Extracting Audit Trail Data of Port Container Terminal for Process Mining

[1,2]Bambang Jokonowo, [1]Riyanarto Sarno, [1]Siti Rochimah
[1]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya
[2]Department of Information Systems, Universitas Mercu Buana, Jakarta
bambang.jokonowo@mercubuana.ac.id, riyanarto@if.its.ac.id, siti@if.its.ac.id

***Abstract*— The process mining approach typically requires a structured format of an event log. In the practical fact, the structured format of the event log is not always directly available in the database information systems. For the evident needs of process mining at container terminals at the modern port. Then the historical data can be methodically taken from the audit trail data plus related event data. The historical data of a life-cycle process are driven by question. However, the audit trail data are typically provided in the standard form of a flat table. Each continuous row invariably contains a data event that has in common the essential characteristics of timestamp attributes as a specific form of possible activity. Therefore, this empirical research proposes a possible method to extract audit trail data terminal operating systems (EXTOS). The extraction result is proved by structured formalization of event logs. The practical implementation of this extraction method was made after observing four port case studies in eastern Indonesia. The event data extraction results have been proven in one modern port to measure import dwell-time container process performance indicators..**

Keywords*— Process mining, extract data, event log, terminal operating system (TOS), dwell-time container, cross-organization

I. INTRODUCTION

Currently, a process model can be displayed properly using process mining techniques [1]. The meaningful result of process models describes the historical reality of identifiable events. Process models can be accurately used for comprehensive analysis and verification business processes. To properly monitor a marked passage of the continuous processes in the automated information system, each historical event is typically recorded in the transaction log or audit trail [2]. Process mining is impossible without proper event logs.

Process mining has become aware a hot research area discussed in recent years [1,3,4]. It became interesting, because its technical ability to instantly discover, monitor and meaningfully improve real processes. Process mining is scientifically a academic discipline of computational intelligence and data mining on one hand, modeling and comprehensive analysis of a complex process on the other hand side. The process model is compared by checking deviation between the possible model and the event log. The practical ability of process model may positively enhance the business processes.

The business process in the port container terminals is a process that all activities are concentrated to serve loading and unloading of containers on the vessel. These containers are unloaded and loaded with high priority. Consequently, the demanding process of loading and unloading of containers at the terminal requires urgency as it critically affects other logistics activities [5]. Process mining analysis has been carried away in the container terminal process to measure Dwell-time performance [6].

Dwell-time is one of the several standard benchmark comparison between the port container terminals and the modern standard for objectively measuring the operational performance. How does a container dwell-time calculate? There are typically two possible kinds of container dwell time: loading and unloading [5]. Dwell time of loading is precisely the elevated span of a container from the terminal entrance gate to cautiously enter the chosen vessel. Dwell-time of unloading is precisely the elevated span of a container starting from the vessel to the gate terminal exit. That's why it needs an event log for the life-cycle of the container coming and going out of the port.

Like many other systems in complex logistics and dependent production, container terminals are characterized by complex, dynamic and uncertain processes [7]. Possible activities are highly dependent on each other [8]. Evrim Ursavas proposed a decision support system for optimizing local operations on the quayside of a container terminal [9]. Yin and Khoo utilized a distributed agent system to determine the planning and scheduling of dynamic ports [10]. Hee-jung YEO and other academic researchers found that the modern facilities and needed services of a terminal level are positively associated with the port performance [11–13]. The import container is mentioned in [14] that time is genuinely needed in the order of the process supply chain in the port. It's completely that this published paper provides container ideas that are only stacked in the stacking area. It is obvious that the service factor is also very significant variables, whereas the considerable size of the container terminals scale itself is not a guarantee by systematic enhancement of their performance reviews.

The specific objective of a terminal operating system, or TOS, is typically to reliably provide a comprehensive set of computerized procedures to carefully manage containers, portable machines and competent people within the facility to enable a seamless link to efficiently and effectively manage the facility. The possible selection of the modern terminal's operating system technology has become critical to economic decision-making at the port container terminal [15].

Container port or modern terminal is powerfully influenced by the operational efficiency of human resources, management team, needed infrastructure, equipment and so forth. Maximal effort on the appropriate level of traffic depends on the operational and technical characteristics of the terminal. Many principal parts involved in the process to decide the port operations. Accurately measuring the dwell-

978-1-7281-2133-8/19/$31.00 ©2019 IEEE

time at the modern port involved more than one official organization. Therefore, process mining should be able to accurately model the inter-organizational processes [16].

