J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr
Copyright ⓒ 2019 KSII

Tailoring Operations based on Relational Algebra for XES-based Workflow Event Logs☆

Jaeyoung Yun[1] Hyun Ahn[1] Kwanghoon Pio Kim[1*]

ABSTRACT

Process mining is state-of-the-art technology in the workflow field. Recently, process mining becomes more important because of the fact that it shows the status of the actual behavior of the workflow model. However, as the process mining get focused and developed, the material of the process mining – workflow event log – also grows fast. Thus, the process mining algorithms cannot operate with some data because it is too large. To solve this problem, there should be a lightweight process mining algorithm, or the event log must be divided and processed partly. In this paper, we suggest a set of operations that control and edit XES based event logs for process mining. They are designed based on relational algebra, which is used in database management systems. We designed three operations for tailoring XES event logs. Select operation is an operation that gets specific attributes and excludes others. Thus, the output file has the same structure and contents of the original file, but each element has only the attributes user selected. Union operation makes two input XES files into one XES file. Two input files must be from the same process. As a result, the contents of the two files are integrated into one file. The final operation is a slice. It divides anXES file into several files by the number of traces. We will show the design methods and details below.

☞ keyword : workflow, process mining, XES, relational algebra, event log

1. Introduction

Process mining is one of a workflow-analyzing technology that uses workflow event logs. Recently, process mining technology becomes more important because of the fact that it shows the status of the actual behavior of the workflow process model(we will call the process model in this paper). We can find out the difference between the designed and running models. Before the process mining is focused, the way to advance the process model is to analyze the structure of the model or information of it [1-3].

As the event log becomes massive, there are some problems in process mining such as too many computations. To avoid this, there should be a very light process mining algorithm, or the event log should be divided into several parts. In this paper, we suggest some operations that edit XES-based workflow event logs and implementation. We designed three operations – select, union, and slice. The overall system and details about each operation are described in section 4. In section 5, experimental results using a simple implementation will be explained.

2. Backgrounds

2.1 Process Mining

Process mining [4-5] is a rising technique in the workflow field. It is a technology that analyzes the workflow event logs, which is an execution history of the process model. It is important because it shows the actual behavior of the process model and gives the direction what the actual model is and how it works. There are three main phases in process mining. First, process discovery. Process discovery is to find out the process model only using the event logs. Then, the second can be done. Second is a conformance checking. In this phase, the discovered model is compared with the original model(i.e.,designed model before running). Finally, update the original model using the

---

1 Div. of Computer Science and Engineering, Kyonggi University, Gyeonggi, 16227, Republic of Korea
* Corresponding author: Kwanghoon Pio Kim (kwang@kgu.ac.kr)
[Received 20 June 2019, Reviewed 15 July 2019(R2 28 August 2019), Accepted 17 September 2019]
☆ Apreliminary version of this paper was presented at ICONI 2018.
☆ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant No. 2017R1A2B2010697).

information obtained from the second phase.

2.2 XES Event Log

As the process mining arises, the event logs are also becoming essential and developed. There are several standard formats to define the workflow event logs such as XWELL [6], BPAF [7], XES [8-9]. In this paper, we use XES (eXtensible Event Stream) formats because it is popular, and there are lots of datasets. The structure of XES is shown in Fig. 1. XES is an XML based format. There are three main tags in XES. <Log> is the highest level tag. It defines a process model. In the <Log>, there is a tag called <Trace>. It defines an instance of the process model defined in the log. Thus, there can be many traces in a log. <Event> which belongs to the trace defines an instance of an activity in the instance of the process model. There also can be many events in a trace. Also, each tag has its attributes to contain the information.

