Original Research Article: full paper

(2018), «EUREKA: Physics and Engineering» Number 4

METHOD OF AUTOMATED CONSTRUCTION AND EXPANSION OF THE KNOWLEDGE BASE OF THE BUSINESS PROCESS MANAGEMENT SYSTEM

Viktor Levykin
Department of Information Control Systems
Kharkiv National University of Radio Electronics
16 Nauka ave., Kharkiv, Ukraine, 61166
levykinvictor@gmail.com

Oksana Chala
Department of Information Control Systems
Kharkiv National University of Radio Electronics
16 Nauka ave., Kharkiv, Ukraine, 61166
oksana.chala@nure.ua

Abstract

The problem of constructing and using the knowledge representation in the process control system is studied. It is shown that when implementing knowledge-intensive business process management, it is necessary to use automated construction and expansion knowledge base to support decision-making in accordance with the current state of the context for the implementation of business process actions. The state of the context is specified as a set of weighted logical facts, the arguments of which are the values of the attributes of the events of the business process log. The sequence of the process implementation at each moment of time is displayed in the form of a probabilistic distribution of the possible rules of executing the actions of the business process in this context. The method of automated construction and updating of the knowledge base of the information system of process control is proposed. The method includes the stages of forming knowledge representation templates, constructing context descriptions, logical facts, constructing rules, and calculating the probability distribution for rules. The method creates opportunities to support decision-making on the management of the business process in the event of a discrepancy between the current implementation of the business process and its model.

Keywords: knowledge-intensive business processes, knowledge base, process management systems, context, event, attribute, cause-effect relationships.

DOI: 10.21303/2461-4262.2018.00676

1. Introduction

Process control systems realize the “horizontal” management of the enterprise by developing business process (BP) models and further managing the BP using these models [1, 2].

Process management of the enterprise provides for the construction, use, analysis and refinement of models of business processes [3, 4]. The effectiveness of process management largely depends on the adequacy of BP models. For traditional business processes with an a priori given structure, the problem of adequacy of the BP model is solved at the analysis stage, after completion of the process. Solving this problem for a class of knowledge-intensive business processes causes considerable difficulties due to the key feature of such processes: they can change the sequence of actions based on personal decisions of the knowledge workers [4, 5]. Knowledge workers make a decision to change the course of the process taking into account the current state of the subject area and use both publicly available explicit knowledge and implicit personal knowledge. The latter usually have the form of contextual causal dependences obtained experimentally [6, 7]. Such knowledge is not included in the business process model [8, 9] and can’t be obtained by traditional methods of knowledge engineering.

Thus, when managing knowledge-capacious business processes, the problem arises of constructing and using the knowledge base in the process control system.

The behavior of business processes is recorded by the process management information system in the form of event logs [10]. The description of each event log contains information about the operation of the process and the context of the execution of this action, which allows to determine the cause-effect relationships between actions and conditions of their occurrence.

This indicates the relevance of constructing such knowledge representation of the information system of process management, which would take into account the features of the knowledge-capacious business process presented in the logs.

2. Literature review

Traditional approaches to the construction of knowledge bases (KB) require considerable time spent by qualified specialists in order to formalize the expert experience in the form of cause-effect dependencies. Therefore, such methods are unsuitable for construction knowledge bases for control systems that operate in real time.

In the last decade methods, approaches and technologies of automated knowledge base construction have been widely spread [11, 12]. Such methods are designed to identify knowledge in large databases available on the Internet [13, 14]. However, these approaches have a significant drawback, which narrows the scope of their use: they focus on the static description of dependencies in the domain, which requires the preservation of previous versions of the dependencies presented in the knowledge base [15].

At the same time, when using the knowledge base to support decision-making in the process management system, it is necessary to update the facts base on the subject area synchronously with the progress of relevant business processes. It is also necessary to represent the multivariance of such facts and connections between them.

To solve the latter problem, a probabilistic representation of knowledge based on Markov logical networks is used [16-19]. The basic idea of such networks is use of patterns of logical dependencies to construct a set of weighted predicates reflecting knowledge of the subject domain [20]. However, when constructing such a representation for process control systems, it is necessary to take into account the features of the event logs as input data for the identification of knowledge. Taking these features into account simplifies the construction and expansion of the knowledge base.

The aim of this article is development of a method for automated construction and updating of the knowledge base in the process management information system that would ensure continuous updating of knowledge based on log analysis.

