IEEE Second International Conference on Data Stream Mining & Processing
August 21-25, 2018, Lviv, Ukraine

# Software Architecture Design of the Real-Time Processes Monitoring Platform

Anatoliy Batyuk
*Dept. of Automated Control Systems*
*Institute of Computer Science*
*and Information Technologies*
*Lviv Polytechnic National University*
Lviv, Ukraine
abatyuk@gmail.com

Volodymyr Voityshyn
*Dept. of Automated Control Systems*
*Institute of Computer Science*
*and Information Technologies*
*Lviv Polytechnic National University*
Lviv, Ukraine
voytyshyn@gmail.com

Volodymyr Verhun
*Dept. of Automated Control Systems*
*Institute of Computer Science*
*and Information Technologies*
*Lviv Polytechnic National University*
Lviv, Ukraine
vverhun@gmail.com

***Abstract* — Understanding of how business processes are executed in real-life is vitally important for a company. Any process leaves a digital footprint that can be transformed into so-called event logs and analyzed with process mining techniques. A software platform with the purpose of near real-time processes monitoring is implemented. Design of the represented platform is based on the lambda architecture combining online and offline process mining algorithms with advanced analytics based on machine learning.**

***Keywords* — *process mining, event data, event logs, business process management, BPM, XES, lambda architecture***

## I. Introduction

Any event in the surrounding world is not by itself but belongs to some processes. IT systems, that have become ubiquities nowadays, help to automate vast amount of various kind of processes either in personal everyday lives or in huge enterprises. Most main stream software development practices did not consider process nature of the tasks they are devoted to automate "hardcoding" logic of workflow steps in source code. Consequently, it has been developed huge amount of software products which from one hand automate quite complex processes but from the other do not incapsulate any explicit definition of the implemented workflows losing the connection between the implementation and real-life. Nonetheless, within the industry field called business process management (or BPM) it has been developed a lot of practices to deal with workflows including their visual modeling (e.g. BPEL, BPMN, Petri nets etc.) and appropriate software implementations supplying wide range of products from powerful business process management systems (e.g. IBM BPM, Oracle BPM) to software components that can be embedded to a particular application (e.g. jBPM, Activiti, Camunda). Another trend that has had considerable influence on software industry is data science. The goal of applying data science techniques is to make software more intelligent obtaining insights from accumulated data. However, like classical software development practices most data science algorithms do not consider process nature of analyzed data.

Process mining is a discipline that fills in the gap between the mentioned above three industry domains. Significant contribution into creation of the academical core of process mining, its further promoting and encouraging industrial applications has been made in Eindhover Technical University (The Netherlands) under direction of professor Wil M.P. van der Aalst.

Guiding principle #1 declared in Process Mining Manifesto [1] states that event data (or event logs) is a primary data source for process mining. Like the entire software development industry process mining has faced with the challenge to deal with increasing amount of event data. In practice datasets are not static but are constantly fed with new data. This circumstance requires to handle data streams in near real-time mode and consequently puts process mining techniques into the position when it is necessary to deal with incomplete process instances.

Current paper is devoted to architecture design of the implemented by the authors software platform with the purpose of near real-time processes monitoring. The visualization and analytics modules of the represented system are impowered with advanced process mining and machine learning algorithms.

The rest of the paper is organized as follows: statement of the technical task is provided in section II; architecture significant requirements are specified in section III; the solution architecture design is described in section IV; in section V technical implementation details are provided; section VI contains the results of validation whether the designed system meets the performance requirements; the built-in analytics module is briefly described in section VII; short overview of the already existing process mining software products is provided in section VIII; conclusion remarks are in section IX.

## II. Task Statement

The software platform represented in current paper is general enough to be applied to wide range of practical tasks related to near real-time processes monitoring. However, it is obvious that it is hardly possible to implement a unified software product that can be applied to online process mining tasks in different business domains without modifications. That is why the described system is designed as an extensible platform with wide range of configuration capabilities so that it can be adopted to a particular application needs with minimal efforts, extended with specific features and integrated with other software systems.

As specified above one of the primary requirement is that the system takes event data from continuous data streams. It is assumed that data streams consist of items in XES format [2].

