Aspect Mining in Business Process Management

Authors:

Amin Jalali

This is the authors version of a work that was submitted/accepted for publication in:

Björn Johansson, Bo Andersson, Nicklas Holmberg (Eds.)
Perspectives in Business Informatics Research
13th International Conference, BIR 2014, Lund, Sweden,
September 22-24, 2014. Proceedings
Lecture Notes in Business Information Processing, Volume 194,
Springer International Publishing, Sweden, pp. 231-245.

The original publication is available at SpringerLink

Aspect Mining in Business Process Management

Amin Jalali

Department of Computer and Systems Sciences
Stockholm University
aj(at)dsv.su.se

Abstract. Automatic discovery of process models from event logs is an important and promising area in Business Process Management. Process models document how business processes should be performed, so they capture different concerns related to business processes. Some of these concerns are not limited to one process model, and they are repeated in many others as well, called cross-cutting concerns. Although many works have been done to enable discovering different process models, there is no investigation about how models with cross-cutting concerns can be discovered from event logs. Therefore, this work proposes an approach to enable discovering these models from event logs. The investigation is performed based on a case-study from the banking domain. The result shows how these concerns hinder existing process discovery techniques, and how the proposed approach can solve the problem.

Keywords: Process Mining, Aspect Mining, Process Discovery, Business Process Management, Aspect Oriented

1 Introduction

Automatic process discovery is an important area of research in the Business Process Management (BPM), since it enables discovering process models from what really have happened rather than what people claim. The reality is recorded in event logs, so process discovery algorithms aim to find the process models from these logs, which results in discovering the models that have been really followed in a business, called de-facto models. Traditionally, processes are discovered through interviews and other techniques, which results in models that describe what should be done rather than what is really happening. These models are called de-jure models. The comparison between these models can help businesses to discover the deviations between the reality and the goals, which points out opportunities for re-engineering business processes. Therefore, it is very promising and important to discover process models from event logs in BPM area.

Process models document various aspects of a business process, which fulfil different concerns. For example, a process model might incorporate the definition of security concerns that should be followed in a business process. Charfi et al. [14] mention auditing, authorization, privacy, and many other concerns as complementary aspects, which should be considered when designing a process

model. Therefore, a process model might contain aspects which are in common between different process models. These aspects are *scattered* in process models, and each process is *tangled* to them. The concerns which introduce the scattering and tangling relation to process models are called *cross-cutting concerns*. These concerns hinder maintenance and managing changes in process models. They also decrease the re-usability of realization of concerns in different process models. Moreover, they challenge enforcing process models to be comply with rules, since a change might be forgotten to be applied in a part of the process which reflects such a concern. Therefore, these concerns should be separated from process models [10, 14, 21].

To separate cross-cutting concerns from process models, Aspect Oriented Business Process Management (AOBPM) introduces a new modularization technique to encapsulate these concerns into modules. It defines how these modules should be related to each other to solve both scattering and tangling problem. As a result, models are less complex, more reusable and easier to maintain [10, 14, 21]. The models can be combined together in order to support the business process enactment, called weaving [22]. Currently, the whole BPM lifecycle is supported for AOBPM, which enables separation of cross-cutting concerns from business processes.

Although the complete BPM lifecycle is enabled by modeling and execution of aspect oriented business process models, the starting point is the design phase, where the designers model business processes using this approach. Indeed, it applies a lot of cost for organizations that already have process models, which are designed in a traditional way. Furthermore, it is not possible to completely discover such models from process logs using existing process mining techniques, since none of them can completely discover process models with duplicate activities [3] - which are scattered across one or different models. This limitation also hinders process mining area since it is common to have process models with activities which are repeated in a model, and this limits the area of application of process mining.

Therefore, this paper investigates how models with cross-cutting concerns can be discovered from event logs. It starts by investigating the problem through a banking case study in which it will be shown how current process mining techniques cannot discover process models that have cross-cutting concerns in section 2. Then, this paper proposes a method with which such process can be discovered in section 3. The method is later applied and demonstrated through the same case study in section 4. Moreover, a critical discussion on related works are given which show why those approaches cannot solve the problem in section 5. Finally, section 6 concludes the paper and shows the direction for future works.

2 Case Study

In this section, we can see how cross-cutting concerns in business processes hinders current process mining techniques when discovering process models from event logs. Thus, this section starts by introducing a dealing process of a bank.

Then, it shows how current techniques in process mining cannot discover this process based on its event logs.

2.1 Case Description

This section describes the scenario of the selected case from the financial domain. The banking case was selected because there are many cross-cutting concerns in banking domain like securty, privacy, logging, etc. To choose appropriate processes, i.e. fairly simple yet representative processes with at least a couple of cross-cutting concerns, an interview with a domain expert from a bank was conducted . For the confidentiality reasons, the bank asked to be remained anonymous. Two processes were selected. Here, one of them is presented namely the *Change asset deal* process. Detailed information about the process was derived through a follow-up interview with the same domain expert.