3. Method of construction and expansion the knowledge base

The method uses the logical-probabilistic representation of knowledge based on Markov logical networks. This view takes into account the following characteristics of the event log:

– the log reflects the properties of the artifacts of the business process, i.e. objects with which this process interacts;

– with each event the log is associated with a timestamp, as well as a set of attribute values;

– the attributes of the log events correspond to the attributes of the BP artifacts;

– a set of attributes represented by attributes of artifacts describe the context for executing the actions of the business process.

The representation of knowledge is as follows:

$$KB_{L,P} = (Af, \{(f_i(Af), w_i)\},$$
$$\{(r_j(F), w_j)\}, P(A = \{\alpha_k^{\tau}\}) \mid C), \quad (1)$$

where Af – set of artifacts; F – set of logical facts f_i; w_i – weight of the logical fact f_i; r_j – a logical rule that operates with logical facts; w_j – weight of the logical rule r_j; α_k^{ϕ} – the value of the artifact property at the time τ the event was recorded in the business process log; $P(A = \{\alpha_k^{\tau}\})$ – the probability of the rules being executed at the currently known values of the properties of artifacts; C – a priori known causal dependencies acting as limitations of the subject domain.

This knowledge representation takes into account the sequence of business process deployment in time. The static aspect is given by facts and rules with arguments from the event log, and dynamic – in the form of the current probability distribution of rules execution in accordance with the logged events.

This allows to solve the problem of predicting the most likely behavior of a business process in the event that the current state is unpredictably changed due to the fact that the performers have changed the sequence of actions specified in the model.

The probability distribution of possible realizations of the business process takes into account the weighted sum of the rules and has the traditional form for Markov chains:

$$P(A = \{\alpha_k^\tau\}) = \frac{1}{Z} e^{\sum_{j=1}^{J} w_j r_j(\{f_i\})}, \quad (2)$$

where Z – the partition function used for normalization.

The method includes the following stages.

Stage 1. Building a data representation and knowledge template by defining classes of artifacts (properties of artifacts), typical facts $f_i(Af)$ and rules $f_i(Af)$ in accordance with the log structure. Logical facts templates reflect the attributes of events. The rule templates correspond to the sequence of events and reflect the relationship between the context and actions (by the state of the actions) of the business process.

The result of this stage is: (1) a subset of predicates that establish logical facts; (2) a subset of predicates that establish logical rules. The number of such predicates is limited due to the fact that all events of the log have the same structure. Let's suggest using a minimal set of three predicates: a predicate that sets the value of the artifact property; a predicate that specifies a set of properties for the event; predicate, which determines the rules of transition between events.

Stage 2. Building/supplement of the description of the domain context. At this stage, the attribute of the log events to the classes of artifacts and their properties is determined. Let's note that when building a knowledge base using relational DBMS, the list of properties of artifacts and unique values of these properties is formed from the process log in tabular form by means of SQL queries.

Stage 3. Building the logical facts by substituting the arguments into the predicates as the values of the event attributes. Such arguments characterize the current state of the business process.

The result of the stage is the knowledge base in the form of a set of logical facts that reflect the normal behavior of the business process.

Stage 4. Calculation of the weights of the logical facts in such way that their values correspond to the probability of the rules being executed on the set of events of the business process log.

As a result of this stage, the rules are adapted to take into account additional information about new log events.

Traditionally, such calculation in Markov logical networks is performed by methods based on gradient descent

Stage 5. The construction of rules, the arguments of which are the logical facts obtained in stage 3. Rules are formed on the basis of a predicate-template by substitution of values of arguments.

Stage 6. Calculation of probabilities for rules based on logical facts, the arguments of which are attributes of the events of the business process log.

As a result of this stage, information on the most likely actions of a business process with context-bound information can be used by decision maker. Taking into account the obtained probability values, information on the abnormal behavior of the business process can be generated.

Stages 2–6 are repeated as the business process runs and new events appear in the log.

4. Experimental procedures

The aim of the experiment is testing the possibility of applying the proposed method for solving decision support tasks when detecting intrusions for processes in computer systems.