Some published papers explained the process mining of logistics at the port [17] and extraction of data source event log [4,18,19]. A comprehensive methodology is also proposed for applying process mining in complex logistics [17]. Wang described an extraction of data from logistic information systems, but did not explain in detail how to perform the extraction of data to be in a structured format of the event logs. Joost Buijs in his master's thesis [18], performed a mapping of relational database information system to produce the class structure of the event log for the specific need of process mining.

However, the source data is not always in a standard form. It is not possible to automatically be converted into the event log structure. Also associated with the practical knowledge to be carefully extracted, the necessary harmony between the intellectual intuition of process mining analyst and assisted database administrator is significant. Leoni and his academic colleagues [4], showed the alignment tool when connecting the observed behavior with the specific behavior being accurately modeled. Alignment is performed by (i) cleaning event log to change the trace used for further analysis, (ii) the diagnosis of event logs to check for conformance, (iii) repairing event logs to ensure that the restrictions are necessary to be proved before it is analyzed..

As a preliminary study of process mining at the port, we have started to conduct a preliminary study on the specific time interval multi-organizational workflow at the port [20]. In this research, we implement a method that shows how to extract data from event logs within a port terminal operating system. This academic paper is divided into seven sub-themes. Chapter one contains a preliminary study of process mining at the modern port. Chapter two provides the motivation and valuable contribution of empirical research for data extraction. Chapter three contains a formalization of the complex structure of event log. Chapter four proposes the practical method of carefully extracting the event log. Chapter five of this method was reasonably made after direct observations from four port case studies in eastern Indonesia and applied to one of the modern ports. Chapter six properly discusses the practical results of the extraction. Chapter seven is a conclusion and further work.

II. Motivation and Contribution

The extraction process represents an inseparable part of the process mining itself. If there is typically an possible error in the extraction of the historical data, it will typically cause an error in the process mining analysis [18]. Moreover, this empirical research proposes a practical method to extract audit trail data of a terminal operating system (EXTOS).

The motivation and contribution of this paper are:

- To adequately explain preprocessing that has been done to measure the key performance of container terminal ports by the published author [6].
- It carefully extracts audit trail data of the terminal operating system (EXTOS) to generate structured event logs.
- It properly selects a process instance of dwell time of unloading containers at port container terminal.

III. Formalization of Structured Event Logs

Definitions A and B are present the formal definition, which obtain a necessary requirement to scientifically prove a tree structure of the event log. The tree structure of event log means a historical process typically consisting of cases. A case typically consists of events such that each event relates to precisely one case. Events within a case are ordered. Events contain attributes as examples of typical names are activity, time, and resource.

In this research, the formalization of structured event log follows to the method of van der Aalst [1]. To be clear the definition of events and attributes we may associate to the activities of the port terminal.

Definition A. Event, attribute: Let $\mathbb{E}$ be the event of a port terminal, i.e., the set of all possible event identifiers at a port terminal. Events may be characterized by various attributes, e.g., an event to move and stack a container may have a timestamp, correspond to an activity in the yard storage area, is executed by a particular person or group. Events that are associated to containers are retrieved and loaded onto a truck, and transported out of the yard storage area and loaded into the vessel, etc.

Let AN be a set of attribute names. For any event $e \in \mathbb{E}$ and name $n \in AN$:
$\#_n(e)$ is the value of attribute n for event e; and $\#_n(e) = \perp$ (null value) if event e does not have an attribute named n.

For convenience, this paper follows the definition from van der Aalst. It is assumed the following standard attributes.
$\#_{activity}(e)$ is the activity associated to event e.
$\#_{time}(e)$ is the timestamp of event e. The timestamps should be non descending in the event log.
$\#_{resource}(e)$ is the resource associated to event e.

It is assumed a domain $\mathbb{T}$, i.e., $\#_{time}(e) \in \mathbb{T}$ for any $e \in \mathbb{E}$. In most situations, activities take time. Therefore, events may point out for example the start or completion of activities. (1)

Definition B. Case, trace, event log: Let $\mathbb{C}$ be the case at port container terminal, i.e., the set of all possible case identifiers at a port container terminal. Cases like events, have attributes.

