ECONTECHMOD. AN INTERNATIONAL QUARTERLY JOURNAL – 2018, Vol. 7, No. 3, 53–58

Models of temporal dependencies for a probabilistic knowledge base

O. Chala

Kharkiv National University of Radio Electronics,
Nauka Ave, 14, Kharkov, 61166, Ukraine
oksana. chala@nure. ua

Received September 01.2018: accepted September 16.2018

Abstract. The article presents models of temporal dependences for constructing probabilistic temporal rules in the Markov Logical Networks. Such rules describe the relations between the states of a control object and taking account the possibility of integrating different approaches of management according to the paradigm of "Enterprise 2.0" knowledge sharing.

The proposed models define constraints and conditions for changing the states of a control object, which allows predicting possible variants of its behavior in relation to the current state and providing decision support based on a choice of the most likely variants.

Key words: temporal dependencies; temporal rule, knowledge base, information control system, event, attribute, event log.

INTRODUCTION

The problem of probabilistic temporal knowledge base construction becomes particularly relevant when integrating existing approaches to enterprise management based on the "Enterprise 2.0" paradigm [1]. This paradigm involves the creation of a network structure for the exchange of personal knowledge of employees of the enterprise.

Personal knowledge and experience are formalized only partially [2], therefore, in accordance with this paradigm, corporate portals and social networks are used to exchange such knowledge [3].

The rapid transfer of knowledge between employees allows, in particular, combining vertically-oriented functional and horizontal-oriented process approach [4] to manage and improve the efficiency of the enterprise.

However, the transfer of knowledge through blogs and social networks uses a very slow mechanism of socialization. This significantly impedes the support of decision-making in the operational management of the enterprise.

An alternative approach to reasonable decision-making is to predict the behavior of the management object based on the use of probabilistic temporal rules. Such rules describe the behavior of the object of management in time, that is, determine the most likely sequences of its states or actions in relation to the current situation in the enterprise.

Templates for constructing these rules are temporal dependencies that describe the typical relationships between states, events or actions of the control object. Therefore, the automated construction of such temporal dependencies is an important task.

THE ANALYSIS OF RECENT RESEARCHES AND PUBLICATIONS

Modern approaches to the automated building of knowledge bases are intended primarily for the formalization of static dependencies in large databases that are available on the Internet [5, 6]. However, when solving managerial problems, it is necessary to consider the sequence of states or actions on the control object. This means that it is necessary to take into account the temporal dependencies between these actions. Such dependencies can be obtained on the basis of analysis of the records of the behavior of the object of management. The specified records are usually presented in the form of log events [7].

At the same time, the existing approaches to the analysis of logs to support management tasks are focused primarily on the construction of graphical models of the behavior of the control object [8]. To find temporary dependencies, an additional expert analysis of such models is necessary [9]. In addition, these approaches do not take into account the probabilistic aspect of temporal dependencies.

The probabilistic representation of knowledge is based on the apparatus of Markov Logic Networks (MLNs) [10, 11].

However, this mathematical apparatus allows constructing probabilistic static dependencies, without taking into account the temporal aspect.

The general method of automated construction of the probabilistic base of temporal knowledge based on the detection of dependencies between events in the log is proposed in the works [12, 13].

However, in these papers, the typical temporal dependences, which are patterns for constructing probabilistic temporal rules, are not detailed. The task of developing models of temporal dependencies needs to be solved.

OBJECTIVES

The aim of the work is to develop models of facts that correspond to events of the control object, as well as typical temporal dependencies between these facts as templates for constructing probabilistic temporal rules in the MLNs.

EVENT PRESENTATION OF INFORMATION ABOUT THE BEHAVIOR OF THE CONTROL OBJECT

Temporal dependencies are patterns for temporal rules and are constructed from sequences of events, each of which has a timestamp. Such sequence of events is formed as a result of processing information from sensors or monitoring processes in computer systems, as well as business processes and processes of teamwork at the enterprise. The sequence of events is written to an event log.

Consider, for example, the situation with an intrusion in the computer system. . Processes of normal operation and computer intrusion have different patterns of behavior. Patterns for the normal, abnormal, and faulty behavior of a computer system can be obtained by analyzing a log containing one or more sequences of events.

Each sequence of events sets a description of the behavior of a computer system during a certain period of time.

An event according to the terminology given in work [14] contains a set of attributes and a timestamp.