During the experiment, we used the logs of the computer system CIDDS-002 [21]. CIDDS-002 contains logs with weekly sequences of events. Each event log is characterized by a set of attributes (Source IP Address, Source Port, Destination IP Address, Destination Port, Transport Protocol, Duration of the flow, Number of transmitted bytes, etc.). The format of the input data is shown in **Fig. 1.**

Date first seen	Duration	Proto	Src IP Addr	Src Pt	Dst IP Addr	Dst Pt	Packets	Bytes	Flow	Flags	Tos	label	attackType	attackI	attack Description
02.08.2017 14:25	0.00	UDP	192.168.220.51	51144	192.168.100.20	20449	1	42	1	.....	0	attacker	scan	1	nmap args: -sU -T 0
02.08.2017 14:25	0.00	UDP	192.168.220.51	51144	192.168.100.11	20449	1	42	1	.....	0	attacker	scan	1	nmap args: -sU -T 0

Fig. 1. Fragment of Event log

Logical facts are represented by conjunctions of attribute values, for example, for a log of two lines, shown in **Fig. 1**, the fact f_1 is:

$$f_1(\text{SourceIPAddress} = «192.168.220.51» \wedge$$
$$\text{SourcePort} = «51144» \wedge$$
$$\text{DestinationIPAddress} = «192.168.100.20» \wedge ...) = \text{true}. \quad (3)$$

Logical facts can be generalized. A generalization of logical facts consists in choosing such number of attributes that would allow to achieve a given generality of the fact without simplifying it. In other words, when constructing a pattern of logical facts, let's remove irrelevant attributes. In particular, the example does not use the timestamps.

Logical rules correspond to the transition between events, for example for facts f_1 and f_2 from the given fragment of the log, the general rule for the transition has the form $r_1(f_1 \wedge f_2)$.

From the given log, sequences of events that correspond to processes of normal functioning and carrying out external intrusions are filtered. These sequences are represented as separate traces. The trace of the normal operation process contains 1011427 events, and the process of external intrusion is 37148 events.

Weights of logical facts are calculated separately for both routes. This makes it possible to compare the likelihoods of the rules for normal operation and for attacks. Initially, let's identify the abnormal behavior of the process. However, abnormal behavior does not always indicate errors or external intrusions. It is possible that this behavior of the process is simply not included in the knowledge base. Therefore, then a comparison is made with the template behavior of the process during the invasion. As a result, explicit invasion processes are cut off. The remaining dependencies require consideration of the decision maker.

We have simplified the definition of weights of logical facts in comparison with traditional methods due to the need for real-time work. Traditional methods are extremely resource intensive.

Therefore, the weights of facts and rules are determined based on the frequency of the appearance of attribute values. **Table 1** contains a subset of weights for some attribute values for both traces.

From this table it is possible to see that the weights of the attributes are radically different for the same values under normal operation and during an intrusion. For example, for the attribute Proto="ICMP", the weights for normal and abnormal behavior are 1000 times different. Significant differences between attribute weights create opportunities for comparing states in normal operation and when an attack occurs.

Weights of the transitions between the event attributes for each pair of attributes are also calculated. These weights are calculated for normal operation and for intrusion.

The weights of attributes are summed when determining the weights of a logical fact and rule.

A comparison of the values of the subsets of the transition weights is presented in **Table 2**.

From this table, it is possible to see a difference in the weights of rules that specify the transitions between attributes for normal operation and for intrusion.

When deciding whether to detect attacks, the probability calculation for the rules of transition between events is performed taking into account the probability of a logical fact describing the state. Then, the obtained probabilities are compared for the normal work process and the intrusion process. Based on the results of the comparison, it is decided to detect an attack on the computer system.

Table 1
Weights of the Event Attributes

Event Attributes	Event Attribute Values	Weights of Event Attribute Value	
		Normal processing	Intrusion
Proto	ICMP	2,27401E-05	0,027242382
	TCP	0,861623231	0,97063099
Src IP Addr	192.168.220.51	0,086117931	0,584365242
	192.168.220.48	0,009647755	0,0
Src Pt	445	0,01490765	0,000457629
	46900	1,18644E-05	0,0

Table 2
Weights of Attribute Values for Transitions Between Events

Event Attributes	Transitions	Weights of Event Attribute Value	
		Normal processing	Intrusion
Proto	TCP–>ICMP	1,58192E-05	1,47835E-01
	TCP–>UDP	1,68079E-05	1,22064E-03
Src IP Addr	192.168.220.48–>192.168.220.51	0,001384183	0,012473248
	192.168.100.11–>192.168.100.20	3,46046E-05	1,257432E-05
Src Pt	445–>34696	0,000375707	0,0
	443–>80	0,059390347	1,056037 E-05

5. Results and Discussion

The result of the work is the method of automated construction and replenishment of the knowledge base of the process control system.

