

# Decision Support Using Belief Network Constructed from Business Process Event Log

Titas SAVICKAS<sup>1\*</sup>, Olegas VASILECAS<sup>1,2</sup>

<sup>1</sup>*Information Systems Department, Vilnius Gediminas Technical University, Lithuania,*

<sup>2</sup>*Institute of Mathematics and Informatics, Vilnius University, Lithuania*

*e-mail: titas.savickas@vgtu.lt, olegas.vasilecas@mii.vu.lt, olegas.vasilecas@vgtu.lt*

Received: January 2017; accepted: September 2017

**Abstract.** Information systems contain a lot of data regarding business process execution history. Use of this data, in the form of an event log, can greatly support business process management. The paper presents an approach to construct Bayesian belief network from an event log that could facilitate decision support in business process execution. The approach is evaluated against multiple event logs by inferring data probabilities occurring in the business processes. The results show that the approach is suitable for the task and could be used in decision support with future research focused on prediction and simulation of business processes.

**Key words:** event log, Bayesian belief network, decision support, probability inference.

## 1. Introduction

Information systems are at the core of any organization in this information age. Big part of business-related data is now at least partially stored electronically and reflects how business processes are being executed in the organization. This data is not only used for controlling business processes, but also to discover knowledge that has previously been unknown, for example, business rules that are not explicitly documented that can only be found by applying data mining methods. The research area focused on using data about business process execution in information systems for analysis is called process mining. The process mining approaches can be used to discover business process models, enhance them or perform conformance checking of process execution versus existing models (van der Aalst *et al.*, 2012). Another use of process mining approaches is to facilitate decision support.

While there are multiple methods for decision support, they focus on a specific task, such as predicting duration, follow-up activities or identify anomalies. In order to solve complex tasks, such as simulation model creation, the methods have to be combined and applied for each of the subtasks (Martin *et al.*, 2014). Since processes are stochastic by nature (Kellner *et al.*, 1999) or their behaviour is unpredictable (van der Aalst *et al.*, 2010), probabilistic models could be applied to model business process data and its' occurrence

---

\* Corresponding author.

in a probabilistic manner. This could facilitate a single model for multi-perspective analysis of process data behaviour. Bayesian belief networks have been applied in multiple areas where there is a need to work with probabilities and dependencies between temporal events or data attributes (Arroyo-Figueroa and Sucar, 1999), but their application in process mining has seen limited use (see Section 2), therefore it is needed to see how it can be applied for decision support in process mining.

In this paper, an approach to create Bayesian belief network from an event log is presented. The discovered Bayesian belief network is used to infer probabilities of data occurring in business processes which could then be used for decision support, e.g. a manager could check probability of an activity in the process to receive status *failed* and react appropriately. There is a plan to use the Bayesian belief network for data generation and behaviour inference in business process simulation that is based not on statistics, but on probabilistic analysis; it could also be used to create initial simulation models from the belief network.

This paper consists of 5 sections. It starts with introduction – problem statement and proposed approach. It is followed by Section 2 with related literature. Section 3 introduces components of the approach and the method on how to construct the Bayesian Belief Network from an event log. Section 3 ends with the description on how inference is done using the created components. Section 4 provides evaluation of the approach – the experiment is defined, experimental environment is introduced and the results of the experiments are presented. The paper ends with conclusions, further research and applications of the presented approach.

## 2. Related Work

Decision support is becoming widely used application of process mining methods. It has been used for activity or parameter prediction, decision rule mining and anomaly detection. Time prediction is one of the most widely used applications of process mining employing different prediction methods such as van Dongen *et al.* (2008), where regression equations based on event logs are used to prepare model for prediction on when the process instance (case) will be finished or generate transition system from an event log which is used for time prediction of a case (van der Aalst *et al.*, 2011). Rogge-Solti and Kasneci (2014) use non-Markovian stochastic Petri nets with elapsed time since last observed event to predict the follow-up event most probable durations. Finally, Verenich *et al.* (2016) used SVM prediction model to eliminate over processing by detecting redundant activities.

Process mining has also been applied in decision analysis. In Rozinat and Aalst (2006) the authors attempt to extract rules for control flow point in the process model based on data in event logs. The rules are extracted using classification algorithms such as C.45. In de Leoni and van der Aalst (2013), the authors use alignment in business processes to extract data flow rules between activities. Instead of knowledge discovery, there are also methods for real-time decision support such as Liu *et al.* (2012) where it is proposed to simulate discovered models for use in decision support. Also, process mining has been

applied to domain specific decision analysis as in Sarno *et al.* (2015) where an approach to use process mining and association rule learning for fraud detection is presented.