For the formation of temporal dependencies on the basis of the analysis of the event log it is useful to allocate the following groups of attributes that are necessary for solving management tasks:

– identification attributes A^{Type};

– attributes of action A^{Work};

– attributes of artifacts $A^{Artifact}$;

– attributes of time A^{Time}.

The event that takes into account the attribute classes has the form:

$$e\left(A^{Type}, A^{Work}, A^{Artifact}, A^{Time}\right). \qquad (1)$$

The identification attributes contain the information about the event type and its identification number: $A^{Type} = \left\{a^{Type}, a^{ID}\right\}$.

The attributes of action contain information about the name and state of the action on the control object: $A^{Work} = \left\{a^{Work}, a^{State}\right\}$. Change the name a^{Work} or states a^{State} action causes the event to be logged.

Attributes of artifacts $A^{Artifact} = \left\{a^{Artifact,k}\right\}$, $k = \overline{1, \left|A^{Artifact}\right|}$ reflect the properties of the elementary components of a complex control object – artifacts. Examples of artifacts: executor of action; the document on which the action is performed; the department of the enterprise where the action was performed; the computer on which the action was performed, etc. The attributes of these artifacts: the title and position of the worker; the name of the device; IP address of the computer, and so on.

It should be noted that in order to generalize an event description it is possible to consider actions as a special form of artifacts, that is $A^{Work} \subseteq A^{Artifact}$ and thus combine these subsets of attributes.

Attributes of time include the time of occurrence of an event or its length: $A^{Time} = \left\{a^{TimePoint}, a^{TimeInterval}\right\}$. The granulation of these attributes is given during configuring the information system. For example, the time of occurrence of an event can be recorded with accuracy to a day, an hour, a minute or a second.

Generally, without taking into account the attribute classification given above, the event has the form: $e\left(\left\{a^k\right\}\right)$, where a^k – k-attribute of an event belonging to one of the subsets in the expression (1).

While constructing temporal dependencies and their implementation within the knowledge base, it is necessary to consider the possible formats of recording event logs. There are three formats of such logs: process; mixed; sensor log.

The log file L in the process format consists of a set of traces π_i: $L = \left\{\pi_i\right\}$. Each of these traces contains a finite sequence of events: $\pi_i = \left\langle e_{i,1}, ..., e_{i,|\pi_i|}\right\rangle$.

Depending on the tasks being solved in the enterprise and the chosen approach to enterprise management, the following information may be included in a trace:

– a sequence of actions of one process on the control object;

– a sequence of states of one computer program;

– a sequence of states of one sensor.

Information from the process log can be used to detect and add to the knowledge base temporal relationships between events for each recorded process. For example, the set of traces of one business process at an enterprise makes it possible to construct weighted temporal rules that establish a relationship between the actions of this process. The weight of these rules is a function from the probability of the appearance of individual process traces on which these dependencies are performed.

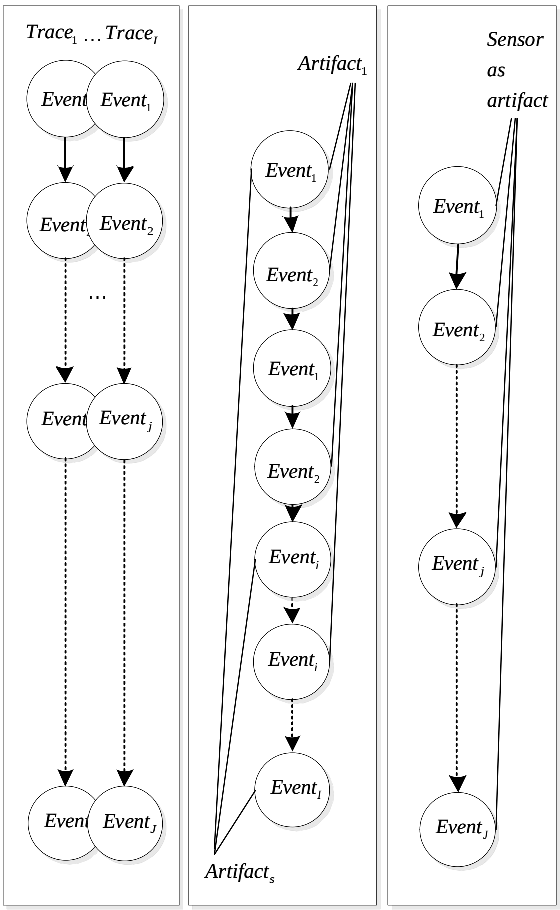
The mixed-format journal contains one single trace. This trace consists of a sequence of events that reflect: actions from different processes on the control object; a sequence of states of all processes in the operating system; states from a given set of sensors, and so on.

