Towards Aspects Identification in Business Process Through Process Mining

Bruna Christina P. Brandão
Federal University of the State of Rio de Janeiro (Unirio)
Avenida Pasteur, 458
Urca, Rio de Janeiro - RJ
+55 (21) 2530-8051.
bruna.christina@uniriotec.br

Flávia Maria Santoro
Federal University of the State of Rio de Janeiro (Unirio)
Avenida Pasteur, 458
Urca, Rio de Janeiro - RJ
+55 (21) 2530-8051.
flavia.santoro@uniriotec.br

Leonardo Guerreiro Azevedo
IBM Research – Brasil

Avenida Pasteur, 138/164
Botafogo, Rio de Janeiro – RJ
+55 (21) 2132-5252.
lga@br.ibm.com

ABSTRACT

In business process models, elements can be scattered (repeated) within different processes, making it difficult to handle changes, analyze process for improvements, or check crosscutting impacts. These scattered elements are named as Aspects. Similar to the aspect-oriented paradigm in programming languages, in BPM, aspect handling has the goal to modularize the crosscutting concerns spread across the models. This process modularization facilitates the management of the process (reuse, maintenance and understanding). The current approaches for aspect identification are made manually; thus, resulting in the problem of subjectivity and lack of systematization. This paper proposes a method to automatically identify aspects in business process from its event logs. The method is based on mining techniques and it aims to solve the problem of the subjectivity identification made by specialists. The initial results from a preliminary evaluation showed evidences that the method identified correctly the aspects present in the process model.

Categories and Subject Descriptors

K.6.0 [**Management of Computing and Information Systems**]: General. D.4.1 [**Operating Systems**]: Process Management.

General Terms

Management.

Keywords

Aspects, Process Mining, Business Process Management.

1. INTRODUCTION

Software code corresponding to the secondary functions, such as, exception handling, time constraints, transaction management, security control, access control (logging), is many times spread all over an application. For example, a data logging functionality in object-oriented programming is implemented in a single class, but is invoked in other methods as a secondary function, becoming a spread code. These spread codes are defined as crosscutting concerns [21].

The aspect-oriented programming (AOP) approach aims to encapsulate spread codes through a construct called Aspect [16]. An Aspect changes the behavior of the code with an additional behavior at one point in the code execution. This additional behavior is called advice, and the execution point is defined as join point [8].

The aspect-oriented paradigm inspired its use in BPM (Business Process Management) with the same goal to modularize the crosscutting concerns spread across the software specification [6]. Crosscutting concerns in BPM are also defined as the interests that cause the scattering and tangling problem. The scattering problem is the repetition of concerns within the process models. The tangling problem means that any change in a concern should be reflected in all processes which use the concern [7] [10] [11] [17] [33]. Thus, in BPM, crosscutting concerns are elements scattered and tangled within process models. They can be part of the core of the process (*e.g.*, business rules) and not only secondary features as in AOP (exception handling, time constraints, check authorization and authentication, transaction control, security control, access control, logging) [18] [25].

Aspects identification is beneficiary to improve process modularization. It facilitates management of the process (reuse, maintenance and understanding). A method to identify aspects automatically can assist in decision making and saving time. Currently, the proposals for identifying aspects laid on manually analysis of business process models by experts [6] [18]. The identification made by humans raises the problem of subjectivity, because it depends on each expert judgment. This problem – the absence of a technical identification method – results in the following research question:

`How to identify automatically aspects in business processes?`

The research hypothesis undertaken is:

`If we apply process mining techniques, then aspects can be identified automatically reducing the subjectivity in the identification.`

In order to address this question, this work aims at developing an automated method to identify aspects. The method applies process mining techniques on information systems event logs. The identification of aspects in logs makes the technique independent from the process model, although it is dependent on the event log structure. A preliminary case study was performed to do a primary evaluation of the method.

SBSI 2015, May 26–29, 2015, Goiânia, Goiás, Brazil.

This paper is divided on 8 sections. The first section is the introduction. Section 2 presents the concepts of aspects in business process management and Section 3 the related work. Section 4 presents a brief introduction of process mining. Section 5 describes the method proposed, Section 6 the exploratory study and Section 7 presents the conclusion.

2. ASPECTS IN BPM

Originally, aspects are the modularization of crosscutting concerns in object-oriented software. The aspect-oriented programming allows the code to be encapsulated and modularized. This programming approach has advantages such as reduction in the code spreading, the responsibilities of each module are more transparent, and the modules are more independent. Evolving the application is easier due to the loose coupling in the aspects implementations, which enables modifications in the application without changing the main functionalities. The systems are easier to develop and maintain because aspect facilitates the integration of new functionalities without causing problems to other parts of the system. Besides, it has low cost to introduce new features [13].

The same logic applies to aspect in process models. Crosscutting concerns in BPM are also defined as the interests that cause the scattering and tangling problem [7] [10] [11] [17] [33]. They are elements scattered and tangled within process models, and they can be the part of the core of the process (*e.g.*, business rules) and not only secondary features as in aspects in oriented-programming.

