Archives of
Transport System
Telematics

Volume 9

Issue 2

May 2016

The alpha algorithm in modeling of the ship's route

M. DRAMSKI
MARITIME UNIVERSITY OF SZCZECIN, Wały Chrobrego 1-2, 70-500 Szczecin, Poland
EMAIL: m.dramski@am.szczecin.pl

ABSTRACT

In this paper the use of alpha algorithm in modeling of the ship's route is described. Based on event log obtained from real data, a Petri net was created. This model let us to review the whole process of the ship's route and take some interesting observations.

KEYWORDS: keywords: process mining, event log, ship's route, alpha algorithm, Petri nets, operational support

1. Introduction

Process mining became one of the most interesting domains in the last few years. This is caused by the huge amount of data that is produced every day and everywhere [1].

In this paper a ship's route was treated as a process. This process is several dozen of cases which consist of some activities. The route data was obtained from Marinetraffic.com web service, so there was a possibility to create a model based on real data.

First thing needed in the process of a construction of a model is to have an event log. This data structure has a form of simple XML file and is the basis for each popular tool (such ProM or Disco) in process mining. Having the event log, the construction of the process model is possible. In this paper a Petri net was created using very popular alpha algorithm.

2. Event log

According to the definition in [1], the event log is the basics of the process mining techniques. It consists of traces and the traces consist of events. So the detailed definition can be described as follows:

A trace is a finite sequence of events σϵε* such that each event appears only once. An event log is a set of cases $L \subseteq \mathbb{C}$ such that each event appears at most once in the entire log. The typical event log contains: case id, event names, timestamp, resources etc.

The example of the event log can be given as follows:

$$L=[\langle a,b,c\rangle,\langle a,b,d\rangle,\langle b,b,c\rangle] \quad (1)$$

The given equation (1) tells us that the process consits of three cases and each case consists of three events. These events are: a, b, c and d. This is a very good way to describe the event log. The form of the equation is a very intuitive description of the process.

2.1. Event log of the ship's route

As mentioned in the introduction, real data was applied to build a process model. This data was obtained based on the real ship's route thanks to the web service Marinetraffic.com [4]. The route chosen is very interesting because the ship travels around the world (started from Szczecin in September 2015 and it still continues it's journey).

At the beginning it is necessary to look at the set of events which will be applied to create a process model. These events are mentioned in the table below. The first column is an ordinal number, the second is the event name and the last one is a shortcut of the event name (shortcuts are used to make the model more clear and visible).

Table 1. The set of the events based on the data obtained from Marinetraffic.com

No.	Event name	Shortcut
1	In Range	a
2	Changed Course	b
3	Arrival	c
4	Departure	d
5	Stopped	e
6	Underway	f
7	Midnight position	g
8	Midday position	h

It was assumed that each case of our event log corresponds to one day of the ship's route. This kind of approach lets us to see what activities are executed during a typical day on the sea. Naturally, it makes sense, when the route is different each day. The same activities in each case are typical for the regular routes such passenger ferries etc. so we can suppose that the model of the proces will be simple. In this paper the first 30 days of the route was considered and the event log obtained looks like the following:

$$\begin{aligned}L=[&\langle a,f,c,h,d,e,g\rangle,\langle a,f,b,b,b,b,c,d,g\rangle,\langle a,h\rangle,\langle g,c,e,d,c,h,d,f,c,c,d,a,g\rangle,\\ &\langle c,d,a,h,d,a,g\rangle,\langle a,h\rangle,\langle g,e,f,b,c,h,g\rangle,\langle h,d,g\rangle,\langle a,h,c\rangle,\langle a,d,h\rangle,\\ &\langle g,h,c,d,e,f,b,e,g\rangle,\langle a,f,e,g\rangle,\langle a,f,e,f,e,c,f,d,h\rangle,\langle g,h\rangle,\langle g,h\rangle,\\ &\langle a,h,c,g\rangle,\langle h\rangle,\langle g,h\rangle,\langle g,d,h\rangle,\langle g,e,f,c,h,g\rangle,\langle h,g\rangle,\langle d,c,e,f,h\rangle,\langle g,c,h,d\rangle,\\ &\langle g,e,f,b,b,h\rangle,\langle g,e,f,e,f,b,e,f,e,c,h\rangle,\langle g,h\rangle,\langle a,f,d,b\rangle,\langle g,h\rangle,\langle g\rangle,\langle g,a\rangle]\end{aligned} \tag{1}$$