The mixed-format journal isn’t divided in to individual processes but contains information about the state change of all known artifacts. Therefore, this journal allows you to allocate and add to the knowledge base relations between events for a life cycle or a cycle of using artifacts. For example, for a computer system log, we can determine typical event patterns for specific IP addresses, ports, etc.

Further, the received temporal relationships make it possible to separate the normal cycles of work with the address and the computer port from abnormal and faulty ones. In a practical sense, this makes it possible to detect interference with the operation of the computer system. Similarly, a sensor that captures traffic on a highway allows allocating dependencies for the normal flow of cars and for the intense traffic, taking into account the time of day, the day of the week.

The event log of the sensor is a simplified version of the mixed log. It contains a single sequence of events for one sensor, which in this case should be considered as an artifact.

The differences in the structure of the event log data are shown in Fig. 1. The process log consists of a sequence of events. Each trace consists of a sequence of events.



a *b* *c*

Fig. 1. The differences between event logs:
f – Process log; b – Mixed log; c – Event log of the sensor

For mixed logs, it is shown that the same events can contain information about the life cycle or cycle of the use of different artifacts. The event log of the sensor contains a cycle of using only one artifact.

MODELS OF LOGICAL FACTORS AND TEMPORAL DIFFERENCES BETWEEN EVENTS

Knowledge of the behavior of the control object can be represented in the form of logical facts and temporal rules.

Logical facts describe the state of the control object at discrete moments of time in accordance with the values of the properties of the artifacts. Temporal rules determine the permissible sequence of states of the control object.

The above description of the behavior of the control object in the form of a sequence of events indicates that the state of the control object at each time point is displayed through the attributes of these events. Therefore, every logical fact ft_j can be represented as a predicate Q on the set attributes of an event. Truthfulness Y_{ft_j} such predicate is determined via the set of attribute values of an event:

$$ft_j = Q\left(\left\{a_j^k\right\}\right),$$

$$Y_{ft_j} = \left\{ \begin{aligned} & a_j^k \,\middle|\, \forall k \exists \alpha_j^k : \\ & \alpha_j^k \in \Lambda^{Type} \vee \alpha_j^k \in \Lambda^{Work} \vee \alpha_j^k \in \Lambda^{Artifact} \end{aligned} \right\}, \quad (2)$$

where $\Lambda^{Type}, \Lambda^{Work}, \Lambda^{Artifact}$ – the set of given attribute values for an event type, action, and artifact.

The logical fact (2) doesn't take into account the time of occurrence of an event. If this fact is an antecedent of the rule, then it takes the truth at the current time. The logical fact that is the consequent will be true at a preset moment in the future.

Temporal rules specify the relations between logical facts. Such rules are formed on the basis of typical temporal dependencies between log events. That is, temporal dependencies act as templates for the formation of temporal rules. Therefore, the necessary condition for building the base of temporal knowledge is the formation of temporal dependencies, which make it possible to describe the behavior of the object of management.

Generally, temporal dependencies have this form:

$$ft_j \,\mathrm{KO}\, ft_m, \quad (3)$$

where: K – quantifier for the sequence of events; O – temporal operator.

The quantifier determines the set of sequence of events on which the temporal dependence (3) will be performed. For temporal rules, it is proposed to use quantifiers of temporal logic *E (Exists)* and *A (All)*. The first quantifier sets that logical fact ft_m will be executed for at least one events sequence of the log. As it shown in Fig. 1, for the process log, the fact ft_m will be executed on at least one trace. For a mixed journal, this fact will be true for at least one cycle of the use of one artifact. For a sensor event log, the fact ft_m will be true at least once in the recorded sequence of events.

The temporal operator specifies the kind of dependencies between the logical facts ft_j and ft_m for a given level of detail.

Taking into account the above structure of the event (1) and the logical fact (2), we distinguish the following levels of detail of the temporal dependencies:

– for events in general, taking into account all attributes;

– for a subset of event attributes;

– for the values of individual attributes.

At all three levels of detail, the following dependencies can be allocated:

– a series of events;

– a couple of events, between which there are other events;

– a cyclical event repetition;

– a presence of a specific event.

The given dependencies are defined by the following set of temporal rules.