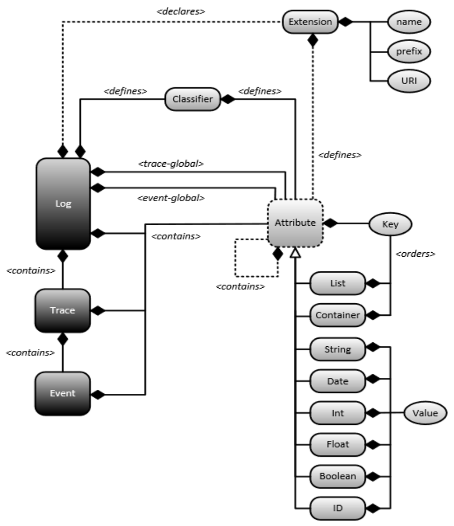


Fig. 1. Structure of the XES [8]

3. Related Works

3.1 Process Cubes

Process cube [10-11] is a data cube with the event logs. It arranges the event logs as a data cube and edits using the OLAP (Online Analytical Processing) operations.Thus, it can manage the event logs easily, and the event logs can be edited for process mining. The OLAP has four operations – slicing, dicing, rolling up and drilling down. Slicing is an operation that parses specific data cells that have a specific value of the dimension. Dicing is similar to the slicing,but it can take several values to parse. Rolling up and drilling down change the level of contents of the dimension.

3.2 Process Mining in Large Data

As the event log datasets become massive, the basic process mining algorithms cannot work because of the computation and resources. The main idea to avoid this problem [12] is to divide the process model and take process mining partly and finally integrate them.

4. XES Log Tailor

XES log tailor is a set of operations that edit and control the XES based event logs. It has three operations – select, union, slice. Select operation parses specific elements in the log that matches the condition and union operation integrates two event logs from distributes process model into one event log file. Also slice operation divides a large log file into several files by the number of traces. Details are in each subsection.

Fig 2 shows the overall structure of the XES log tailor. XES log tailor consists of three components –query compiler, tailor, and log handler. Query compiler interprets the input query and calls the appropriate tailor operation. Each operation has its query similar to the one used in DBMS [13]. Thus, when the user puts the query to XES tailor, the query compiler takes it and calls the proper operation with conditions. Tailor has actual operations such as select, union, and slice. It makes a resultant XES file using the directions from the query compiler. Before that, there should be needed that which XES log file will be processed. It is the query compiler's job to get which log file to handle and the log handler actually manages the log file. The log handler reads the log file and gets the basic structure of the log file such as the version of XML, names of attributes. Thus, the output file has the same structure as the input file.

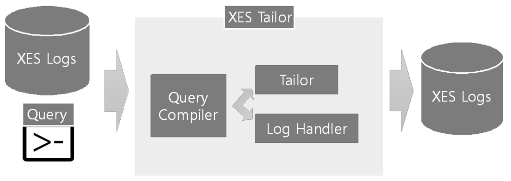


Fig. 2. Structure of XES Tailor

In the subsection, we define each operation and explain how it works. Each operation has two parts. First is the structure of the query, and the second is the operation. The query is used as an input format by the user to call the operation. It is very similar to the SQL because the select and union operations are also based on the relational algebra.

4.1 Select Operation

Select operation parses the specific data from the input event log. Definition 1 shows the structure of the select query. The query obtains <sel-list>, <elem>, and <condition>. <sel-list>is a list of attributes that belongs to <elem>. The output file only has these attributes. <elem> can be a trace or event. If <elem> is an event, traces which have the parsed events are also parsed because a trace is required to identify the event. On the other hand, if the <elem> is a trace, the event is not necessary. <condition> is a set of conditions that defines what to parse. In each condition has attribute_key, operator, and attribute_value. Attribute_key is the name of an attribute,and attribute_value is the value of it. For example, if we want to select the whole events(i.e., events with all attributes) which are performed in ‘Seoul’ city, the query should be like that.