As it can be seen each case (except some short cases consisted of the events g and h) is different. It tells us that the route day doesn't always mean the same activities. It is natural for long routes across the big water areas such seas, oceans etc. If the ship doesn't enter into the port or the restricted area, it can be assumed that the route is simple and there is no need to consider a lot of activities. This observation can be confirmed in practice by observing different ship's routes (among others using Marinetraffic.com).

Each case consists of some activities. Sometimes there are only two or even one activity. The order is also not the same every time. Anyway, some regularities can be observed e.g. e is very often followed by f, b can be repeated few times, some cases are the same etc.

2.2. The XES data format

Initially, the natural way to express the processes was the XML format. This approach let to represent the data in well organized and easy to analyze way.

Event logs, as they occur in practice and research, can take a lot of different forms. Each system can have it's own architecture and can use own data formats. The conclusion is that the lack of the standard, causes a lot of problems in process mining. So, the XES (Extensible Event Stream) format was involved [7].

The XES standard is a XML-based format which is characterised by these four features:

- simplicity - there is a need to use the simplest possible way to represent information, XES files should be generated easily and fast;
- flexibility - XES should capture logs from different environments, not looking at the domain etc.;
- extensibility - it must be easy to add some features to the standard in the future;
- expressivity - the information can't be lost or the loss should be as minimal as possible.

The very detailed description of the XES data format can be found at [7].

XES is not the unique form to express the event log. Sometimes other formats are used such CSV, text files etc. But why XES is so important ? The answer is simple. The most of popular tools such ProM [6] or Disco [3] support XES. Some tools support only this format (ProM).

The data showed in this article was initially written as MS Excel spreadsheet. Excel files are not supported by the tools, so the conversion to XES format was applied. This conversion was carried out using the XLS2XES Converter tool which was originally developed by the author of this paper in Python programming language. The details of the conversion are not the subject of this paper - it will be published in the nearest future.

3. The alpha algorithm

The α-algorithm is the simple way to obtain the Petri net from the event log. The input is of course the event log and the output is the set of places, transitions and arcs between them. This approach scans the event log for particular patterns. For example, if a is followed by b but b is never followed by a, then it is assumed that there is a causal dependency between a and b.

Typical patterns found by the α-algorithm are:

- sequence patterns;
- XOR-split and XOR-join patterns;
- AND-split and AND-join patterns.

Let L be an event log over $T\subseteq\mathcal{A}$,$\alpha(L)$ is defined as follows [1]:

$$T_L = \{t \in T \mid \exists_{\sigma\epsilon L} t\epsilon\sigma\} \tag{3}$$

$$T_I = \{t \in T \mid \exists_{\sigma\in L} t = first(\sigma)\} \tag{4}$$

$$T_O = \{t \in T|\exists_{\sigma\in L} t = last(\sigma)\} \tag{5}$$

$$\begin{gathered}X_L = \{(A,B)|A \subseteq T_L \land A \neq \emptyset\ \land B \subseteq T_L\ \land B \neq \emptyset\ \land\\ \land\ \forall_{a\in A}\forall_{b\in B} a \rightarrow\ _L b\ \land\ \forall_{a_1,a_2\in A} a_1\#\ _L a_2\ \land\ \forall_{b_1,b_2\in B} b_1\#\ _L b_2\}\end{gathered} \tag{6}$$

$$Y_L = \left\{(A,B) \in X_L|\forall_{(A',B')\in X_L} A \subseteq A' \land B \subseteq B' \Rightarrow (A,B) = (A',B')\right\} \tag{7}$$

$$P_L = \left\{p_{(A,B)}|(A,B) \in Y_L\right\}\ \cup \{i_L, o_L\} \tag{8}$$

$$\begin{gathered}F_L = \left\{\left(a, p_{(A,B)}\right)\middle|(A,B) \in Y_L \land a \in A\right\} \cup \left\{\left(p_{(A,B)}, b\right)|(A,B) \in Y_L\ \land\\ \land\, b \in B\right\}\ \cup \{(t, o_L)|t \in T_o\}\end{gathered} \tag{9}$$

$$\alpha(L) = (P_L, T_L, F_L) \tag{10}$$

The equations above can be described as follows:

1. Each activity in L corresponds to a transition in $\alpha(L)$.
2. Fix the set of start activities.
3. Fix the set of end activities.
4. Find pairs (A,B) of sets of activities such that every element $a \in A$ and every element $b \in B$ are casually related.
5. Delete from the set X_L all pairs (A,B) that are not maximal.
6. Determine the places, transitions and arcs for the final Petri net model.

The *α-algorithm* has some week points: can't detect implicit places, has problems with detecting the loops etc. [vdaalst]. Besides for each the same two event logs it always results with the same model.

In this paper an application of this algorithm is presented. In the next section the model of the process obtained using ProM tool is described.

4. The model of the process

Fig. 1. ProM main window [6]

As mentioned in the previous section, for creating the process model, ProM[1] tool was used. This is an extensible framework that supports a variety of process mining techniques in the form of plug-ins. ProM is continously developed and other functionalities are beeing added. The tool is written in Java programming language and thanks to this feature, is accessible for all platforms supporting Java Virtual Machine. The other advantage is that ProM is free of charge. At the Fig. 1 the main window of ProM is illustrated.

4.1. The model

First of all it is necessary to import the event log (in XES format) to ProM. It can be done by using import button in the top right corner of the main window. We have to be sure that our XES file is correct. One of the most significant defects of ProM is that even the smallest error causes the failure of import process. This is the reason why the conversion from other formats has to be done without any errors. Other known and widely used tools (not only in process mining) have some capabilities to ingore or even correct errors. ProM unfortunately not.

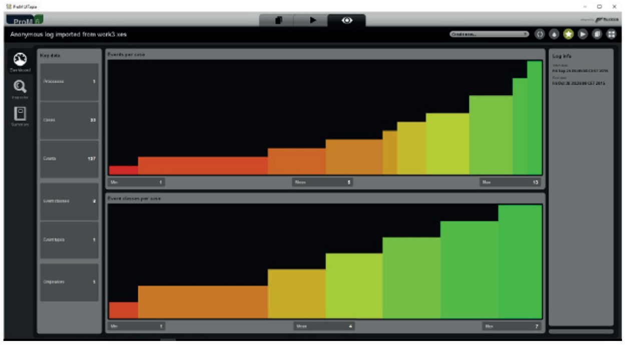

Fig. 2. The event log imported into ProM [own study based on: 6]

At the Fig. 2 the result of importing the event log is presented. ProM offers some useful functions to do some pre-processing analysis. It can be read that this event log has 30 cases with the total number of events equal to 137. The 8 classes of events are visible, so this is the confirmation that the conversion to XES format was done correctly.

Now the Petri net can be constructed. ProM offers the plug-in called AlphaMiner and this tool was used. The obtained model is illustrated at the Fig. 3

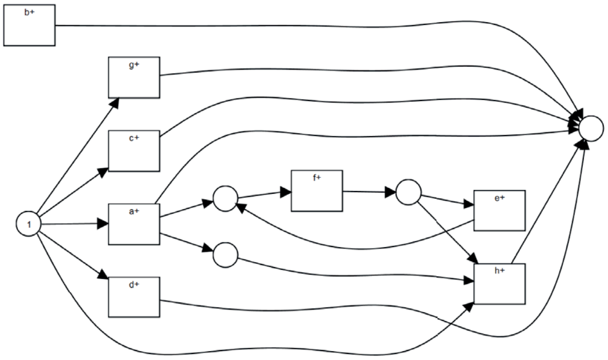

Fig. 3. The model obtained using α-algorithm (30 cases) [own study]

The figure above shows one of the defects of -algorithm. In the event log we can find the fact that event b occurs in the loops. Besides it is never the start or end event. The AlphaMiner plugin couldn't detect these loops. The model says that the execution of event b is impossible. No arc is incoming into this transition, so the token can't arrive here. Also some cases are impossible such No further analysis is required (but of course can be carried out) to confirm that the model is not good. What are the solutions ? Other algorithms can be used, some manual corrections can be made etc. But here we try to increase the total number of cases to 59. And the obtained model looks like at the Fig. 4.