A sequential pair of events is given by the temporal operator $X(NeXt)$. Depending on the quantifiers, we obtain conditions or constraints on the truth of the logical fact ft_m in the case of the truth of the fact ft_j.

The constraint is set by a quantifier $A(All)$:

$$ft_j\ AX\ ft_m. \tag{4}$$

In accordance with the constraint (4), for all known sequence of events in the case of the truth of the fact ft_j the next discrete moment of time will be true the fact ft_m. That is, the constraint specifies the acceptable behavior options for the control object.

For a process log, this temporal restriction is set on all traces π_i. For the mixed log, the restriction is set for the artifact af_s. That is, dependence (4) must be performed for a set of events sequences $\{\varphi_{s,o}\}$, which describes all existing cycles of using this artifact. In the case of a sensor log, the constraint (4) will also be true for all possible cycles of using this sensor.

Temporal dependence, which determines the probable condition of the truth of the fact ft_m directly in the fact ft_j, is given by a quantifier $E(Exists)$:

$$ft_j\ EX\ ft_m. \tag{5}$$

This dependence will be at least executed on one of the traces for the process log, one cycle of usage for one of the artifacts for the mixed log and one cycle for the sensor log.

A pair of logical facts, between which in time may be true other logical facts, is defined by the temporal operator $F(Future)$. Similarly to expressions (4) and (5), the dependence data are both act as the constraint and the condition of the truth of the fact ft_m in the future after the truth of the fact ft_j.

The constraint is given by quantifier A and has the form:

$$ft_j\ AF\ ft_m. \tag{6}$$

The condition of the truth of the fact in the future has the form:

$$ft_j\ EF\ ft_m. \tag{7}$$

Cyclical repetition of the fact ft_j matches the consistent pairs of events that have the same value of attributes, with the exception of attributes a^{ID} and A^{Time}.

The cycle as a constraint and as a condition is given in pairs of facts ft_j as follows:

$$ft_j\ AX\ ft_j, \tag{8}$$

$$ft_j\ EX\ ft_j. \tag{9}$$

The transition from the cycle to subsequent events is determined by means of the temporal operator $U(Until)$.

The cycle as a constraint and as a condition is given in pairs of facts ft_j as follows:

$$(ft_j\ AX\ ft_j)U\ ft_m, \tag{10}$$

$$(ft_j\ EX\ ft_j)U\ ft_m. \tag{11}$$

In accordance with expressions (10) and (11), in discrete, sustained moments of time will be a true fact ft_j until a fact ft_m becomes true.

Similarly, cycles that involve the challenge of individual and possibly different procedures are set using a combination of operators F and U:

$$(ft_j\ EF\ ft_j)U\ ft_m, \tag{12}$$

$$(ft_j\ AF\ ft_j)U\ ft_m. \tag{13}$$

The availability of a specified event is determined by the existence of a corresponding logical fact:

$$A\ ft_j, \tag{14}$$

$$E\ ft_j. \tag{15}$$

Temporal dependencies at the second level of detail take into account the condition Φ^a of selection of a subset of event attributes when determining each logical fact. This condition is given as follows:

$$\Phi^a = \begin{cases} true,\ if\ a_j^k \in A'^{Type} \vee a_j^k \in A'^{Work} \vee a_j^k \in A'^{Artifact} \\ false, otherwise \end{cases}, \tag{16}$$

where: $A'^{Type}, A'^{Work}, A'^{Artifact}$ – the subset of allowable for rules of the second level of attributes.

The temporal dependencies at the third level of detail take into account the condition Φ^α of selection of a subset of admissible attribute values of an event in determining each logical fact:

$$\Phi^\alpha = \begin{cases} true,\ if\ \alpha_j^k \in \Lambda'^{Type} \vee \alpha_j^k \in \Lambda'^{Work} \vee \alpha_j^k \in \Lambda'^{Artifact} \\ false, otherwise \end{cases}, \tag{17}$$

where: $\Lambda'^{Type}, \Lambda'^{Work}, \Lambda'^{Artifact}$ – the subset of admissible attribute values for the third level rules.

A complete list of temporal dependencies is given in Table 1.

The given list of temporal dependencies and logical facts constitute a logical component of the temporal knowledge base. These dependencies in the MLNs determine the set of possible temporal rules. Such rules are formed by substituting specific values of event attributes for predicates (2).

According to the paradigm of MLNs, the probabilistic component is formed by determining the weight of the facts and rules. These weights depend on the probability of such rules emergence in the event log. For example, for a process log, the weight of the rules is determined by the probability of the appearance of traces of the process π_i [13]. For a mixed log, a rule weight depends on the likelihood of implementing cycles using the relevant artifacts.