- SELECT * FROM event WHERE place=Seoul;

Definition 1. The structure of query SELECT

```
SELECT 〈sel-list〉 FROM 〈elem〉 WHERE 〈condition〉;
〈sel-list〉        := 〈attribute〉, 〈sel-list〉
                   := 〈attribute〉
〈elem〉            := "trace"
                   := "event"
〈condition〉       := 〈condition〉 AND 〈condition〉
                   := 〈attribute_key〉 = | 〉 | 〈
                                  〈attribute_value〉
```

The algorithm of select operation is defined in Algorithm 1. It reads the input file, and when it finds anelement, it buffers the element and checks this attribute to meet the condition. If the element meets the condition, it writes this element to the output file.

Algorithm 1. Select Operation

```
1:  Input: XES file, a condition_list, a sel_list
2:  for all elements ∈ XES file do
3:      for all attribute ∈ element do //checks all attributes in
                                                          element
4:          if attribute not ∈ sel_list then //not to select

5:              removeAttribute()
6:          if attribute ∈ condition_list then
7:                  checkCondition() //check this attribute meets
                                                       condition
8:      if elementIsMatched()
9:          readElementToOutputFile()
```

4.2 Union Operation

Union operation integrates two XES files that are of thesame process model but a different part. If the process model is executed in a distributed system or the original XES file is divided, several XES files that came from the same process modelare created. This situation is good for process mining analysts because it can help to reduce the calculation and resources. However, if he is available totake all these logs at once, it is better to use the integrated one file. The union operation is useful to such a situation.

There are basically two ways to divide the process model. One is horizontally dividing, and the other is vertically dividing. Divide the process model horizontally means that the process model is divided based on the disjunctive point. In other words, at a disjunctive point, each branch is a part of the process model, and when each branch is executed separately, the model is divided horizontally. Vertically dividing is to divide the process model at a specific activity. Thus, the former and latter are part of the process model. Fig 3 shows these divisions and their union operation.

There are two divisions, and the union operation also matches to these divisions. Thus, we designed two union operations – horizontal union and vertical union. The horizontal union is for the horizontal division. As shown in Fig 3, in the horizontally divided process model, each part

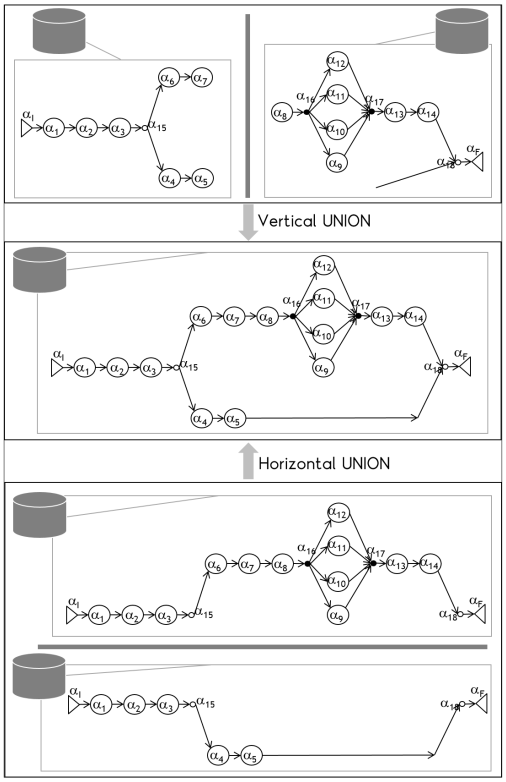


Fig. 3. Illustraion of union operation

of the process model has a complete process instance (i.e.,each part has the same start activity and end activity as the original process model). Therefore, each XES file has its traces because trace defines each process instance. So, the horizontal union operation gathers all the traces from two input log files and integrates them. In the vertically divided process model, each part has the same process instances but different activities. Therefore, in XES files, each file has the same traces but different events. Thus, the vertical union operation gathers the events and make complete traces.

4.3 Slice Operation

Slice operation divides anXES log file into several XES log files by the number of traces. It is used when the log file is too large to analyze. The query of slice operation is defined in Definition 2. It gets fileName to slice and gets the number of traces. For example, if the input file has 200,000 traces and slice by 10,000 (i.e., numOfTrace is 10,000), a total of 20 sliced XES files created.