By adding new cases some problems were solved. First of all there is no non-reachable transitions. Besides the event b can occur in loops.

[1] More informations about ProM and it's plug-ins can be found at http://www.promtools.org

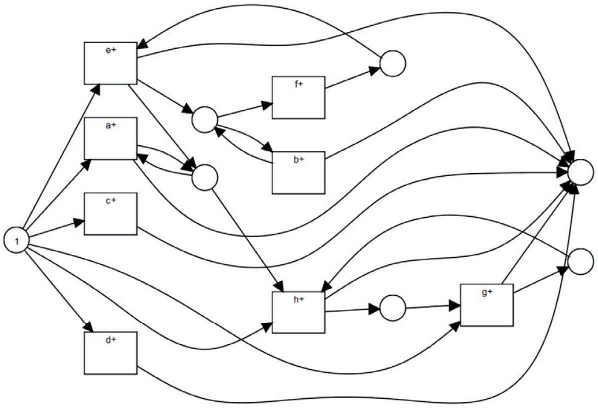

Fig. 4. The model obtained using α-algorithm (59 cases) [own study]

4.2. The evaluation of the model using footprints

At this point the model is evaluated using the simple method of footprints. Two tables are created - for the model and for the event log. The evaluation is to compare these footprints.

Table 2. The footprint of the event log (59 cases) [own study]

	a	b	c	d	e	f	g	h
a	#	#	→	||	→	→	||	||
b	#	||	→	←	→	←	#	→
c	←	←	||	||	||	||	||	||
d	||	→	||	#	||	||	||	||
e	←	←	||	||	#	||	||	→
f	←	→	||	||	||	#	#	||
g	||	#	||	||	||	#	#	||
h	||	←	||	||	←	||	||	#

Now, the footprint of the model is necessary:

Table 3. The footprint of the process model (59 cases) [own study]

	a	b	c	d	e	f	g	h
a	||	#	#	#	#	#	||	→
b	#	||	#	#	#	→	#	#
c	#	#	#	#	#	#	#	#
d	#	#	#	#	#	#	#	#
e	#	→	#	#	#	||	#	→
f	#	←	#	#	||	#	#	#
g	#	#	#	#	#	#	#	||
h	←	#	#	#	←	#	||	#

The quality of the model can be calculated using the formula:

$$Q = \frac{64 - 45}{64} = 0.296875 \tag{11}$$

5. Conclusion

As it has been proven using the equation (11), the model is underfitted. The footprints are coherent only in 19 cells of 64. It means that less then one third of traces in the event log is modeled in the proper way.

The α-algorithm is not able to design the model with repeated transitions. Each of them occurs only once.

Besides, the choose of the data is also significant. This time each day of the ship's route was treated as a single case. There's always a possibility to do some changes in this field. There is a need to report that some events in the event log occured few minutes before the midnight or few minutes after. This coincindence can also influence on the event log and the way how the traces look like.

In this paper the use of the α-algorithm in modeling of the ship's route was described. The results tell that there is a need to do more research, but initial ones seem to be very promising. It is necessary to think about the organization of the event log.

The improvement of model's quality is very important. Process mining can be very helpful in the transport. It provides a strong operational support [2] which can influence on the economic factors of transport companies.

Bibliography

[1] v.d. AALST W.M.P., Process mining: discovery, conformance and enhancement of business processes, Springer-Verlag Berlin Heidelberg 2011.

[2] DRAMSKI M., Wsparcie operacyjne w transporcie w kontekście process mining, Logistyka 3/2015.

[3] http://www.fluxicon.com [date of access: 12.12.2015].

[4] http://www.marinetraffic.com [date of access: 12.12.2015].

[5] http://www.processmining.org [date of access: 12.12.2015].

[6] http://www.promtools.org [date of access: 12.12.2015].

[7] http://www.xes-standard.org [date of access: 12.12.2015].