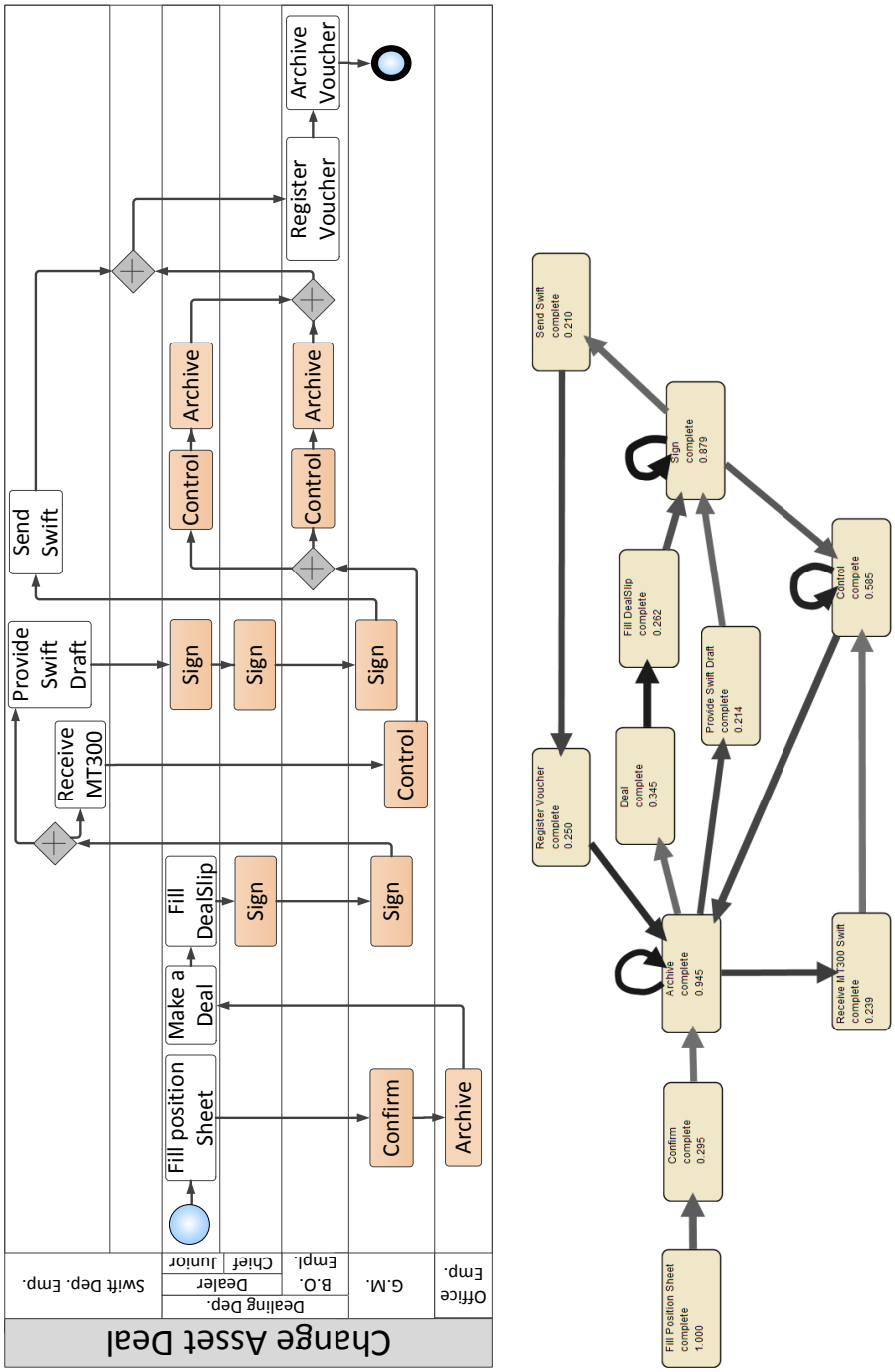
Generally, the assets of the bank are in two forms, cash and non-cash. Cash assets are either in the form of the account balances of the bank or the marketable securities. The `Change asset deal` process (see Figure 1(a)) handles deals for exchanging assets of the bank from one currency to another. The process starts with a *junior dealer* filling in a position sheet (`Fill position sheet` activity). Then, the *general manager* confirms the position sheet, which will be archived by the *office employee*. Then, a *junior dealer* makes the deal and fills in a deal slip. Next, both a *chief dealer* and the *general manager* sign the deal slip.

Afterwards, two parallel sets of activities are performed. On the one hand, the dealt amount of money is sent to the external partner of the deal. For this, first an *employee of the Swift department* provides a swift draft for sending the money. Then, for security purposes, the *junior dealer*, *chief dealer* and *general manager* sign the swift draft. Finally, an *employee of the Swift department* sends out the swift. On the other hand, the dealt amount of money should be received. This part starts when an *employee of the Swift department* receives an MT300 swift message. The *employee* sends this message to the general manager. The *general manager* makes an order to the Back office department and to the dealer to control the swift message. These messages are controlled separately. When each one of them has been controlled, the messages are archived (separately). When the deal is made, a *back office employee* registers a voucher in the accounting system. Finally, the deal is archived.