Definition 2. The structure of query UNION and SLICE
H_UNION 〈fileName〉 〈fileName〉;
V_UNION 〈fileName〉 〈fileName〉;
SLICE 〈fileName〉 〈numOfTrace〉

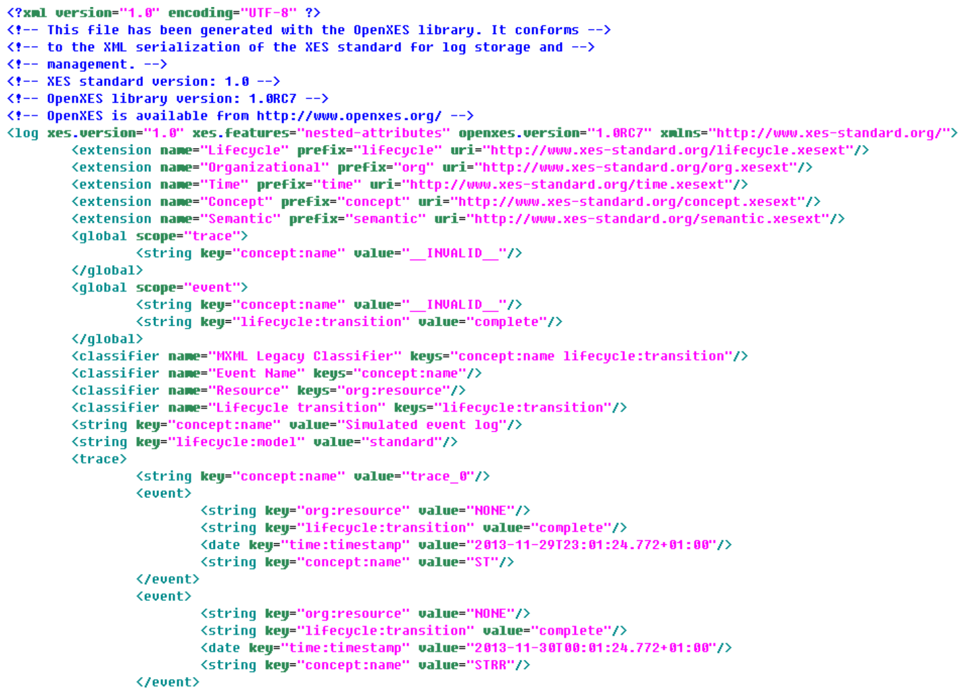
5. Experiments

We implemented operations in the XES log tailor and used to show how they work. Our dataset is the large bank transaction process dataset [14] which is XES based synthetic data. It is a simulated log data from the artificially made process model. Among the dataset, we use the 2000-all-nonoise version, which has 2000 traces without noises.Fig 4 shows the overall structure of the dataset. In the dataset, the traces haveonly one attribute ‘concept:name’, which defines the ID of the trace. Event has four attributes ‘org:resource’, ‘lifecycle:transition’, ‘time:timestamp’, and ‘concept:name’. ‘org:resource’ defines the organization of the activity. However, this is set to ‘NONE’because the data is synthetic. ‘lifecycle:transition’ defines the status of activity execution. ‘time:timestamp’ defines thetime when this activity performed and ‘concept:name’ defines the ID of the activity.

We use tailoring operations to this dataset. Fig. 5 and 6 show the results. Fig. 5 shows the result of tailored data by the select operation. We parsed the start activities from the original dataset. Also,we parsed only timestamp and ID of the events without any further information. In Fig.5, we can see that each trace has only one event whose name is ‘ST’ with only two attributes. The query is as below.