Bayesian probabilistic models have previously been used in process mining area, but not for general decision support. Ping *et al.* (2010) presented an approach to build Bayesian networks with data about event sequences and their temporal probabilities as additional nodes. The approach is specific to temporal anomalies and does not provide insight on how to detect general anomalies. In Rogge-Solti *et al.* (2013) the authors combine stochastic Petri nets, alignments and Bayesian networks for repairing event logs. The approach uses Bayesian inference to detect most likely timestamps for each event, but the approach uses only durations of activities and does not build a general model of the whole process. Sutrisnowati *et al.* (2015) build a general Bayesian belief network and a CPT building approach. It uses the built model to detect when a process in a ship port will be late. The approach uses Heuristic Miner to discover dependency graph and removes loops to generate the directed acyclic graph. It does not employ stateful information. In van der Spoel *et al.* (2012), the authors use multiple probabilistic methods for process predictions and one of them is Bayesian Naive Classifiers. The authors do not build a full Bayesian belief network nor explain data used, but for very complex log it shows prediction rate average between 30% and 45%.

### **3. Bayesian Belief Network Construction**

Business processes are by nature complex and stochastic, therefore it's useful to analyse them using probabilistic methods which do not operate on clear-cut rules. The answers provided are with some degree of certainty. This way of thinking reflects real-life decision making where not all conditions are known and the process execution is not always governed by business rules, but by the context of the process.

Usual approach for building decision support systems is to collect expert knowledge and create a model which could be used for decision support. Expert knowledge collection is a manual labour and it needs to be automated. For this reason, data on historical business process execution can be employed for automated knowledge discovery. Process mining deals with process on how to re-use data existing in information systems regarding business process execution for discovering previously unknown knowledge. This section describes how Bayesian belief network is constructed from an event log and is used for inference to support decision process.

#### *3.1. Bayesian Belief Network*

One of probabilistic methods available for such analysis is Bayesian belief network (Darwiche, 2008). It allows to represent a set of variables and their conditional dependencies via a directed acyclic graph. For example, given some known information about client, whether the process will execute successfully.

DEFINITION 1. A Bayesian network over variables  $X$  is a pair  $(G, \Theta)$  where  $G$  is a directed acyclic graph (DAG) over variables  $X$  and  $\Theta$  is a set of conditional probability tables (CPTs).

The Bayesian belief network could be used to answer questions important to business process owners by exploiting general probability calculations:

- Probability of evidence  $P(X|e)$  can be used to answer questions such as “*What’s the chance for insurance claim to be declined for someone aged 20–30 years old?*”
- Most probable explanation  $MPE(e) = \arg \max_x Pr(x, e)$  can be used to answer “*What is the probability for process to end now given the current state?*”
- Maximum a posteriori hypothesis  $MAT(e, M) = \arg \max_x Pr(m, e)$  could be used for “*What’s the most probable outcome of a claim check if the claimant is aged 23 years old and made the claim in Vilnius?*”

The belief network contains two components – a DAG that represents variable dependency and CPTs that represent probabilities. Based on previous section, this is very suitable for event log transformation. In the event log, event sequences are stored and each unique event could be represented as a node in a DAG and data dependency of events could be represented by the CPTs. While nodes transformation is not a hard task, arc between nodes creation is not so easy, because they reflect conditional dependency in the Bayesian belief network and transitions between events in the business process, therefore the task of DAG creation becomes complex.

### 3.2. Event Log

Process Mining methods focus on applying data-mining methods on data existing in information systems that represent historical execution of business processes (Verbeek *et al.*, 2010). This data comes in a form of an event log which consists of data collected from various sources. There are a few ways to represent event logs, but the most common one is the XES file format (Günther and Verbeek, 2014) – a standardized and extensible file format. It is extensible and allows addition of domain specific data about business process execution. The event log contains general information on execution of the business process, such as trace identifier to identify process instance and a list of events with occurrence timestamp and identifier (Table 1). Each trace and event might also contain any other additional data related to the behaviour, e.g. client names, ages, locations, system specific information such as subsystem, server, etc.

In the scope of this paper, Event Log definition used for transformation, is based on van Dongen *et al.* (2008) and adapted from previous work (Savickas and Vasilecas, 2014) and is defined as follows:

DEFINITION 2. An event log over a set of activities  $A$  and time domain  $TD$  is defined as  $L_{A,TD} = (E, C, M, V, \mu, \alpha, \gamma, \beta, \succ)$ , where:

- $E$  is a finite set of events,

Table 1  
Fragment of an exemplary event log with data.

Trace ID	Event	Timestamp	Attribute resource	Attribute claimant	Attribute status
1	Incoming_claim	2014.01.05 08:05		1	
1	Register_claim	2014.01.05 08:30	A	1	
1	End	2014.01.05 13:57	B	1	Reject
2	Incoming_claim	2014.01.07 13:07		2	New client
2	Register_claim	2014.01.07 13:37	A	2	
2	Initiate_payment	2014.01.10 11:15	B	2	Payed
2	End	2014.01.10 11:17	B	2	Closed

- $I$  is a finite set of traces,
- $N$  is a finite set of attribute names,
- $V$  is a value space of attributes,
- $M : N \times V$ : is a finite set of attributes,
- $\mu : E \rightarrow M$  is a function assigning each event with attributes and their values,
- $\alpha : E \rightarrow A$  is a function assigning each event to an activity,
- $\gamma : E \rightarrow TD$  is a function assigning each event to a time stamp,
- $\beta : E \rightarrow C$  is a surjective function assigning each event to a case,
- $name : E \rightarrow N$  is a function identifying the name of an event and  $name(ev) = v : (v \in V, n \in N : (v, n) \in \mu(e) \wedge n = \text{"concept : name"})$
- $>\subseteq E \times E$  is the succession relation, which imposes a direct ordering of the events in  $E$ ,
- $>\subseteq^+ E \times E$  is the succession relation, which imposes a total ordering of the events in  $E$ .