978-1-5386-2874-4/18/$31.00 ©2018 IEEE


Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

The received event data is stored “forever” in the system’s internal storage. Manual or automate data archiving outside the system is out of scope.

One of the main functional requirements is to support process flow chart visualization discovered by means of process mining techniques [3]. The visualized process model has to be updated in near real-time mode in accordance with receiving events from data streams. A similar system with real-time dashboards is described in [4]. Dealing with process model concept drift [5] is out of scope now and planned for the future.

The analytics module should include features defined in the online process mining framework [6]. The system supports prediction when a process instance completes, suggestions of next steps which are considered as optimal by the system and alerting if actual state of a process instance breaks predefined rules. The specified analytics features function in near real-time mode.

## III. Non-Functional Requirements

Non-functional requirements to technical architecture of the platform are specified in current section. The requirements listed below are a subset of the quality attributes defined in [7]. The taken software architecture building approach is based on the attribute-driven design method [8].

### A. *Performance*

Characteristics of the performance are defined by the requirement which states that the system has to process event data in near real-time mode (see section II). For current task the latency and throughput are highly important. Latency is the interval from the time of receiving of an event till the time when the end user sees the changes caused by the event (e.g. in the process model visual representation). In turn, throughput refers to number of events processed by the system during a certain period of time.

### B. *Scalability*

In current context scalability stands for ability to variate latency and throughput of the system according the changes of number of received events. From practice standpoint it is necessary to decide whether the system is intended to deal with BigData or “small” data. The reason of necessity to make such decision on the architecture design phase is that the implementation and maintenance cost of a BigData solution is much higher in comparison with the similar solution for “small” data. So, the decision is: the described platform is not intended to deal with BigData. Implementation of the platform modification with BigData support is planned for the future.

### C. *Interoperability*

The represented platform should be so-called cloud agnostic which means that it can be deployed at clouds of different providers (e.g. Azure, AWS, Google Cloud Platform) or use on-premises infrastructure of a customer.

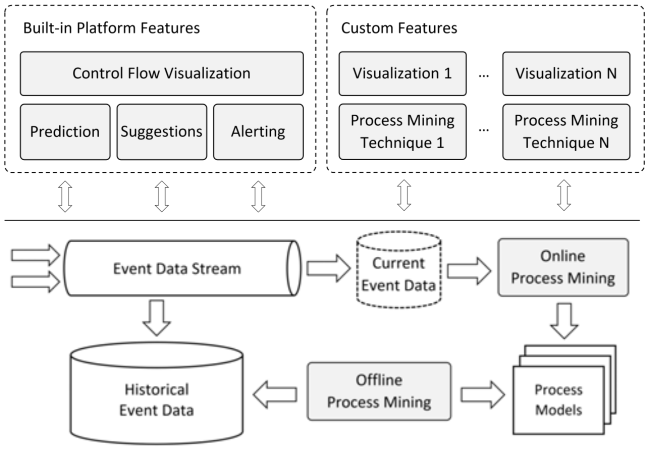


Fig. 1. Architecture concept of the platform

### D. *Extensibility*

As mentioned above the designed software is not a product ready to use without any modifications but it is an extensible platform with predefined architecture and implemented basic built-in functionality. This means that the platform is going to be extended with specific features necessary for a particular customer (e.g. adding an anomaly detection module with appropriate visualization and alerting functionalities).

### E. *Configurability*

In current context configurability means that the platform is flexible enough to be adopted to needs of specific applications without changing the source code. For example, rules for the alerting feature can be defined considering specific of a monitored process.

## IV. Architecture Concept

High level architecture design of the platform is represented in current section. The concept meets the functional and non-functional requirements specified in sections II and III respectively.

As it is already defined the major requirement is near real-time processing of event data. There are two architecture patterns that address such a task: (a) lambda architecture [9] and (b) kappa architecture [10]. The core idea of the first pattern is that data processing is split into two layers: (a) speed and (b) batch. The speed layer is accountable for handling newly received data in near real-time mode whilst the batch layer deals with accumulated historical data. Kappa architecture is derived from the lambda. The main difference is that batch layer is omitted in kappa architecture simplifying the implementation of the pattern. Hence, kappa architecture is not applicable for the tasks that need batch processing.