Let AN be a set of attribute names. For any event $c \in \mathbb{C}$ and name $n \in AN$:

$\#_n(c)$ is the value of attribute n for case c; and (2)
$\#_n(c) = \perp$ (null value) if case c does not have an attribute named n. (3)

Each case has a special mandatory attribute *trace*:

$\#_{trace}(c) \in \mathbb{E}^*$, it is assumed $\#_{trace}(c) \langle \ \rangle$, i.e., traces in a log contain at least one event; and

$\hat{c} = \#_{trace}(c)$ is shorthand for referring to the trace of a case. (4)

A trace is a finite sequence of events $\sigma \in \mathbb{E}^*$ such that event appears only once, i.e., for $1 \leq i < j \leq |\sigma|: \sigma(i) \neq \sigma(j)$. (5)

An *event log* is a set of cases $L \subseteq \mathbb{C}$ such that each event appears at most once in the entire log, i.e., for any $c_1, c_2 \in L$ such that $c_1 \neq c_2$: $\partial_{set}(\hat{c}_1) \cap \partial_{set}(\hat{c}_2) = \emptyset$. (6)

If an event log contains timestamps, then the ordering in a trace should respect these timestamps, i.e., for any $c \in L$, i *and* j such that $1 \leq i < j \leq |\hat{c}|$: $\#_{time}(\hat{c}(i)) \leq \#_{time}(\hat{c}(j))$. (7)

$\{\#_{activity}(e) \mid c \in L \wedge e \in \hat{c}\}$ is the set of all activities appearing in log L. (8)

$$\left\{a \in \mathbb{A} \;\middle|\; \begin{array}{c} c \in L \wedge a = \#_{activity}(\hat{c}(1)) \wedge a = \#_{activity} \\ (\hat{c}(|\hat{c}|)) \end{array}\right\}$$

is the set of all activities that served as start and end activity for the same case. (9)

IV. Extraction Method of Event Log

Terminal operating system, TOS, represent a system that operates the existing processes in the modern port. For the organizational need of data audit, the system stores a table called the audit trail, then extract the audit trail in the event log to produce structures. Event logs are generated in accordance with the event data structure for the specific need of process mining, namely life cycle process. Furthermore, the event log that is in conformity with the structure converted into a standard event log, i.e., a extensive file XES.

Database transaction applications do not automatically make the event log, but typically keep a log data for audit requirements Fig 1. Audit trail data is typically a comprehensive database in flat files. Data log store various transaction codes and the timestamp of the historical event. But for the needs of "process mining" the data is driven based on the question-driven, namely the process of the selected instance, because it is possible that the audit trail data is not enough so that additional incomplete data is needed.

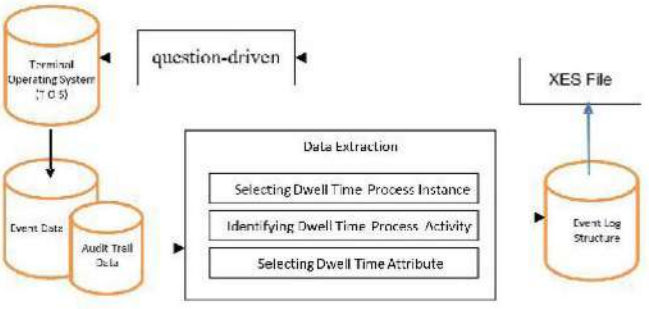

Fig 1. Data Extraction

To sort the log data into a sequence of processes, the log data need to be changed to conform to structure of event log. Therefore, when designing an event log based audit trail data there are several steps that must be performed.

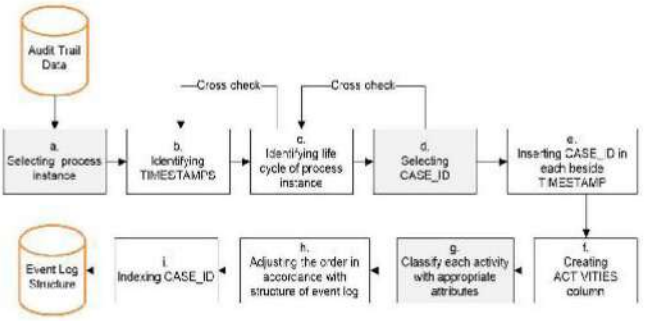

Fig 2. Detail Data extraction Framework

The data extraction is described in Fig 2. There are several steps that may only be done by a "process mining analyst" and a "data administrator", which are marked by a gray box (Figure 2: a, d, g). The other boxes may be done automatically using computer applications. For the current study, the extraction is processed using SQL Database. The following are the steps of the extraction mentioned above:

a. Selecting process instance.