### 3.3. State Transition System

One of the components of the Bayesian belief network is a directed acyclic graph. Business processes expose complex behaviour, such as parallelism, repeated execution of activities, loops and others. This causes standard process models that are based on graph theory to be unusable for Bayesian belief networks since they can expose the same cyclic behaviour and are not acyclic.

Event logs contain data on events that have occurred in the process. The information on the sequencing of specific events is hidden in the log and needs interpretation to understand what events can and what events cannot follow each other. In order for those event sequences to be transformed into a directed acyclic graph, a labelled state transition system can be used. In this transition system, each event has a unique label and no cycles are formed, because repeated events have unique labels. This way, event sequences are represented as a unique path between states, where a state is never accessed twice. For example, sequences  $\{a, b, c, b, d, e\}$  and  $\{a, b, c, d, e\}$  in an event log would be represented as state transition sequences  $\{\{a, b\}, \{b, c\}, \{c, b1\}, \{b1, d\}, \{d, e\}\}$  and  $\{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}\}$ .

Van der Aalst *et al.* (2011) used state transition system to predict process instance duration and each state was for either a set of events that occurred, or event sequences that

have occurred. We believe that process work-flow analysis is best represented by event sequences, therefore we choose to represent each state as a sequence of events and their data.

**DEFINITION 3.** Given a state representation function  $l^{state}$ , an event representation function  $l^{event}$  and a partial trace  $\sigma$ , a labelled transition system is defined as  $TS = (Y, E, T)$  where  $Y = \{l^{state}(hd^k(\sigma)) | \sigma \in L \wedge 0 \leq k \leq |\sigma|\}$  is the state space and  $hd^k(\sigma)$  is a “head” of event sequence in a trace of first  $k$  elements.  $E = \{l^{event}(\sigma(k)) | \sigma \in L \wedge 1 \leq k \leq |\sigma|\}$  is the set of events labels, and  $T \in Y \times E \times Y$  with  $T = \{l^{state}(hd^k(\sigma)), l^{event}(\sigma(k+1)), l^{state}(hd^{k+1}(\sigma)), l^{event}(\sigma(k)) | \sigma \in L \wedge 0 \leq k \leq |\sigma|\}$  is the transition relation.  $Y^{start} = \{l^{state}(\epsilon)\}$  is the singleton of initial states and  $Y^{end} = \{l^{state}(\sigma) | \sigma \in L\}$  is the set of final states.

The definitions of trace state and event state are not clearly defined in van der Aalst *et al.* (2011), therefore we introduce the definitions for use in the CPT construction.

Event state describes attributes and their values that belong to the specific occurrence of an event in a trace, therefore we can reuse the definition of  $\mu$  in the event log definition:

**DEFINITION 4.** Event state is defined as  $l^{event}(e) = \mu(e)$ ,  $e \in E$ ,  $\mu(e) \in M$  and it describes attributes and their values of a specific event.

A state of a trace is a collection of event states, therefore it can be defined as:

**DEFINITION 5.** Trace state for a partial trace is represented as a set of event states  $l^{state}(\sigma) = \{(e, M_e, e_{previous}) | \sigma \in T, e \in E \wedge \forall \alpha(e) = t, M_e \in M \wedge \mu(e) = M_e, e_{previous} > e\}$ .

While the labelled state transition system in the referenced paper is used for some general attribute prediction, it does not provide any prediction functions for attribute or sequence predictions, therefore it is used only as directed acyclic graph representation in Bayesian belief network. The state representations are used for observations used in conditional probability calculations.

### 3.4. Event Log Transformation to Bayesian Belief Network

For a Bayesian belief network to be constructed, we start with an event log. Usually, process execution data is stored in information systems in many different places and forms. Data collection regarding business processes is always a context-dependent task. That is because each organization has unique information system implementations and their business process differ from one organization to another. Due to this reason, our approach ignores the data collection task and assumes that an event log is present as defined in Section 3.2.