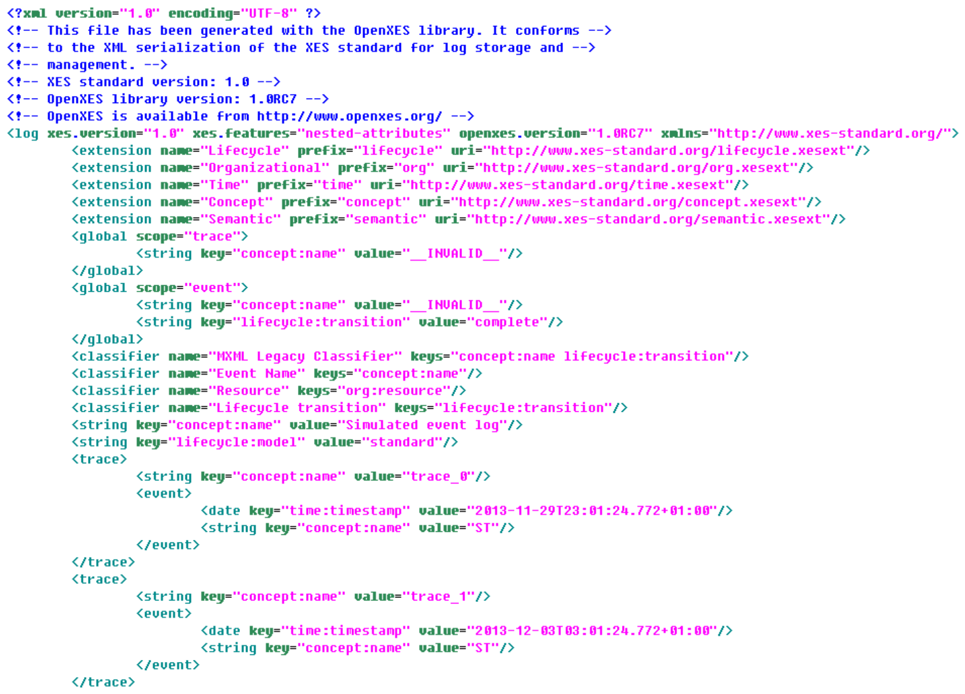
- SELECT concept:name, time:timestamp FROM event WHERE concpet:name=ST;

Fig. 6 shows the result of tailored data by slice operation. We sliced by 500 traces. In Fig 6, we can see that the ID of the first trace is ‘trace_500’ because the file is the second file. Traces 0 to 499 are in the first file. The query is as below.