Determining a scope of the process is based on the flow of a particular business process. Characterization of the business process applies a life-cycle process from start to finish , accordance with the definition (9). Each life-cycle is called process instance.

b. Identifying TIMESTAMPS columns.

After specifying the name of a process instance, the next grouping column that contains TIMESTAMP is performed. Usually, the form of a timestamp contains the timestamp of doing an activity, accordance with the definition (1).

c. Identifying the life-cycle of process instances.

Identifying life-cycle process instance is performed to make a sequence the activity. The order of activity is identified by the timestamp of the event. Therefore, sorting a process is performed based on the timestamp column. The events are then checked to determine the beginning and end of the activity. Formally, this step is the same as the definitions (4) and (5). We need to make sure this step, by comparing the results of the previous step.

d. Selecting a column case_ID.

A case consists of events such that each event relates to precisely one case. Case_ID is made to ensure that every event in a process instance runs from start to end, according to the definition (6). We need to make sure this step, by comparing the results of the previous step.

e. Inserting case_ID column in each beside TIMESTAMP columns.

We put case_ID column next to each column of TIMESTAMP. The goal is that every change in the order of time, was involved in a process instance. That is, it is still associated with the same case_ID. This is consistent with the definition (4).

f. Creating a column for ACTIVITIES column.

Among each group case_ID and TIMESTAMP column, it is made ACTIVITIES column. We writte the activities can be customized with the name of the column headings TIMESTAMP. Typically, the TIMESTAMP column headings are tailored to the activities undertaken. Column group of activities overwritten with the TIMESTAMP column headings. The purpose of this step is to create a group of

activity columns based on events. In accordance with the convenience $\#_{activity}(e)$ is the activity associated with the event e. This step by definition (7), that the order of a trace can be based on events.

g. Classifying each activity with the appropriate attributes.

The content of data log is not sequentially according to the activity. Accordance with the definition (2) and (3), data administrator assistance is needed. They can help to adjust data to be extracted and grouped by activity data.

h. Adjusting the order of columns in accordance with the structure of event log.

Each group of activities and attributes are arranged to form the structure of the event log. Rows of columns are based on the structure, i.e., case_ID, ACTIVITIES, TIMESTAMP, and the corresponding attributes.

i. Indexing by case_ID.

The final step of extracting the audit trail data is to index data based on case_ID. This structure is the standard to establish the event log. There are other stages to selecting and filtering, namely to clean incomplete data. This step uses tools PROM[1] this step is not discussed in this paper Thus, the process mining is expected to be done next. All of these steps are important because errors in the formation of the structure event log will cause errors when going in verification and analysis.

V. CASE STUDY

In the case study, we extracted one of the four container ports.. We show the table which is arranged vertically Table 1. The audit trail and additional data tables can be used to measure the performance. Therefore, we use the method of extracting the terminal operating system (EXTOS).

Table 1. Flat log data

CONTAINER_KEY	1.063.411	1.063.412	1.063.413
CONTAINER_NO	SI'NU2951987	SI'NU2955350	SI'NU2980680
VES_ID	ARPE0035	ARPE0035	ARPE0035
VES_CODE	ARPE	ARPE	ARPE
VOY_NO	1400B	1400B	1400B
CTR_I_E_T	I	I	I
CTPS_YN			
CTR_ACTIVE_YN	H	H	H
CTR_SIZE	20	20	20
CTR_TYPE	DRY	DRY	DRY
CTR_STATUS	FCL	FCL	FCL
CTR_INTERN_STATUS	09	09	09
DISC_LOAD_TRANS_SHIFT	DISC	DISC	DISC
GROSS	26.800	26.800	19.100
GROSS_CLASS	X	X	H

To generalize the use of this method we create the following algorithm:

In algorithms 1. Flat tables are grouped according to event labels. Label events are names of certain infinite activities. The output is still a flat table but already in groups.

