Models and Method for Estimate Information-Time Characteristics of Real-Time Control System

Kosolapov Anatolii

Dnipro National Rail University, Dnipro, 49010, Ukraine

Abstract — *In the work proposes models for describing and calculating the characteristics of automated real-time control systems at the conceptual design stage. The models are based on the proposed notion of a φ - transaction.*

Keywords — *φ -transaction, conceptual design, real-time control systems.*

I. INTRODUCTION

This modern control systems are geographically and functionally distributed systems of large sizes, combining thousands of computers and working in real time. When creating and improving such systems, the role of conceptual or system design increases [1], [2], [4], [6]. Under these conditions, the process of selecting design decisions becomes more complicated due to the lack of basic data and models for evaluating the information and time characteristics of data processing and transmission processes. Special attention should be paid to real-time systems, in which the violation of time constraints can lead to emergency situations [1], [4], [5], [11]. The paper proposes simple graph-analytical models for engineering evaluation of the following characteristics of computer systems: the processors performance, the amount of information flow, the processing time of applications. The basis of the models is the concept of φ-transaction. The developed models and methods are part of Framework Conceptual Design of Complex Real-Time Management System (CoDeCS) [1].

Recent trends in the transition to cloud systems do not remove the relevance of the problem of evaluating the characteristics of computer systems and their elements (processors, switches, communication channels, databases). After all, cloud systems are information infrastructures with a new data processing architecture [17, 18]. The use of cloud systems for enterprise management translates their computer systems into a class of systems operating in real time [18]. The study of the characteristics of such systems is usually associated with the use of complex and expensive modeling packages. This limits their application, especially at the stages of conceptual design [1]. Therefore, simplified methods are proposed that allow obtaining acceptable parameter values for making design decisions [1], [19]. The models and analytical methods described in the work allow calculations in the environment of spreadsheets on tablets and smartphones.

II. RELATED WORK

The massive development of real-time systems began in the 1970s. This is due to the emergence of reliable computers of the third generation. At this time, there is an introduction of the process control system in industry and transport [7]. One of the main characteristics of control computers is their speed, since it is necessary to ensure the execution of functions for a limited time [8]. At this time, a large number of analytical and simulation methods for estimating the speed of control computers and various performance metrics appear: linear regression models, benchmark programs, or mixes, queuing system models, Petri net models and others, MIPS, FLOPS metrics [8], [10], [11], [12], [13]. The first systems were centralized systems. But with the advent of industrial computers and microcontrollers, there has been a transition to hierarchical geographically and functionally distributed systems, where the role of information flows that affect productivity increases [4], [5]. At the same time, the concept of a transaction similar to that used in banking systems was used in some automated process control systems [3], [14]. For the first time, the author systematized the methods for evaluating the performance of processors in their selection under various design conditions (detailed description of the tasks to be solved) [1], [2]. The relevance of this is confirmed by [15], [16]. Another one of the requirements for models was the simplicity of their calculation in the environment of spreadsheets on gadgets. This is important for the conceptual design of computer systems with a lot of variations in the choice of processors.

.

III. THE CONCEPT OF THE Φ-TRANSACTION AND ITS PROPERTIES

For the study and design of real-time information management systems of sorting stations, we introduce the concept of φ-transaction.

φ-transaction is a logically unified sequence of functional-algorithmic and program blocks (FPB) of operations, which is processed completely from the moment an event occurs in the control system that requires processing until the end of its processing with the output of a message and/or control management action.

φ-transaction must own ACID properties [3]. It is an abbreviation of four words, which means atomicity, consistency, isolation and durability.

Atomicity. Atomicity is a unit of work. In relation to the φ-transaction, this means that either the entire unit of work will be successfully completed, or nothing will be changed.

Consistency. The states before the start of the φ-transaction and upon its completion must be correct. During the φ-transaction state may have intermediate values.