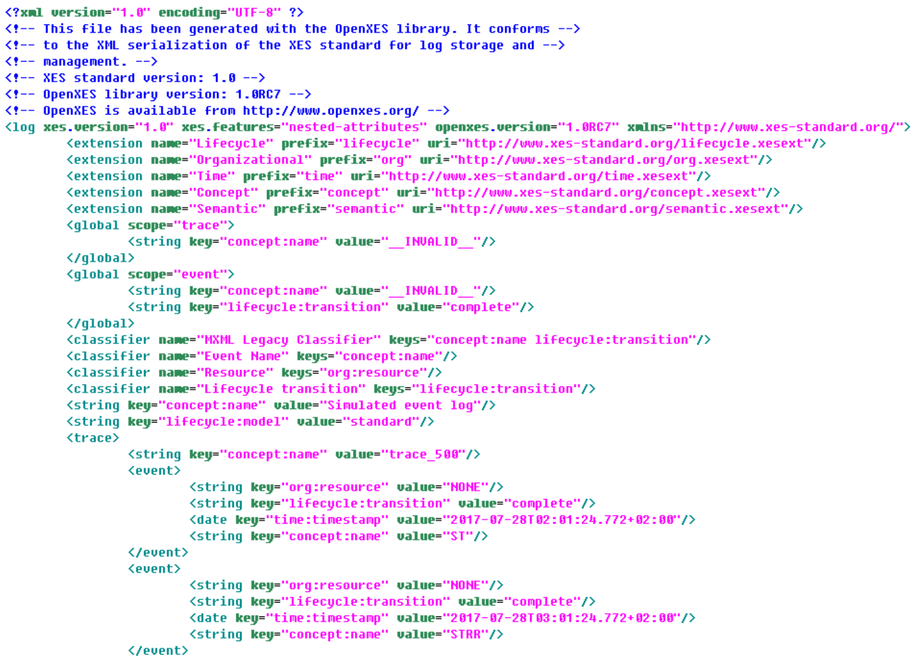
```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This file has been generated with the OpenXES library. It conforms -->
<!-- to the XML serialization of the XES standard for log storage and -->
<!-- management. -->
<!-- XES standard version: 1.0 -->
<!-- OpenXES library version: 1.0RC7 -->
<!-- OpenXES is available from http://www.openxes.org/ -->
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7" xmlns="http://www.xes-standard.org/">
        <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
        <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
        <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
        <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
        <extension name="Semantic" prefix="semantic" uri="http://www.xes-standard.org/semantic.xesext"/>
        <global scope="trace">
                <string key="concept:name" value="__INVALID__"/>
        </global>
        <global scope="event">
                <string key="concept:name" value="__INVALID__"/>
                <string key="lifecycle:transition" value="complete"/>
        </global>
        <classifier name="MXML Legacy Classifier" keys="concept:name lifecycle:transition"/>
        <classifier name="Event Name" keys="concept:name"/>
        <classifier name="Resource" keys="org:resource"/>
        <classifier name="Lifecycle transition" keys="lifecycle:transition"/>
        <string key="concept:name" value="Simulated event log"/>
        <string key="lifecycle:model" value="standard"/>
        <trace>
                <string key="concept:name" value="trace_0"/>
                <event>
                        <string key="org:resource" value="NONE"/>
                        <string key="lifecycle:transition" value="complete"/>
                        <date key="time:timestamp" value="2013-11-29T23:01:24.772+01:00"/>
                        <string key="concept:name" value="ST"/>
                </event>
                <event>
                        <string key="org:resource" value="NONE"/>
                        <string key="lifecycle:transition" value="complete"/>
                        <date key="time:timestamp" value="2013-11-30T00:01:24.772+01:00"/>
                        <string key="concept:name" value="STRR"/>
                </event>
```