As it can be seen, the activities that represent cross-cutting concerns are repeated in this model. The event logs which is the result of execution of this model has also repetitive activities. Next section shows the result of applying the current process mining algorithm on this log.

2.2 Discovering the Case Study Process

This section shows the result of applying two process mining algorithm on event logs of the case study. The log was cleaned, so it is noise free. The log was also structured according to the eXtensible Event Stream (XES) standard, adopted



(a) Change Asset Deal Process

(b) The process model discovered through applying Fuzzy Miner

Fig. 1. Case Study Result

by the IEEE Task Force on Process Mining [2]. In this standard, each event represents the execution result of an activity. The sequences of events which are related to a specific case, i.e. the process instance, are grouped into a trace. The events in a trace are ordered according to their execution times. Therefore, the event log consists of number of traces that represent different execution scenarios of a process model.

Two process mining algorithms are selected to be applied on the log, i.e. Fuzzy algorithm and Alpha algorithm. Fuzzy algorithm is selected since it can discover process models from event logs with noise [17]. Alpha algorithm is used since it supports discovering process models from noise free event logs [4].

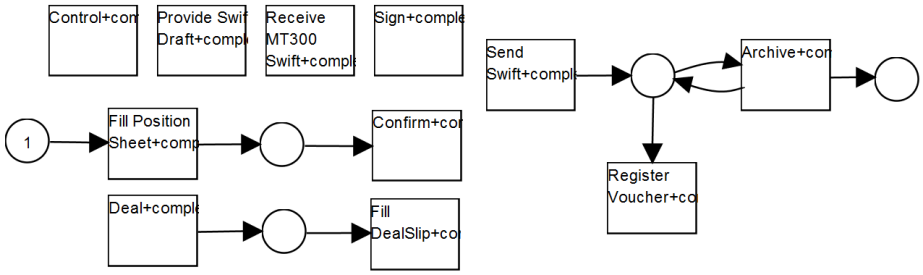


Fig. 2. The process model discovered through applying Alpha Miner

Fig. 1(b) shows the process model that is discovered through using Fuzzy Miner in ProM [16]. The fuzzy miner is the plug-in that implements the fuzzy algorithm to discover process models from event logs which might have noise. Although the result is a process model, it does not represent the correct process that generates the log (compare it with Fig. 1(a)). Indeed, the discovered process is not correct and the result is useless.

Fig. 2 shows the process model that is discovered through using Alpha Miner in ProM. The alpha miner is the plug-in that implements alpha algorithm to discover process models from a noise free event logs. As it can be seen, the discovered process is fragmented, and it cannot show the proper control-flow of the process.

In the next section, this paper proposes an approach with which process models with cross-cutting concerns can be discovered from event logs.

3 Approach

This section proposes an approach to enable discovering process models which contain cross-cutting concerns. The approach is a cyclical process which contains four phases, i.e. i) cross-cutting concerns discovery, ii) cross-cutting concerns elimination, iii) business process discovery, and iv) relation discovery (see Fig. 3).

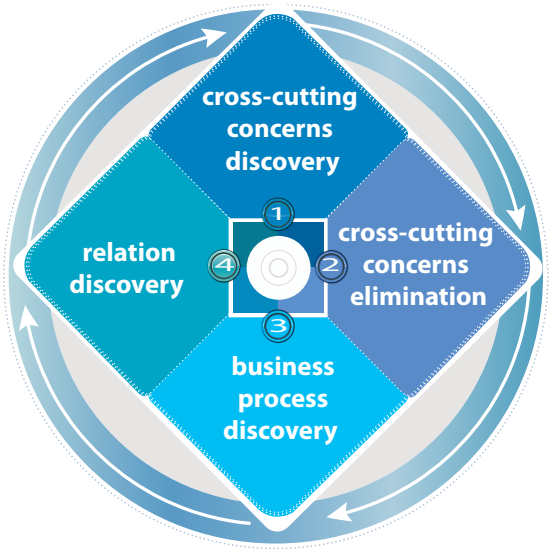


Fig. 3. Aspect Mining Approach for BPM

3.1 Cross-cutting concerns discovery

In this phase, cross-cutting concerns should be discovered. These concerns can be discovered manually or automatically. In *manual discovery* of cross-cutting concerns, the concerns can be identified through traditional methods of process identification and discovery like the interview. For example, cross-cutting concerns can be expressed as policies in organizations. In *automatic discovery* of cross-cutting concerns, the concerns should be discovered by an algorithm. These concerns can be represented as events in the log file recording the enactment result of a process or several process models.