The designed platform definitely requires batch layer since process mining techniques (e.g. the implemented process discovery algorithm, see section VII) use historical event data. This is the reason behind choosing lambda architecture as a primary design pattern.



Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Architecture concept of the platform is depicted on Fig. 1. Parts of the lambda architecture are adopted to the purposes of current task. Offline and online process mining components represent batch and speed layers of the lambda architecture respectively. Another important aspect of the designed architecture is that events of incomplete process instances are kept in a separate storage ("Current Event Data" on Fig. 1) so that relevant data is accessed with minimal latency by the online process mining techniques.

Since lambda architecture is mostly applied to BigData tasks it possible to modify the platform to deal with BigData without changes in its conceptual design. A similar lambda architecture-based BigData system is represented in [11].

## V. Technical Solution

### A. Components Model

The components model of the platform (Fig. 2) is the next step of the design process after architecture concept. The model is designed flowing the decision that the platform is not intended to deal with BigData (see section III). Event data stream is supposed to be a message queue. One of the benefits of applying this pattern is that it ensures reliable message delivery. The online process mining algorithms are handlers that listen to the queue for new messages. Received event data are persisted in the database. The main requirement to this database is to be optimized for time series data. Additionally, the most recent event data (including events of incomplete process instances) is cached in an in-memory database which significantly minimizes latency of processing and visualizing this data.Results of execution of the integrated process mining and machine learning algorithms are stored in a NoSQL database. The reason of this decision is that such kind of databases supports unstructured data and are fast on reading. Presentation layer of the platform is implemented as a pluggable single page web application. The server side is composed with microservices and exposes a RESTful API for the frontend.

### B. Technology Stack

The service side technology stack is mostly Java-based. One of the reason behind this is that most process mining algorithms are implemented with Java [12]. Another reason is high quality of cross-platform support provided by Java. The frontend side is built with HTML 5 and CSS 3 using the latest JavaScript standard – ECMAScript 6. In particular, process model visualization is implemented with SVG and D3JS. The business rules engine component is intended to address the configurability requirements (see section III). The DMN engine from the Camunda platform is used for its implementation.

## VI. Performance Measurement of the Message Queue

As a messaging technology RabbitMQ v.3.7.4 was chosen. The reason of taking this particular message broker is that it is an open source mature solution with scalability and high-availability support. Additionally, RabbitMQ is not intended to work with BigData which means that it consumes not so much resources as a similar BigData solution (e.g. Apache Kafka). Since the message queue is a single point of failure of the platform a load test was performed to measure its capabilities. The performance test was executed on the following environment:

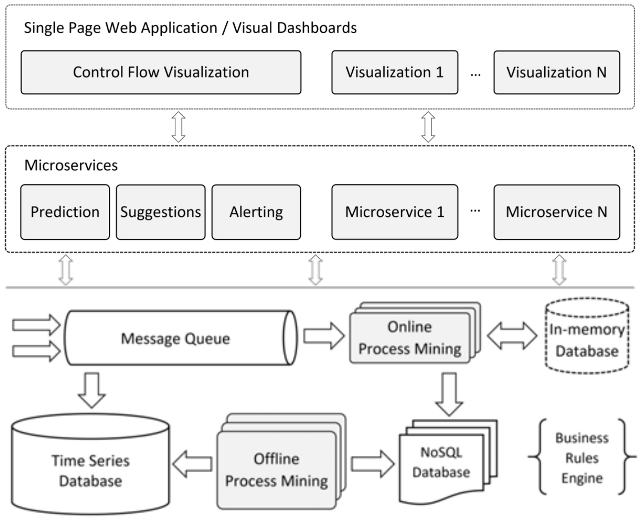


Fig. 2. Components model of the platform

RabbitMQ v.3.7.4 (one producer and one consumer), Linux CentOS 7, 12 GB RAM, Intel Core i7 – 4500U 1.8. GHz. The RabbitMQ management plugin was used to collect the metrics. The results of the test (Table 1) proves that RabbitMQ v.3.7.4 is stable and supports acceptable level of performance.