After collecting data and creating an event log, the construction of belief network can be started. The overall approach is depicted in Fig. 1 and is done in a sequence of steps as follows:

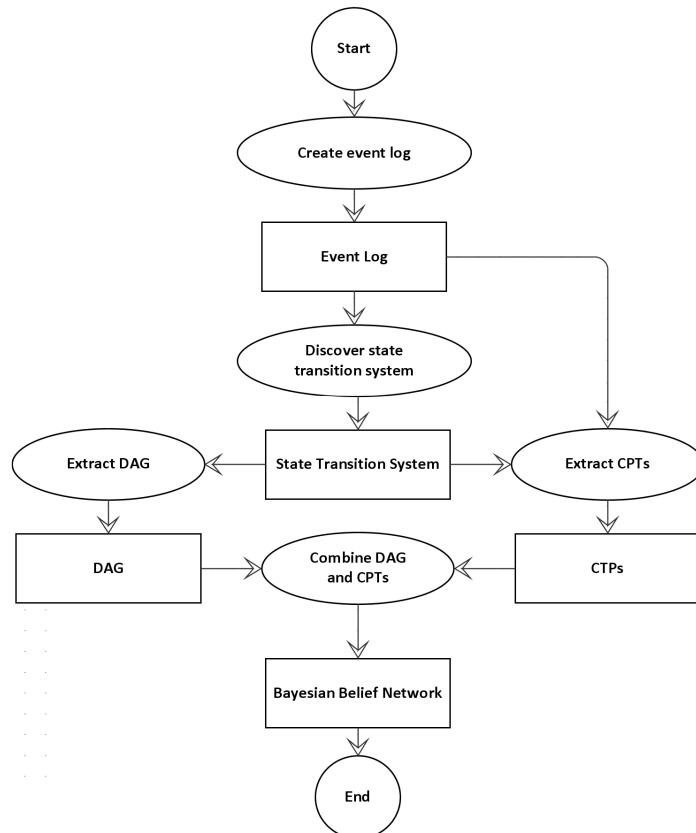


Fig. 1. Bayesian belief network construction.

1. From the event log  $l$  state transition system  $T_l$  is discovered;
2. A DAG  $G$  is extracted from the state transition system  $T_l$ . It is done by removing any state data in the state transition system. This leaves a Directed Acyclic Graph;
3. Conditional Probability Tables  $\Theta$  are constructed;
4. The DAG  $G$  and CPTs  $\Theta$  are combined into Bayesian belief network  $(G, \Theta)$ .

CPT aggregates data in the event log for each event and its' parents into a single table where each attribute combination is assigned a probability. Therefore, is is defined as:

**DEFINITION 6.** A Conditional Probability Table of an event is defined as  $\theta = (A_e, V_e, \omega)$  where  $A_e = \{x : \exists e_i \exists e_j | e_j \in E \wedge e_i \in E \wedge e_j \prec e_i \wedge \beta(e_i) = \beta(e_j) \wedge x = \mu(e_j)\}$  is the attribute space of event and its predecessors in the event log and  $V_e = \{x : \exists e_j | e_j \in E \wedge x = \mu(e_j)\}$  is the set of values that belong to the attributes of the events and  $\omega = P(v \in V_e | a \in A_e)$  is the probability function for each possible attribute node related to attribute value set of parent node.

The CPTs for the DAG are constructed as follows:

1. Data attributes and their values  $M_e$  from an event log are collected for each event  $e$  with identical partial traces  $\sigma$ ;
2. For each event  $e_{previous}$ , data attributes and their values  $M_{previous}$  from an event log are collected;
3. CPT  $\theta_e$  is constructed where each row represents a unique set of attribute subsets  $N \times V \in M_e \cup M_{previous}$  and each row has a probability of  $\omega = \frac{\text{count\_of\_times\_seen}}{\text{total\_count\_of\_event\_occurrences}}$ .

### 3.5. Business Process Inference

Business processes, once automated in an information system, have a controlled work-flow. During this work-flow, performers of the process generate data with regard to the process execution, such as location, organizational resource or other domain specific data, e.g. student group, faculty, etc. This data, once taken as a whole, allows to detect causality between events or between data parameters.

Usual approaches for analysing business process execution is to use statistical data, such as averages, maximums, minimums, sums, frequencies and others. While this does provide a means to infer how the process behaves, it could be superficial, because it might not take conditional dependencies between data. For this, we believe that Bayesian inference could be used for decision support, because it provides reasonable expectations.

Bayesian inference derives the posterior probability using well known Bayesian inference formula (Darwiche, 2008):

$$\frac{P(H|D) = P(D|H) \times P(H)}{P(D)}$$

In here,  $P(E|H)$  is the posterior probability of a hypothesis  $H$  based on evidence  $D$ , which is a consequence of two antecedents – the prior probability  $P(H)$  and a likelihood  $P(D|H)$  with a marginal likelihood  $P(D)$ .

For business processes, the hypothesis is any set of event attributes and values whose probability we would like to infer, for example, “*what is the probability of the claim status to be declined?*”, where *claim* is the event, *status* is the attribute and *declined* is the value.

**DEFINITION 7.** Hypothesis of a business process is defined as  $H_t \in N \times V, \exists e_i \in E, h \in H_t : h \in \mu(e_i)$  – a set of event’s attributes and value pairs which have been observed in the past in the log.