Aspects have similar meaning as services in a SOA (Service-Oriented Architecture) context. Services are defined as parts or entire functions of a system that may be available to another system and can be invoked through messages [19]. The services must operate in an independent way of other services, and should have a well-defined interface. Services are logical representations of elements (*e.g.*, activities, business rules, business requirements) in the business process that can be mapped on input, processing and output. When those elements can be performed automatically or by human using systems (in a semi-automated approach), they can be made available as services in SOA. Then the difference from service to aspect is that a service represents elements of business process that can be performed automatically or supported by systems. On the other hand, aspects can also manual elements performed manually.

Cappelli *et al.* [6] propose a meta-language called Aspect Oriented Process Modeling Language (AOPML), which is independent of any business process language. The use of this meta-language improves business process model modularity. In another work, Cappelli *et al.* [7] proposed developed a notation of AOPML specifically to handle BPMN (Business Process Modeling Notation) [20] models. They called this particular notation as Aspect-Oriented Business Process Modeling Notation (AO-BPMN), which uses aspect-oriented paradigm to extend the BPMN. In summary, they proposed the introduction of graphical elements in the notation through new roles for the lanes and relationships. In that case, the crosscutting concerns are represented separated in a vertical lane, orthogonal to the processes. It uses a new connector called relationship cross, which represent the composition between the transverse elements of interest and process model. However, this approach has proved to be confusing for analysts, due to the many lines connecting elements, so there are some works which are trying to improve this representation.

Figure 1 illustrates the business process model “send articles to reviewers” modularized by aspects using AO-BPMN. In this process, the aspect “Log information” is reused when there is the need to store information, e.g., in the execution of the activities “Send invitation”, “Receive invitation”, “Reply invitation” and “Receive response”. In each activity execution, the aspect “Log information” activates its own execution intervening in the activity within the process, and storing information about the execution of the activity.

The modularization of processes through aspects brings the same benefits of aspect-oriented programming. Processes modularization provide a better understanding of the process, enables reusability of parts between processes that share the same "interests", including key features such as business rules.

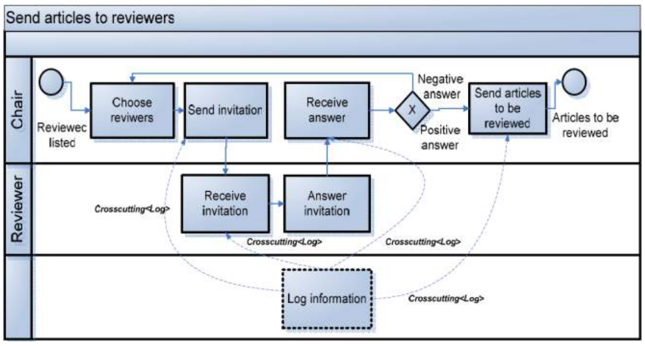


Figure 1 - Process Model with aspects modularization [6].

The implementation of aspect-oriented programming normally consists of [27]:

- A component language for components programming;
- One or more aspect languages for aspects programming;
- An aspects combiner (aspect weaver) – to combine the artifacts of the programming language and the aspects;
- A program written in the components language;
- One or more programs written in the aspects language

Figure 2 illustrates the relationship of the artifacts present in the aspect implementation.

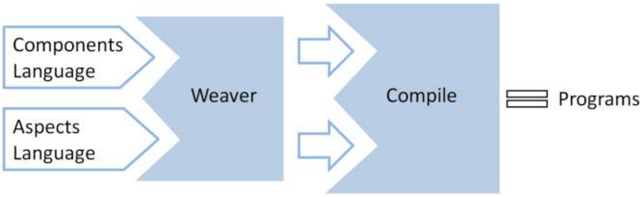
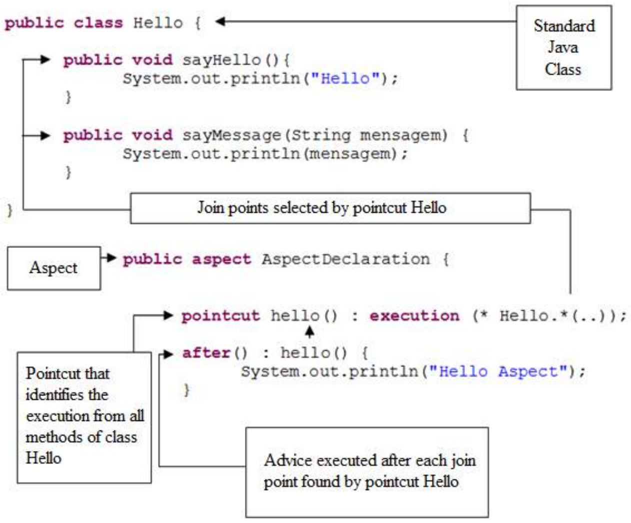


Figure 2 – Weaver relationship.

