EFFECTIVE CONSTRUCTIVE MODELS OF IMPLICIT SELECTION IN BUSINESS PROCESSES

Nataliya Golyan, Vera Golyan, Olga Kalynychenko

***Abstract**: Wide introduction of the process-oriented systems for today is connected with the fact that they allow to formalize knowledge about business processes at the enterprises companies, organizations. At the same time, construction of business processes' formal models require a lot of time and material costs, and it is the subject to influence of the human factor because it is often carry out with participation of experts and executors the purposes of which can be not coincided with the purposes of modeled business processes and enterprise overall aims. The given contradiction can lead to a divergence between real business process and its developed model. It can lead to construction of inadequate models of business processes.*

One of the important approaches to the decision of the given problem is realized on the basis of methodology of business processes intelligent analysis. The intelligent analysis is directed on reception of really carried out business processes' models on the basis of research of events registration magazines of such processes.

***Keywords**: business processes, Petri net, implicit selection, logic net, predicate.*

***ACM Classification Keywords**: C.4 Performance of systems - Modeling techniques, J.1 Administrative data processing - Business*

Introduction

Nowadays we can see a transformation to the controlling process from traditional functional, this requires formalization of existing business processes for development of their hierarchic structure using typical fragments of a processes. That is why structure and characteristics refining of business processes is a necessary condition for investigation and development of intellectual data analysis methods [Agrawal, 1998].

The purpose of data analysis is to extract information from the logs of the business process in order to formalize the actual behaviour of BP. Such an analysis is particularly important when logged occurring sequence of events, but the process is partially or completely formalized – i.e. executors can make a decision on how to continue the process on the basis of existing local information and data of the business process rules. It is also important to note that the measures are taken on the basis of local information and personal knowledge of the decision may lead to deviations from the normal operation of the business process [Aalst, 2003].

Currently, a number of efficient algorithms for events analysis and logs basing on business process models were build [Medeiros, 2003], [Weijters, 2004]. But at the same time present studies can’t allow to solve a serious problem in the area of intelligent analysis of the processes - the problem of implicit selection identifying in structure of business processes that require, first of all, the construction of formal models of such structures.

1 Set of a problem

In a real operating business processes we can distinguish two types of selection procedures, sequences that represent different types of dependencies: explicit and implicit.

Explicit dependences [Desel, 1995] represent the direct causal connection between the procedures. In the event log of the business process such procedures usually reflected in pairs.

Example: To start the procedure P2 is necessary to complete the procedure P1.

Implicit dependences represent the indirect cause-effect relationship between the procedures. For these relationships the relationship between P1 and P2 directly from the event log is not visible.

For example, procedure P1 and P2 can be linked through a chain of procedures <P3, P5>.

This determines the importance of formalizing the implicit selection of typical designs.

Therefore, the challenge is to get a formal algebraic structures ща implicit logic model selection based on analysis of the main features of standard procedures, sequences with an implicit choice in business processes, as well as allocation failures such fragments existing algorithms.

2 Constructs models of an implicit choice

Formally, we define the explicit and implicit dependencies between procedures of business process by means of the algebra of finite predicates.

We introduce the variables x1, x2, ..., xn, indicating the status of procedures P1, P2, ..., Pn. These variables are defined on some finite set of possible values of state procedures. For example, $x1 \in \{a, b, c\}$, where x1=a means “procedure P1 is not done”, x1 = b – “procedure P1 is executing”, x1 = c – “P1 procedure done”.

We also introduce predicates L1, L2,…, Lk, denoting pairwise links (if they exist) between the procedures P1, P2, ..., Pn. For the above example the relationship between the procedures P1 and P2 will be described by predicate L1 (x1, x2).

Constructions, which implement implicit choice, are characterized by the contradiction between the choice of several alternatives and the need to synchronize selected action with already running in a business process. In other words, the situation is characterized by a combination of an implicit choice of designs and choice of timing, as shown in Fig. 2.1.

