



## Integrating dependency with DEVS in the process mining

Yan Wang, Grégory Zacharewicz, David Chen, Mamadou Kaba Traoré

### ► To cite this version:

Yan Wang, Grégory Zacharewicz, David Chen, Mamadou Kaba Traoré. Integrating dependency with DEVS in the process mining. NICST'2015 New Information Communication Sciences and Technology for Sustainable Development, Sep 2015, Bordeaux, France. hal-01551430

**HAL Id: hal-01551430**

**<https://hal.science/hal-01551430>**

Submitted on 30 Jun 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Integrating dependency with DEVS in the process mining

Yan Wang<sup>(a)</sup>, Grégory Zacharewicz<sup>(b)</sup>, David Chen<sup>(c)</sup>, and Mamadou Kaba Traoré<sup>(d)</sup>

<sup>(a),(b),(c)</sup>IMS, University of Bordeaux, 33405 Talence Cedex, France

<sup>(d)</sup>LIMOS, Université Blaise Pascal, 63173 Aubiere Cedex, France

<sup>(a)</sup>yan.wang@etu.u-bordeaux.fr, <sup>(b)</sup>gregory.zacharewicz@u-bordeaux.fr,  
<sup>(c)</sup>david.chen@ims-bordeaux.fr, <sup>(d)</sup>traore@isima.fr

**Abstract.** Process mining is used as an intelligent technique to improve process model by extracting event log. DEVS is specified as one of the resulting process model for simulation and it is transformed from transition system by using region-based approach in the process mining. However, the discovered DEVS model is pretty lacking and has a negative effect on the overall process flow. This paper reviews current approaches and proposes a time-related method dependency. Using an example of event logs, we calculate the value of the dependency and integrate with DEVS model. We also discuss about the time criteria.

**Keywords:** DEVS; process mining; dependency; time

## 1 Introduction

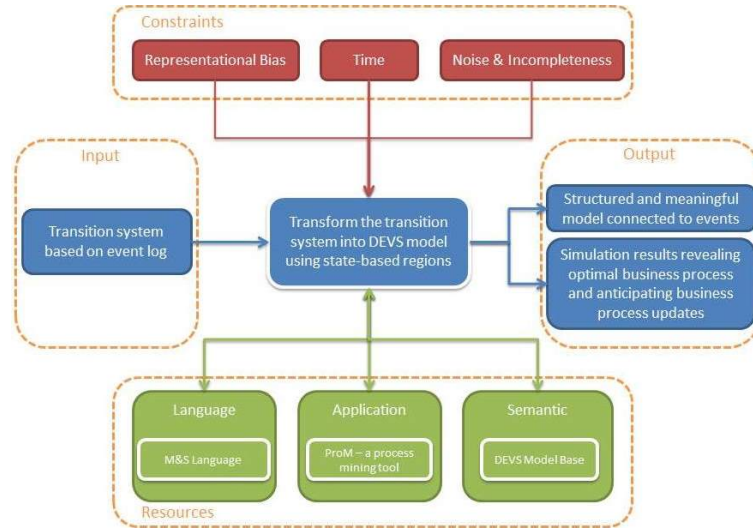
Nowadays, the development of the information systems cannot be separated by the operational process. Moreover, more and more events with abundant process information are recorded. Despite the omnipresence of event data, most of the organizations analyze and build models based on expert assumption rather than observed fact. Thanks to process mining[1], it provides the methods to discover monitor and improve actual processes by using event data to extract process-related information. Process mining combines process model-driven approaches with data mining. It includes process discovery, process conformance and process enhancement.