Table 1. Temporal dependencies

Dependency type	Constraints	Conditions
Sequence of events or actions	$ft_j\ AX\ ft_m$	$ft_j\ EX\ ft_m$
A pair of events or actions between which there are other events (actions)	$ft_j\ AF\ ft_m$	$ft_j\ EF\ ft_m$
Cyclical repetition of an event or action	$ft_j\ AX\ ft_j$ $(ft_j\ AX\ ft_j)U\ ft_m$	$ft_j\ EX\ ft_j$ $(ft_j\ EX\ ft_j)U\ ft_m$
Presence of a certain event	$A\ ft_j$	$E\ ft_j$

It should also be noted that we propose to set the weight of constraints equal to ∞, since according to the rules of the MLNs the probability of executing rules with such weight is equal to 1 [13].

The weight of conditions is determined by the method presented in [11].

Thus, the given dependencies give an opportunity to implement the base of temporal knowledge.

The general sequence of creation of the knowledge base contains the following stages: definition of artifacts of the domain with the log events attributes; definition of classes of artifacts; definition of logical facts according to specific values of attributes of events; definition of temporal rules for given logical facts; determining the weight of the facts and the temporal rules.

EXPERIMENT

A fragment of the process log describing the service of an electronic device is considered. The fragment contains a sequence of events for a priori time interval. Events are organized according to their time of emergence. Attributes of events in this fragment are not given to simplify the construction of temporal rules.

The log consists of two distinct sequences of events, the first sequence being repeated three times, and the second is twice. The first sequence has the following semantics: application for repair of the device (1); disassembly of the device (2); fault diagnostics (3); reconciliation of repairs with the customer (4); purchase of a knot or a part (5); replacement of a knot or a part (6); assembly of the device (7); payment and transfer of the device to the customer. In the second sequence, an event (9) is used additionally - cleaning the device from dust.

The events sequences are presented in Table 2.

The purpose of the experiment is to test the possibility of constructing temporal rules to create a knowledge base based on the use of the obtained temporal dependencies.

Table 2. Events sequences

Sequences	Number of the repetitions
$\langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\rangle$	3
$\langle e_1, e_2, e_9, e_7, e_8\rangle$	2

In this experiment, we consider the dependencies of the level of events in general, since the input log does not contain complete information about the attributes of the events. Therefore, each fact ft_j will be considered as a predicate, the argument of which is the event e_j in general:

$$ft_j = Q(e_j). \tag{18}$$

The predicate will be true if there is an appropriate event in the input sequence. For example, both sequences contain an event e_1. Therefore $ft_1 = true$ for each sequence.

The resulting temporal rules for this journal, built on the templates presented in Table 1, are given in Table 3.

Table 3. Temporal rules

Dependency type	Constraints	Conditions
Sequence of events or actions	$ft_1\ AX\ ft_2$, $ft_7\ AX\ ft_8$	$ft_2\ EX\ ft_3$, $ft_3\ EX\ ft_4$, $ft_4\ EX\ ft_5$, $ft_5\ EX\ ft_6$, $ft_6\ EX\ ft_7$, $ft_2\ EX\ ft_9$, $ft_9\ EX\ ft_7$
A pair of events or actions between which there are other events (actions)	$ft_1\ AF\ ft_7$, $ft_1\ AF\ ft_8$, $ft_2\ AF\ ft_7$, $ft_2\ AF\ ft_8$	$ft_2\ EF\ ft_4$, $ft_2\ EF\ ft_5$, $ft_2\ EF\ ft_6$, $ft_3\ EF\ ft_5$, $ft_3\ EF\ ft_6$, $ft_3\ EF\ ft_7$, $ft_3\ EF\ ft_8$, $ft_4\ EF\ ft_6$, $ft_4\ EF\ ft_7$, $ft_4\ EF\ ft_8$, $ft_5\ EF\ ft_7$, $ft_5\ EF\ ft_8$, $ft_6\ EF\ ft_8$, $ft_9\ EF\ ft_8$
Cyclical repetition of an event or action	–	–
Presence of a certain event	$A\ ft_1$, $A\ ft_2$, $A\ ft_7$, $A\ ft_8$	$E\ ft_3$, $E\ ft_4$, $E\ ft_5$, $E\ ft_6$, $E\ ft_9$

Analysis of the resulting table allows us to draw the following conclusions.