Algorithm 1: (Grouping activities becomes an event label)
Input: Tabel Flat_01
Output: Tabel Flat_02

```
1  max_attr  /* maximum attribute columns */
2  max_raw /* maximum raw */
3  TD  /* TIMESTAMP or DATE */
4  DATA_attr  /*attribute data correlated with activity*/
5
6  for i ← 1 to max_raw do
7          CASE_ID = candidate Case ID ∧ attributes ≠ TD
8          Write CASE_ID
9      for j ← 1 to max_attr, attr ≠ null do
10         if attr = CASE_ID then
11               attr = attr + 1
12         else if attr = TD, attr ≠ null then
13               AKTIVITAS = TD  /*an event label
14               TIMESTAMP = TD
15               Write AKTIVITAS, TIMESTAMP
16                 attr = attr + 1
17                    while attr = DATA_attr, attr ≠ null do
18                       DATA_attr =  DATA_attr
19                       Write DATA_attr
20                        attr = attr + 1
21                    end while
22          end if
23     end for j
24     baris = baris + 1
25 end for i
```

In algorithm 2. The flat table is then changed to an event log file. The event log file matches the standard structure for extracting mining processes. Which can subsequently be converted into XES format.

Algorithm 2: (Formation of event log CASEID, ACTIVITIES, TIMESTAMP, DATA_attr)
Input: Tabel Flat_02
Output: Event Log_01

```
1  max_raw  /* raw_record */
2  label_event  /* activity group */
3  DATA_attr  /* attribute data correlated with activity */
4  col = 1
5  for i ← 1 to max_raw do
6      for j ← 1 to col
7                CASE_ID = CASE_ID
8                 write CASE_ID
9                   AKTIVITAS = label_event
10                  TIMESTAMP = timestamp
11                 write AKTIVITAS, TIMESTAMP
12                       k = 4
13                      while ( col = k, k = last )  do
14                      DATA_attr =  DATA_attr
15                         Write DATA_attr
16                          k = k + 1
17                       end while
18               col = col + 1
19       end for
20            raw = raw + 1
21 end for
```

a. Selecting process instance.

Process instance that is chosen represent a process of unloading containers. Container unloading process is selected to calculate dwelling time unloading of containers in the port area. The process time starts from the container are unloaded from the vessel and the ends of the container out of the port gate.

b. Identify TIMESTAMPS columns.

c. Identifying the life-circle of process instance.

Based on the process of unloading container we carry out identification and determine the dwell time process.

d. Selecting a column case_ID

We select the column CONTAINER_KEY as case_ID. The goal is to know the life-cycle process of unloading container. CONTAINER_KEY column is made to keep the

[1] http://www.promtools.org

cycle of the process from start to finish in a process instance. Make sure double-check with the previous step.

e. Insert column case_ID in each beside DATE column.

We insert and align the columns CONTAINER_KEY as case_ID on any TIMESTAMP column. In the process of unloading containers, CONTAINER_KEY as case_ID.

f. Creating a column for ACTIVITIES column.

Create and insert an activity column between case_ID and TIMESTAMP. Insert a column ACTIVITY between columns CONTAINER_KEY and column DISC_DATE. Insert a column ACTIVITY between columns CONTAINER_KEY and column LOAD_DATE. Insert a column ACTIVITY between columns CONTAINER_KEY and column STACK_DATE and so on. In this study the name of the activity is not known, then the naming of activity adapted to the DATE column name. For example, naming DISC_ derived from DISC_DATE column, LOAD_ derived from LOAD_TIMESTAMP and so on. The purpose of the adjustment of the event name, is when sorted by CONTAINER_KEY will be clustered based on the activities performed.

g. Classifying each activity with the appropriate attributes.

We need to complete the activity with the corresponding attribute group. For example, the activity STACK_ equipped with attributes: YARD_BLOCK, YARD_SLOT, YARD_ ROW, and YARD_TIER.

h. Adjusting the order of the columns in accordance with the tree structure of event log.

We need to categorize the activities and attributes, then moved underneath to be merged. The columns are combined in accordance with the tree structure of the event log. Sorted as follows, case_ID, ACTIVITY, TIMESTAMP, and attributes according to the group activities Fig 3.