The Discrete Event System Specification (DEVS)[2] provides a hierarchical and modular formalism to describe the event system. It contains the basic components like explicit time, inputs, outputs, states and functions. DEVS is also specified as the resulting process model in the process discovery[3]. As DEVS has the advantage of explicit time and separating model simulation, a region-based approach is proposed to convert the transition system directly to DEVS. However, there are still two missing part which need to be enriched for the DEVS model in process mining, DEVS semantic and considered time. The DEVS process model is not strongly controllable to simulate business process, meaning that no controller or methods provide the desired states without choosing some

bad states. It presents all the possible business process in the DEVS model without considering how to find the next transition. The objective of this paper is to integrate the dependency relation with DEVS model which has a strong relationship with the time. The new method provides more precise information for transitions.

First we will depict the whole structure of DEVS model in the process mining to provide a general background of this research in section 2. And the previous work region-based approach will also be introduced. Section 3 will give some related researches. Section 4 will explain how to calculate the dependency in DEVS model and how it works. At last we conclude the paper in section 5.

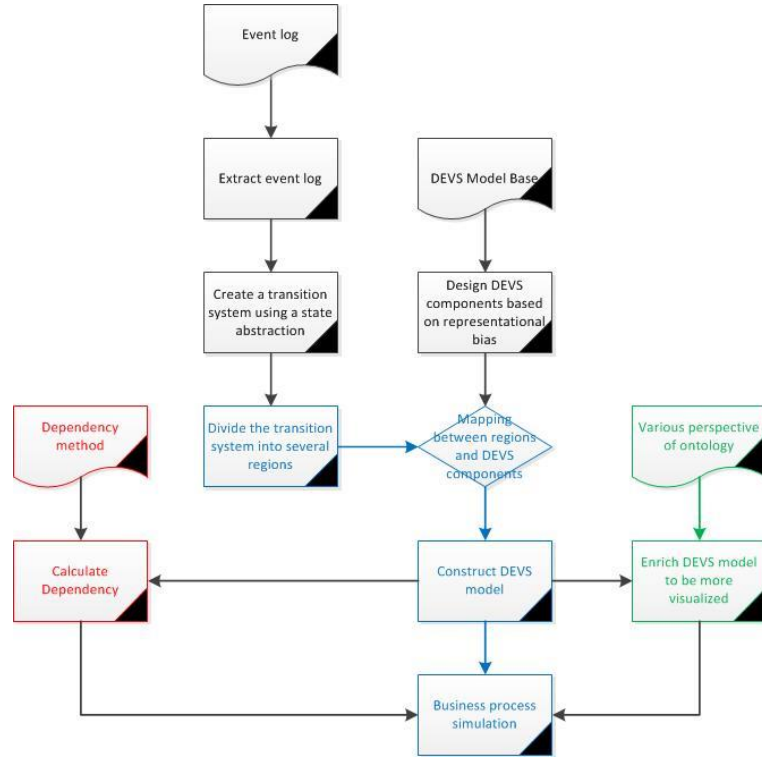
## 2 DEVS in process mining



**Fig. 1.** DEVS-based process mining structure

Figure 1 shows the general background of the whole research. We reuse the discovery technique and take the transition system instead of Petri Nets as input. The transition system is discovered by the traces in the event log. In the resource part, we need to provide the DEVS model base, in another words semantic DEVS to support the region-based approach. The process mining tool ProM is used as the development environment for simulation. The representational bias, noise and incompleteness[1] are the main constraints when we try to transform the event log into other formalism with the optimal and desired results. According to the representational bias, DEVS model base provides several solutions. For example, parallel DEVS models with multiple inputs and outputs and time segments can solve the problem of concurrency, a switch network can

solve the problem of OR-splits, generator coupled model can solve the problem of loop and duplicate actions. All these models will be designed in the DEVS model base. The noise defines event log contains rare and infrequent behavior not representative for the typical behavior of the process. The incompleteness defines event log contains too few events to be able to discover some of the underlying control-flow structures. Besides, time is another very important constraint which will lead to desired state scheduling. This paper focuses on providing new time-related methods to develop the DEVS models. At last, we wish to implement a structured and meaningful model used for anticipating optimal business process.