To discover the cross-cutting concerns from a log file of a process model, an algorithm should search for processes with duplicate tasks. Finding these tasks helps to identify cross-cutting concerns, since these concerns are sometimes repeated in a process model. Li et al. [23], proposes an extension to alpha algorithm to discover repetitive activities in an event log, which is extended by Gu et al. [26] to discover short loops. Their extensions do not work in all situations, yet it can help in some others. In our case study, this approach did not help to discover repetitive activities. Section 5 discusses the problem of this algorithm in more detail. Alves de Medeiros [25] proposes a genetic algorithm to tackle duplicate tasks, called Duplicates Genetic Algorithm (DGA). This algorithm helps to discover some processes with duplicate tasks automatically. Herbst et al. [20] also proposes an algorithm which can help in identifying duplicate tasks in process models. Moreover, there are many works in programming area which can be considered to inspire extending this area, e.g. [8, 28].

To discover the cross-cutting concerns from a log file of multiple process models, no solution exist currently. This is due to current situation in process mining area in which "there are no process discovery techniques that produce overarching models able to relate and analyze different groups and process variants" [1]. However, it is expected that the introduction of process cubes, which support vertical and horizontal decomposition of event logs, enables such investigation [1].

3.2 Cross-cutting concerns elimination

Identifying cross-cutting concerns enables separation of their events from event logs. Therefore, the logs will contain only the main activities for each process model. It should be mentioned that the whole approach is cyclical, so all concerns may be identified and their events may be eliminated through several cycles.

3.3 Business process discovery

The cleaned event logs can be used by existing process mining algorithms to discover process models. There are many algorithms that can be used, for example alpha algorithm can be used to identify process models from noise free event log [4], and fuzzy miner can be used to discover process models from event logs which have noises [17].

3.4 Relation discovery

The relation between cross-cutting concerns and process models can be discovered manually or automatically. These relations are called *pointcuts* in aspect oriented paradigm.

In *manual discovery*, pointcuts can be identified through traditional methods like the interview. In *manual discovery*, pointcuts should be identified and added to process models through *process enhancement* technique. Process Enhancement is a technique "to extend or improve an existing process model using information about the actual process recorded in some event log" [2]. Here, information about the control-flow perspective of the model and the cross-cutting concerns are available as a result of previous phases. Moreover, complete information regarding the execution of these events exists in the event log. Therefore, the relation between the process model and cross-cutting concerns can be identified through investigating the event logs. This information can be added to the process model as pointcuts which defines these relations. Fig. 4 shows the pointcut discovery phases.

4 Demonstration

The proposed approach is applied on event logs of the case study. The logs are noise free and cleaned. The demonstration is performed in four phases, i.e.

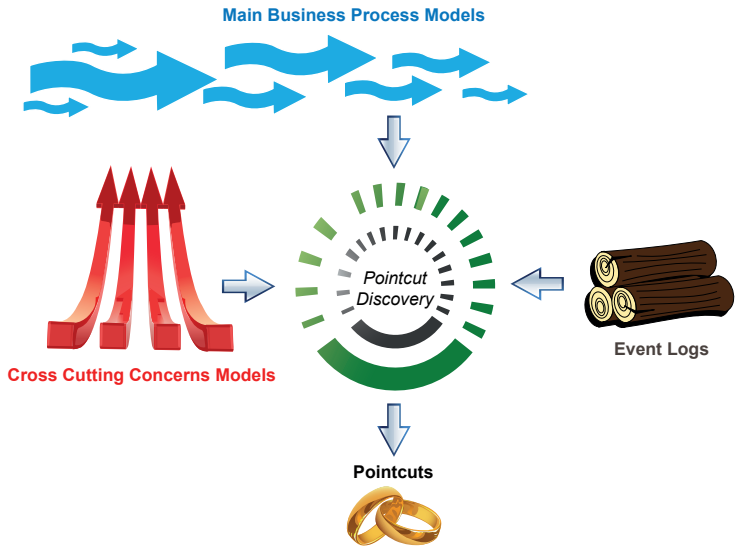


Fig. 4. Pointcut Discovery Process

i) cross-cutting concerns discovery, ii) cross-cutting concerns elimination, iii) business process discovery, and iv) relation discovery.

4.1 Cross-cutting concerns discovery

To discover cross-cutting concerns, manual cross-cutting concerns discovery is followed through interview, because there is not currently any algorithm that can be used for automatic discovery of aspects. There are four scenarios in this business process that require some sort of security checks, i.e. i) the position sheet should be confirmed before making a deal, ii) the deal slip should be signed for further processing, iii) the received swift message should be controlled, and iv) the swift draft should be signed before it can be sent.