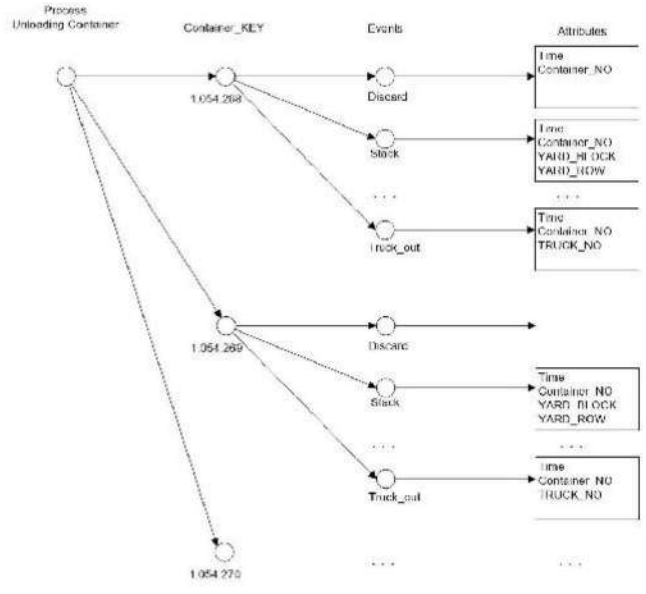

Fig 3. Structure event logs

i. Indexing by case_ID

The last work of the framework is the sort by case_ID. In the process of unloading containers, CONTAINER_KEY be case_ID in a case Table 2.

Table 2. Data event log

process instance

CASE_ID	ACTIVITIES	TIME_STAMP	CONTAINER_NO	TRUCK_NO	YARD_BLOCK
1.054.268	DISC_	01/01/2014 01:01:43	SPNJ2844660		
1.054.268	STACK_	01/01/2014 01:13:09	SPNJ2844660		A
1.054.268	SP2_KE_	01/01/2014 08:57:43	SPNJ2844660		
1.054.268	UPDATE_TIME	01/01/2014 10:03:14	SPNJ2844660		
1.054.268	TRUCK_OUT_	01/01/2014 10:43:17	SPNJ2844660	HT04	
1.054.269	DISC_	01/01/2014 00:59:24	SPNJ2846488		
1.054.269	STACK_	01/01/2014 01:13:54	SPNJ2846488		A
1.054.269	SP2_KE_	01/01/2014 08:57:51	SPNJ2846488		
1.054.269	UPDATE_TIME	01/01/2014 10:06:56	SPNJ2846488		
1.054.269	TRUCK_OUT_	01/01/2014 11:07:30	SPNJ2846488	HT03	
1.054.270	DISC_	01/01/2014 01:01:32	SPNJ2845554		
1.054.270	STACK_	01/01/2014 01:11:34	SPNJ2845554		A
1.054.270	SP2_KE_	01/01/2014 08:57:57	SPNJ2845554		
1.054.270	UPDATE_TIME	01/01/2014 10:02:43	SPNJ2845554		
1.054.270	TRUCK_OUT_	01/01/2014 10:43:27	SPNJ2845554	HT04	
1.054.271	DISC_	01/01/2014 00:01:46	SPNJ2846955		
1.054.271	UPDATE_TIME	01/01/2014 00:01:46	SPNJ2846955		
1.054.271	TRUCK_OUT_	01/01/2014 00:11:14	SPNJ2846955	HTTL	

The next stage is to convert into XES format. XES (eXtensible Event Stream) is a standard syntax for the operationalization of the event log[2]. So that the file XES is discovered for process mining. After the extraction of the structured form event log, we use the disco[3] tool to convert the file format into XES as shown in Fig 4 .

```
<trace>
   <string key="concept:name" value="1054197"/>
   <string key="creator" value="Fluxicon Disco"/>
   <event>
        ....
        <date key="time:timestamp" value="2014-01-0
        <string key="ACTIVITIES" value="Start"/>
        <string key="CONTAINER_NO" value="Start"/>
        <string key="TRUCK_NO" value="Start"/>
        <string key "YARD_BLOCK" value "Start"/>
        ....
   </event>
   <event>
```

Fig 4. Fragment of an XES file

VI. DISCUSSION

This discussion describes the important things in the log data extraction, i.e., selection process instance, determining case_ID, specified the name criteria of an activity, addition of attributes.