In the context of aspect-oriented programming, components are abstractions provided by the language that allow the implementation of a system functionality (procedures, classes, functions, objects) [27].

The aspect language supports the implementation of desired features clearly and succinctly, providing the programmer structures needed to describe the behavior of aspects and situations in which they must act [27]. AspectJ is an example of an aspect language. It extends the Java language with new structures to support the modular implementation of crosscutting concerns. New elements are: (i) *join points* represent execution points; (ii) pointcuts represent sets of join points; (iii) advice represents the methods attached to pointcuts. Aspects are modular units of crosscutting implementations composed of pointcuts, advice and own statements of the Java language [15].

Figure 3 illustrates a Java class using aspect elements. The Java class Hello has two join points *sayHello* and *sayMessage*. They will be intercepted by the aspect *AspectDeclaration*, which has the pointcut *hello* referencing all methods of Hello class. This pointcut has an advice, which defines that the command *hello* should be to be executed after the accomplishment of the join point found by the pointcut [13]. In other words, after the execution of each method of *Hello* class the *hello* method should be executed. The combination of the Java code with the Aspect code is done in a process named as weaving.



```
public class Hello {

    public void sayHello(){
          System.out.println("Hello");
    }

    public void sayMessage(String mensagem) {
          System.out.println(mensagem);
    }
}

public aspect AspectDeclaration {

    pointcut hello() : execution (* Hello.*(..));

    after() : hello() {
             System.out.println("Hello Aspect");
    }
```

Standard Java Class
Join points selected by pointcut Hello
Aspect
Pointcut that identifies the execution from all methods of class Hello
Advice executed after each join point found by pointcut Hello

Figure 3 - Relationship between a class and one aspect [adapted from [13].

The identification of aspects is the first step to be done before the implementation or the modeling of aspects. There are three approaches in to identify aspects: AOP clone, cluster and analysis fan-in. However, there is not a technical approach to identify automatically the crosscutting concerns in business process. Most approaches are manual – executed by experts. The next section presents the existing approaches of aspects in business processes.

3. RELATED WORK

The proposals for identifying aspects in business process are in general based on manual identification by experts [6] [18]. Other approaches for aspect identification are on requirements context, which identify aspects automatically in text, but they not consider non-functional aspect of business process instead of functional aspects as our approach does.

Jalali [18] proposes that process models with cross-cutting concerns can be discovered from the event logs. After demonstrating the current techniques of process mining cannot discover processes models with cross-cutting concerns, he proposes a method to discover this kind of processes models. He mentions automatically discovery by mining techniques to discover duplicated activities. However, he makes the aspect discovery manually by interviewing business process specialists, and not automatically as we propose in our work.

Rago *et al.* [22] propose de identification of aspects using natural language processing techniques from use cases. Their analysis is focused on the relationships among terms in use cases (*e.g.*, verbs, direct objects) that often hint crosscutting behaviors. This approach suggests that these techniques can solve issues related to synonyms, vagueness and ambiguity in text, as reported by other aspect mining approaches.

Sampaio *et al.* [24] propose the identification of aspects from natural language processing based on the identification of business requirements. Their work presented an approach for mining aspects from requirements-related documents based on NLP (Natural Language Processing) techniques that enable an efficient context sensitive analysis of textual documents. The documents can vary from very informal textual documents, such as interviews and high level descriptions of the system, to more structured documents such as use case textual descriptions or viewpoint descriptions.

Campos *et al.* [5] discussed the identification of aspects via inspection of use cases descriptions. They suggested a use case template and a checklist to process the description. The template was designed to include the sections in which aspect candidates were identified, use the step numbering that indicates where the aspect candidates should be inserted (join points). Their work emphasizes that the aspects candidates are normally found in the extensions of the description.

Souza *et al.* [28] recommended the identification of services in a SOA (Service-Oriented Architecture) from a business process model that was designed in an aspect-oriented fashion (AO-BPM). There are other approaches for service identification from business process models and for service composition from an aspect-oriented approach [9]. However, all of them are concerned with services identification and not with aspect identification.

Although there are approaches for aspects identification in business requirements and use cases ([5], [22], [24]), and to use aspects to identify services ([1], [2] and [28]), our approach differentiates by identifying aspects from events logs. The main benefit of this approach is to identify aspects automatically in the business level. That is, in the business process models, the obvious crosscutting concerns (non-functional requirements) normally are not present. Our method tries to identify the scattering and tangling problem within a process execution focusing on the functional requirements instead of non-functional requirements.

4. PROCESS MINING

Process mining is a discipline that relates data mining and process intelligence. Process intelligence is the discovery, analysis and verification if the process is effective for the improvement of business. Process mining is performed from data recorded in event logs, containing information that was created during the execution of the process [30]. Therefore, a process discovered through mining event logs is called AS-IS process, how the process really runs. On the other hand, a modeled processes is called TO-BE process, it presents how the process should run [12].