These four scenarios can be defined through three mechanisms that expressed three cross-cutting concerns, i.e. `Sign`, `Confirm` and `Control`. As the `Sign` mechanism, if the document is a swift draft, the junior dealer, chief dealer and general manager should sign it sequentially; however, if it is a dealslip, only chief dealer and general manager require to sign it. This concern is modeled as a process model which is called Sign (see Fig. 5). As the `Confirm` mechanism, the general manager can only confirm decisions, and each confirmation should be archived by the office employee. This concern is modeled as a process model which is called Confirm (see Fig. 5). As the `Control` mechanism, the general manager should control all swifts which are received. Then, the swifts message should be controlled and archived by the back-office employee. If the received swift is related to a deal, the dealer who imitated the deal should also control and archive the swift. This concern is modeled as a process model which is called Control (see Fig. 5). All of these concerns represent security aspects of the process model, so they are grouped together in a module, called *security aspect*.

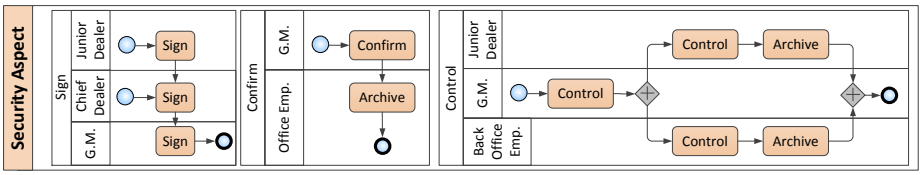


Fig. 5. The result of manual discovery of cross-cutting concerns

4.2 Cross-cutting concerns elimination

In this step, all events which are related to one of the activities in cross-cutting concerns are removed from event log. Therefore, the new event logs only contains information about the core part of the change asset deal process.

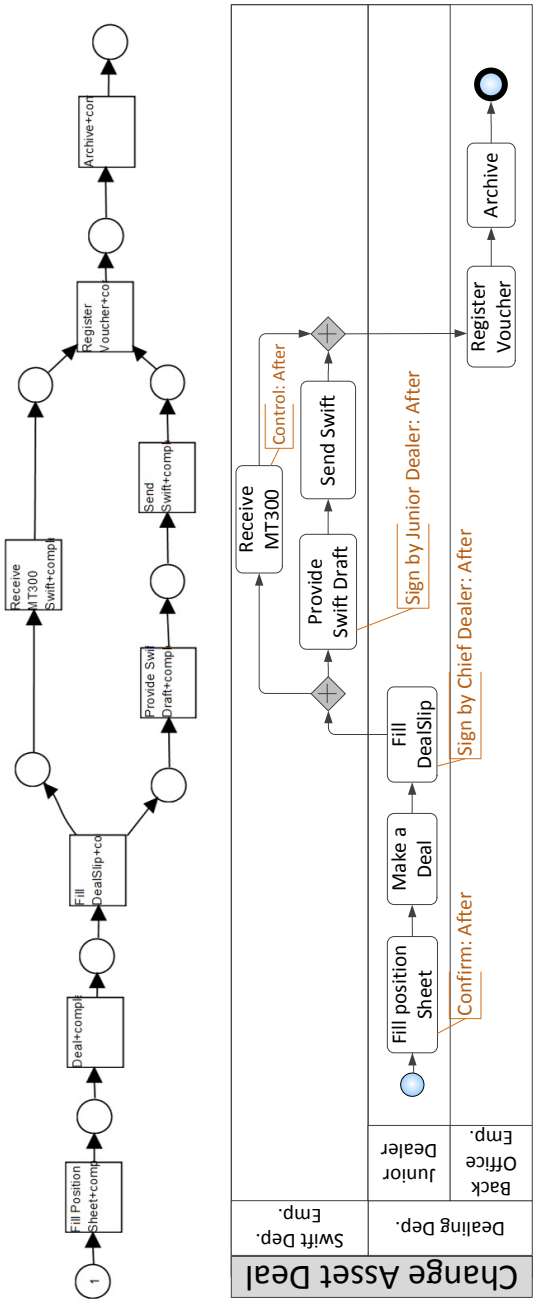
4.3 Business process discovery

The cross-cutting concerns free event log is used to discover the change asset deal process. Fig. 6(a) shows the result of process discovery by applying Alpha Miner in ProM. The same result was achieved by applying Fuzzy Miner. As it was expected, the results are the same since the log file does not contain noises. The discovered process represents the change asset deal process without cross-cutting concerns. It starts when a junior dealer fills a position sheet. Then, (s)he should make a deal, and fill the deal slip. Afterwards, the swift department employee provides the swift draft and sends the swift. In parallel, the swift department of the corresponding bank sends the related swift, so the swift department employee also receives a swift with MT300 message. Finally, the back-office employee should register the voucher and archive it.

4.4 Relation discovery

We already discovered both the deal asset process model and models of cross-cutting concerns. However, it is not clear how these models should be related to each other. Therefore, manual pointcut discovery is used to discover the relations. It should be noted that automatic pointcut discovery is not possible now, since it requires further research to define an algorithm for such a discovery. Some of these relations are specified when we discovered cross-cutting concerns, e.g. the general manager should control all swifts which are received. Thus, in this step all relations should be investigated and completed.