The method uses process logs as a source data. As a result of using the method, the following components of the knowledge base are created.

First, a description of the normal behavior of processes in the domain is formed in the form of a set of logical facts. These facts reflect the relationships between the attributes of the log events. This set of facts can be used to identify abnormal behavior of business processes and further adjust the management of such processes.

Secondly, the rules for executing the business process are formed, reflecting the links between the logical facts. This set of rules should be used to identify the reasons for deviations in the behavior of the business process. The identified set of reasons is represented in the form of a subset of attributes that specifies the conditions for the abnormal operation of the business process.

Advantages of the proposed method consist in the ability to combine strategies for identifying abnormal behavior of a business process based on knowledge of its normal behavior and identifying misused behavior based on relevant rules to support decision making in process management.

The disadvantage of the method lies in the fact that the completeness and consistency of the knowledge base largely depends on the quality of the log: the number of event attributes, as well as the correctness of the event timestamps.

The method can also be used to analyze the activities of arbitrary process-oriented systems that form logs. In particular, the proposed method can be used to detect intrusions in a computer system. At the same time, a knowledge base is generated about the behavior of processes (running programs), including dependencies related to known intrusions.

6. Conclusions

The problem of formalization and use of knowledge in the business process management system is considered. The expediency of the automated construction and adaptation of the knowledge base with the purpose of its further use for supporting decision making on business process management is grounded.

The proposed method has the following differences from traditional approaches. First, in constructing facts and rules, the log structure is taken into account: the state of the context for executing the actions of the business process is expressed through the attributes of the events, and the actions through the causal dependencies between the events, which allows to take into account a fixed number of event attributes and thereby simplify the calculation of the weights of the logical dependencies. Secondly, the method provides a cyclical refinement of the weights of facts and rules as the process progresses and new events appear in the log, which allows to continuously adapt the knowledge base to real-time management needs.

In practical terms, the method provides opportunities to improve the management of knowledge-intensive business processes and the ability to identify abnormal and misuse process behavior. Improving the effectiveness of process control is achieved by using causal dependencies between the context and the actions of the process to build new paths for its implementation. The detection of anomalous and misuse behavior of the process is carried out on the basis of a comparison of the probabilities of newly formed rules when new events occur.

References

[1] Van der Aalst, W. M. P. (2011). Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Berlin Heidelberg, 352. doi: https://doi.org/10.1007/978-3-642-19345-3

[2] Vom Brocke, J., Rosemann, M. (Eds.) (2015). Handbook on Business Process Management 1. Introduction, Methods, and Information Systems. Springer-Verlag Berlin Heidelberg, 709. doi: https://doi.org/10.1007/978-3-642-45100-3

[3] Müller, D., Reichert, M., Herbst, J. (2007). Data-Driven Modeling and Coordination of Large Process Structures. Lecture Notes in Computer Science, 131–149. doi: https://doi.org/10.1007/978-3-540-76848-7_10

[4] La Rosa, M., Dumas, M., ter Hofstede, A. H. M., Mendling, J. (2011). Configurable multi-perspective business process models. Information Systems, 36 (2), 313–340. doi: https://doi.org/10.1016/j.is.2010.07.001

[5] Gronau, N. (2012). Modeling and Analyzing knowledge intensive business processes with KMDL: Comprehensive insights into theory and practice (English). Gito, 522.

[6] Brocke, J. vom, Zelt, S., Schmiedel, T. (2016). On the role of context in business process management. International Journal of Information Management, 36 (3), 486–495. doi: https://doi.org/10.1016/j.ijinfomgt.2015.10.002

[7] Van der Aalst, W. M. P. (2014). Process Mining in the Large: A Tutorial. Lecture Notes in Business Information Processing, 33–76. doi: https://doi.org/10.1007/978-3-319-05461-2_2

[8] 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 (IDAACS). doi: https://doi.org/10.1109/idaacs.2013.6662657

[9] Gunther, C. W., Ma, S. R., Reichert, M., Aalst, W. M. P. V. D., Recker, J. (2008). Using process mining to learn from process changes in evolutionary systems. International Journal of Business Process Integration and Management, 3 (1), 61. doi: https://doi.org/10.1504/ijbpim.2008.019348

[10] Chalyi, S., Levykin, I., Petrychenko, A., Bogatov, I. (2018). Causality-based model checking in business process management tasks. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). doi: https://doi.org/10.1109/dessert.2018.8409176

[11] Niu, F., Zhang, C., Re, C., Shavlik, J. W. (2012). DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. VLDS, 25–28.