There are three approaches of process mining: process discovery, process analysis and process verification. The most common algorithm for process discovery is the alpha algorithm [12]. Alpha algorithm discovers the process by mapping the relationships between the activities present in the process according to their orders in the event logs. Initially, it maps the basic relations of precedence, then the causal relationships, the potential parallelism relations and relations of no direct succession.

The second approach which uses process mining is process performance analysis. The analysis is based on four dimensions: Time, Cost, Quality and Flexibility. The time dimension checks the lifecycle of activities and the waiting time to execute them. The cost dimension can be analyzed if there are price details present in the logs, e.g., the cost to execute each activity from the process on the current instance. The quality dimension analyzes the process quality. The flexibility dimension is the range of variation the process enables, that is, it analyzes whether the process discovered (as-is) allows paths which are or aren't desirable [12].

The third approach is the process verification of conformity, in which the process is checked in relation to its constraints of exclusivity, order and mandatory issues. The constraint of exclusivity checks activities that exclude others activities – *e.g.*, activity of order accepted and order rejected. The constraint of order checks the order of the activities (*e.g.*, rent equipment before checking availability). The mandatory constraint checks activities which must be executed for any purpose of the organization - example: review request for rent in order to control the cost.

Process mining is applied in events log generated by information systems. An IEEE process mining task force created a standard log format to be able to be use in any process mining tool. Van der Aalst *et al.* [31] standardized XES (www.xesstandard.org), a standard logging format that is extensible and supported by the OpenXES library and by tools such as ProM, XESame, and Nitro. The metamodel of XES is illustrated in Figure 4.

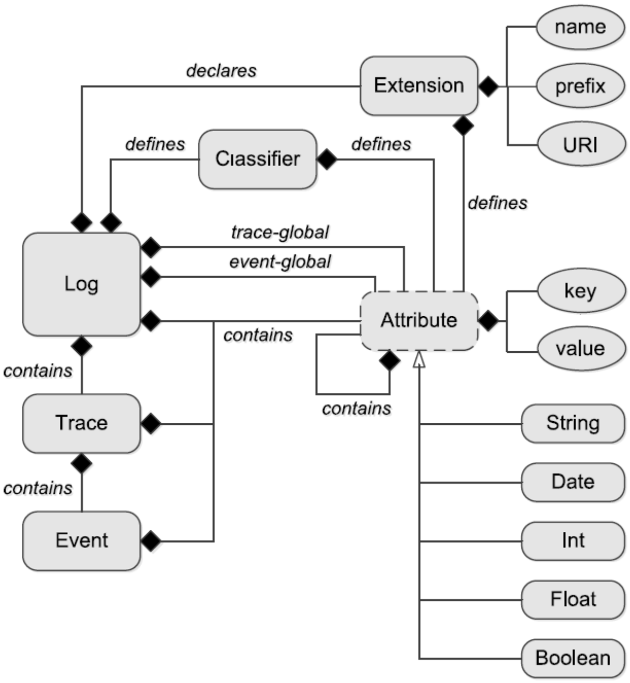
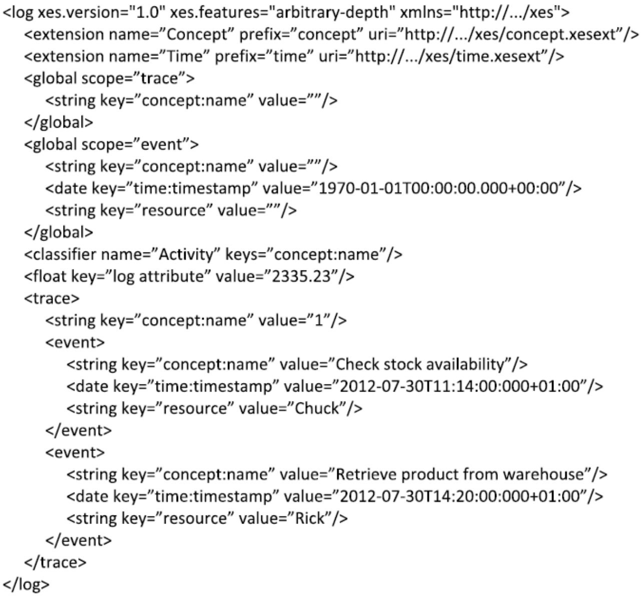


Figure 4 - XES Metamodel [12].