Fig. 6(b) shows the discovered change asset deal process, in which the relations to cross-cutting concerns are specified through annotations. As it can be seen, the position sheet should be confirmed after junior dealer fills it. Moreover, the dealslip should be signed after junior dealer fills it. The swift draft should be signed by junior dealer, chief dealer and general manager before the swift is sent, and the received swift should be controlled. This process and cross-cutting concerns which are shown in Fig. 5 complement each other.



(a) The process model discovered through applying Alpha Miner

(b) Change Asset Deal process with pointcuts

Fig. 6. Demonstrating the proposed approach in the case study

5 Related Works and Discussions

This section explains related works in aspect mining through three categories, i.e. discovering duplicate tasks from event logs in BPM, discovering cross-cutting concerns from business process models, and aspect mining in programming. The first category explains current attempts to discover duplicate tasks from event logs in BPM area, and it explains a counter example for these works. The second category introduces some works which enable discovering cross-cutting concerns from business process models. The third category shows the similar paradigm in programming area which tries to discover cross-cutting concerns from programs' codes and traces.

5.1 Discovering duplicate tasks from event logs in BPM

Discovering duplicate tasks is one of the challenges and open issues in process mining area [3, 24]. Li et al. [23], propose an extension to alpha algorithm to discover duplicate tasks in event logs, which is also extended by Gu et al. [26] to discover short loops. This algorithm is called α*. However, their extensions do not work in all situations. It cannot discover duplicate activities in our case study correctly, which can be consider as a counter example.

Their approach follows two steps, i.e. i) finding duplicate tasks, and ii) discovering a process model. To find duplicate tasks, the approach makes a predecessor and successor table, called P/S-table [23]. Each trace is denoted by δ_i, where i represents the number of trace that contains events. $\delta_i(t, n)$ represents a function which specify the nth occurrence of event t in δ_i. For each of these occurrence, the predecessor and successor of the event can be retrieved by T_P and T_S correspondingly. The approach proposes three rules with which the duplicate tasks claimed to be discovered from event logs, i.e. Rule (1), (2) and (3). The U in these formulae represents the set of duplicate tasks.

(1) IF $((T_P \neq T_{P'})$ AND $(T_S \neq T_{S'})$ AND $(T_P \neq T_{S'})$ AND $(T_S \neq T_{P'})$ AND (not $T_P \Sigma w T_{P'}))$ THEN $<\delta_i(t, N_1), \delta_j(t, N_2)> \in$ U

(2) IF $((T_P = T_{S'})$ AND $((T_{P'} \neq pred(\delta_i, T_P))$ OR $(T_S \neq succ(\delta_j, T_{S'})))$ THEN $<\delta_i(t, N_1), \delta_j(t, N_2)> \in$ U

(3) IF $((T_S = T_{P'})$ AND $((T_{P'} \neq pred(\delta_j, T_{P'}))$ OR $(T_{S'} \neq succ(\delta_i, T_S)))$ THEN $<\delta_i(t, N_1), \delta_j(t, N_2)> \in$ U

The case study in this paper shows that this formula cannot work. To explain it, we can consider a very simple process shown in Fig. 7 which contains tasks a,b,c,a and b which should be followed in sequence. It is clear that all traces of executing this process has the same occurrence of events, so for each i and j representing a trace number: $\delta_i = \delta_j$.

The duplicate task a cannot be discovered through the first rule since:

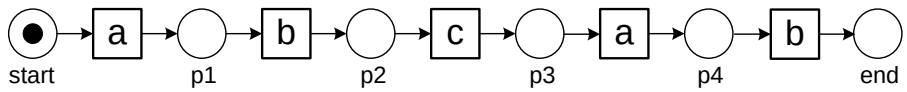


Fig. 7. Counter Example for duplicate task identification rules in α*-algorithm

$$\text{For rule 1:}$$
$$\frac{\begin{array}{c} T_S = \delta_i(a,1) = b \\ T_{S'} = \delta_j(a,2) = b \end{array}}{\therefore T_S = T_{S'} \not\Rightarrow a \in U}$$

The duplicate task a cannot be discovered through the second rule since:

$$\text{For rule 2:}$$
$$\frac{\begin{array}{c} T_P = \delta_i(a,1) = \emptyset \\ T_{S'} = \delta_j(a,2) = b \end{array}}{\therefore T_P \neq T_{S'} \not\Rightarrow a \in U}$$

The duplicate task a cannot be discovered through the third rule since:

$$\text{For rule 3:}$$
$$\frac{\begin{array}{c} T_S = \delta_i(a,1) = b \\ T_{P'} = \delta_j(a,2) = c \end{array}}{\therefore T_S \neq T_{P'} \not\Rightarrow a \in U}$$

Thus, this algorithm cannot discover duplicate tasks in all circumstances.