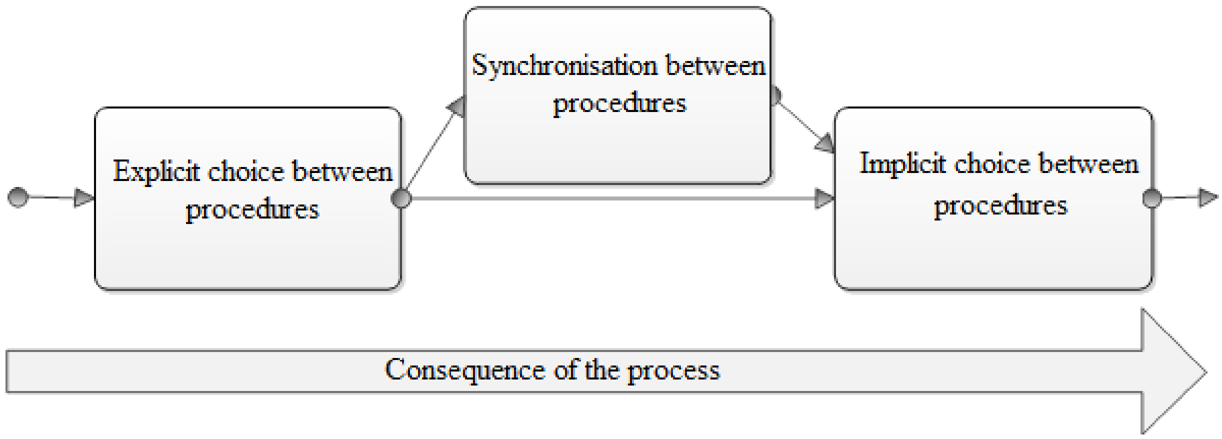

Fig . 2.1 The situation of an implicit choice between the procedures in business process

As can be seen from Fig. 2.1, the implicit choice depends on the choice of various procedures at the earlier stages of the process and is carried out after the results of the previous synchronization with the current selection procedures. This synchronization is necessary in order to have all the input conditions for the final procedures, between which there is an implicit choice in the course of a business process.

The problem of identifying structures of implicit choice in the tasks of business processes' intelligent analysis defined by the fact that the existing algorithms, the mathematical basis of which are Petri's nets, particularly their extension WF-nets [Li, 2003], usually can't process such constructions.

Consider a typical situation in more detail and implicitly selection problems that arise in identifying following situations.

In Fig. 2.2 presents the situation, according to which the final result of the current fragment of the business process depends on an implicit choice between procedures P4 and P5.

The implementation of this fragment begins in the case, the initial condition C1 is done. Then an explicit choice is executing between the procedures P1 and P2. The results of the procedures P1 or P2 must be synchronized with the result of the procedure P3, after which the choice between procedures P4 and P5 is done. Since the choice of one of the procedures determines by the choice of P1 or P2 at an earlier stage of implementation, we find that the choice between P4 and P5 is implicit.

The problem of recognition of this fragment is as follows. Existing algorithms usually do not recognize the connection between the procedures P1 and P4, P2 and P5. Therefore, the resulting model does not appear the situation is an implicit choice between P4 and P5, determined by the choice between P1 and P2. In other words, without taking into account the links between the procedures P1 and P4, P2 and P5 existing approaches to the analysis display generated model, a set of four sequences of the procedures of the form.

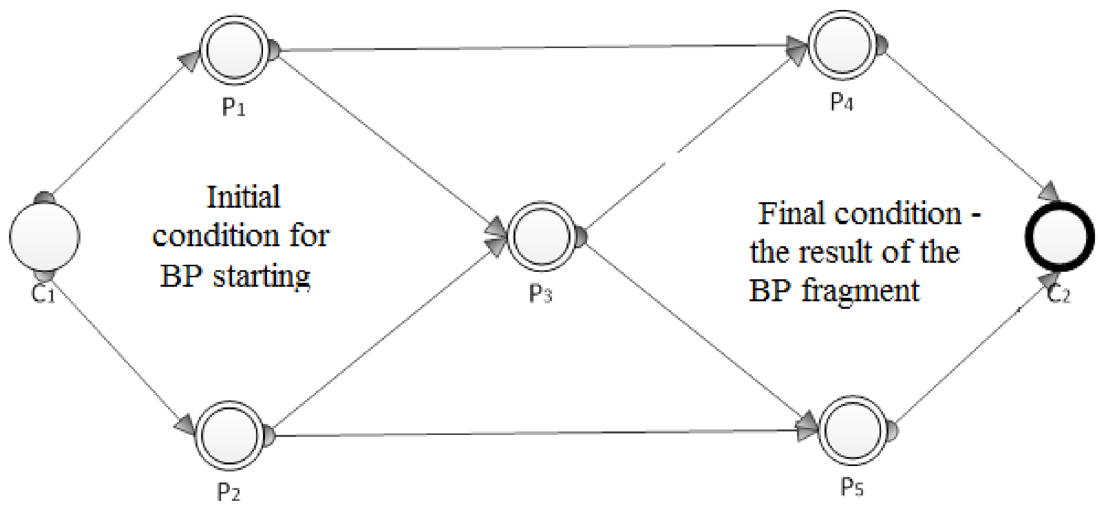

Fig. 2.2 – The situation of an implicit choice between P4 and P5

The problem of recognition of this fragment is as follows:

{ <P1, P3, P4, >, <P2, P3, P5>, <P1, P3, P5, >, <P2, P3, P4>}. (2.1)

At the same time in the real process, the links between P1 and P4, P2 and P5 valid only sequences

{ <P1, P3, P4, >, <P2, P3, P5>}. (2.2)

Way of solving this problem is as follows. The developed method for detecting design implicit choice must fix each previously made a free choice in the future track of the relationship between each pair of treatments.

Formally, this can be done with the mathematical apparatus of the algebra of finite predicates [Bondarenko, 2004] as a logical network [Shabanov-Kushnarenko, 2004], [Leschinski, 2004], ie solution to the problem of determining the optimal sequence of the procedures is reduced to solving a system of predicate equations L1, L2,…, Lk, set above.

The work of this logical network would be in a cyclic sets test of all of the modeled business process links between the procedures, which at each step of the calculations will be expressed in the generation of a set of procedures that must be done to continue the business process.

We introduce the variables x1, x2,…, x5, indicating the status of procedures P1,…, P5. These variables can take values from {0, 1}, respectively, which means performance or failure of procedures. To construct a logical network finds a system of binary predicates L1, L2,…, Lk, which describes the execution logic of the piece of business processes, shown in Fig. 2.2. For this purpose, we introduce an intermediate variable t, which contains information about explicit and implicit election procedures. The variable t can take values from {0, 1, 2, 3} denoting respectively the performance of procedures P1, P2, P1 and P3, P2 and P3.

The system predicate that describes the logical network model of the first situation:

$$\begin{cases} L_1(x_1, x_3), \\ L_2(x_1, t), \\ L_3(x_2, x_3), \\ L_4(x_2, t), \\ L_5(x_3, t), \\ L_6(t, x_4), \\ L_7(t, x_5). \end{cases} \quad (2.3)$$

The corresponding logical network is represented in fig. 2.3:

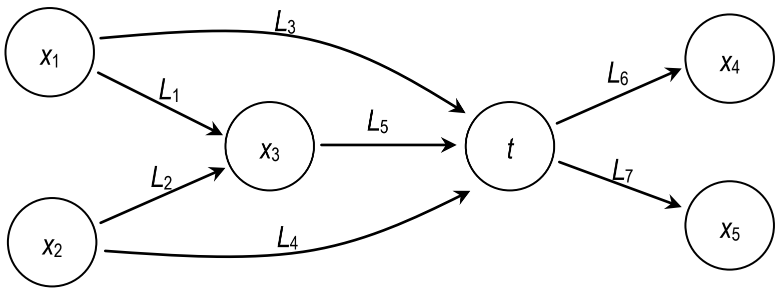

Fig. 2.3 – The logical network of the 1st typical situation

The initial logical network nodes are the variables x1, x2, final - variables x4, x5. At the beginning of the logical network of all its variables, the modeling procedure of a business process, are set to 0 - no procedure is performed. In the course of a logical network, i.e. computation corresponding to this network of binary predicates, some or all of the variables x1, x2,…, x5 logical network takes the values of 1 - all procedures are performed. The variable t takes a value corresponding to the realized in the business process execution sequence of procedures.

Consider the example of the logical network built on Measure:

0 cycle (beginning of the network): x1=0,…, x5=0, t is not defined.

1 cycle: x1=1, t=2, other nodes unchanged.

2 cycle, 1 semi cycle (network status): x3=1.

3 bar (end of the network): x4=1

Obviously, if the initial action in the network will be carrying out the procedure P2, then at the end of the network will result x5 = 1.

The second typical situation is an implicit choice of the sequence of procedures is presented in Fig. 2.4.

In this situation there are two implicit choice - between the procedures P4 and P5, as well as for P3 and P5.

Sequence of this fragment is as follows. The initial condition C1 is an input for a single procedure P1. As a result, its performance can be parallel (in any order), follow the procedures P2 and P3 respectively.

Further, there are several implementation options.