Each XES file represents one log. Each log contains multiples traces, which contains multiples events. Log, traces and events contain different attributes. An attribute type can be string, date, int, float or boolean key-value element. Attributes refers to a global definition. For example, a global definition could state that each trace must have the attribute name. Another global definition could state that each event must have the attributes name, timestamp and resource. The classifier maps one or more attributes of an event to a label that is used in the output of the analysis tool [12]. For example, the log has a classifier defining that the attribute name is an activity classifier, which means that each event in the log with a name will be classified as an activity in the analysis tool. Figure 5 shows a XES file example.



```
<log xes.version="1.0" xes.features="arbitrary-depth" xmlns="http://.../xes">
    <extension name=”Concept” prefix=”concept” uri=”http://.../xes/concept.xesext”/>
    <extension name=”Time” prefix=”time” uri=”http://.../xes/time.xesext”/>
    <global scope=”trace”>
        <string key=”concept:name” value=””/>
    </global>
    <global scope=”event”>
        <string key=”concept:name” value=””/>
        <date key=”time:timestamp” value=”1970-01-01T00:00:00.000+00:00”/>
        <string key=”resource” value=””/>
    </global>
    <classifier name=”Activity” keys=”concept:name”/>
    <float key=”log attribute” value=”2335.23”/>
    <trace>
        <string key=”concept:name” value=”1”/>
        <event>
            <string key=”concept:name” value=”Check stock availability”/>
            <date key=”time:timestamp” value=”2012-07-30T11:14:00:000+01:00”/>
            <string key=”resource” value=”Chuck”/>
        </event>
        <event>
            <string key=”concept:name” value=”Retrieve product from warehouse”/>
            <date key=”time:timestamp” value=”2012-07-30T14:20:00:000+01:00”/>
            <string key=”resource” value=”Rick”/>
        </event>
    </trace>
</log>
```

Figure 5 - XES file example [12].

The minimum information required in the log to be able to do process mining is a trace id, timestamp and activity. This work assumes that the log has at least this information and it is the XES structure.

5. ASPECT IDENTIFICATION METHOD

In the literature, the identification of aspects in an aspect-oriented programming is most achieved by three approaches (Clone, Cluster and Fan-in analysis) according to Barbosa [3]. The Clone approach aims to discover the methods in the code that can be clones of others, *i.e.*, duplicated code. The Cluster approach aims to group patterns of methods by their execution. For example, if the methods A, B and C are always performed together, they are grouped as an aspect candidate. Finally, the Fan-in approach calculates the fan-in measure of methods to identify aspects, *i.e.*, the approach counts the number of times the method is invoked by other methods. And when a method is invoked more times than a defined threshold, it is identified as an aspect candidate.

Our approach is to identify aspects in business process model inspired in the first two software engineering methods (clones and clusters). So, we propose a method to identify aspects automatically from event logs by the clone and cluster approach.

The proposal is based on the following assumptions. The first assumption is the event log structured in XES format. The second assumption is the number of different business process logs, which should be equal or higher than two, *i.e.*, our method only discovers aspects stuck between processes, not within the same process. So, we assume a minimum of two logs. The last assumption is the abstraction level, that is, the logs must present the same level of abstraction of the elements. For example, if a process about acquiring products has the activities "go to supermarket", "buy items in the list", and "return home", and the another related process has the activities "park car in supermarket lot", "get cart", "put milk in cart", "put bread in cart", "go to cashier", and so on, then the proposed method will not be able to identify aspects due to the level of the process abstraction (process details) is different.

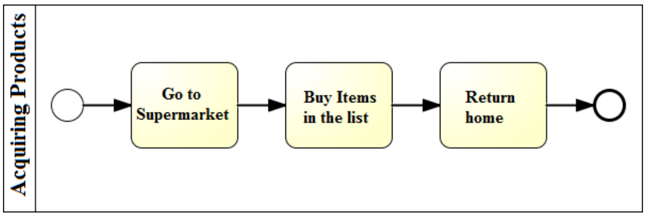


Figure 6 - Acquiring products process more abstract.

Figure 6 e Figure 7 illustrates the same process; however they have different abstraction levels. For example, the activity "Buy items in the list" from Figure 6 contain the activities "Get cart", "Put milk in cart", "Put bread in cart" and "Go to cashier". Neither one of the processes is wrong; they only have their descriptions with different details.

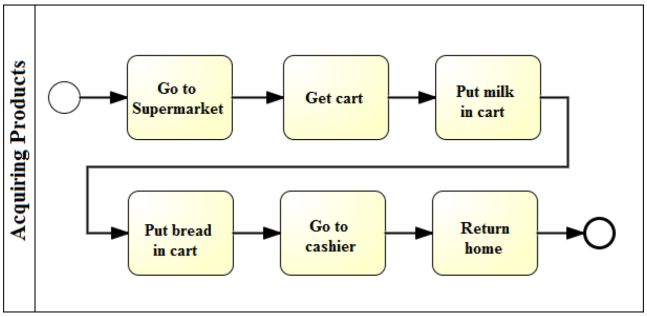


Figure 7 - Acquiring products process less abstract.

5.1 Method Description

Figure 8 depicts the method, which first reads the logs and extracts process elements. Then, it applies the clone approach to identify aspect candidates considering elements with same names or synonym names. Afterwards, it applies the cluster approach, which identifies aspect groups considering the order of execution of the aspects candidates identified by the "Apply clone approach" step, *e.g.*, two or more aspect candidates that are executed in sequence are identified as an aspect group if the group appears more than a threshold percentage defined by the user.