5.2 Discovering cross-cutting concerns from process models

Although cross-cutting concerns can be discovered from event logs, they can also discovered from process models that are designed through traditional modelling techniques. There are few works in BPM area that support separation of cross-cutting concerns from process models,e.g. [15, 5, 6]. These works apply queries on process models definitions, and they try to discover cross-cutting concerns which exist within these models.

5.3 Aspect mining in programming

There are a lot of researches about Aspect Mining in Aspect Oriented Programming area, e.g. [7, 9, 12, 13, 19]. These researches can be divided into two categories, i.e. static and dynamic program analysis techniques.

Static program analysis techniques aim to discover aspects out of source codes. Two approaches exist in these techniques, i.e. discovering advices without any previous knowledge or with previous knowledge. For example, 'Formal Concept Analysis of Identifiers' is utilized by some researchers to discover cross-cutting concerns from Smalltalk and Java codes using the FCA algorithm [11, 29]. These techniques require that the user have some assumption regarding the structure of advices. Other researchers used 'Natural Language Processing on Source Code' to solve this problem [27]. Furthermore, there are

other works that discover cross-cutting concerns from codes without any previous knowledge about concerns [18]. There are also other techniques based on detecting clones in codes, which are categorized in these techniques [18]. The advantages of static approaches are that they consider all possible execution scenarios in source codes.

Dynamic program analysis techniques aim to discover aspects out of program traces of method invocations. There are also different works that have been done in this area. For example, discovering the cross-cutting concerns by detecting recurring execution patterns [8], or applying formal concept analysis to achieve this goal [28]. These works can be used as inspiration for enabling automatic cross-cutting concerns discovery in BPM area.

6 Conclusion

This paper proposes an approach that enables discovering business process models which have cross-cutting concerns. The approach is a cyclical process which has four phases, i.e. i) cross-cutting concerns discovery, ii) cross-cutting concerns elimination, iii) business process discovery, and iv) relation discovery. In *cross-cutting concerns discovery*, the cross-cutting concerns should be discovered either manually or automatically. The events of these concerns should be eliminated from event log in the *cross-cutting concerns elimination* phase. The eliminated log can be used for discovering the main process model in *business process discovery* phase. Finally, the relations between cross-cutting concerns and the main process model should be discovered in *relation discovery* phase, which can be done manually or automatically. This approach is a cyclical process, which means that it might be repeated several times that all cross-cutting concerns are separated.

The approach is applied and demonstrated through a case study, which shows how these sort of models can be discovered. The finding is important since it is common to have repetitive and duplicate tasks in business processes, and this situation is a limitation of process mining currently. It means that process mining cannot deal with these sorts of processes, yet this approach can enable discovering these processes as well.

As future works, more experiments are needed to be conducted to evaluate the proposed method. Moreover, it is important to investigate i) how cross-cutting concerns can be discovered automatically from event logs, ii) how duplicate tasks can be discovered automatically from event logs, iii) how the relations between cross-cutting concerns and the main process model can be discovered automatically from the event log.

Acknowledgement

I would like to appreciate professor Paul Johannesson for his valuable comments on this work.

References

1. W.M.P. van der Aalst. Process cubes: Slicing, dicing, rolling up and drilling down event data for process mining. In Minseok Song, MoeThandar Wynn, and Jianxun Liu, editors, *Asia Pacific Business Process Management*, volume 159 of *LNBIP*, pages 1–22. Springer International Publishing, 2013.
2. W.M.P. van der Aalst et al. Process mining manifesto. In F. Daniel, K. Barkaoui, and S. Dustdar, editors, *BPM Workshops*, volume 99 of *LNBIP*, pages 169–194. Springer Berlin Heidelberg, 2012.
3. W.M.P. van der Aalst and A.J.M.M. Weijters. Process mining: a research agenda. *Computers in Industry*, 53(3):231 – 244, 2004.
4. W.M.P. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: discovering process models from event logs. *Knowledge and Data Engineering, IEEE Transactions on*, 16(9):1128–1142, Sept 2004.
5. Seyed Mehdi Reza Beheshti. *Organizing, Querying, and Analyzing Ad-hoc Processes Data*. PhD thesis, University Of New South Wales Sydney· Australia, 2012.
6. S.M.R Beheshti, B. Benatallah, and H.R. Motahari-Nezhad. Enabling the analysis of cross-cutting aspects in ad-hoc processes. In Camille Salinesi, MoiraC. Norrie, and scar Pastor, editors, *Advanced Information Systems Engineering*, volume 7908 of *LNCS*, pages 51–67. Springer Berlin Heidelberg, 2013.
7. S. Breu. Extending dynamic aspect mining with static information. In *Source Code Analysis and Manipulation, 2005. Fifth IEEE International Workshop on*, pages 57–65, 2005.
8. S. Breu and J. Krinke. Aspect mining using event traces. In *Automated Software Engineering, 2004. Proceedings. 19th International Conference on*, pages 310–315, Sept 2004.
9. S. Breu and Th. Zimmermann. Mining aspects from version history. In *Automated Software Engineering, 2006. ASE'06. 21st IEEE/ACM International Conference*, pages 221–230. IEEE, 2006.
10. C. Cappelli, F.M. Santoro, J.C.S. do Prado Leite, T. Batista, A.L. Medeiros, and C.S.C. Romeiro. Reflections on the modularity of business process models: The case for introducing the aspect-oriented paradigm. *BPM Journal*, 16:662–687, 2010.
11. M. Ceccato, M. Marin, K. Mens, L. Moonen, P. Tonella, and T. Tourwe. A qualitative comparison of three aspect mining techniques. In *Program Comprehension, 2005. IWPC 2005. Proceedings. 13th International Workshop on*, pages 13–22. IEEE, 2005.
12. M. Ceccato, M. Marin, K. Mens, L. Moonen, P. Tonella, and T. Tourwé. Applying and combining three different aspect mining techniques. *Software Quality Journal*, 14(3):209–231, 2006.
13. M. Ceccato, M. Marin, K. Mens, L. Moonen, P. Tonella, and T. Tourwé. Applying and combining three different aspect mining techniques. *Software Quality Journal*, 14(3):209–231, 2006.
14. A. Charfi, H. Müller, and M. Mezini. Aspect-Oriented Business Process Modeling with AO4BPMN. In T. Khne et al., editor, *Modelling Foundations and Applications*, volume 6138 of *LNCS*, pages 48–61. Springer, 2010.
15. C. Di Francescomarino and P. Tonella. Crosscutting concern mining in business processes. *Software, IET*, 5(6):552–562, 2011.
16. B.F. Dongen, A.K.A. Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. Aalst. The prom framework: A new era in process mining tool support. In Gianfranco Ciardo and Philippe Darondeau, editors, *Applications and Theory of Petri Nets 2005*, volume 3536 of *LNCS*, pages 444–454. Springer Berlin Heidelberg, 2005.