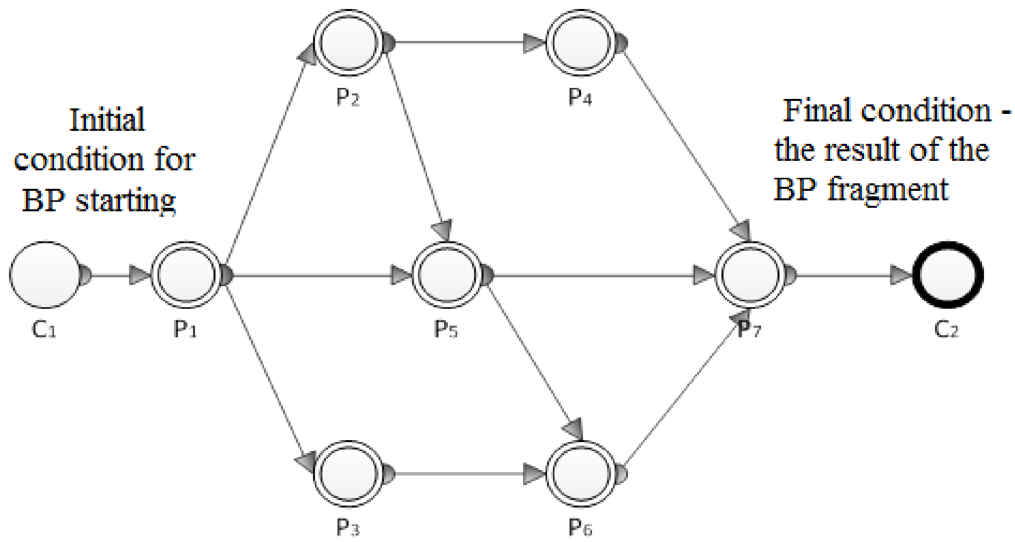

Fig. 2.4 – The situation of an implicit choice between the procedures P4 and P5 , P3 and P5

Option 1. In that case, the procedure P2 is executed before the procedure P3 then the procedure P3, P4 and P5 can be performed in any order, independently of each other. Then you can choose from two parallel branches - a procedure performed P6, or both of the procedures for P6 and P7.

Option 2. If the procedure P3 is executed before the procedure P2, then it becomes a further order of the hard-coded: P2 should be executed after the P4 and P5, P is performed after the P6, and then - P7.

Thus, the possible sets of sequences of the procedures can be written as the following tuples

{ <P1, P2, P5, P6, P7 >, <P1, P2, P3, P4, P6, P7 >, <P1, P3, P2, P6, P4, P7 >,

<P1, P2, P4, P3, P6, P7 >, <P1, P3, P6, P2, P4, P7 >} (2.3)

The problem of identifying the fragment is as follows. Existing algorithms for data analysis does not recognize the connection between the procedures P1 and P5, P5 and P7, which can lead to delays and deadlocks in process models, in particular, if a pair of parallel processes P4 and P5 will be performed only one procedure P5.

The direction of solving this problem is associated with the identification of the cause - effect relationship between the procedures P1 and P5, P5 and P7 on the basis of the analysis of event log.

We construct algebraic-logical model of the second typical situation of an implicit choice of the sequence of procedures similar to the first type situation. We introduce the variables x1, x2,…, x7, indicating the status of procedures P1,…, P7. These variables can take values from {0, 1}, respectively, which means performance or failure of procedures.

To construct a logical network determine a system of binary predicates L1, L2,…, Lk, which describes the execution logic of the piece of business processes, shown in Fig. 2.4. For this purpose, we introduce an intermediate variable t, which contains information about the sequence of explicit and implicit election procedures. The variable t can take values from {0, 1} by the rule: t = 0, if the procedure P2 is executed before the procedure P3; t = 1, on other way the procedure P2 is executed after the procedure P3.

System of binary predicate describes the logical network of the second type of implicit choice situation, consists of 12 predicates L1, L2,…, L12. The corresponding logical network shown in Fig. 2.5

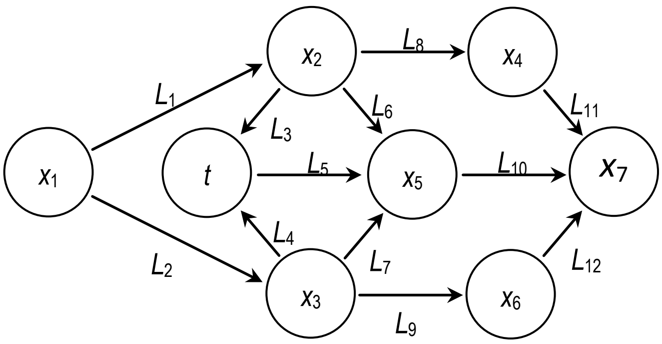

Fig. 2.5 – The logical network of the 2nd typical situation

Consider the example of the logical network built on following cycles:

0 cycle (beginning of the network): x1 = 1, all other nodes unchanged.