Discussion in term of selection process instance:

In selecting process instance we have to identify the business processes to be obtained from event data. Determination of the selection process instance is critically significant, because it determines the scope of data to be in the extraction. In the database or large log data, assorted process sequence spread in many tables and columns. For example, process instances: unloading containers, inspection of goods by customs, placement of containers in the yard, and others.

Discussion in term of determine case_ID:

In the structure of the event log there is an item that is referenced in any activity. We can instantly know the life-cycle of a process from a single case_ID. The case_ID is typically to maintain that the item is produced from beginning to properly complete.

If we are wrong in accurately determining case_ID, then we will find the potential error at the time of analysis. We know there was a possible mistake when we do process discovery. For example, if we choose CONTAINER_NO as case_ID, we will find in a process instance two containers. This is due to the fact, that the same container can be

[2] http://www.xes-standard.org/

[3] http://fluxicon.com/

repeatedly placed at same time. Therefore, the process of unloading container using CONTAINER_KEY as case_ID.

Discussion in term of making activity name:

It could be possible that an audit trail log data Table *1* as the data did not mention the name of the event. Hence, the title of the column that contains the timestamp can be used as the name of the event. Usually the name of the column headings represent the timestamp of events conducted. For example, the column headings LOAD_DATE means LOAD events.

Discussion of additional attributes:

Each event has attributes of data. In the audit trail of data Table *1* attributes associated with related events. But the data in the form of a data flat. Therefore, the data administrator is required to classify the data. We take the selected timestamp column and adjust any attributes associated with those events. The same thing can be done to look for linkages events with attributes.

VII. Conclusion and Future Work

Data extraction audit trails Terminal Operating Systems EXTOS is absolutely necessary for the practical need of process mining. The historic structure of the resulting event log will be determined to discover process mining. The event log is properly used to measure the performance of the dwell time process of unloading containers (container import dwell-time [6]). Framework extraction is properly used to avoid errors of analysis and verification [18]. Starting from the selection process of data instances, with identifying the activity and selection of the attributes is combined in accordance with their respective activities.

This empirical research is limited solely to the material processes in the port terminal. For future extraction research it is necessary to involve data messages from other organizations involved in this import dwell-time process. The import dwell-time process at the container terminal port typically involves several official organizations.

Acknowledgments

The author would like to thank Memet Ismartadianto and Wawan Dharmawan for their valuable contributions and support towards this case study at port container terminals, data collection and constructive discussion.

References

[1] W.M.P. der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, 1st ed., Springer Publishing Company, Incorporated ©2011, 2011.

[2] M. Jans, M. Alles, M. Vasarhelyi, The case for process mining in auditing: Sources of value added and areas of application, Int. J. Account. Inf. Syst. 14 (2013) 1–20. doi:http://dx.doi.org/10.1016/j.accinf.2012.06.015.

[3] W. van der Aalst, A. Adriansyah, A.K.A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J.C. Bose, P. van den Brand, R. Brandtjen, J. Buijs, A. Burattin, J. Carmona, M. Castellanos, J. Claes, J. Cook, N. Costantini, F. Curbera, E. Damiani, M. de Leoni, P. Delias, B.F. van Dongen, M. Dumas, S. Dustdar, D. Fahland, D.R. Ferreira, W. Gaaloul, F. van Geffen, S. Goel, C. Günther, A. Guzzo, P. Harmon, A. ter Hofstede, J. Hoogland, J.E. Ingvaldsen, K. Kato, R. Kuhn, A. Kumar, M. La Rosa, F. Maggi, D. Malerba, R.S. Mans, A. Manuel, M. McCreesh, P. Mello, J. Mendling, M. Montali, H.R. Motahari-Nezhad, M. zur Muehlen, J. Munoz-Gama, L. Pontieri, J. Ribeiro, A. Rozinat, H. Seguel Pérez, R. Seguel Pérez, M. Sepúlveda, J. Sinur, P. Soffer, M. Song, A. Sperduti, G. Stilo, C. Stoel, K. Swenson, M. Talamo, W. Tan, C. Turner, J. Vanthienen, G. Varvaressos, E. Verbeek, M. Verdonk, R. Vigo, J. Wang, B. Weber, M. Weidlich, T. Weijters, L. Wen, M. Westergaard, M. Wynn, Process Mining Manifesto, (2012) 169–194. doi:10.1007/978-3-642-28108-2_19.