The hypothesis is not limited to a single  $\{e, m\}$  tuple. Since business processes can drift and mutate in time, we limit the possible choices for hypothesis only to those attribute and value pairs which have been seen before in the trace.

Since hypothesis contains multiple possible elements, the prior probability is calculated as a product for each of the attribute values to occur with no conditions, i.e.

$$P(H) = \prod_i P(h_i) = \prod_i \omega(h_i).$$



In standard statistical methods, only the number of times when  $H|E$  occurred would be used for inference, but this is not really useful for decision support, because it does not take into account the marginal likelihood and only shows a number of times it has been seen regardless of the likelihood of each of the parameters, i.e. not only how often this hypothesis has been seen before, but also how likely is it to be seen given current evidence. Therefore, for inference, we need to use the evidence likelihood. We assume that the inference is done in the context of a single process instance, therefore the evidence is the current state of the trace of the process.

**DEFINITION 8.** Given a partial trace  $\sigma$  and the current state of a process  $l_{state}(\sigma)$ , the evidence for a hypothesis is defined as a set of events that have occurred in the current partial trace and their attribute value pairs  $D = \{(e_i, m_i) | e_i \in \sigma, (e_i, m_i) \in E \times M, e_i \in \sigma, m_i \in \mu(e_i)\}$ .

Given the definition of the evidence, marginal likelihood can be calculated as a sum of all probabilities for the evidence to occur with the subsets of hypothesis, i.e.

$$\begin{aligned} P(D) &= \sum_i P(D|h_i) \times P(h_i) = \sum_i \left( \prod_j P(d_j|h_i) \right) \times P(h_i) \\ &= \sum_i \left( \left( \prod_j \omega(d_j)/\omega(h_i) \right) \times \omega(h_i) \right). \end{aligned}$$

Finally, the prior likelihood is the probability to see the evidence given the hypothesis and it can be calculated as  $P(D|H) = \frac{|D \times H \in M|}{|D|}$ , i.e. the number of times the evidence has been seen together with the hypothesis divided by the number that the evidence has been seen in general.

Having all of the components, we get the final inference formula:

$$P(H|D) = \frac{\frac{|\omega(D \cap H)|}{|\omega(D)|} \times \prod_i \omega(h_i)}{\sum_i \left( \left( \prod_j \omega(d_j)/\omega(h_i) \right) \times \omega(h_i) \right)}.$$

#### 4. Evaluation of the Approach

The presented approach is to be used for decision support and allow to preemptively identify most probable execution path of a process. Usually in the real-life scenarios, there would be some before-known hypotheses whose probabilities should be identified in order to understand whether the execution is going in the “right” direction. Some of the exemplary hypotheses could be to identify whether some state will be reached such as event *end*, whether the state will contain some data such  $\{\text{“status”}, \text{“successful”}\} \in \mu(e_{done})$ .

Table 2  
Parameters of the used event logs.

Log	Traces	Unique events	Total events	Attributes
SL	3512	9	20339	2–6
DL	13087	36	262200	3–4
ML	1156	289	59083	12

#### 4.1. Experiment Definition

For formal verification of the approach, domain specific questions should be ignored and there needs to be an objective testing. Therefore it was decided to test the approach with event logs and calculate probabilities of already known values and see whether it is capable of achieving high probability rates.

The event logs used in experiments should be with multiple complexity levels. For this, a synthetic log (SL) and two publicly available logs were chosen (Table 2). The publicly available logs were of a Dutch Financial Institution event log (DL) taken from Business Process Intelligence Challenge 2012 (van Dongen, 2012) and Municipality event log (ML) from Business Process Intelligence Challenge 2015 (Van Dongen, 2015). The ML log contains time stamp or unrelated attributes *activityNameNL*, *dateFinished*, *dueDate*, *planned*, *datestop* and during experiments they are ignored.

The experiments were done independently for each event log as follows:

1. The event log is transformed into two subsets – 80% and 20%;
2. The 80% subset is used for discovering belief network;
3. Leftover set of the remaining 20% is used for experimental testing;
4. Average probability and standard deviation for the experiment is calculated;
5. Experiment is repeated for 4 more times with different subsets of event logs to create  $k$ -fold = 5 results.

The experiment itself is performed by imitating the execution of a business process. The system iterates through each event creating a partial trace with a state  $l^{state}(\sigma)$ , where  $\sigma$  is currently iterated part of the trace in the event log. Knowing what the last event with a state  $l^{event}(\sigma(k))$  is, we calculate  $P(l^{event}(\sigma(k)) | l^{state}(hd^{k-1}(\sigma)))$  – the probability for the event's state, given the already occurred events in the partial trace. Probability calculations are done only when  $|\sigma| > 0$ , i.e. at least one event is in the partial trace. This is done, because we are not interested in the first event in the trace – its' probability does not have any conditional dependencies, therefore it does not test the approach.