17. ChristianW. Günther and WilM.P. Aalst. Fuzzy mining adaptive process simplification based on multi-perspective metrics. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management*, volume 4714 of *LNCS*, pages 328–343. Springer Berlin Heidelberg, 2007.
18. Kris Gybels and Andy Kellens. An experiment in using inductive logic programming to uncover pointcuts. In *First European Interactive Workshop on Aspects in Software*, 2004.
19. L. He and H. Bai. Aspect mining using clustering and association rule method. *International Journal of Computer Science and Network Security*, 6(2A):247–251, 2006.
20. Joachim Herbst and Dimitris Karagiannis. Workflow mining with inwolve. *Computers in Industry*, 53(3):245 – 264, 2004. Process / Workflow Mining.
21. A. Jalali, P. Wohed, and C. Ouyang. Aspect oriented business process modelling with precedence. In J. Mendling et al., editor, *BPMN 2012*, volume 125 of *LNCS*, pages 23–37. Springer, 2012.
22. A. Jalali, P. Wohed, Ch. Ouyang, and P. Johannesson. Dynamic weaving in aspect oriented business process management. In Robert Meersman, Herv Panetto, Tharam Dillon, Johann Eder, Zohra Bellahsene, Norbert Ritter, Pieter Leenheer, and Deijing Dou, editors, *CoopIS 2013*, volume 8185 of *LNCS*, pages 2–20. Springer Berlin Heidelberg, 2013.
23. Jiafei Li, Dayou Liu, and Bo Yang. Process mining: Extending α-algorithm to mine duplicate tasks in process logs. In K.C.C. Chang, W. Wang, L. Chen, C.A. Ellis, C.H. Hsu, A.C. Tsoi, and H. Wang, editors, *Advances in Web and Network Technologies, and Information Management*, volume 4537 of *LNCS*, pages 396–407. Springer Berlin Heidelberg, 2007.
24. A.K.A. Medeiros, W.M.P. Aalst, and A.J.M.M. Weijters. Workflow mining: Current status and future directions. In R. Meersman, Z. Tari, and D. Schmidt, editors, *CoopIS 2003*, volume 2888 of *LNCS*, pages 389–406. Springer Berlin Heidelberg, 2003.
25. de AKA Medeiros. *Genetic process mining*. PhD thesis, Technische Universiteit Eindhoven, 2006.
26. Chun qin Gu, Hui you Chang, and Yang Yi. Workflow mining: Extending the α-algorithm to mine duplicate tasks. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 1, pages 361–368, July 2008.
27. D. Shepherd, L. Pollock, and T. Tourwé. Using language clues to discover crosscutting concerns. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–6. ACM, 2005.
28. P. Tonella and M. Ceccato. Aspect mining through the formal concept analysis of execution traces. In *Reverse Engineering, 2004. Proceedings. 11th Working Conference on*, pages 112–121, Nov 2004.
29. T. Tourwe and K. Mens. Mining aspectual views using formal concept analysis. In *Source Code Analysis and Manipulation, 2004. Fourth IEEE International Workshop on*, pages 97–106. IEEE, 2004.