**Fig. 2.** Main process based on the structure

Figure 2 shows the whole process of Modeling & Simulation by integrating DEVS model with process mining. The blue part of the figure has already done in the last paper. First we need to extract the useful information from the list of event log. The event log is recorded by the XES standard. Using the state abstraction, we can automatically construct a transition system based on the event log. Meanwhile, it is necessary to construct the DEVS model base. It has atomic model and coupled model with special functions which we have talked

in the last paragraph. These models will be used as components to support the transformation. The main part of this approach is to implement the transformation between transition system and DEVS model. Table 1 shows the mapping between transition system and DEVS. To implement the transformation, we divide the transition system as several regions. In this case all activities can be classified into entering the region, leaving the region and non-crossing. An activity cannot be entering the region in one part of the transition system and exiting the region in some other part. As DEVS model has the characteristic of separating model simulation, each region can be mapped with one component in the model base, each activity will be recognized as input event or output event. Then all the components will be connected by the internal or external transitions to build a complete DEVS model. The green part of the figure shows the extension of ontology based on various perspectives (organizational perspective, time perspective, data perspective, etc.) to be more meaningful and visualized like a map. So customers can simulate business process, identify various components and supervise the operating process. The red part of the figure is the main work of this paper and it will be shown in more detail in section 4.

**Table 1.** Mapping between transition system and DEVS

Transition System	DEVS
Region	Atomic model
State	State
Transition	Internal transition
	External transition
Activity	Input event
	Output event

### 3 Related research

In the process mining[1], heuristic mining algorithms try to discovery the dependency graph from the event log. It first offers a list of event log examples under limited time. Then it calculate the frequency between any pair of activities and the value of the dependency relation based on dependency measure. At last it derives the dependency graph using a threshold and the states are connected with the dependency.

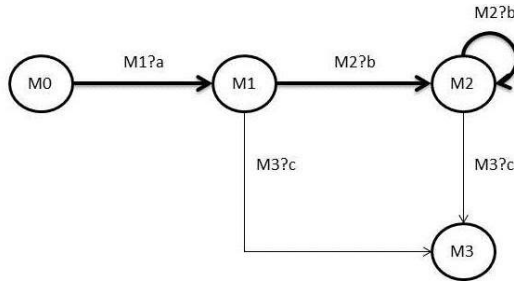
In order to mine useful information , many researchers propose new approach and strategy to address the issue based on the dependency graph. Das et al.[4] propose the feature selection approaches to reduce the complexity of the overall process by allowing the data mining system to focus on what is really important. The proposed method encompasses a new strategy in dimensionality reduction by attribute clustering based on the dependency graph of the attributes. The method strongly based on the graph theory and information gain, an established theory of measuring uncertainty and quantified the information contained in the system. Zimmermann et al..[5] use the complexity of dependency graphs

for predicting failures at statistically significant levels. This helps for resource allocation and decision making.

Some researchers use dependency graph to represent different respectives. Wang et al.[6] propose a novel document representation model and a new similarity measure for documents which calculates the similarity based on dependency graphs. The similarity measure is applied to the Group average Agglomerative Hierarchical Clustering (GAHC) algorithm and promising results are obtained in the document clustering experiments.

#### 4 Dependency based DEVS model

The discovered DEVS process model in the process mining is not comprehensive. For example, there are two transitions which start from the same atomic model but point to the different destination. We don't know which transition we can choose so we can't predict the business process based on the current DEVS model. In order to find the desired state and the next anticipating transition, we propose a method based on dependency with the criteria of time.