After completing the experiment for each of the event log for the 5 times, the average probability is calculated for each event. This allows to see what is the general capability of the approach. Results, where probability is equal to 0 or events that have occurred less than 5 times are rejected as noise. The probability equal to 0 is rejected because it has some data that has never occurred in the event log, therefore decision support for such cases is impossible and it does not answer whether the approach is any good. The rarely occurring events are also rejected, because they do not appear frequently enough for reliable results.

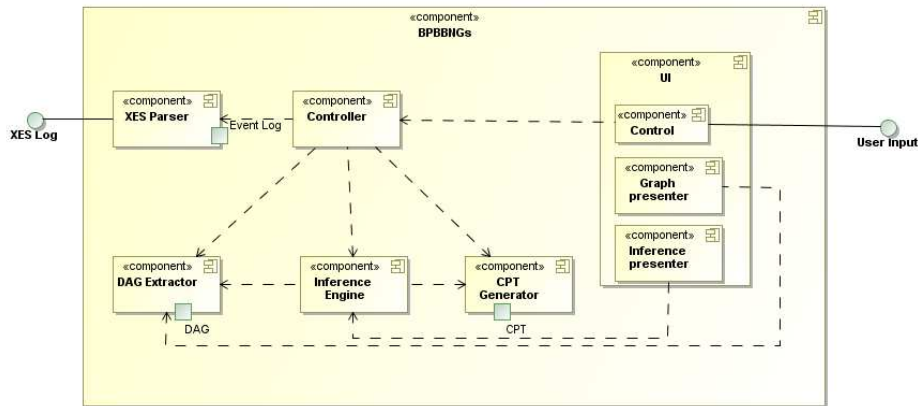


Fig. 2. Architecture of the prototype implementing the proposed approach.

#### 4.2. Experimental Environment

The selected experiment process allows to see how the proposed approach behaves with different attribute counts and differing complexity of the event logs. The approach was implemented in a prototype tool. The prototype tool is called BBNBs (Business process Belief Network enGine) and implemented using .NET framework. The BBNBs is a tool designed to receive an input event log of a business process, transform it into belief network and allow inferences on the belief network. The overall architecture is shown in Fig. 2.

The main component responsible for the behaviour of the tool is controller and it exposes the behaviour to the GUI. Graphical user interface is used by users to perform actions like setting source files, performing observations, or previewing extracted graphs and inference results to the user.

The initial task is to receive input in the XES format from an external system. The specific input format was chosen, because, as described previously, most of the time information systems have no clear event logs of business processes and the data might be heterogeneous. The XES parser component loads the data into the BBNBs and makes the event log accessible in memory.

Afterwards, a component for each of the steps described in previous section is present – DAG Extractor for extracting labelled transition system and the directed acyclic graphs from an event log, CPT Builder for creating conditional tables and inference engine which is responsible for observing variables and performing inferences. Inference Engine component is responsible for making inferences on the generated belief network. It is used by UI component to make required inferences and allow extraction of knowledge about business processes.

Table 3  
Probability inference results.

Log	Inferences taken into account	Total inferences/ total events in the log	Events observed/ events in the log	Precision
SL	13262	16811/20339	8/9	$0.77 \pm 0.26$
DL	95097	147070/262200	33/36	$0.52 \pm 0.35$
ML	11787	43435/59083	42/289	$0.95 \pm 0.16$