Fig. 4. Sample of 2000-all-nonoise.xes



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This file has been generated with the OpenXES library. It conforms -->
<!-- to the XML serialization of the XES standard for log storage and -->
<!-- management. -->
<!-- XES standard version: 1.0 -->
<!-- OpenXES library version: 1.0RC7 -->
<!-- OpenXES is available from http://www.openxes.org/ -->
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7" xmlns="http://www.xes-standard.org/">
        <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
        <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
        <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
        <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
        <extension name="Semantic" prefix="semantic" uri="http://www.xes-standard.org/semantic.xesext"/>
        <global scope="trace">
                <string key="concept:name" value="__INVALID__"/>
        </global>
        <global scope="event">
                <string key="concept:name" value="__INVALID__"/>
                <string key="lifecycle:transition" value="complete"/>
        </global>
        <classifier name="MXML Legacy Classifier" keys="concept:name lifecycle:transition"/>
        <classifier name="Event Name" keys="concept:name"/>
        <classifier name="Resource" keys="org:resource"/>
        <classifier name="Lifecycle transition" keys="lifecycle:transition"/>
        <string key="concept:name" value="Simulated event log"/>
        <string key="lifecycle:model" value="standard"/>
        <trace>
                <string key="concept:name" value="trace_0"/>
                <event>
                        <date key="time:timestamp" value="2013-11-29T23:01:24.772+01:00"/>
                        <string key="concept:name" value="ST"/>
                </event>
        </trace>
        <trace>
                <string key="concept:name" value="trace_1"/>
                <event>
                        <date key="time:timestamp" value="2013-12-03T03:01:24.772+01:00"/>
                        <string key="concept:name" value="ST"/>
                </event>
        </trace>
```

Fig. 5. Sample of tailored 2000-all-nonoise.xes by select operation



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This file has been generated with the OpenXES library. It conforms -->
<!-- to the XML serialization of the XES standard for log storage and -->
<!-- management. -->
<!-- XES standard version: 1.0 -->
<!-- OpenXES library version: 1.0RC7 -->
<!-- OpenXES is available from http://www.openxes.org/ -->
<log xes.version="1.0" xes.features="nested-attributes" openxes.version="1.0RC7" xmlns="http://www.xes-standard.org/">
        <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
        <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
        <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
        <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
        <extension name="Semantic" prefix="semantic" uri="http://www.xes-standard.org/semantic.xesext"/>
        <global scope="trace">
                <string key="concept:name" value="__INVALID__"/>
        </global>
        <global scope="event">
                <string key="concept:name" value="__INVALID__"/>
                <string key="lifecycle:transition" value="complete"/>
        </global>
        <classifier name="MXML Legacy Classifier" keys="concept:name lifecycle:transition"/>
        <classifier name="Event Name" keys="concept:name"/>
        <classifier name="Resource" keys="org:resource"/>
        <classifier name="Lifecycle transition" keys="lifecycle:transition"/>
        <string key="concept:name" value="Simulated event log"/>
        <string key="lifecycle:model" value="standard"/>
        <trace>
                <string key="concept:name" value="trace_500"/>
                <event>
                        <string key="org:resource" value="NONE"/>
                        <string key="lifecycle:transition" value="complete"/>
                        <date key="time:timestamp" value="2017-07-28T02:01:24.772+02:00"/>
                        <string key="concept:name" value="ST"/>
                </event>
                <event>
                        <string key="org:resource" value="NONE"/>
                        <string key="lifecycle:transition" value="complete"/>
                        <date key="time:timestamp" value="2017-07-28T03:01:24.772+02:00"/>
                        <string key="concept:name" value="STRR"/>
                </event>
```

Fig. 6. Sample of tailored 2000-all-nonoise.xes by slice operation - Second file of total 4 files

- SLICE 2000-all-nonoise.xes 500

We can also make the original data file from the sliced files using horizontal union operation.

6. Conclusions

In this paper, we introduceXES tailoring operations based on relational algebra and its experimental results. Select operation parses specific elements in the log that matches the condition and union operation integrates two event logs from distributes process model into one event log file. The implemented tool obtains an XES-based event log, and its output is also an XES event log. Therefore, the output can be used in other process mining tools thatuse the XES based event logs without any further preprocessing. Also, the operations are activated by a query which is similar to the one used in DBMS it is easy to use and can be automated through a script. As future work, there are some challenges. First, in union operation, we do not think about the complicated situation such as bothhorizontally and vertically divided processes and their logs. The second is in the slice operation. We will extend the slice operation to get more criteria for dividing files diversely. Finally, we will design other operations.

References

[1] J. Kim, et al., "An Estimated Closeness Centrality Ranking Algorithm and Its Performance Analysis in Large-Scale Workflow-supported Social Networks," KSII Transactions on Internet and Information Systems, Vol. 10, No. 3, pp. 1454-1466, 2016. https://doi.org/10.3837/tiis.2016.03.031

[2] S. Park and K. P. Kim, "A Closeness Centrality Analysis Algorithm for Workflow-supported Social Networks," Journal of Internet Computing and Services, Vol. 14, No. 5, pp. 77-86, 2013. https://doi.org/10.7472/jksii.2013.14.5.77

[3] D. Pham, H. Ahn and K. P. Kim, "Discovering Temporal Work Transference Networks from Workflow

Execution Logs," Journal of Internet Computing and Services, Vol. 20, No. 2, pp. 101-108, 2019.
https://doi.org/10.7472/jksii.2019.20.2.101

