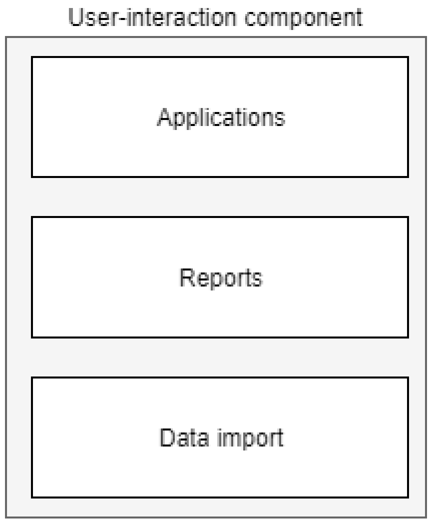

Fig. 6. User-interaction component

The “Applications” subsystem implements the functionality of interacting with data, according to rules from “Applications setup” subsystem.

The “Reports” subsystem should be capable of transferring the “hard” copy of a document from system to user, according to the template from “Reports setup” subsystem.

The “Data import” subsystem should provide the functionality of mass data input from different sources, most common choice is, obviously, Excel or Google Sheets.

Some informational-analytical processes can be held in real time, but huge amounts of evaluations should be processed in background.

The structure of proposed “Background processes” component is presented on Figure 7.

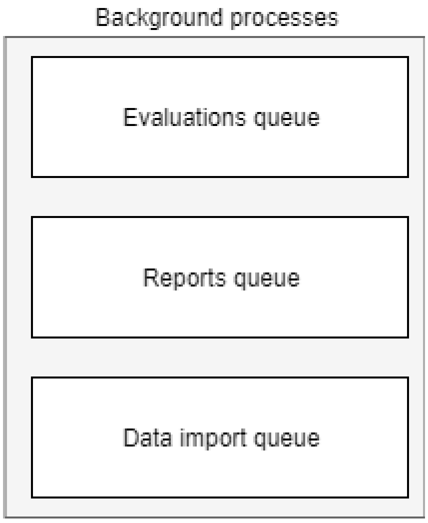

Fig. 7. Background processes component

This component includes:

- “Evaluation queue” for mass processing scripts from “Internal language” subsystem;
- “Reports queue” for mass creation of “hard” copies;
- “Data import queue” for mass processing of input data.

IV. Example of Methodology Implementation

ZAO “GIAP-DISTcenter” was established in 1995, the main activity is to ensure industrial safety of hazardous production facilities.

When carrying out large projects, such as the creation of control systems for technical devices, including as part of unique work to increase the intervals between overhauls of oil and gas and chemical complexes, the forces of relevant leading institutes and expert organizations of the country are consolidated to fulfill the assigned tasks.

To ensure the reliability and safety of industrial plants, it is necessary to take into account the accumulated scientific knowledge and some existing approaches, supported by good engineering practice, which are included in international standards and guidelines.

The developed methodological documents, such as ICTE 1-002-14, ICTE 3-002-14, ICTE 3-003-14, as well as software created by own IT department, allow to optimize the process of analyzing the actual condition of the equipment, timely identify critically dangerous objects and provide industrial safety.

Based on many years of experience and analysis of existing systems, it was decided to develop own industrial analytical systems for data collection and processing, which allows to solve specific problems in the field of ensuring industrial safety of the enterprise, as well as in the field of information support of various business processes in the enterprise. The modularity and flexibility of the system allows to quickly adapt it for implementation at a particular enterprise.

GIAP-DISTcenter, the industrial information system for data collection and processing, is a client-server application that can be accessed both from stationary computers and mobile devices, both from a local computer network and via the Internet, which significantly increases the flexibility of working with it and also increases the efficiency of access to the necessary data.

This system includes the following main subsystems:

- the subsystem for managing users and user groups, responsible for providing users with access to the system and setting their privileges;
- the subsystem for constructing a “Type tree” responsible for the formation of a data structure;
- the subsystem for constructing a “Object tree” that allows managing the data entered into the system;
- the subsystem for configuring and displaying user “Applications”, which is responsible for setting up and ensuring the functioning of the business processes of the system;
- the subsystem of “Internal language” processing;
- the subsystem of units of measurement, providing the conversion of data from one dimension to another;
- the data conversion subsystem that allows bringing complex-structured data, for example, received from diagnostic devices, into the desired form;
- “Reports” subsystem responsible for generating documents in the system;
- the subsystem for constructing two-dimensional and three-dimensional schemes, providing visualization of system objects;
- the “Data import” subsystem that allows loading data into the system from various sources;
- neural network supervisor;
- the “Background processes” subsystem.

The general views of the user interface of “Type tree” and “Internal language” subsystems are presented on Figures 8 and 9 [7] .