[12] Nakashole, N., Weikum, G. (2012). Real-time Population of Knowledge Bases: Opportunities and Challenges. Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX 2012). Montreal, Canada, 41–45.

[13] Galárraga, L., Heitz, G., Murphy, K., Suchanek, F. M. (2014). Canonicalizing Open Knowledge Bases. Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management – CIKM '14. doi: https://doi.org/10.1145/2661829.2662073

[14] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. (2008). Freebase. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data – SIGMOD '08. doi: https://doi.org/10.1145/1376616.1376746

[15] Shin, J., Wu, S., Wang, F., De Sa, C., Zhang, C., Ré, C. (2015). Incremental knowledge base construction using DeepDive. Proceedings of the VLDB Endowment, 8 (11), 1310–1321. doi: https://doi.org/10.14778/2809974.2809991

[16] Huynh, T. N., Mooney, R. J. (2008). Discriminative structure and parameter learning for Markov logic networks. Proceedings of the 25th International Conference on Machine Learning – ICML '08. doi: https://doi.org/10.1145/1390156.1390209

[17] Richardson, M., Domingos, P. (2006). Markov logic networks. Machine Learning, 62 (1-2), 107–136. doi: https://doi.org/10.1007/s10994-006-5833-1

[18] Niu, F., Zhang, C., Re, C., Shavlik, J. (2012). Scaling Inference for Markov Logic via Dual Decomposition. 2012 IEEE 12th International Conference on Data Mining. doi: https://doi.org/10.1109/icdm.2012.96

[19] Singla, P., Domingos, P. (2006). Entity Resolution with Markov Logic. Sixth International Conference on Data Mining (ICDM'06). doi: https://doi.org/10.1109/icdm.2006.65

[20] Van Haaren, J., Davis, J. (2012). Markov network structure learning: A randomized feature generation approach. Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 1148–1154.

[21] Niu, F., Ré, C., Doan, A., Shavlik, J. (2011). Tuffy. Proceedings of the VLDB Endowment, 4 (6), 373–384. doi: https://doi.org/10.14778/1978665.1978669

EXPLANATION OF THE ALGORITHMS FOR DISPLAYING 3D FIGURES ON THE COMPUTER SCREEN

Alexander Zhyrytovskyi
Institute of Computer Technologies, Graduate School
Open International University of Human Development "Ukraine"
23 Lvivs'ka str., Kyiv, Ukraine. 04071
i.am.zhirik@gmail.com

Abstract

Not long time ago people can only dream about the thing that they can see 3-dimensional world inside of the screen of the computer. Computers at that time can display only 256 colors on their screens, they can work only with integer numbers, and their computation speed was not fast enough. This dream comes true with the help of Id Software company. This company started to build its games by using of simplified 3D graphics. All these simplified 3D graphics were calculated by the software 3D rendering algorithms which were not published by these company. At that time there was a global goal to speedup calculations for 3D graphics and to make it look more realistic. The solution for this comes from hardware companies. Hardware companies started to work on hardware 3D accelerators, which can render 3D graphics much faster than software algorithms. These 3D accelerators are fully patented, so that no one knows how they are implemented inside. At that time there were proposed two 3D rendering interfaces: Direct3D and OpenGL. And manufacturers of graphic cards started supporting both of these interfaces. Lots of game development companies started to use these interfaces. And also Id Software company started to redesign its game engine to support these interfaces to take place in game market. From that time, everybody started to use 3D accelerators with built-in 3D rendering features and as a result everybody forgot about software 3D rendering algorithms.

Nowadays the situation come in the following way, that most of the algorithms which are used in 3D games are not published on paper, and all the 3D accelerator internal algorithms are patented. All modern 3D graphics are fully conserved by the video cards, so that 3D rendering could not be developed in other way.

The goal of this article is explanation of how 3D graphics can be displayed on the screen of the computer

Keywords: 3D, graphics, rendering, mesh, z-buffer, texture, projection, perspective, triangulation, triangle drawing.

DOI: 10.21303/2461-4262.2018.00680