**Fig. 3.** Main process based on the structure

Consider an externally coupled system  $N = \langle X, Y, M, EIC, EOC, IC \rangle$ , where  $M_i \in M$  is an atomic model  $M_i = \langle X_i, A_i, Y_i, \delta_{ext,i,j}, \delta_{int,i,j}, \lambda_i, ta_i, D_{i,j} \rangle$  where  $j$  is the destination atomic model.  $D_{i,j}$  is the value of the dependency for each of the  $\delta_{ext,i,j}$  or  $\delta_{int,i,j}$ . Given a proper time duration, we can get a list of event logs for example  $L = [ (a,b)^5, (a,b,c)^3, (a,c)^{10}, (a,b,b)^5 ]$ .  $a,b,c$  are different activities and they will be transformed into external or internal events. According to the methods in [3], we can get the DEVS process model as shown in figure 3.  $M_0, M_1, M_2, M_3$  are the atomic models and the arcs are external transitions. The frequencies of external event and internal event in the same atomic model are the same because the internal event will absolutely invoke the external event. Assuming all the activities in the example are external events and we can calculate the frequencies:  $\delta_{ext,0,1} = 23$ ,  $\delta_{ext,1,2} = 13$ ,  $\delta_{ext,2,2} = 5$ ,  $\delta_{ext,2,3} = 3$ ,  $\delta_{ext,1,3} = 10$ . Based on the equation 1 and 2 which are inherited from heuristic mining algorithm[1], we can calculate the value of dependency:  $D_{0,1} =$

0.96,  $D_{1,2} = 0.93$ ,  $D_{2,2} = 0.83$ ,  $D_{2,3} = 0.75$ ,  $D_{1,3} = 0.91$ . If  $D_{i,j}$  is close to 1, then there is a strong positive dependency between  $i$  and  $j$ . If  $D_{i,j}$  is close to -1, then there is a strong negative dependency between  $i$  and  $j$ . Using dependency, figure 3 provides a more intuitive visualization. The thickness of the arcs corresponds to the frequencies of the corresponding paths. Such visualizations are important to get insight into the desired process flow.

$$D_{i,j} = \frac{\delta_{i,j} - \delta_{j,i}}{\delta_{i,j} + \delta_{j,i} + 1} \quad \text{if } i \neq j \quad (1)$$

$$D_{i,j} = \frac{\delta_{i,i}}{\delta_{i,i} + 1} \quad \text{if } i = j \quad (2)$$

Meanwhile, we can also get the time duration from the event log. We select the earliest event time and the latest event time from the list of event logs as the time duration  $ta_i$ . We make a criteria of the time that if the state is in the  $ta_i$ , we will take  $D_{i,j}$  into consideration. If not, we will reduce the value of  $D_{i,j}$ .

## 5 Conclusion

This paper introduces the whole structure of specifying DEVS model as the resulting process model in the process mining. The original DEVS model can be built up based on the region-based approach. Then this paper focuses on the extension of DEVS model on time-related method. In order to predict the desired process flow, it proposes a dependency method with providing a more intuitive visualization based on a study case. Time criteria is also taken into consideration.

## References

1. Van der Aalst W.M.P.: Process mining: discovery, conformance and enhancement of business processes. Springer Science & Business Media, pp. 1-25 (2011)
2. Wainer G.A., al.: Discrete-event modeling and simulation. Chapter 16, an introduction to DEVS standardization. CRC Press, pp. 393-425 (2010)
3. Wang Y., Zacharewicz G., Chen D., Traoré M.K.: A proposal of using DEVS model for process mining. The 27th European Modeling and Simulation Symposium(2015)
4. Das A.K., Sil J., Phadikar S.: Attribute Clustering and Dimensionality Reduction Based on In/Out Degree of Attributes in Dependency Graph. Second International Conference, SEMCCO, pp. 372-380(2011)
5. Zimmermann T., Nagappan N.: Predicting Subsystem Failures using Dependency Graph Complexities. ISSRE '07 Proceedings of the 18th IEEE International Symposium on Software Reliability, pp. 227-236(2007)
6. Wang Y., Ni X., Sun J.T., Tong Y., Chen Z.: Representing document as dependency graph for document clustering. CIKM '11 Proceedings of the 20th ACM international conference on Information and knowledge management, pp. 2177-2180(2011)