5.1.1 Extract Process Elements

The method begins reading the N number of logs using Open XES [14], an open source library in Java to store and manage event log data. The structure of Java classes was based in the implementation of the algorithm presented by Buijs *et al.* [4], which identifies a single process model from different event logs. The output of this step is a list of process elements by log.

5.1.2 Apply Clone Approach

After loading the logs, the method finds the events that exist in at least two logs by its names. Since it is probable that equals events has synonyms names, we implemented the clone approach (discovering semantically similar methods) with natural language techniques. This method is based on the technique proposed by Richetti *et al.* [23], which uses natural language processing to discover log activities that have similar meaning in order to simplify the declarative processes models. Our proposal does not take in consideration elements that are semantically similar, but only synonyms because of the abstraction problem. For example, if there is "Send invitation" and "Receive invitation" activities in the business processes, our method verifies if they "send" and "receive" are synonyms. It does not classify these activities as aspects because the words are not synonyms, although the activities names are semantically similar. The output of this step is a list of aspects candidates.

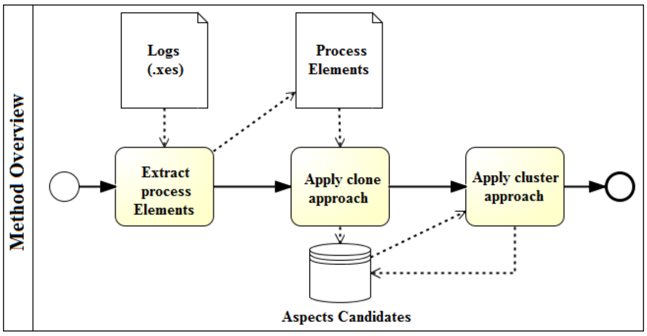


Figure 8 - Method Overview.

5.1.3 Apply Cluster Approach

The cluster approach was based on the plugin Pattern Abstractions in ProM. ProM is an open-source framework for implementing process mining tools in a standard environment. It also provides process mining algorithms, which includes the Pattern Abstraction algorithm [32]. It provides a functionality to automatically discover significant abstractions of the activities and pre-process the log using those abstractions. Since this plugin groups activities to make the process more abstract, we only used this functionality. So, after the aspects identification by the clone approach, our method seeks to group the same aspects if they were executed in a sequence. For example, if the clone method finds three aspects (A, B and C), the cluster approach will verify if they were executed in a sequence in a certain number of cases. Our method receives as input the frequency it is supposed to group. That is, if the user wants to group aspects that is executed in a sequence above 70%, our method verifies if aspect A and aspect B are executed in sequence in 70% of the cases. If it is true, then the method suggests grouping them. The output of this step is a list of groups of aspects candidates.

6. EXPLORATORY STUDY

The scientific methodology employed in this work was conducted in 3 stages. In the first stage, it was defined the research problem and the hypothesis, and it was designed the solution approach. At the next stage, a wide theoretical background research was performed, looking for existing techniques or inspiration to be used in the proposed solution. In the next step, the solution was implemented. In the third stage, a preliminary case study was performed to do an initial evaluation. This evaluation corresponded to a comparison of results from an example of the literature. That is, we use the same processes used by Tavares and Marinho [29] to check if our method identifies the same aspects identified by the experts in [29]. The results are presented as follows.

For the exploratory study, we used the business process of an administrative department of an university. The processes were already modeled and we use a tool to simulate them, *i.e.*, to generate synthetic logs corresponding to that processes' models. The first process represents how student obtain the discipline program. The discipline program has the goal to present the content of the discipline including the bibliographical references, themes and topics developed in the discipline. The second process represents how the student can require the school record. Both processes represent the path the student has to follow to obtain those services at this university department.

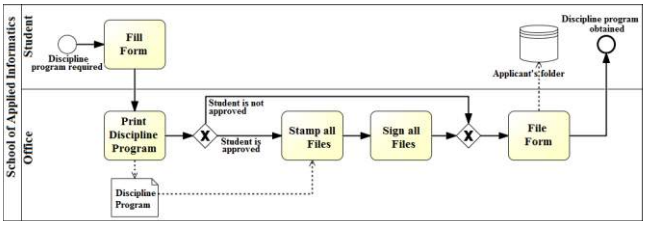


Figure 9 – Require discipline program process.

Figure 9 and Figure 10 illustrate the two processes. Figure 9 represents the process to require the discipline program. The discipline program process begins with a student requiring it and filling the form. The next step is the secretary printing the discipline program, after this activity, the secretary checks if student is approved in the required discipline. If the student is approved, the secretary stamp and sign the files and then file the form. If the student is not approved, the secretary skips the stamp and sign activities, and files the form. The process ends when the student goes to the office to get the discipline program. Figure 10 represents the process to require the school record. This process is very similar to the discipline program process. It also begins with the student filling the form. The secretary prints the school record, stamp and signs the files and files the form. The process also ends when the student goes to the office to get the school record.