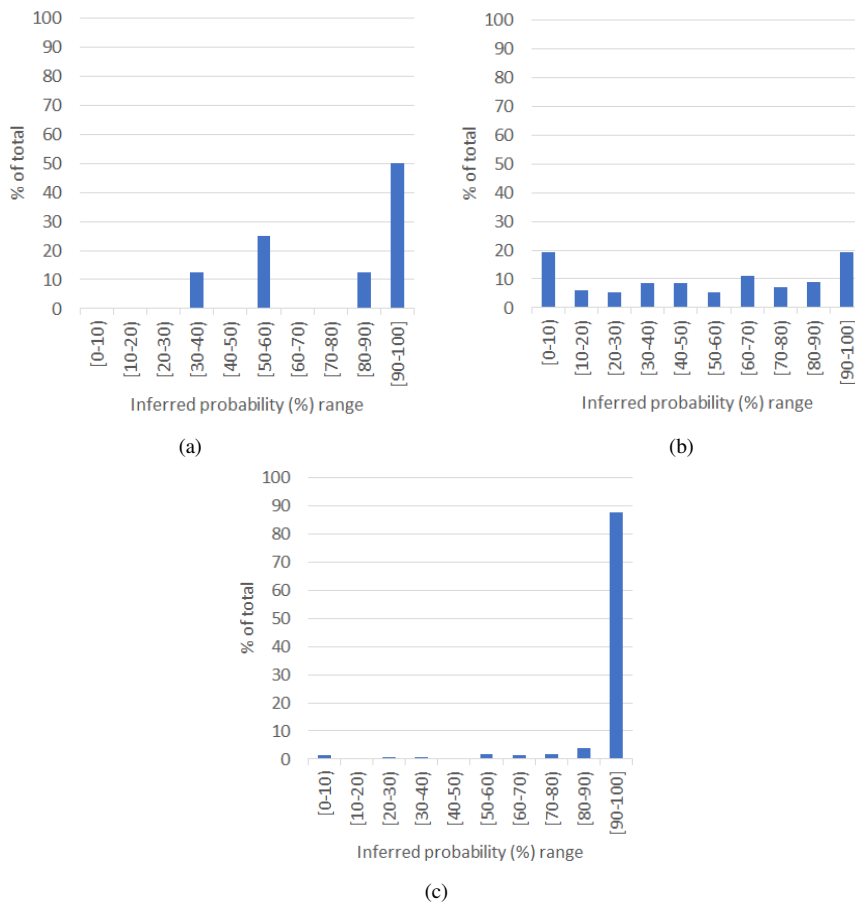


Fig. 3. Inference results of (a) SL, (b) DL, (c) ML.

#### 4.3. Experimental Results

The experiments resulted in a total of 17750 trace runs with a total of 207316 events (Table 3). From all of the probability inferences, 87170 of them were rejected as unsuitable, because they had data not available in the training set, were anomalous with  $P(D|H) = 0$  or because they were the first event in the trace. The inference results are visualized in Fig. 3.

From all of the inferences taken into account, the highest average probability was for synthetic log, as expected. This was due to the underlying process being rather simple. The inferences were successful on average 77% of time with 4 of the events having inferred probability on average higher than 99% with deviation <1%. Other events had the average probabilities spanning from 31% to 82% with deviation ranging from  $\pm 38\%$  to  $\pm 49\%$ .

Other processes were much more complex, having many more possible data variations. This resulted in a lower average probability. In the case of DL log, the events contained barely 3–4 data attributes, therefore their causal dependency is arguable. This resulted in average probability of 52%, but 10 out of 36 events had average inference probability higher than 80%. 3 events in the DL log were ignored in inferences, because either they were always the first event or have occurred less than 5 times in each of the test sets.

The ML log has the lowest results regarding calculations taken into account (11787 out of 43435), but the process itself is the most complex, because the log has 289 unique possible events and only 1156 traces in total. This causes the belief network to be under-trained due to such complex structure and low amount of data. Also, it had usable inferences only on 42 out of possible 289 events. Ignoring that, it had average probability of 95%. Even more so, in total 39 events out of 42 taken into account had the average probability higher than 80%.

To sum up, the experiments show that the approach is usable, but it relies heavily on data – in case attribute values are observed that have never been observed, or if there is limited amount of historical observations which do not fully cover the process behaviour, the approach has limited use. But for events whose behaviour is expressed in the event log, the proposed approach shows great results and allows to answer questions important to the execution of processes – whether events are expected in the process instance, what data might occur there and others.

## **5. Conclusions**

The paper presents an approach on how to construct a Bayesian belief network from an event log and perform inferences on the constructed network for business process decision support. The approach takes an event log, creates a system state transition which is then used to create directed acyclic graph and combines the directed acyclic graph with the data in the event log to construct Bayesian belief network. The created network has been evaluated using 3 event logs with multiple complexity levels to test whether inferences are reliable and can be used for decision support. The main conclusions of the paper are:

- The presented approach allows construction of Bayesian belief network from an event log;
- The approach, when used for decision support, provides, on average, 52% to 95% probabilities for actual data, proving that the approach can be used for inferences;
- The approach is dependent on data quantity in the event log and its' expressiveness.

As can be seen, the approach provides satisfying probability inferences which can be used for decision support. There is still a need to improve the approach and make it more suitable for very complex processes, where there can be a lot of event types but a relatively

small amount of events in the log. Further plans of the research is planned to see how to the approach can be applied to automatically predict business process behaviour, i.e. what events will occur and with what data attributes. Also, it is planned to research how the approach can be used to generate initial business process simulation models, therefore reducing human labour required to create such models.