[4] K. P. Kim, "Mining Workflow Processes from Distributed Workflow Enactment Event Logs," Knowledge Management & E-Learning: An International Journal (KM&EL), Vol. 4, No. 4, pp. 528-553, 2013.
https://doi.org/10.34105/j.kmel.2012.04.038

[5] W. van der Aalst, et al. "Process Mining Manifesto," in Proc. of the International Conference on Business Process Management. Springer, Berlin, Heidelberg, 2011.
https://doi.org/10.1007/978-3-642-28108-2_19

[6] M. -J. Park and K. -H. Kim. "XWELL: A XML-based Workflow Event Logging Mechanism and Language for Workflow Mining Systems," in Proc. of the International Conference on Computational Science and Its Applications. Springer, Berlin, Heidelberg, 2007.
https://doi.org/10.1007/978-3-540-74484-9_76

[7] M. zur Muehlen, and K. D. Swenson. "BPAF: A Standard for the Interchange of Process Analytics Data," in Proc. of the International Conference on Business Process Management. Springer, Berlin, Heidelberg, 2010.
https://doi.org/10.1007/978-3-642-20511-8_15

[8] C. W. Günther and E. Verbeek, "XES Standard Definition", Technische Universiteit Eindhoven University of Technology, Netherlands, 2014.
https://research.tue.nl/en/publications/xes-standard-definition

[9] G. Acampora, et al., "IEEE 1849: The XES Standard", IEEE Computational Intelligence Magazine, Vol. 12, No. 2, pp. 4-8, 2017.
https://standards.ieee.org/standard/1849-2016.html

[10] W. M. P. van der Aalst, "Process Cubes: Slicing, Dicing, Rolling Up and Drilling Down Event Data for Process Mining," in Proc. of the Asia-Pacific Conference on Business Process Management. Springer, Cham, 2013.
https://doi.org/10.1007/978-3-319-02922-1_1

[11] A. Bolt, and W. M. P. van der Aalst. "Multidimensional Process Mining using Process Cubes," Enterprise, Business-Process and Information Systems Modeling, pp. 102-116, Springer, Cham, 2015.
https://doi.org/10.1007/978-3-319-19237-6_7

[12] W. M. P. van der Aalst, "Process Mining in the Large: A Tutorial," European Business Intelligence Summer School. Springer, Cham, 2013.
https://doi.org/10.1007/978-3-319-05461-2_2

[13] H. Garcia-Molina, et al, Database System: The Complete Book, Department of Computer Science Stanford University, 2009.

[14] J. Munoz-Gama, "Large Bank Transaction Process, " Universitat Politècnica de Catalunya (Barcelonatech). Dataset, 2014.
https://doi.org/10.4121/uuid:c1d1fdbb-72df-470d-9315-d6f97e1d7c7c

◐ 저 자 소 개 ◑

Jaeyoung Yun
2018 B.S. in Computer Science, Kyonggi University
2018 ~ Present, M.S. student in Computer Science, Kyonggi University
Research Interests : process mining, deep learning.
E-mail : j@kgu.ac.kr

Hyun Ahn
2011 B.S. in Computer Science, Kyonggi University
2013 M.S. in Computer Science, Kyonggi University
2017 Ph.D. in Computer Science, Kyonggi University
2018 ~ Present, Assistant Professor of the Dept. of Computer Science and Engineering at Kyonggi University
Research Interests : Business Process Management, business process intelligence, process mining.
E-mail : hahn@kgu.ac.kr

Kwanghoon Pio Kim
1984 B.S in Computer Science, Kyonggi University
1986 M.S. in Computer Science, Chungang University
1994 M.S. in Computer Science, University of Colorado at Boulder
1998 Ph.D. in Computer Science, University of Colorado at Boulder
1998 ~ Present, Professor of the Dept. of Computer Science and Engineering at Kyonggi University
Research Interests : CSCW, workflow systems, Business Process Management, process mining, enterprise social network analysis.
E-mail : kwang@kgu.ac.kr