TABLE I. Performance Measurement of RabbitMQ v.3.7.4

| Message Size, bytes | Measured RabbitMQ Metrics | | |
|---|---|---|---|
| | *Average Latency, ms* | *Average Message Publish Rate, msg/s* | *Average Message Delivery Rate, mgs/s* |
| 100 | 3800 | 55100 | 20015 |
| 300 | 4100 | 49800 | 1893 |
| 100000 | 9055 | 4400 | 980 |

## VII. Built-in Analytics Features

As specified in section II flow chart visualization based on a process discovery algorithm is a built-in feature of the platform. One of the oldest and most well-known process discovery technique is the alpha algorithm [6]. It takes event data and produces a Petri net. However, the alpha algorithm is not the best choice for real-life processes with a lot of transitions especially if the results are for business domain experts who are not process mining professionals. The Fuzzy Miner algorithm [13] is more suitable for such cases. The efficiency of this algorithm has been proved by experience of well-known process mining software like Disco [14] and Celonis [15]. Additionally, comparison of the Fuzzy Miner with some other process discovery algorithms is done in [16]. Considering the facts above the offline process discovery implemented within the scope of the platform is based on the Fuzzy Miner algorithm. In turn, the online process discovery implementation follows the ideas outlined in [17]. The built-in prediction analytics is used to forecast completion time of a process instance. The feature is developed upon a combined time series forecasting information technology based on fuzzy experts' evaluation [18] and analysis of dynamic processes [19]. The suggestions feature recommends an optimal process flows and suits the human behavior (which is important if people are involved into monitored processes). Another requirement to this



Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

feature is the ability of incremental learning from the event data stream. To meet these requirements an implementation of a neuro-fuzzy model [20, 21] is integrated.

## VIII. Overview of Existing Process Mining Software

Process mining is a relatively new academic discipline and its software implementations began to gain popularity in the market not so long ago. The oldest process mining tool is ProM [12] which is an open source Java-based framework. Scientists are the target audience of this application. Disco [14] is a commercial process mining tool. It includes the most useful algorithms. This product is used by experts from business domains who are not process mining professionals. Another process mining product is Celonis [15]. It is a fast-growing German startup. Target consumers' audience of Celonis is medium and big enterprises. This product supports offline and online process mining and has connectors to other software, for example SAP [22]. The main difference between the software system developed by the authors and Celoins is that Celonis is a product with a set of features delivered to all its customers whilst the represented platform is a ground for custom development with predefined architecture and initial set of built-in features.

## IX. Conclusions

The implemented platform has been integrated with an energy efficiency management system [23] as an extension with the purpose to monitor real-life processes on the operator control level. The visualization feature has provided visibility on the processes executed within the energy management system and generated alerts once a process instance breaks the predefined restriction rules. From practice standpoint it is necessary to include the following algorithms to the set of built-in features: (a) conformance checking [6] and (b) handling process concept drifts [5]. Another way of the platform's evolution is to design and implement a version with the purpose to support Big Data.

## References


[1] W. M. P. van der Aalst, et al., "Process mining manifesto," in Business Process Management Workshops. BPM 2011. Lecture Notes in Business Information Processing, Berlin, German, vol. 99, pp. 169-194, 2011.

[2] IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams, IEEE Std 1849-2016, 2016.

[3] W. M. P. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: discovering process models from event logs," in IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128-1142, 2004.

[4] A. Batyuk and V. Voityshyn, "Business Processes Monitoring by Means of Real-Time Visual Dashboards," 6th International Academic Conference on Information, Communication, Society 2017, Lviv, Ukraine, pp. 204-205, 2017.

[5] R. P. Jagadeesh Chandra Bose, Wil M. P. van der Aalst, I. Žliobaitė, and M. Pechenizkiy, "Handling Concept Drift in Process Mining," in Advanced Information Systems Engineering. CAiSE 2011. Lecture Notes in Computer Science, London, UK, vol. 6741, pp. 391-405, 2011.

[6] W.M.P. van der Aalst, Process mining: data science in action. Berlin Heidelberg: Springer-Verlag, 2016.