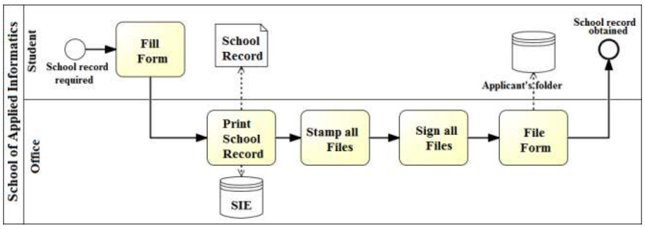
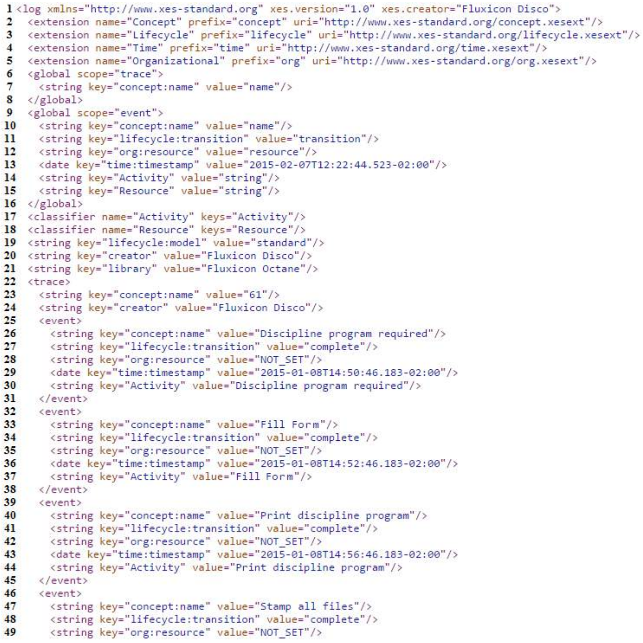


Figure 10 – Require school record process.

We used a log simulator (BIMP Simulator [12] [26]) to create a log to test our implementation. BIMP Simulator is a research prototype available as a part of the modeling platform of the BPM Academic Initiative. This tool allows simulating complex real-world business processes in large-scale scenarios. The system was designed and implemented by Marlon Dumas *et al.* at the University of Tartu [26]. This simulator takes as input a BPMN process models in XML format produced by other process modeling tools such as Signavio Process Editor or OpenText Provision [12]. Figure 11 illustrate part of the generate log. There is the tag log (line 1), which represents the log itself, whereupon it contains the tags trace that contains the tags events. Also, there are the tags of characterization of the log, for example the tag global defines what information each elements of the log must have. In Figure 11, there are two global definitions, one to the trace scope (lines 6 to 8) and other to the event scope (lines 9 to 16). The global definition of trace is requiring that each trace in the log contains a name. The global definition of event is requiring that each event in the log contains a name, a lifecycle transition, resource, timestamp and activity. After the tags of the log characterization, there are the traces tags that contain the events tags. In Figure 11, there is a tag trace with the attribute name and creator which contains 4 events tags (lines 22 and onwards). The first event tag contains "discipline program required" as value from the name field, and it has the others attributes (line 25).



```
1 <log xmlns="http://www.xes-standard.org" xes.version="1.0" xes.creator="Fluxicon Disco">
2   <extension name="Concept" prefix="concept" uri="http://www.xes-standard.org/concept.xesext"/>
3   <extension name="Lifecycle" prefix="lifecycle" uri="http://www.xes-standard.org/lifecycle.xesext"/>
4   <extension name="Time" prefix="time" uri="http://www.xes-standard.org/time.xesext"/>
5   <extension name="Organizational" prefix="org" uri="http://www.xes-standard.org/org.xesext"/>
6   <global scope="trace">
7     <string key="concept:name" value="name"/>
8   </global>
9   <global scope="event">
10    <string key="concept:name" value="name"/>
11    <string key="lifecycle:transition" value="transition"/>
12    <string key="org:resource" value="resource"/>
13    <date key="time:timestamp" value="2015-02-07T12:22:44.523-02:00"/>
14    <string key="Activity" value="string"/>
15    <string key="Resource" value="string"/>
16  </global>
17  <classifier name="Activity" keys="Activity"/>
18  <classifier name="Resource" keys="Resource"/>
19  <string key="lifecycle:model" value="standard"/>
20  <string key="creator" value="Fluxicon Disco"/>
21  <string key="library" value="Fluxicon Octane"/>
22  <trace>
23    <string key="concept:name" value="61"/>
24    <string key="creator" value="Fluxicon Disco"/>
25    <event>
26      <string key="concept:name" value="Discipline program required"/>
27      <string key="lifecycle:transition" value="complete"/>
28      <string key="org:resource" value="NOT_SET"/>
29      <date key="time:timestamp" value="2015-01-08T14:50:46.183-02:00"/>
30      <string key="Activity" value="Discipline program required"/>
31    </event>
32    <event>
33      <string key="concept:name" value="Fill Form"/>
34      <string key="lifecycle:transition" value="complete"/>
35      <string key="org:resource" value="NOT_SET"/>
36      <date key="time:timestamp" value="2015-01-08T14:52:46.183-02:00"/>
37      <string key="Activity" value="Fill Form"/>
38    </event>
39    <event>
40      <string key="concept:name" value="Print discipline program"/>
41      <string key="lifecycle:transition" value="complete"/>
42      <string key="org:resource" value="NOT_SET"/>
43      <date key="time:timestamp" value="2015-01-08T14:56:46.183-02:00"/>
44      <string key="Activity" value="Print discipline program"/>
45    </event>
46    <event>
47      <string key="concept:name" value="Stamp all files"/>
48      <string key="lifecycle:transition" value="complete"/>
49      <string key="org:resource" value="NOT_SET"/>
```