First, temporal constraints determine the logical facts that are mandatory for all variants of the behavior of the object of management for a given interval of time. Also, the restrictions impose compulsory temporal dependences for this subset of logical facts. Temporal constraints give out immutable fragments in the behavior of the object of management.

Secondly, the conditions for occurrence of events set logical facts and rules that are executed for individual sequences of events. That is, these dependencies determine variables in the behavior of the control object.

The combination of facts, constraints and conditions allows you to predict possible variants of the behavior of the object of management, in particular new sequences of actions that haven't been recorded in the event log yet.

In compliance with the theory of MLNs, the probability of every new variant of the behavior of the control object is calculated. This allows you to select from a subset of the most likely variants and thereby simplify the support for making managerial decisions.

The proposed approach can complement traditional approaches in identifying advantages in decision support tasks [15].

CONCLUSIONS

1. This article addresses the problem of automated construction of probabilistic knowledge bases in the tasks of enterprise management using temporal dependencies.

2. The novelty of the obtained results is that the proposed models of temporal dependencies are patterns for probabilistic temporal rules in the MLNs and they allow automating the construction of these rules based on the events log analysis to solve the problem of automated probabilistic knowledge base construction.

3. The practical significance of the developed models of temporal dependencies is that the weighted rules derived from them are used to construct an ordered set of possible options for the further behavior of the control object in the current situation, which allows for more effective management decisions.

REFERENCES

1. **Bughin J. 2008.** The rise of enterprise 2.0. Journal of Direct, Data and Digital Marketing Practice, 9(3), 251–259.
2. **Kalynychenko O., Chalyi S., Bodyanskiy Y., Golian V., Golian N. 2013.** Implementation of search mechanism for implicit dependences in process mining. 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems. Available: https://doi. org/10.1109/idaacs.2013.6662.
3. **Christidis K., Mentzas G., Apostolou D. 2012.** Using latent topics to enhance search andrecommendation in Enterprise Social Software. Expert Systems with Applications, 39(10), 9297–9307.
4. **Vom Brocke J. 2015.** Handbook on Business Process Management 1. Introduction, Methods, and Information Systems. Springer-Verlag Berlin Heidelberg, p. 709 doi:10.1007/978-3-642-45100-3
5. **Shin J., Wu S., Wang F., De Sa C. Zhang C., R´e C. 2015.** Incremental Knowledge Base Construction Using DeepDive. 41 th International Conference on Very Large Data Bases (VLDB). Vol. 8(11).
6. **Niu F., Zhang C., Re C. 2012.** DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. VLDS, 25–28.
7. **Chalyi S., Levykin I., Petrychenko A. and Bogatov I. 2018.** Causality-based model checking in business process management tasks. Proc. IEEE 9th International Conference on Dependable Systems, Services and Technologies DESSERT'2018. Ukraine, Kyiv. May 24–27, 478 – 483.
8. **Van der Aalst W. M. P. 2014.** Process Mining in the Large. A Tutorial. Business Intelligence. Springer Science + Business Media, 33–76. doi:10.1007/978-3-319-05461-2_2
9. **Gronau N., Thim C., Ullrich A., Weber E. 2016.** A Proposal to Model Knowledge in Knowledge-Intensive Business Processes. BMSD 2016: 6th Int. Symposium on Business Modeling and Software Design. doi:10.5220/0006222600980103.
10. **Richardson M., Domingos P. 2006.** Markov logic networks. Machine learning, 62(1-2), 107–136. doi: 10.1007/s10994- 006-8633-8.
11. **Lowd D., Domingos P. 2007.** Efficient weight learning for Markov logic networks. European Conference on Principles of Data Mining and Knowledge Discovery. Knowledge discovery in databases: PKDD 2007.
12. **Levykin V., Chala O. 2018.** Method of automated construction and expansion of the knowledge base of the business process management system. EUREKA: Physics and Engineering, 4, 29–35.
13. **Levykin V., Chala O. 2018.** Method of determining weights of temporal rules in markov logic network for building knowledge base in information control system. EUREKA: Physics and Engineering, 5, 29–35.
14. **Christian W. Gunther, Eric Verbeek. 2014.** XES Standard Definition. 24.
15. **Beskorovainyi V. V., Berezovskyi H. 2017.** Identification of preferences in decision support systems. Econtechmod. An international quarterly journal. Vol. 6, No. 4, 15–20.