[4] M. de Leoni, F.M. Maggi, W.M.P. van der Aalst, An Alignment-based Framework to Check the Conformance of Declarative Process Models and to Preprocess Event-log Data, Inf. Syst. 47 (2015) 258–277. doi:10.1016/j.is.2013.12.005.

[5] K.H. Kim, H.-O. Günther, Container terminals and terminal operations, in: P.K.H. Kim, P.D.H.-O. Günther (Eds.), Contain. {Terminals} {Cargo} {Systems}, Springer Berlin Heidelberg, 2007: pp. 3–12. http://link.springer.com/chapter/10.1007/978-3-540-49550-5_1 (accessed November 21, 2014).

[6] B. Jokonowo, R. Sarno, S. Rochimah, B. Priambodo, Process mining : measuring key performance indicator container dwell time, Indones. J. Electr. Eng. Comput. Sci. 16 (2019) 401–411. doi:10.11591/ijeecs.v16.i1.pp401-411.

[7] M. Amrou, B. Hassan, Process Mining for port container terminals : The state of the art and issues, ASD 2018 Big Data Appl. 12th Ed. Conf. Adv. Decis. Syst. (2018).

[8] S. Hartmann, Scheduling reefer mechanics at container terminals, Transp. Res. Part E Logist. Transp. Rev. 51 (2013) 17–27. doi:10.1016/j.tre.2012.12.007.

[9] E. Ursavas, A decision support system for quayside operations in a container terminal, Decis. Support Syst. 59 (2014) 312–324. doi:10.1016/j.dss.2014.01.003.

[10] X.F. Yin, L.P. Khoo, C.-H. Chen, A distributed agent system for port planning and scheduling, Adv. Eng. Informatics. 25 (2011) 403–412. doi:10.1016/j.aei.2010.10.004.

[11] H.J. Yeo, Competitiveness of asian container terminals, Asian J. Shipp. Logist. 26 (2010) 225–246. doi:10.1016/S2092-5212(10)80003-3.

[12] R. Rahardianto, R. Sarno, G. Intani Budiawati, Performance Time Evaluation of Domestic Container Terminal Using Process Mining and PERT, Proc. - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, ISemantic 2018. (2018) 469–475. doi:10.1109/ISEMANTIC.2018.8549768.

[13] R. Setiawan, R. Sarno, A.C. Fauzan, Evaluation of container forecasting methods for analyzing port container terminal performance using agent-based simulation, 2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018. 2018-Janua (2018) 286–291. doi:10.1109/ICOIACT.2018.8350718.

[14] M. Gaete G., M. C. González-Araya, R. G. González-Ramírez, C. Astudillo H., A Dwell Time-based Container Positioning Decision Support System at a Port Terminal, Proc. 6th Int. Conf. Oper. Res. Enterp. Syst. (2017) 128–139. doi:10.5220/0006193001280139.

[15] M. Acciaro, P. Serra, Strategic Determinants of Terminal Operating System Choice: An Empirical Approach Using Multinomial Analysis, Transp. Res. Procedia. 3 (2014) 592–601. doi:10.1016/j.trpro.2014.10.038.

[16] J. Claes, G. Poels, Merging event logs for process mining: A rule based merging method and rule suggestion algorithm, Expert Syst. Appl. 41 (2014) 7291–7306. doi:http://dx.doi.org/10.1016/j.eswa.2014.06.012.

[17] Y. Wang, F. Caron, J. Vanthienen, L. Huang, Y. Guo, Acquiring logistics process intelligence: Methodology and an application for a Chinese bulk port, Expert Syst. Appl. 41 (2014) 195–209. doi:http://dx.doi.org/10.1016/j.eswa.2013.07.021.

[18] C.A.M. Buijs, Mapping Data Sources to XES in a Generic Way, 2010. http://www.processmining.org/_media/xesame/xesma_thesis_final.pdf (accessed April 9, 2019).

[19] D. Calvanese, T.E. Kalayci, M. Montali, A. Santoso, OBDA for log extraction in process mining, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 10370 LNCS (2017) 292–345. doi:10.1007/978-3-319-61033-7_9.

[20] B. Jokonowo, Process Mining-Soundness Workflow Multi-organizations, in: n.d. http://ieomsociety.org/ieom2014/pdfs/219.pdf (accessed April 9, 2019).