Isolation. Isolation means that the φ-transactions that are executed simultaneously are isolated from the state that changes during the φ-transaction A

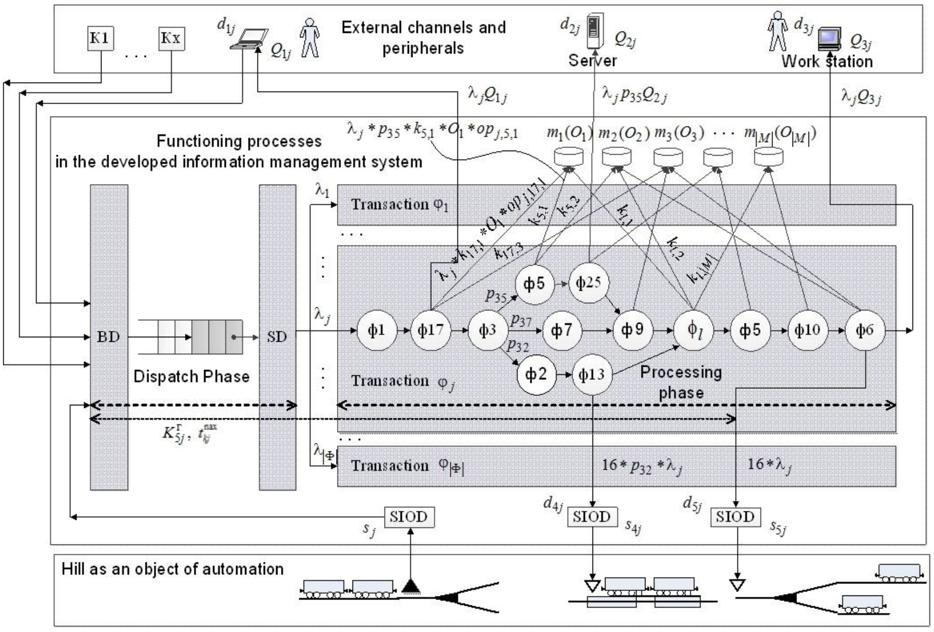

Figure 1: φ-transaction in automation systems for sorting stations

cannot see the intermediate state of the φ-transaction B until it is completed.

Durability Upon completion of the φ-transaction, its result should be recorded on an ongoing basis. This means that if the microcontroller or the control computer fails, the state should be resumed after their restart.

From the known definitions, the concept of φ-transaction differs by the extended description necessary for calculating the characteristics of an ACS at the early stages of its design, which includes for the set of processed by the system of φ-transactions:

- description of the main arrays of the database M ($\forall m_i \in M \ \{O_i\}$, where, O_i $i \in \overline{1,|M|}$ is the size of the array or the amount of memory in bytes), that used when performing φ-transactions;

- description of the set of events E that require the response of the control system; this is start-up φ-transactions ($\forall \varphi_j \in \Phi \ \{s_j, \lambda_j\}$, where $s_j \in S_o$ is the set of input signals and messages associated with the corresponding events E, λ_j is the average intensity of the start of the j-th φ-transaction);

- description of a set of conditions and requirements for successful completion of transactions $\forall \varphi_j \in \Phi \ \forall d_k \in D \ \{op_{kj}, Q_{kj}, K^{\text{r}}_{kj}, t^{\text{rp}}_{kj}\}$, where d_{kj} is the k-th information transceiver from the j-th φ-transaction, $op_{kj} = \{$ a message $msg_{kj} \in Msg^{out}$ or control action as a signal $s_{kj} \in S^{out}$ ...}, Q_{kj} - the amount of information in bytes that is transmitted to the k-th transceiver from the j-th φ-transaction, K^{r}_{kj} is the availability factor and the maximum execution time of the φ-transaction j for the k-th transceiver t^{nax}_{kj} ;

- description of each FPB that is part of the system $\forall \phi_l \in \text{FPB} \ \forall m_i \in M \ \forall com_n \in \square \ \{k_{i,l}, op_{i,l}, K_{nl}\}$, where $\text{FPB} = \{\phi_l \,|\, l \in \overline{1,|\text{FPB}|}\}$ the set of FPB systems, $M = \{m_i \,|\, i \in \overline{1,|M|}\}$ - arrays of the database,

$\square = \{com_n \mid n = \overline{1,|\square|}\}$ - a set types of commands in a mix of commands of the designed system, $0 \le k_{i,l} \le 1$ - coefficient array utilization (what part of the array it is used for one implementation of the FPB l); $op_{j,l,i} = \{\mathrm{rd,wt}\}$ - operations read and write with memory i when executing the l-th FPB in the j-th φ-transaction, K_{nl} - number of commands of type n in the l-th FPB.

A general description of the designed system with the help of φ-transactions is shown in Fig. 1. In the figure the following notation is used: buffering disciplines (BD), service disciplines (SD), device input / output signals (SIOD).

In the FPB set that is included in the φ-transaction, vertices are singled out in which multiple branching of the data processing process is possible (for example, $\phi 3$ has three options for continuing calculations). In this case, the probability of choosing a direct branching is given, the sum of which is 1 (in the example $p_{35} + p_{37} + p_{32} = 1$).

All described characteristics of φ-transactions are used in the design process of automation systems for

Table 1. Description of functional program blocks (FPB)

Options	**Functional program blocks**					
	ϕ_1	ϕ_2	...	ϕ_l	...	$\phi_{\lvert FPB \rvert}$
Mix command of system	**Number of operations in FPB, thousand commands**					
com_1	K_{11}	K_{12}		K_{1l}		$K_{1\lvert FPB \rvert}$
com_2	K_{21}	K_{22}		K_{2l}		$K_{2\lvert FPB \rvert}$
. . .						
com_n	K_{n1}	K_{n2}		K_{nl}		$K_{n\lvert FPB \rvert}$
. . .						
$com_{\lvert \square \rvert}$	$K_{\lvert \square \rvert 1}$	$K_{\lvert \square \rvert 2}$		$K_{\lvert \square \rvert l}$		$K_{\lvert \square \rvert \lvert FPB \rvert}$
Arrays and their volumes	**Array utilization and type of operation**					
$m_1(O_1)$	$k_{1,1}$, $op_{1,1}$	$k_{1,2}$, $op_{1,2}$		$k_{1,l}$, $op_{1,l}$		$k_{1,\lvert FPB \rvert}$, $op_{1,\lvert FPB \rvert}$
$m_2(O_2)$	$k_{2,1}$, $op_{2,1}$	$k_{2,2}$, $op_{2,2}$		$k_{2,l}$, $op_{2,l}$		$k_{2,\lvert FPB \rvert}$, $op_{2,\lvert FPB \rvert}$
. . .						
$m_i(O_i)$	$k_{i,1}$, $op_{i,1}$	$k_{i,2}$, $op_{i,2}$		$k_{i,l}$, $op_{i,l}$		$k_{i,\lvert FPB \rvert}$, $op_{i,\lvert FPB \rvert}$
. . .						
$m_{\lvert M \rvert}(O_{\lvert M \rvert})$	$k_{\lvert M \rvert,1}$, $op_{\lvert M \rvert,1}$	$k_{\lvert M \rvert,2}$, $op_{\lvert M \rvert,2}$		$k_{\lvert M \rvert,l}$, $op_{\lvert M \rvert,l}$		$k_{\lvert M \rvert,\lvert FPB \rvert}$, $op_{\lvert M \rvert,\lvert FPB \rvert}$

Table 2. Arithmetic operations time, ns

Argument type	**int**				**double**		
Operation	^	+	*	/	+	*	/
Pentium 4 2,5 GHz	0,78	0,86	5,6	19	2	2,8	15
CoreDuo 1,8 GHz	1,1	1,1	2,2	4,2	1,7	2,7	18
Pentium M 1,8 GHz	1,1	1,1	2,2	8,7	1,7	2,7	17
Athlon 64X2 2,5 GHz	1,4	1,4	1,6	17	1,8	1,8	8,3

Table 3. The number of cycles required to execute instructions on Intel P4 +

Instruction	XOR	ADD	MUL	DIV	FADD	FMUL	FDIV (double)
Processing delay	0,5-1	0,5-1	3-18	22-70	5-6	7-8	38-40
Time of processing	0,5	0,5	1-5	23-30	1	2	38-40

Table 4. Characteristics of typical algorithms

Class of algorithms (tasks)	Input parameters of algorithms	The resulting characteristics		
		Internal connectivity	**The number of middle operations**	**Specific volume**
Initial processing of scale measurements	*n* - number of measurements, *m* - size of the calibration table	$n+m$	nm	m
Statistical analysis	*n* - number of measurements, *m* - number of factors, number of coefficients	nm	nm^2	n
Optimization (linear prog., Etc.)	*m* - number of variables, *n* - number of restrictions	$n(m+n)$	$n^2(m+n)$	n
Calculation of technical and economic characteristics	*n* - number of data, *m* - the number of constants of the formula	$n+m$	nm	m

sorting stations in accordance with the proposed approach of CoDeCS [1]. In fact, the set of φ-transactions (Φ) is the M-model of the system [6].

IV. METHODS FOR CALCULATING THE CHARACTERISTICS OF REAL-TIME COMPUTER SYSTEMS DURING THE CONCEPTUAL DESIGN PROCESS

The initial data for calculating the characteristics of the designed control system are:

- description of functional program blocks, which includes a list of FPB, the arrays used by them and the coefficients of their use during each FPB, the number of operations of the selected types in each FPB (Table 1);
- description of φ-transactions in the form of oriented vertex and edge-weighted graphs with a description of their parameters (Fig. 1);
- description of the pre-selected type of microprocessor and the execution times of the operations of the types mentioned above (examples of microprocessor descriptions are in Tables 2, 3 [10]).

All initial data are presented in tabular form for further processing.

It should be noted that the FPB are either programs (if this block is taken from the developer's program library), or algorithms (for known control algorithms), or functions (for new tasks). This determines the accuracy of the estimate of K_{nl} and $k_{i,l}, op_{i,l}$.

To select a set of commands □ and get K_{nl}, you can use the "POET" technique [7]. For new functions, there are characteristics of typical algorithms that are recommended for the IMS (information management system) design stages (Table 4) [4].

To calculate the speed of computing systems in order to compare and select them, a large number of various models, methods and benchmarks are used (benchmark - Linpack, SPEC, SPECfp95, SPECweb97, TPC, WinMark, Winstone 97, etc.), but most of them focused on already developed systems and they are almost impossible to use in the early stages of design [4], [5], [10].

The developed methodology proposes two methods for evaluating the performance of microprocessors — $\tilde{W}_i$ the average performance of the i-th model for a mix of tasks of the system and $\tilde{W}_i^{\varphi}$ the average performance of the I model for a mix of φ-transactions of the control system that is designed.

For the automated control systems of the hump yard, the author has already identified frequency mix of commands, first described in [7]. To take into account new algorithms and control programs, the necessary adjustment of the frequency characteristics of the mix. It is recommended to use the command sets from Tables 2, 3.

The performance on the frequency mix of commands when designing a system for a pre-selected I of type of microprocessor (MCP) is calculated as follows:

$$\tilde{W}_i = 1 / \sum_{n=1}^{||} \xi_n \tau_{ni}, (1)$$

where ξ_n is the frequency or probability of command occurrence in the process of system operation; τ_{ni} is the average execute time the n command on the microprocessor of the i type; the unit of measurement of performance is the number of operations on the mix of system tasks per second – op/s .

The value of ξ_n is calculated as

$$\xi_n = K_n / \sum_{h=1}^{||} K_h . (2)$$

Here K_h is the number of h-type commands that are processed by the system in the process of performing φ-transactions.

K_h is calculated from formula (3) taking into account the data K_{nl} for each FPB from Table 1 and the structure of all φ-transactions processed by the system, taking into account the intensities of their launch, described in Fig. 1.

$$\forall h \in || \quad K_h = \sum_{\forall \varphi_j \in \Phi} \lambda_j * \sum_{\forall \phi_l \in \varphi_j} k_{lj} * K_{hl} . (3)$$

In formula (3), $0 < k_{lj} \le 1$ is the coefficient that takes into account the decrease in the number of operations in the φ-transaction j in the FPB l due to the previous ramifications and cycles (in the weighted graph). This coefficient is calculated in the process of transforming the structure and calculating the characteristics of the φ-transactions.

The obtained average performance of the i-th model on a mix of tasks of the future system $\tilde{W}_i$ is used in the proposed method of CoDeCS [1] to compare different types of MCPs in their choice, as well as for resource-saving methods for searching rational structures of decentralized control systems for sorting hill [6,7].

For service-oriented real-time systems that have a limit on the reaction time or the maintenance time of each φ-transaction, it is proposed to calculate the average performance of the i-th model on a mix of φ-transactions, or $\tilde{W}_i^{\varphi}$, which is measured in the number of weighted φ-transactions per second ("$\varphi - \text{tps}$" - φ-transaction per second) or per minute - "$\varphi - \text{tpm}$". To do this, we use the expression (4).

$$\tilde{W}_i^{\varphi} = \sum_{j=1}^{|\Phi|} \lambda_j / \sum_{j=1}^{|\Phi|} \lambda_j T_{ji} . (4)$$

In this formula, since the condition for the selected i-th type of the MCP (or clock frequency) must be met

$$\forall j \in |\Phi| \quad T_{ji} \le t_j^{\max}, (5)$$

then according to (4) it is possible to make a choice for a project of this type of MCP minimum performance (and cost) at which (.5) is performed. The T_{ji} is the time of the implementation of this j-th φ-transaction on i-th MCP.

The developer's knowledge of the values of $\forall j \in |\Phi| \ T_{ji}$ in the process of designing a real-time control system allows for the selected i-th type of microprocessor to estimate what part of the processor load is provided by providing the j-th service (j-th φ-transaction), formula (6).

$$\rho_{ji} = \lambda_j T_{ji} \ \text{and for} \ \forall j \in |\Phi| \quad \rho_{ji} \le 0{,}6. (6)$$

Carrying out the distribution of processed φ-transactions on the subsystems obtained as a result of structural optimization [1]. It is necessary to ensure that the total service load of each subsystem in the project $f_p \in \mathrm{F}$ does not exceed the permissible limit (for processors it is 0.6), i.e.

$$\forall f_{pi} \in \mathrm{F}_i \quad R_{pi} \le 0{,}6. (7)$$

Sometimes the lower limit of loading (8) is also established in order to efficiently use computational resources and reduce the total cost of IMS.

$$\forall f_{pi} \in \mathrm{F}_i \quad 0{,}1 \le R_{pi} \le 0{,}6. (8)$$

The construction of φ-transactions allows you to evaluate some of other characteristics that are very important for the design of the ASC, in particular, those related to the information flows in the future system. In the ASC we will distinguish three types of information flows: message flows (msg), control action (signals , s_k) flows, and reading (rd) and writing (wt) flows from / to the arrays of the future database.

The information flow of messages $\pi_{jd_k}^{msg}$, which is formed by each φ-transaction j, and transmitted to the peripheral device d_k:

$$\pi_{jd_k}^{msg} = \sum_{\forall \phi_l \in \varphi_j \wedge \exists d_k} \lambda_j \hat{p}_{jl} Q_{ld_k j} . (9)$$

The total flow of messages to the transceiver d_k during operation of the system will be:

$$\Pi_{d_k}^{msg} = \sum_{j=1}^{|\Phi|} \pi_{jd_k}^{msg} . (10)$$

Flows of control actions, or s_k signals are output through the SIOD on the devices of the control object. For one j-transaction and when encoding one action with 2 bytes (16 bits) they will be:

$$\pi_{s_{kj}} = \sum_{\forall \phi_l \in \varphi_j \wedge \exists s_k} 16 \lambda_j \hat{p}_{jl} . (11)$$

In the system as a whole, the signal flow s_k will be:

$$\Pi_{s_k} = \sum_{j=1}^{|\Phi|} \pi_{s_{kj}} . (12)$$

Information flows associated with reading / writing data from / to array i, when performing the j-th transaction, can be calculated as follows:

$$\pi_{ji}^{\text{rd/wt}} = \sum_{\forall \phi_l \in \varphi_j} \lambda_j \hat{p}_{jl} k_{j,l,i} O_i op_{j,l,i} . (13)$$

The total information flow to the array i in the process of the system can be calculated by the formula:

$$\Pi_i^{\text{rd/wt}} = \sum_{j=1}^{|\Phi|} \pi_{ji}^{\text{rd/wt}} . (14)$$

The total information flow to all arrays or the flow to the central database of the ASC will be:

$$\Pi_M^{\text{rd/wt}} = \sum_{i=1}^{|M|} \Pi_i^{\text{rd/wt}} . (15)$$

In addition, φ-transitions allow determining the priorities for processing applications, the average waiting time for applications in the queue, its length and other parameters of the organization of the system's functioning.

V. CONCLUSIONS

The model of the φ-transactions proposed in the article and the methods of approximate calculation of the temporal and informational characteristics of distributed real-time computer systems at the stages of conceptual design differ from the known approaches. They are simple and can be implemented on smartphones and tablets in a spreadsheet environment. The proposed models were obtained on the basis of exponential flows of applications and the disciplines of their maintenance, which makes it possible to obtain overestimated values of system characteristics, which is especially important for real-time systems. The considered models are part of the CoDeCS framework, covering the whole complex of tasks of conceptual design of distributed automated control systems [1]. Almost all tasks are solved in a spreadsheet environment or using mobile applications. The developed framework CoDeCS is constantly being improved with the development of information technologies and systems. It has found application in the creation of a number of automation systems for control and management for the railways of Russia and Ukraine.

REFERENCES

[1] A. Kosolapov, and D. Loboda D., "Framework Conceptual Design Of Complex Real-Time Management System (Codecs)", European Journal Of Advances In Engineering And Technology, 5(8), pp. 559-566, 2018.

[2] A. Kosolapov, and D. Loboda, "Semiotic- Agent-Ontological Approach To Design Intellectual Transport Systems", American Journal Of Engineering Research (Ajer), Volume-7, Issue-7, pp.205-209, 2018.

[3] C. Nagel, B. Evjen, J. Glynn, M. Skinner, and K. Watson, Professional C# 2005 with .NET 3.0, 1416 p., Wrox, 2005.

[4] V. Valkov, Microelectronic control computing systems. System design and engineering, Mechanical Engineering. Leningrad Department, 224 p., 1990.

[5] A. Kosolapov, and M. Kosolapova, "To the question of the structural design of automated systems", Mathematical Modevian, Science Journal, Dniprodzerzhinsk, DDTU, № № 2 (5), pp. 124-128, 2000.

[6] A. Kosolapov, E. Shafit, and L. Safris, "Some characteristics of microprocessors and micro-computers and technological algorithms in control systems for composition dissolution on a hill", Trudy. Interuniversity thematic collection, Rostov-on-Don, RIZHT, No. 168, pp. 20-23, 1982.

[7] A. Kosolapov, "Development and application of mathematical modeling methods in the analysis and design of microprocessor control systems for technological processes of dissolution of compositions on hills", M. Rus. thesis,. VNIIZhT (Central Research Institute of Ministry of Railways), Russia, Febr., 1984.

[8] S. Mayorov, G. Novikov, and T. Alive, Fundamentals of the Theory of Computing Systems, Book for Universities, High School, 408 p., 1978.

[9] P. Trifonov, Computer science. Construction and analysis of algorithms, Book for Universities, St. Petersburg, Peter, 95 p., 2007.

[10] B. Tsilker, and S. Orlov Computer Organization and Systems, Book for High Schools, St. Petersburg, Peter, 654 p., 2006.

[11] P. Heidelberger, and S. S. Lavenberg, "Computer Performance Evaluation Methodology, IEEE Ttransactions Computers, Vol. c-33, No. 12, pp. 1195-1219, 1984.

[12] D. J. Lilja, Measuring computer performance: a practitioner’s guide, Cambridge University Press, 15 p. 2000.

[13] M. Zelkowitz1,. and V. Basili1, "Measuring Productivity on High Performance Computers", Presented at International Symposium on Software Metrics, Como, Italy, September. p. 113-123, 2005.

[14] V.A Vtorin,. Automated process control systems. Basics of PCS, St. Petersburg, St. Petersburg State Forestry Academy named after S. M. Kirov. 182 p., 2006.

[15] S. R. Alford, Computer Systems Engineering Management. CRC Press, 372 p., 2017

[16] B.S. Blanchard, and J. E Blyler, System Engineering Management, Fifth Edition, John Wiley & Sons, 568 p. 2016.

[17] Anand Dohare, B.Mallikarjuna, Tulika, D. Arun Kumar Reddy, Mohd.Shahjad, and D. Mallikarjuna Reddy, "Avoid Delay of Packets and Improve the Behavior of Switch/Router under Self-Similar-Type Variable Input Traffic". International Journal of Computer Trends and Technology (IJCTT), Volume 67, Issue 6, pp. 111-116, 2019.

[18] Raja Muhammad Ubaid Ullah#, Kevan A. Buckley,Mary Garvey, and Jun Li "Cloud Computing Adoption in Enterprise: Challenges and Benefits", International Journal of Computer Trends and Technology (IJCTT), Volume 67, Issue 6, pp. 93-104, 2019.

[19] A. Kosolapov, "Reliability of Fuzzy System Evaluation Methodology Using Different Types of Vague Sets", Science and progress of transport, Dnepropetrovsk National University of Railway Transport, 2 (44), pp. 17-27, 2013