1 cycle: t=1, x3=1, all other nodes unchanged.

2 cycles: x2=1, all other nodes unchanged.

3 cycles: x4=1, x5=1, all other nodes unchanged.

4 cycles: x6=1, all other nodes unchanged.

5 cycles (end of the network): x7=1.

In this example, the logic network includes the issue of the procedures appropriate to the 2nd version of the procedure P2 executed after the procedure P3. Cycle 1 is calculated similarly.

Business process model in the form of a logical network, as shown above, generates at any given time a number of procedures that must be done to continue the business process. When performing any procedure, the situation changes once that immediately displays the change in the state of the logical network and thus change the set of procedures that must be done to continue the business process.

The third typical situation an implicit choice of the sequence of procedures is presented in Fig. 2.6.

In this situation, there is an implicit choice between the procedures P3 and P5. Sequence of this fragment is as follows. After the procedure P1 independently can be performed procedures P2 and P3. The final procedure for this fragment, P5 can be executed after the cycle P3 → P4 and the procedure P2.

As we can see, the possible sets of sequences of the procedures can be represented by the following tuples:

{ < P1, P2, P5 >, <P1, P2, P3, P4, P5>,

<P1, P3, P2, P4, P5>, <P1, P3, P4, …, P3, P4, P2, P5>}. (2.5)

The solution of the problem considered in this case involves finding the relationship between the procedures P1 and P5.

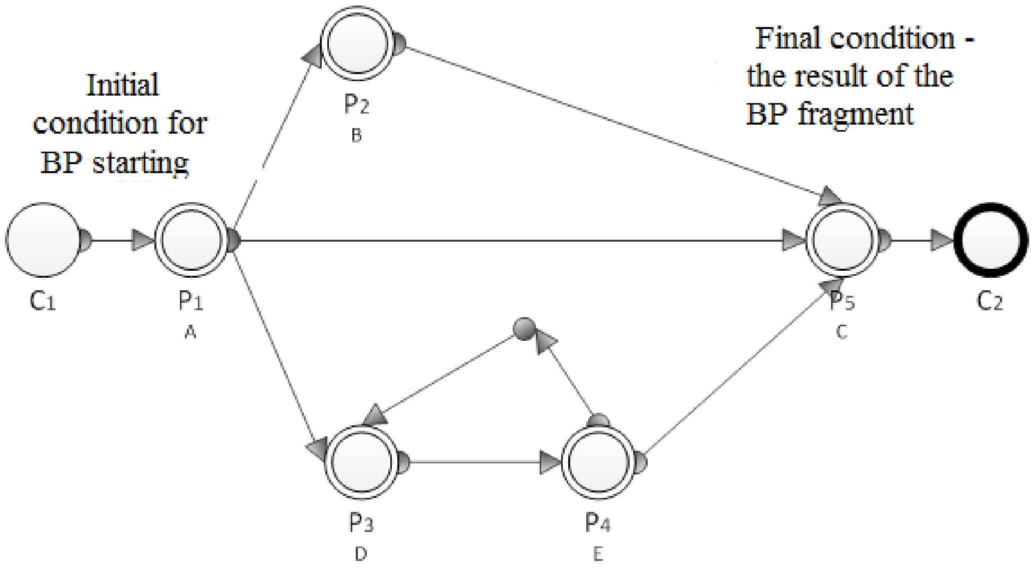

Fig. 2.6 – The situation of an implicit choice between the procedures P3 and P5

Lets construct a logical network of third typical situation an implicit choice of the sequence of procedures similar to the situation considered above. We introduce the variables x1, x2,…, x5, indicating the status of procedures P1,…, P5. We find the system binary predicate L1, L2,…, Lk, which describes the execution logic of the movie business processes depicted in Fig. 2.6. For this purpose, we introduce an intermediate variable t, which contains information about implicit choice between procedures P3 and P5. The variable t can take values from {0, 1} by the rule: t = 0, if the procedure is executed before the procedure P3 P5; t = 1, if the procedure is executed after the procedure P3 P5.

System of binary predicate that describes the logical network of the second type of implicit choice situation, consists of seven predicates L1, L2, ..., L7. The corresponding logical network shown in Fig. 2.7:

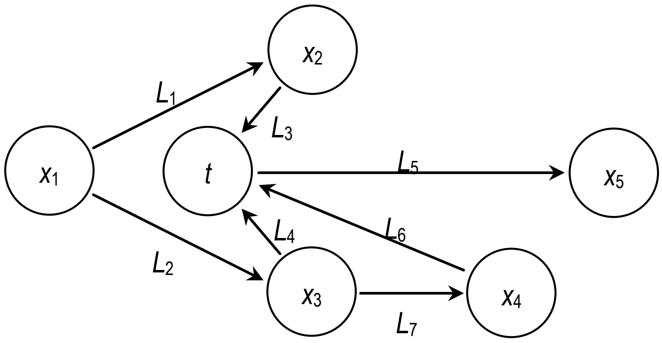

Fig. 2.7 – Logical network of third typical situation

Consider the example of the logical network built on following cycles:

0 cycle (beginning of the network): x1=1, all other nodes unchanged

1 cycle: x2=1, all other nodes unchanged.

2 cycles: x3=1, t=0, the remaining nodes unchanged.

3 cycles: x4=1, all other nodes unchanged.

4 cycles (end of the network): x5=1.

Conclusions

Algebraic-logical models of implicit choice are logical nets have been shown as a predicates system, which depicts correlation between procedures of indirect construction. Such models allow depict implicit connections between existing procedures, which are not directly shown in an event of business process

Using the examples of built logical nets we can see, that in major amount of cycles counts only one predicate, that means that such logical nets work in sequence mode, and they have a prior to other models of same processes of vary predicates. However in shown examples were studied only small "key" fragments of existing business processes, and on practice models usually include many procedures, which can be executed at the same time, in that case usage of model with binary equations system is obvious.

The second feature of built logical net is in similar introduction of additional variable, which characterizes all deviations in work logics of a business process and models are showing the sequence of procedures execution.

Practical aspect of received results is in follows. Realization of a logical net, which realize construction of implicit choice, allows receive a log of event registration which depicts implicit relations between procedures, which gives a condition for development of detecting methods for constructions with implicit choice

Bibliography

[Agrawal, 1998] R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. In Sixth International Conference on Extending Database Technology, pages 469-483, 1998.

[Aalst, 2003] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering, 47(2):237-267, 2003.

[Medeiros, 2003] A.K.A. de Medeiros, W.M.P. van der Aalst, and A.J.M.M. Weijters. Workflow Mining: Current Status and Future Directions. In R. Meersman, Z. Tari, and D.C. Schmidt, editors, On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, volume 2888 of Lecture Notes in Computer Science, pages 389 -406. Springer-Verlag, Berlin, 2003.

[Weijters, 2004] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128-1142, 2004.

[Desel, 1995] J. Desel and J. Esparza. Free Choice Petri Nets, volume 40 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, UK, 1995.-256 p.

[Li, 2003] M. Z. J.Q. Li, Y.S. Fan. Timing constraint workflow nets for workflow analysis. IEEE Transactions on Systems, Man, and Cybernetics, part A: Systems and Humans, 33(2):179–193, March 2003.

[Bondarenko, 2004] M.F. Bondarenko, Yu. P. Shabanov-Kushnarenko About algebra of predicates // Problems of bionics, 2004. - Vol. 1. – p. 176-185.

[Shabanov-Kushnarenko, 2004] S.Yu. Shabanov-Kushnarenko, M.F. Bondarenko, V.A. Chikina, Yu. P. Shabanov-Kushnarenko Models of Languages // Problems of bionics, 2004. - Vol. 1. – p. 128-139.

[Leschinski, 2004] V.A. Leschinski, S.Yu. Shabanov-Kushnarenko, M.F. Bondarenko, Z.V. Dudar, I.A. Yefimova About Brainlike Machines // Radioeletronics and Informatics. 2004. № 2. C. 12-20.

Authors' Information

Olga Kalynychenko - PhD, associate professor of Software department, Kharkov National University of Radio Electronics Ukraine; Lenin av., 14, 61166 Kharkov, Ukraine; e-mail: okalinichenko@mail.ru

Vera Golian - PhD, Software department, Kharkov National University of Radio Electronics Ukraine; Lenin av., 14, 61166 Kharkov, Ukraine; e-mail: veragolyan@yandex.ru

Nataliya Golian – PhD student of Software department, Kharkov National University of Radio Electronics Ukraine; Lenin av., 14, 61166 Kharkov, Ukraine; e-mail: veragolyan@yandex.ru