[7] M. Barbacci, M. H. Klein, T. H. Longstaff, and C. B. Weinstock, Quality Attributes. SEI at Carnegie Mellon University, Pittsburgh, Pennsylvania, Rep. CMU/SEI-95-TR-021, 1995.

[8] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice, 3rd ed. Addison-Wesley Professional, 2012.

[9] "Lambda Architecture", [Online]. Available: http://lambda-architecture.net/. [Accessed: 25 Mar 2018].

[10] "kappa-architecture.com", [Online]. Available: http://milinda.pathirage.org/kappa-architecture.com/. [Accessed: 25 Mar 2018].

[11] A. Batyuk and V. Voityshyn, "Apache storm based on topology for real-time processing of streaming data from social networks," 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP 2016), Lviv, Ukraine, pp. 345-349, 2016.

[12] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "ProM 6: the process mining toolkit," in Proceedings of the Business Process Management 2010 Demonstration Track, Hoboken NJ, USA, vol 615, pp. 34-39, 2010.

[13] W.G. Christian, W.M.P. van der Aalst, "Fuzzy Mining – Adaptive Process Simplification," in Proceedings of the 5th International Conference on Business Process Management (BPM 2007), Brisbane, Australia, vol 4714, pp. 328-343, 2007.

[14] Ch. W. Günther, and A. Rozinat, "Disco: Discover Your Processes," in Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012), Tallinn, Estonia, vol 940, pp. 40-44, 2012.

[15] F. Veit, J. Geyer-Klingeberg, J. Madrzak, M. Haug, and J. Thomson, "The Proactive Insights Engine: Process Mining meets Machine Learning and Artificial Intelligence," in 15th International Conference on Business Process Management (BPM 2017). BPM Demo Track and BPM Dissertation Award, Barcelona, Spain, vol 1920, pages 5, 2017.

[16] A. Rozinat, "ProM Tips - Which Mining Algorithm Should You Use?", 2010. [Online]. Available: https://fluxicon.com/blog/2010/10/prom-tips-mining-algorithm/. [Accessed: 25 Mar 2018].

[17] A. Burattin, "Process Mining for Stream Data Sources,". in Process Mining Techniques in Business Environments. Lecture Notes in Business Information Processing, Cham, Springer, vol 207, pp. 177-204, 2015.

[18] O. Mulesa, F. Geche, A. Batyuk, and V. Buchok, "Development of Combined Information Technology for Time Series Prediction," in Advances in Intelligent Systems and Computing II (CSIT 2017), Lviv, Ukraine, vol 689, pp. 361-373, 2017.

[19] P. Bydyuk, A. Gozhyj, I. Kalinina, and V. Gozhyj, "Analysis of unsertainty types for model building and forecasting dynamic processes," in Advances in Intelligent Systems and Computing II (CSIT 2017), Lviv, Ukraine, vol 689, pp.66-82, 2017.

[20] Ye. Bodyanskiy, I. P. Pliss, D. Peleshko, Yu. Rashkevych, and O. Vynokurova, "Hybrid Generalized Additive Wavelet-Neuro-Fuzzy-System and its Adaptive Learning," in The Eleventh International Conference on Dependability and Complex Systems DepCoS-RELCOMEX., Brunow, Poland, pp. 51-61, 2016.

[21] Ye. Bodyanskiy, G. Setlak, D. Peleshko, and O. Vynokurova, "Hybrid Generalized Additive Neuro-Fuzzy System and its Adaptive Learning Algorithms," 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw, Poland, pp. 328-333, 2015.

[22] "Showcase: SAP Process Mining by Celonis", [Online]. Available: https://www.sap.com/developer/showcases/process-mining-by-celonis.html. [Accessed: 25 Mar 2018].

[23] T. Teslyuk, I. Tsmots, V. Teslyuk, M. Medykovskyy, and Y. Opotyak, "Architecture and Models for System-Level Computer-Aided Design of the Management System of Energy Efficiency of Technological Processes at the Enterprise in Advances," in Intelligent Systems and Computing II. CSIT 2017, Lviv, Ukraine, vol 689, pp. 538-557, 2017.




Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua