An Approach to Developing Ontology-Based Tools for Event Series Analysis

Anton Platunov[1][a], Lyudmila Lyadova[1][b], Nada Matta[2][c], Viacheslav Lanin[3][d] and Elena Zamyatina[1][e]

[1]*Department of Information Technologies in Business, HSE University, Perm, Russian Federation*
[2]*University of Technology of Troyes, Troyes, France*
[3]*Perm State University, Perm, Russian Federation*

Keywords: Event Series, Data Analysis, Event Attributes, Event Logs, Process Mining, Multifaceted Ontology, Rules Design, Rules Interpreter, Event Log Generation.

Abstract: Existing process mining methods allow to investigate processes in different domains. Besides mandatory event attributes like as identifier, activity, and timestamp, additional event attributes can be present in data sources. The analysing dynamics of changing the values of additional attributes allows to get important information on the system. The applications must be developed by programmers with programming languages to implement new methods of analysis. An approach to develop tools based on the use of algorithm designers and expression builders like those included in MS Office applications is proposed in the paper. Their use does not require programming skills. The implementation of the approach is based on a multifaceted ontology, including descriptions of the rules for developing functions, as well as a description of functions for generating and analysing event logs in accordance with these rules. The user interface for developing rules and the algorithm for their interpreting are implemented in the research prototype of the application.

1 INTRODUCTION

Typically, Process Mining methods are used to analyse the business processes of enterprises, where it is possible to obtain structured information from user workplaces to generate event logs. However, at present, the scope of these methods has expanded significantly: they use to solve the problems of analysing social networks, analyse processes in the healthcare process organization, etc. New applications address the challenges of extracting data from unstructured or semi structured heterogeneous information sources to generate event logs. Another feature of these domains is that additional attributes specific to the appropriate domains are defined for events. Analysing the values of these event attributes allows to investigate the dynamics of the behaviour of complex systems and to evaluate developing a set of process in them.

New analysis methods are implemented through the development of programs, in particular plugins for the ProM system. Researchers must be proficient in the programming languages (Java, Python, etc.) which used for the applications developing, or must involve professional programmers, specialists in information technologies in the methods designing. This approach to application development also complicates the implementation of methods, slows down research.

The task of developing tools that implement the low-code principle becomes relevant. An approach to creating a knowledge-driven analytical platform based on domain specific modelling (DSM) is proposed. Visual domain specific languages (DSL) to develop algorithms and research scenarios can be used. These languages correspond domain specificity, reflect object characteristics and domain limitations. Language toolkits are included in the

[a] https://orcid.org/0009-0008-7500-6278
[b] https://orcid.org/0000-0001-5643-747X
[c] https://orcid.org/0000-0001-8729-3624
[d] https://orcid.org/0000-0002-0650-2314
[e] https://orcid.org/0000-0001-8123-5984

Platunov, A., Lyadova, L., Matta, N., Lanin, V. and Zamyatina, E.
An Approach to Developing Ontology-Based Tools for Event Series Analysis.
DOI: 10.5220/0012259100003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 2: KEOD*, pages 323-330
ISBN: 978-989-758-671-2; ISSN: 2184-3228

platform software. The using DSLs is available to researchers, specialists in specific subject areas who do not know professional software development tools. The advantages of this approach are flexibility, the ability to dynamically configure to solving new tasks and to changing requirements. The problem of creating such language toolkits is the need to implement model editors and model transformation tools or model interpreters.

An alternative approach is to develop tools based on the use of algorithm designers and expression builders like those included in MS Office applications. These tools are available for users who have the skills to work with the appropriate MS Office tools, where users can develop own complex algorithms for processing and analysing data using standard application functions, and the existing set of functions can be expanded by users (like the plugins development).

The challenge of the project is creating a research prototype of tools based on the low-code principles for analysing event series with attributes of various types.

The tasks of the approach implementation:

1. Developing a language to describe user functions and research scenarios.
2. Designing representation of functions and research scenarios, defining data structures for implementing them.
3. Developing a parsing and interpretation algorithms for the functions and research scenarios.
4. Designing a user interface for function builder.

This paper focuses on the solving these tasks.

2 RELATED WORKS

Let's review approaches to solving the most interesting tasks in development of process analysis tools considering the values of event attributes, the dynamics and behaviour of these attributes.

The papers (Cremerius, 2021; Cremerius, 2022; Cremerius, 2023) consider the issues of analysing processes with methods taking into account the values of attributes specific to domain and their behaviour. The authors note that besides mandatory event attributes like as identifier, activity, and timestamp, additional event attributes can be present in data sources. For example, researchers can analyse additional attributes of events specific for their domain such as human resources, costs, and laboratory values. The analysis of attribute values is especially important in healthcare processes, where a huge amount of data is generated (such as vital signs or laboratory measurements representing a patient's wellbeing). Analysis of developing these measurements depending on which treatment activities were conducted can help to evaluate different treatment paths, to select one that might result in better patient wellbeing than others. Process mining traditionally focusses on the workflow control aspects. However, processes not only include activities and their orderings, but also the data generated and manipulated during process executions. Any process activity generates data, but this information does not play the role in process mining that it deserves. Data-enhanced process models are described by authors. They give formal definitions of event, event log, trace, process model and process variants. Event attribute selection and aggregation are defined. Authors focus on healthcare processes. Nevertheless, the developed methods can be applied to different domains, where event attributes are available.

In the papers (Bano, 2020; Bano, 2021) authors also argues that despite event logs capturing behavioural information (most discovery algorithms focus on process control-flow), they are a rich source of domain specific data useful for analysis data-flow perspective. Usually, this data is not represented explicitly in process models, but it provides valuable contextual information. A semi-automatic approach to discover a data model that complements traditional process mining techniques with domain specific information is proposed.

Another important problem is the processing of uncertainty of the event attribute values (Pegoraro, 2019). In the paper the setting of uncertain event logs is analysed: quantified uncertainty in the logs is recorded together with the corresponding data. Authors define a taxonomy of uncertain event logs and models and examine the challenges that uncertainty poses on process discovery and conformance checking. They show how upper and lower bounds for conformance can be obtained aligning an uncertain trace onto a regular process model.

The paper (Kampik, 2022) presents the results of survey of challenges and perspectives around event log generation in process mining. The responses of an industry experts analysed by authors indicate that particularly relevant challenges exist in data integration and quality. Authors argue that process mining can benefit from a systematic integration traditional methods and wide-spread business intelligence approaches.

The problems of generating event logs

(Mitsyuk, 2017) and their preprocessing (Marin-Castro, 2021) are devoted to the articles that consider tasks of forming event logs based on data obtained from various sources (from data bases (Calvanese, 2016) and network messages of trading systems (Carrasquel, 2021), social networks (Lanin, 2021; Peña-Araya, 2015; Ritter, 2012) and mass media (Abrosimova, 2018; Shalyaeva, 2016; Shalyaeva, 2017), etc.).

However, the implementation of the proposed methods requires the involvement of professional developers who have skills of using software development tools and programming languages (Java, Python, etc.). It complicates development and slows down research with using Process Mining tools.

The authors of papers (Lyadova, 2022; Zayakin, 2023) propose an approach to developing an ontology-driven analytical platform that includes toolkits for creating visual domain specific languages (DSL) that are customised to solve tasks in research domains. Created languages reduce the requirements for knowledge and skills for researchers in the field of software development. The use of DSL minimizes errors because domain restrictions are included in DSL, language semantics complies with characteristics of domain objects and corresponds to existing domain restrictions.

An approach to solving the problem of the creating DSL through automating DSL metamodels generation, customising languages based on the multifaceted ontology, is proposed in the paper (Kulagin, 2022).

However, this approach also requires developers to have skills in knowledge engineering, in formal languages and grammars that are necessary for creating metamodels of basic languages, developing domain ontologies. An alternative approach to developing ontology-driven tools for generating event logs and analysing event series that do not require programming skills is proposed in this paper.

3 ONTOLOGY-BASED TOOLS FOR THE EVENT SERIES ANALYSIS

The requirements for the developed tools are determined based on the descriptions given in the paper (Zayakin, 2022), which provides event-time series and event series definitions. Examples of generating event logs based on data, extracted from various sources, according to rules described by users, as well as examples of rules for analysing processes, based on event logs containing additional attributes, are included in paper.

The designed tools shall meet the following functional requirements:

1. Determining the data sources that will be used to create event series containing additional attributes (numerical indicators).
2. Describing the rules for generating event logs based on numerical attributes processing.
3. Describing the rules for calculating numerical event characteristics based on data, extracted from various sources, and parameters, set by users, and aggregating these characteristics with events.
4. Interpreting rules defined by users.

The event log structure must meet the XES format used in ProM. Rules described by users should be stored in the ontology.

3.1 Description of Functions for Generation and Analysis of Event Series

To implement functions for generation and analysis of event series, a formal language for describing rules should be developed.

The description of the rules (functions for generating and analysing events) is based on the definition of the superposition of functions.

The algorithm for solving any task can be represented as a function that transforms the input data into a result. The results of function calculation can be used as input data for calculating other functions. Thus, any complex algorithms can be implemented as superposition of functions. It is needed to define set of basic functions to implement this approach.

This definition allows to simplify the grammar of the language for describing rules and to develop a universal user interface for developing and interpreting rules.

When developing a language, two types of rules should be distinguished, which may have structural differences, therefore, each of them should be described separately:

1. Rules for calculating events for generating event logs. The user defines rules by which the types of events are determined based on the values of input data (Table 1). The result of applying these rules is event log containing events with calculated event types and timestamps (Table 2).
2. Rules for processing event logs extended with numeric attributes. The user defines rules that allow analysing behaviour of attributes associated with

events in the event log. The results of the analysis according to such rules are included in the process model (calculated characteristics are associated with events).

Table 1: Example of input data (COVID-19 incidence).

Region / Town	Confirmed Cases	Recovered	Deaths	Date
St. Petersburg	511 389	1 931	95	22.07.2021
Moscow	1 477 871	7 587	102	22.07.2021
St. Petersburg	509 446	1 928	103	21.07.2021
Moscow	1 473 584	8 006	104	21.07.2021
...	...	...	...	...
St. Petersburg	245 870	3 058	75	01.01.2021
Moscow	815 676	6 334	71	01.01.2021

Table 2: Examples of events determined based on incidence rates.

Region / Town	Confirmed Cases	Event Type	Date
St. Petersburg	511 389	A dramatic increase of the confirmed cases number	22.07.2021
Moscow	1 477 871	A dramatic increase of the confirmed cases number	22.07.2021
St. Petersburg	509 446	A mid-range increase of the confirmed cases number	21.07.2021
Moscow	1 473 584	A dramatic increase of the confirmed cases number	21.07.2021
...	...		...
St. Petersburg	245 870	A dramatic increase of the confirmed cases number	01.01.2021
Moscow	815 676	A dramatic increase of the confirmed cases number	01.01.2021

The requirements for the language intended for description of rules for generating and analysing event logs are determined by the requirements for the user interface, which should ensure the simplicity, availability of the developed tools for non-programmer users: rules should be developed in designers that have a unified interface close to the interface of designers (expression builders) in Microsoft Office applications.

The rule definitions can include built-in operations and functions implemented in the ontology. The operands of these function can be numbers and text strings, arrays of values, time series, event logs, the structures of which are defined in the ontology, as well as the results of calculating functions defined in the system.

Thus, rules (functions) described by users to generate event logs using data obtained from specified sources and to preprocess event logs with additional attributes are defined as a superposition of functions:

$$f = \sigma(f_0, f_1, ..., f_n),$$

where the function f is defined as a superposition of the functions $f_0, f_1, ..., f_n$; and functions $f_1, ..., f_n$ have their own parameter sets for calculation: $f_1(x^1_1, x^1_2, ..., x^1_{k1})$, …, $f_n(x^n_1, x^n_2, ..., x^n_{kn})$, and the result is calculated as a function

$$f(x^1_1, x^1_2, ..., x^1_{k1}, \ldots, x^n_1, x^n_2, ..., x^n_{kn}) = \\ = f_0(f_1(x^1_1, x^1_2, ..., x^1_{k1}), \ldots, f_n(x^n_1, x^n_2, ..., x^n_{kn})).$$

All the functions are partial, that is, not everywhere defined – there can be such combinations of argument values for which the values of functions $f(x^1_1, x^1_2, ..., x^1_{k1}, \ldots, x^n_1, x^n_2, ..., x^n_{kn})$ do not exist: at least one of the values $f_1(x^1_1, x^1_2, ..., x^1_{k1})$, …, $f_n(x^n_1, x^n_2, ..., x^n_{kn})$ does not exist; or these values exist and are equal to $b_1, ..., b_n$, but does not exist value of $f_0(b_1,..., b_n)$.

Thus, the method of describing the algorithm for calculating a function, the type of rules is defined.

To describe the rules for generating event logs and their preprocessing, a language has been developed. The formal language grammar includes more than 50 syntactic rules that define non-terminal symbols of the language. The language defines, along with standard types for programming languages, data types for describing time stamps, time series, events with additional attributes and event logs that can be processed or generated using functions, etc. The target grammar symbol is "Function".

The designed functions include operations on values of standard types (arithmetic operations, operations on text strings, comparison operations, etc.) and built-in functions (*MAX*, *MIN*, *AVG*, *SUM*, etc.). To simplify the interface of the expression builder, operations *AND*, *OR*, *XOR* are implemented as functions by analogy with the corresponding functions of MS Excel (this simplifies the considering operation priorities when forming expressions and parsing the user-defined function, when interpreting (calculating) the built function). The expression designer also allows to use the functions *SWITCH* or *IFTHEN* (to determine alternative calculations depending on the specified conditions), *FOREACH* (to process data in a loop).

The grammar described is of type $LL(1)$.

Based on the described rules, algorithms for parsing and interpreting functions described by users have been developed. When analysing the descriptions of functions, the algorithm of left-recursive descent is implemented.

The grammar of the rule description language is developed using Backus-Naur forms. However, a representation in the form of Wirth diagrams is more suitable for including rules in the ontology.

3.2 Ontology of Rules (Functions)

The developed grammar of the language for function description (rules for generating and analysing event logs) includes more than fifty syntactic rules. Grammar rules are included in the ontology.

Ontology of rules contains descriptions of two type functions (rules defined by users). Fragment of the main classes is shown in Figure 1 and Figure 2.

An example of a rule description for calculating event types when generating an event log is shown in Figure 3.

When analysing the descriptions of functions developed by users according to the rules included in the ontology, a representation of algorithms for calculating functions in the form of a tree is built (Figure 4). The interpretation (function calculation) is implemented through traversing the constructed tree.

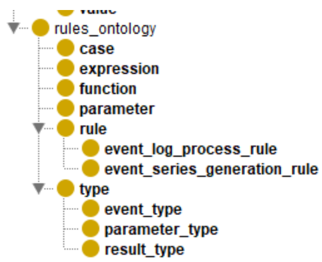

Figure 1: Fragment of the class hierarchy of rules.

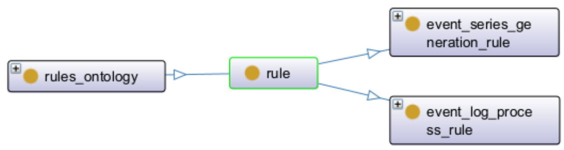

Figure 2: Classification of rules as ontograph fragment.

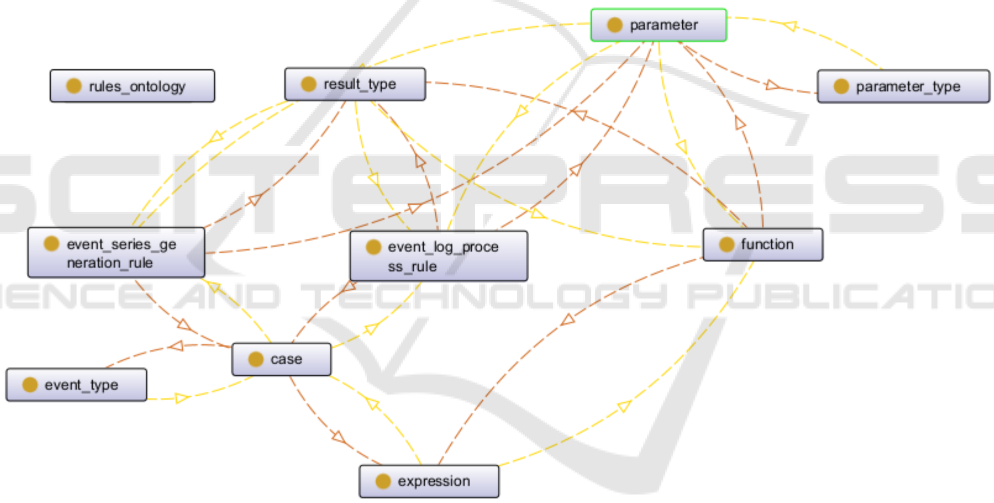

Figure 3: An example of a rule description as ontograph.

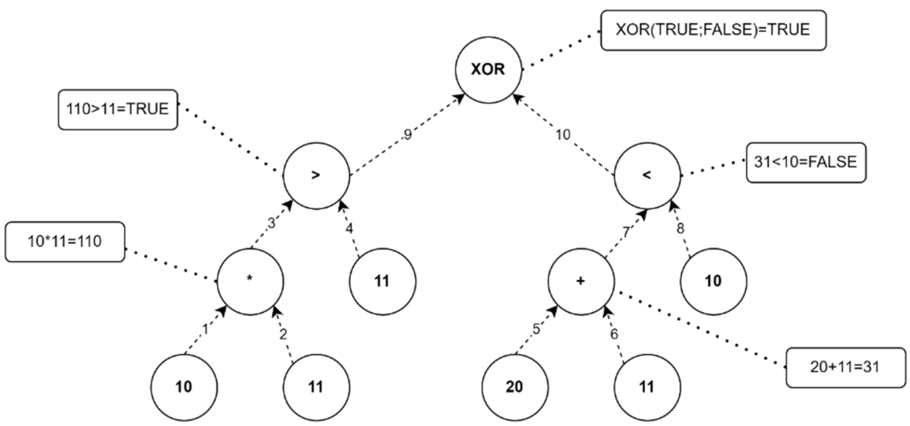

Figure 4: An example of the algorithm for calculating function *XOR* represented as a tree.

3.3 Implementing Research Prototype

The research prototype is developed in C#. A user interface is implemented to develop rules (functions) for generating and analysing event logs.

The research prototype structure is shown in Figure 5. The modules of the application implement the functions described above.

The user form of the event calculation rule builder is presented in Figure 6. Using the drop-down list, the user determines the rule type (event calculation rule or event log processing rule). User describes the list of formal parameters where name for each parameter and type are defined (for calculating events it is *TimeSeries* type; for the event log it is *EventLog* type). User can add formal time series constraint parameters to the timestamp using the *Add Boundaries* buttons. Also, the user can write a comment that will describe the rule. Alternative calculation options can be specified for different conditions. The form controls allow to define all elements of functions defined in grammar rules.

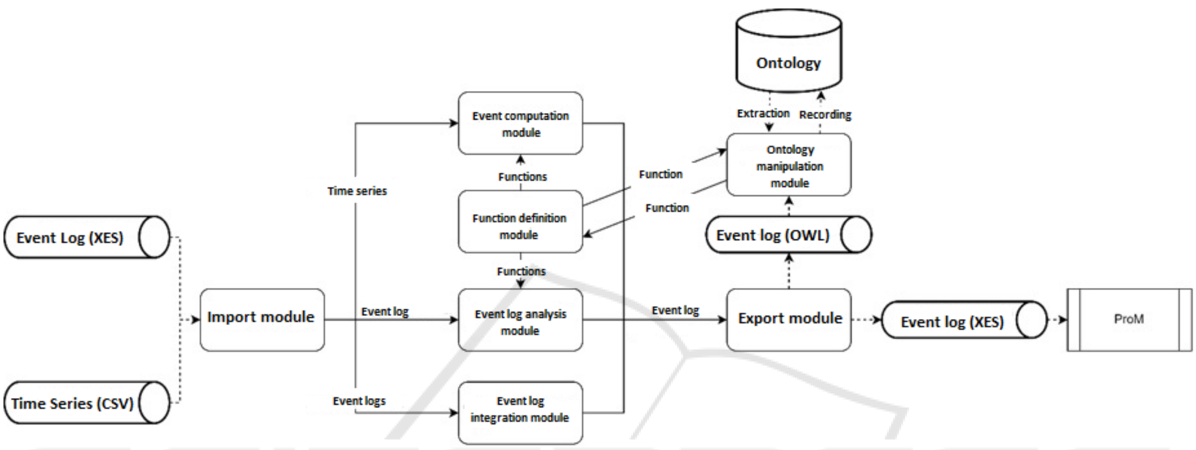

Figure 5: Structure of the research prototype.

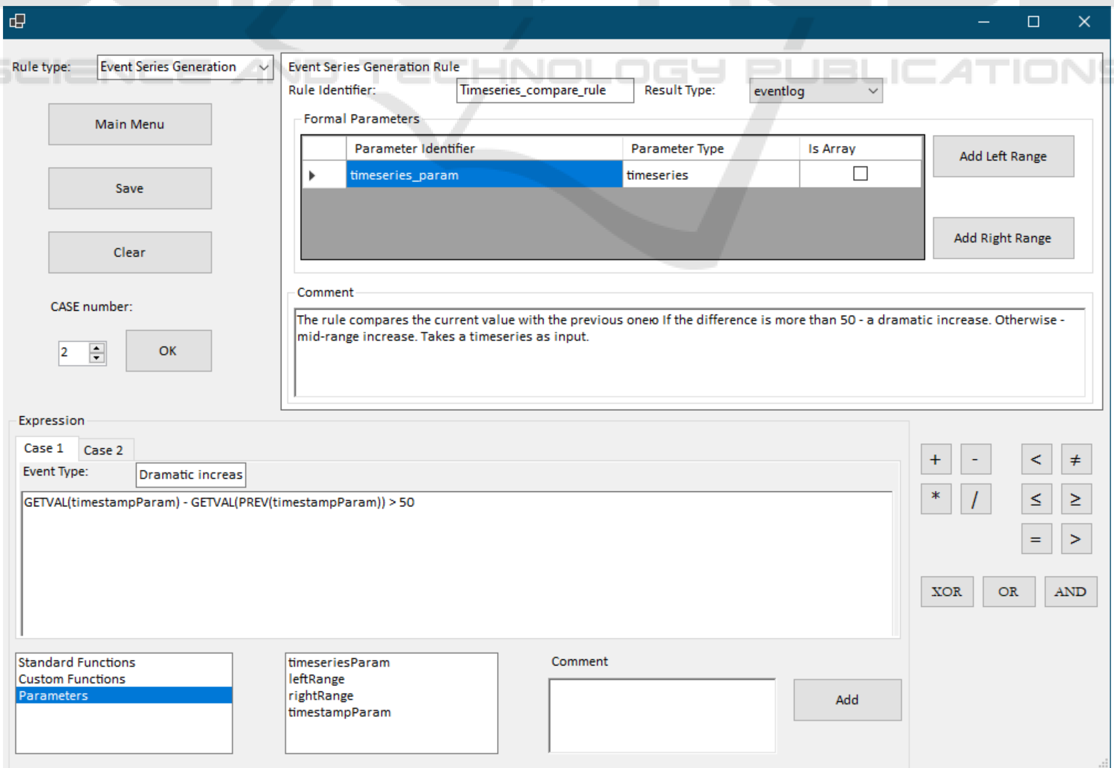

Figure 6: The user form of the event calculation rule builder.

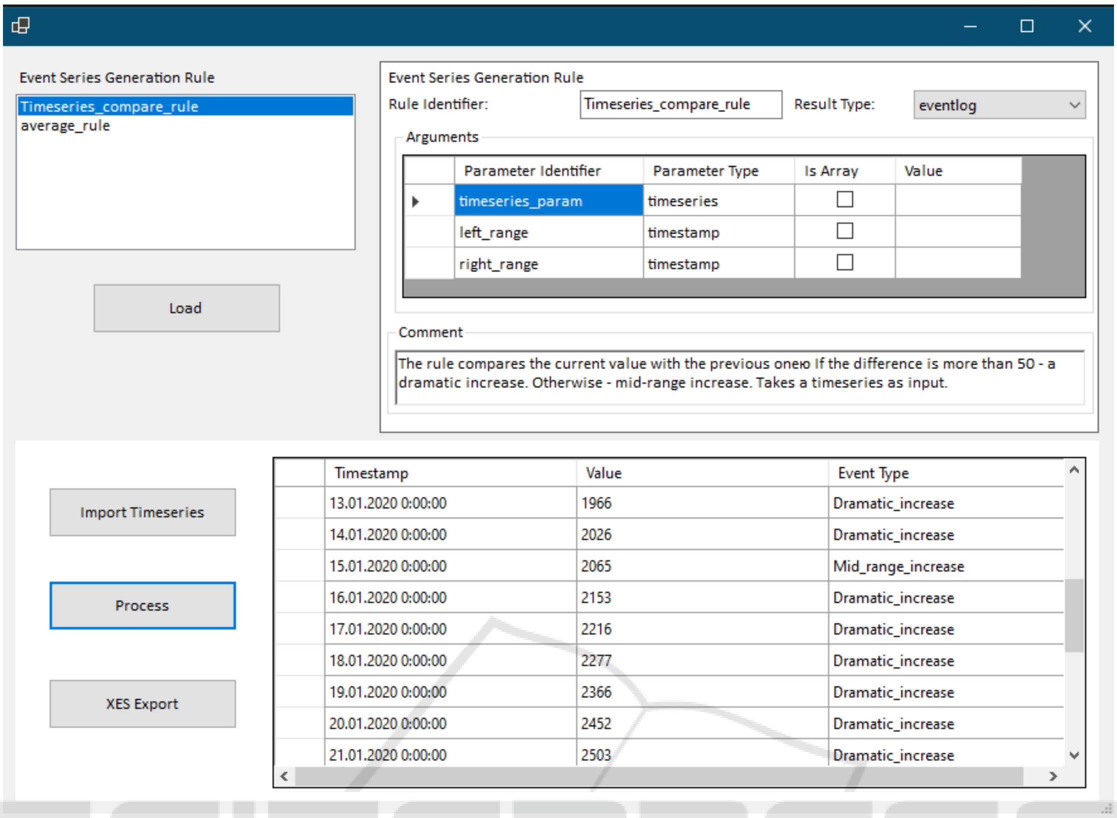

Figure 7: The result of generating an event log.

The result of generating an event log according to the rules specified by user is presented in Figure 7.

The results of generating and processing event logs according to the rules specified by users can be uploaded to files in XES format (Figure 8) and exported for analysis to ProM.

```
<trace>
  <string key="concept:name" value="Case3.0" />
  <event>
    <string key="concept:name" value="UNDEFINED" />
    <date key="time:timestamp" value="2020-01-01T00:00:00.000" />
    <float key="float" value="1230" />
    <string key="concept:eventType" value="Mid_range_increase" />
  </event>
  <event>
    <string key="concept:name" value="UNDEFINED" />
    <date key="time:timestamp" value="2020-01-02T00:00:00.000" />
    <float key="float" value="1292" />
    <string key="concept:eventType" value="Dramatic_increase" />
  </event>
```

Figure 8: The result of generating an event log (fragment).

4 CONCLUSIONS

The main result is a research prototype of tools for generating and analysing event logs. Experiments have shown the practical significance of the proposed approach. The developed tools have demonstrated the universality of the created formal language, flexibility, and accessibility of the software for users. The developed tools can be integrated with process analysis tools (for example, implemented as a ProM plugins). The set of built-in functions that are used at interpreting rules can be extended. As experiments, the methods described in the related works (Pegoraro, 2019; Kampik, 2022; Mitsyuk, 2017; Marin-Castro, 2021; etc.) can be implemented with created software.

REFERENCES

Abrosimova, P., Shalyaeva, I., Lyadova, L. (2018). The Ontology-Based Event Mining Tools for Monitoring Global Processes. In: *Proceedings of the IEEE 12th International Conference on Application of Information and Communication Technologies (AICT 2018)*. Almaty, 2018. Pp. 108-113.

Bano, D., Weske, M. (2020). Discovering Data Models from Event Logs. In: *Proc. International Conference on Conceptual Modeling*. Vienna, Austria, 2020. DOI: 10.1007/978-3-030-62522-1_5.

Bano, D., Zerbato, F., Weber, B. Weske, M. (2021). Enhancing Discovered Process Models with Data

Object Lifecycles. In: *Proc. IEEE 25th International Enterprise Distributed Object Computing Conference (EDOC)*. Gold Coast, Australia, 2021, pp. 124–133. DOI: 10.1109/EDOC52215.2021.00023.

Calvanese, D., Montali, M., Syamsiyah, A., van der Aalst W.M.P. (2016). Ontology-Driven Extraction of Event Logs from Relational Databases. In: *Business Process Management Workshops. BPM 2016. Lecture Notes in Business Information Processing*, vol 256. Springer, Cham.

Carrasquel, J.C., Chuburov, S.A., Lomazova, I.A. (2021). Preprocessing Network Messages of Trading Systems into Event Logs for Process Mining. In: *Tools and Methods of Program Analysis. TMPA 2019. Communications in Computer and Information Science*, vol 1288. Springer, Cham. Pp. 88-100.

Cremerius, J., Weske, M. (2021). Data-Enhanced Process Models in Process Mining. Preprint. DOI: 10.48550/arXiv.2107.00565.

Cremerius, J., Weske, M. (2022). Change Detection in Dynamic Event Attributes. In: *Business Process Management Forum. BPM 2022. Lecture Notes in Business Information Processing*. Vol. 458. Springer, Cham. Pp. 157–172. DOI: 10.1007/978-3-031-16171-1_10.

Cremerius, J., Weske, M. (2023). Context-Aware Change Pattern Detection in Event Attributes of Recurring Activities. In: *Intelligent Information Systems. CAiSE 2023. Lecture Notes in Business Information Processing*. Vol. 477. Springer, Cham. DOI: 10.1007/978-3-031-34674-3_1.

Kampik, T., Weske, M. (2022). Event Log Generation: An Industry Perspective. In: *Enterprise, Business-Process and Information Systems Modeling. BPMDS EMMSAD 2022. Lecture Notes in Business Information Processing*. Vol. 450. Springer, Cham, pp. 123–136. DOI: 10.1007/978-3-031-07475-2_9.

Kulagin, G., Ermakov, I., Lyadova, L. (2022). Ontology-Based Development of Domain-Specific Languages via Customizing Base Language. In: *Proceedings of the IEEE 16th International Conference on Application of Information and Communication Technologies (AICT2022).* Washington DC, DC, USA: IEEE, 2022. 6 pp. DOI: 10.1109/AICT55583.2022.10013619.

Lanin, V., Lyadova, L., Zamyatina, E., Vostroknutov, N. (2021). An Ontology-Based Approach to Social Networks Mining. In: *Proc. 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2021. Vol. 2: KEOD. Lisbon : SciTePress, 2021. Pp. 234-239. DOI: 10.5220/0010716600003064.

Lyadova, L., Suvorov, N., Zayakin, V., Zamyatina, E. (2022). An Ontological Approach to the Development of Analytical Platform Language Toolkits. In: *Proceedings of the IEEE 16th International Conference on Application of Information and Communication Technologies (AICT2022)*. Washington DC, DC, USA: IEEE, 2022. 6 pp. DOI: 10.1109/AICT55583.2022.10013576.

Marin-Castro, H.M., Tello-Leal, E. (2021). Event Log Preprocessing for Process Mining: A Review. In: *Applied Sciences*. 2021, 11(22), 10556. DOI: 10.3390/app112210556.

Mitsyuk, A.A., Shugurov, I.S., Kalenkova, A.A., van der Aalst, W. M.P. (2017). Generating event logs for high-level process models. In: *Simulation Modelling Practice and Theory*. Vol. 74, 2017, pp. 1–16. DOI: 10.1016/j.simpat.2017.01.003.

Peña-Araya, V. (2015). Galean: Visualization of Geolocated News Events from Social Media. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '15). ACM New York. 2015. Pp. 1041-1042.

Pegoraro, M., van der Aalst, W.M.P. (2019). Mining Uncertain Event Data in Process Mining. In: *Proc. International Conference on Process Mining (ICPM)*. Aachen, Germany, 2019, pp. 89–96. DOI: 10.1109/ICPM.2019.00023.

Ritter A. (2012). Open domain event extraction from twitter. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012. Pp. 1104-1112.

Shalyaeva, I., Lyadova, L., Lanin, V. (2016). Events Analysis Based on Internet Information Retrieval and Process Mining Tools. In: *Proceedings of 10th International Conference on Application of Information and Communication Technologies (AICT2016)*. Baku, 2016, pp. 168-172.

Shalyaeva, I., Lyadova, L., Lanin, V. (2017). Ontology-Driven System for Monitoring Global Processes on Basis of Internet News. In: *Proceedings of IEEE 11th International Conference on Application of Information and Communication Technologies (AICT2017)*. Moscow, 2017, pp. 385-389.

Zayakin, V., Lyadova, L., Sminov, M., Lanin, V., Matta, N., Zamyatina, E. (2022). Event Series Generation and Analysis Based on Multifaceted Ontology. In: *Proceedings of the IEEE 16th International Conference on Application of Information and Communication Technologies (AICT2022).* Washington DC, DC, USA: IEEE, 2022. 6 pp. DOI: 10.1109/AICT55583.2022.10013573.

Zayakin, V.S., Lyadova, L.N., Lanin, V.V., Zamyatina, E.B., Rabchevskiy, E.A. (2023). An Ontology-Driven Approach to the Analytical Platform Development for Data-Intensive Domains. In: *Knowledge Discovery, Knowledge Engineering and Knowledge Management. IC3K 2021. Communications in Computer and Information Science*. Vol. 1718, ch. 8, pp. 129-149. Springer, Cham. DOI: 10.1007/978-3-031-35924-8_8