Within the framework of “Internal language” subsystem, an internal linear programming language is implemented, as well as the following sets of functions:

- mathematical functions;
- logical functions;
- functions of interactions with dates (getting today's date, getting the difference between dates, setting the date, getting the maximum/minimum date);
- statistical functions;
- functions for working with strings;
- conversion functions (converting a string to a number, converting Arabic numbers to Roman, and vice versa).

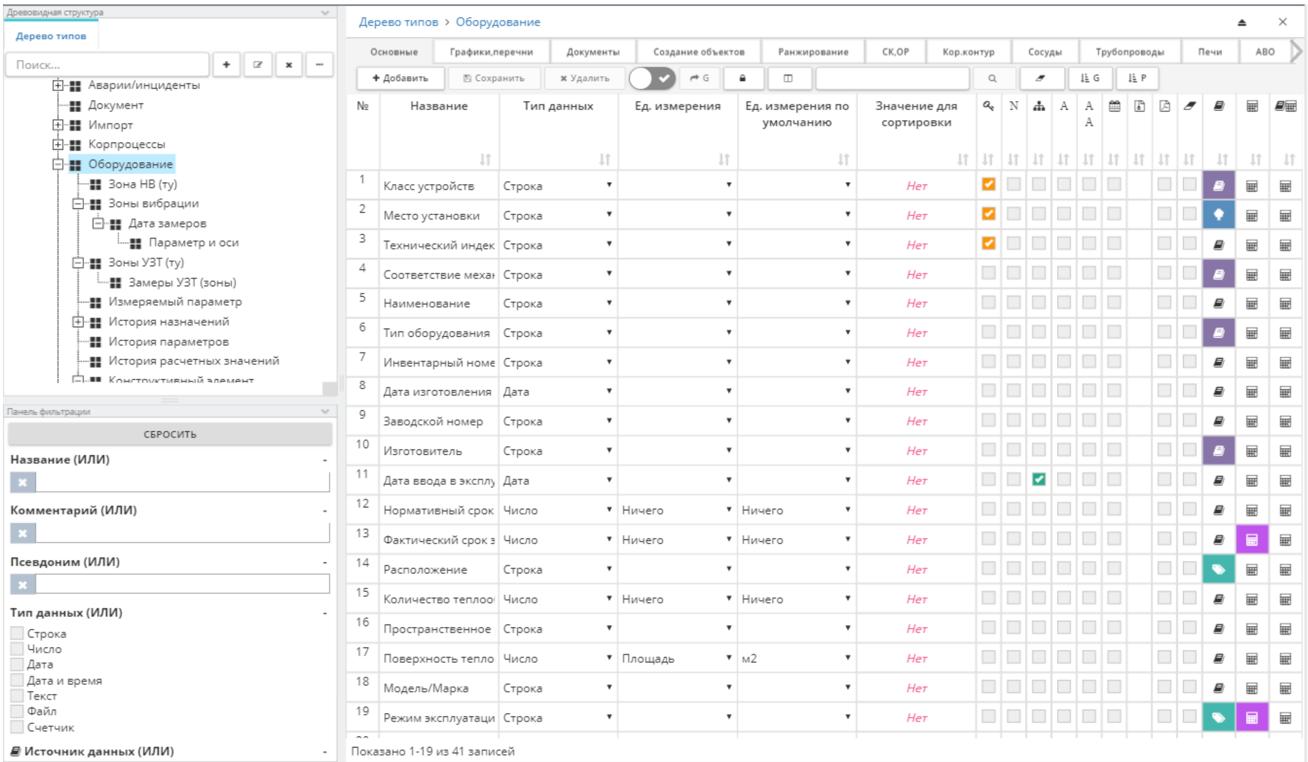

Fig. 8. “Type tree”

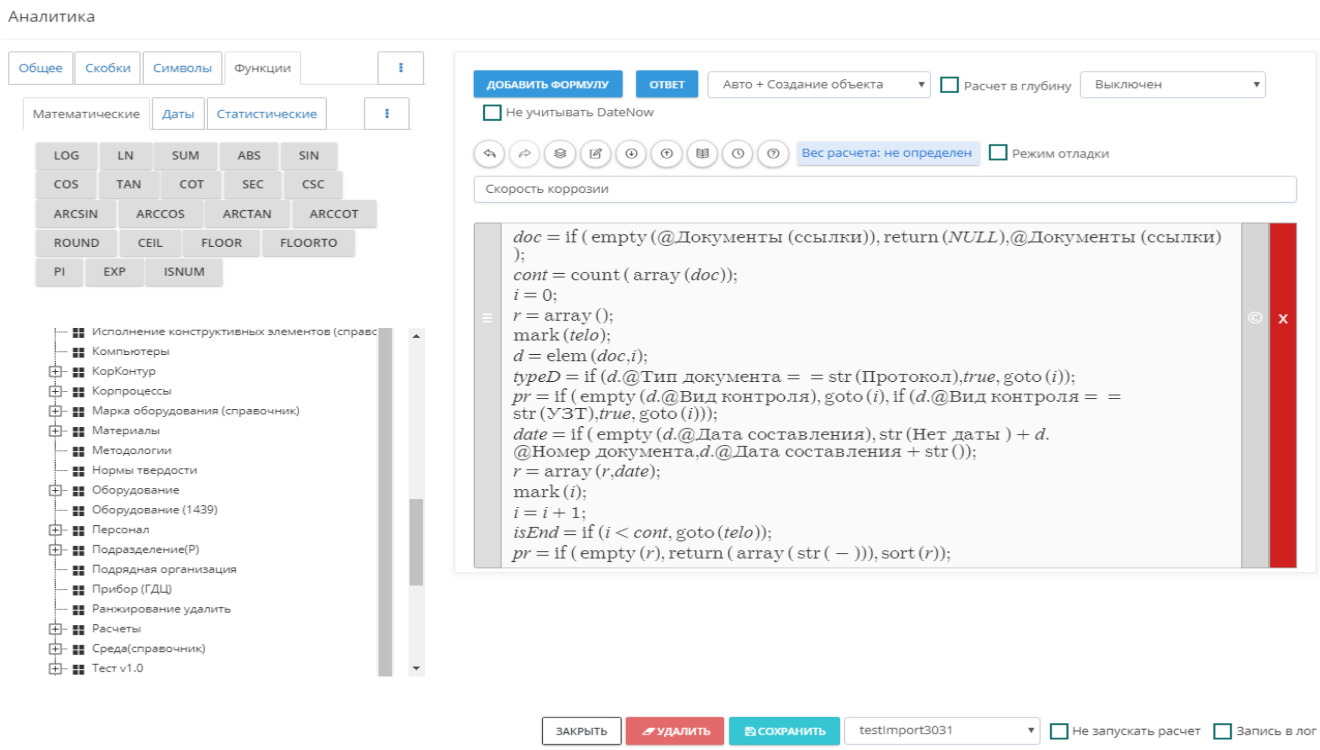

Fig. 9. “Internal language”

CONCLUSION

In this paper, the issues of automation of complex technical-organizational systems were described. The traditional and proposed approaches have been compared. It was shown, how the proposed approach grants the elimination of semantic gaps in the transfer of information from an expert to a knowledge engineer and from a knowledge engineer to developers as well, as other disadvantages. The main disadvantage of proposed methodology – is the complexity of development of described software-instrumental environment. Also, the example of the implementation of the methodology – the industrial information system of collecting and processing data GIAP-DISTcenter has been presented.

REFERENCES

[1]. Bertrand Meyer. Object-oriented software construction, volume 2. Prentice hall New York, 1988.

[2]. Kim Wald´en and Jean Marc Nerson. Seamless object-oriented software architecture.Prentice-Hall, 1995.

[3]. Misnik Anton, Krutalevich Siarhei, Intelligent decision support in an industrial data collection and processing system. XVI All-Russian Scientific Conference "Neurocomputers and their application." Abstracts. 2018 p. 204-205

[4]. Prakapenka Siarhei, Misnik Anton, Decision support system for analysis risks of failure of technical systems. XVI All-Russian Scientific Conference "Neurocomputers and their application." Abstracts. 2018 p. 205-206

[5] Misnik Anton, Krutolevich Sergey, Prakapenka Siarhei, Lukjanov Eugene Methodology for Development of Industrial Analytical Systems for Data Collection and Processing. Proceedings of the 14th International Conference on Interactive Systems: Problems of Human-Computer Interaction. Ulyanovsk, Russia, September 24-27, 2019. p. 223-231

[6]. Andrew S. Tanenbaum, Maarten van Steen. "Distributed Systems. Principles and paradigms". Prentice Hall, Inc., 2002

[7]. Nancy G Leveson, Mats Per Erik Heimdahl, Holly Hildreth, and Jon Damon Reese. Requirements specification for process-control systems. Software Engineering, IEEE Transactions on, 20(9):684–707, 1994.

[8]. Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju. Web Services. Concepts,Architectures and Applications. Springer-Verlag, 2004

[9]. Misnik Anton, Krutolevich Sergey, Prakapenka Siarhei, Lukjanov Eugene Methodology for Development of Industrial Analytical Systems for Data Collection and Processing. Proceedings of the 14th International Conference on Interactive Systems: Problems of Human-Computer Interaction. Ulyanovsk, Russia, September 24-27, 2019. p. 223-231