## References

- van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russell, N. (2010). Business process simulation: how to get it right? In: *Handbook on Business Process Management*. Springer, Berlin, Heidelberg, pp. 313–338.
- van der Aalst, W.M.P., Schonenberg, M.H., Song, M. (2011). Time prediction based on process mining. *Information Systems*, 36(2), 450–475.
- van der Aalst, W.M.P., Adriansyah, A., Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., van den Brand, P., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., de Leoni, M., Delias, P., van Dongen, B.F., Dumas, M., Dustdar, S., Fahland, D., Ferreira, D.R., Gaaloul, W., van Geffen, F., Goel, S., Günther, C., Guzzo, A., Harmon, P., ter Hofstede, A., Hoogland, J., Ingvaldsen, J.E., Kato, K., Kuhn, R., Kumar, A., La Rosa, M., Maggi, F., Malerba, D., Mans, R.S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Motahari-Nezhad, H.R., zur Muehlen, M., Munoz-Gama, J., Pontieri, L., Ribeiro, J., Rozinat, A., Seguel Pérez, H., Seguel Pérez, R., Sepúlveda, M., Sinur, J., Soffer, P., Song, M., Sperduti, A., Stilo, G., Stoel, C., Swenson, K., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, E., Verdonk, M., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, T., Wen, L., Westergaard, M., Wynn, M. (2012). Process mining manifesto. In: *Business Process Management Workshops*, Springer, Berlin, Heidelberg, pp. 169–194.
- Arroyo-Figueroa, G., Sucar, L.E. (1999). A temporal Bayesian network for diagnosis and prediction. In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., pp. 13–20.
- Darwiche, A. (2008). Chapter 11. Bayesian Networks. In: van Harmelen, F., Lifschitz, V., Porter, B., (Eds.), *Handbook of Knowledge Representation*, Vol. 3. Elsevier, Amsterdam, pp. 467–509.
- van Dongen, B.F., Crooy, R.A., van der Aalst, W.M.P. (2008). Cycle time prediction: when will this case finally be finished? In: *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp. 319–336.
- van Dongen, B.F. (2015). *BPI Challenge 2012*. Eindhoven University of Technology. Dataset.
- van Dongen, B.F. (2015). *BPI Challenge 2015 Municipality 5*. Eindhoven University of Technology.
- Günther, C.W., Verbeek, E. (2014). *XES Standard 2.0*. Eindhoven.
- Kellner, M.I., Madachy, R.J., Ra, D.M., Raffo, D.M. (1999). Software process simulation modeling: why? what? how? *Journal of Systems and Software*, 46(2), 91–105.
- Martin, N., Depaire, B., Caris, A. (2015). The use of process mining in a business process simulation context: overview and challenges. In: *CIDM 2014 2014 IEEE Symposium on Computational Intelligence and Data Mining*, pp. 381–388.
- de Leoni, M., van der Aalst, W.M.P. (2013). Data-aware process mining: discovering decisions in processes using alignments. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, pp. 1454–1461.
- Liu, Y., Zhang, H., Li, C., Jiao, R.J. (2012). Workflow simulation for operational decision support using event graph through process mining. *Decision Support Systems*, 52(3), 685–697.
- Ping, J., Chen, Y., Chen, B., Howboldt, K. (2010). A robust statistical analysis approach for pollutant loadings in urban rivers. *Journal of Environmental Informatics*, 16, 35–42.
- Rogge-Solti, A., Mans, R.S., van der Aalst, W.M., Weske, M., (2013). Improving documentation by repairing event logs. In: *IFIP Working Conference on The Practice of Enterprise Modeling*. Springer, Berlin, Heidelberg, pp. 129–144.
- Rogge-Solti, A., Kasneci, G. (2014). Temporal anomaly detection in business processes. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 8659. Springer, pp. 234–249.
- Rozinat, A., van der Aalst, W.M.P. (2016). Decision mining in business processes. *Beta, Research School for Operations Management and Logistics*, 2006.

- Sarno, R., Dewandono, R.D., Ahmad, T., Naufal, M.F. (2015). Hybrid association rule learning and process mining for fraud detection. *IAENG International Journal of Computer Science*, 42(April), 59–72.
- Savickas, T., Vasilecas, O. (2014). Bayesian belief network application in process mining. In: *Proceedings of the 15th International Conference on Computer Systems and Technologies – CompSysTech '14*, Vol. 883. ACM, pp. 226–233.
- van der Spoel, S., Van Keulen, M., Amrit, C. (2012). Process prediction in noisy data sets: a case study in a dutch hospital. In: *International Symposium on Data-Driven Process Discovery and Analysis*. Springer, Berlin, Heidelberg, pp. 60–83.
- Sutrisnowati, R.A., Bae, H., Song, M. (2015). Bayesian network construction from event log for lateness analysis in port logistics. *Computers & Industrial Engineering*, 89, 53–66.
- Verbeek, H.M.W., Buijs, J.C., van Dongen, B.F., van der Aalst, W.M. (2010). Xes, xesame, and prom 6. In: *Forum at the Conference on Advanced Information Systems Engineering (CAiSE)*. Springer, Berlin, Heidelberg, pp. 60–75.
- Verenich, I., Dumas, M., La Rosa, M., Maggi, F.M., di Francescomarino, C. (2016). Minimizing overprocessing waste in business processes via predictive activity ordering. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 9694, Springer, pp. 186–202.

**T. Savickas** has a master's degree in information system engineering acquired in 2013 and currently pursues doctorate degree in Vilnius Gediminas Technical University in the area of informatics engineering. Current research is focused on process mining and its application in business process analysis and simulation.

**O. Vasilecas** is a full professor in Information System Department of the Vilnius Gediminas Technical University (VGTU) and a researcher in Vilnius University Institute of Mathematics and Informatics. He has many years of practical and research experience in information system development. Current research areas include business, information and software systems engineering; knowledge based information systems; business process modelling and simulation; systems theory and engineering, modern databases.