Figure 11 - Generated log.

Based on the logs generated for these two processes by BIMP tool, the proposed method extracts the events from the log. The output of this step is a list containing all the events: "Discipline program required", "Fill Form", "Print Discipline Program", "Stamp all Files", "Sign all Files", "File Form", "Discipline program obtained", "School record required", "Fill Form", "Print School Record", "Stamp and Sign all Files", "File Form".

The next step is the clone approach, which runs through the list searching for equals and synonyms events. In this case study, this approach found four aspects candidates ("Fill Form", "Stamp all files", "Sign all files" and "File Form").

The final step from our method is to apply the cluster approach, which runs through the log searching to group the aspects candidates found by the previous step (the clone approach). In this case study, the threshold to seek the aspects that were executed in sequence was one hundred percent. That is, the cluster approach search through the log checking if the three aspects ("Fill Form", "Stamp all files", "Sign all files" and "File Form") were always executed in sequence. This approach found three aspects ("Stamp all files", "Sign all files" and "File Form") which can be group as one aspect, since in the context of these processes; this three aspects are executed in sequence.

The final results of this exploratory study using the business process of an administrative department of an university illustrated in Figure 9 and Figure 10 were four aspects. It also recommends grouping three of these aspects, since they are always executed in sequence.

Tavares and Marinho [29] used these two same processes as examples to propose improvements on the AO-BPM notation. They used different process models, their models from obtain the discipline program and require the school record were more detail. That is, their models have more activities describing the process. In their work, they found seven aspects ("Deliver form", "Request director signature", "Stamp document", "Sign document", "Return document to the office", "Remove document from the office", "Wait student to pick up document"), wherein the aspects "Request director signature", "Stamp document", "Sign document", "Return document to the office" are group as one aspect and "Remove document from the office", "Wait student to pick up document" are also group in another aspect.

The aspects found in our method correspond to the aspects modelled by Tavares and Marinho's work [29]. Our aspect "Fill form" corresponds to the same activity "Deliver form" from Tavares and Marinho's processes. The same with our group "Stamp all files", "Sign all files" and "File Form" that corresponds to "Request director signature", "Stamp document", "Sign document" and "Return document to the office". Since the process model we used didn't have details of how the student get the document required, our method couldn't find the aspects correspondent of "Remove document from the office" and "Wait student to pick up document" from Tavares and Marinho's work.

This initial case study shows that our method could do the same aspect identification process than a human identification. However, the details from the processes affect the results. That is, the level of abstraction from the processes should be the same to our method work. For example, if we used one process from Tavares and Marinho's work and one process from our work, the method wouldn't discover any aspects because the level of details from the processes are different. The evaluation of this method was qualitative, dependent of specialist opinion to check if the aspects found were accurate. A quantitative evaluation would only be possible if we had a templet available. However, this initial results show that our method could have been used by their work to assist and to facilitate the aspects identification.

This exploratory study has some threats to the validity of the method. The use of simulated logs of trivial processes rather than using actual logs of complex processes is one threat, but there is an ongoing study using real-life logs. The last threat the way activities names are written. Our proposal assumes activities names written accordingly to good practices, such as verb plus noun.

7. CONCLUSION

The use of aspects in models of business processes improves the process modularization, facilitates maintenance, understanding, modeling and reuse parts of the process. Therefore, it facilitates management of the process. However, it is a new field; so, the existing approaches of aspect-oriented models in BPM are being developed following different lines of work. In existing approaches, the identification is performed manually.

This research proposed an approach for automatic identification of aspects from event logs (the process as it was executed - as-is). Jalali [17] pointed out that the mining techniques to discover duplicated activities within one process are an ongoing research with many open questions. In our work, we developed a solution to fill this gap by discovering aspects among different processes. The initial results show that our method could be used to identify duplicated events through mining the processes logs.

This research provides with a technical method to identify aspects in business process management without depending on experts. The initial results are optimistic and for future work would be develop the analysis fan-in approach. Another future work is semantic analysis of the names of the activities reducing the complexity of synonyms algorithm and improving its specificity. Besides, we propose the analysis of activities relations, such as successors and their predecessors of the activities as a way to prioritize the search for aspects. Our next step is to prepare a broader case study to apply the evaluation. The future evaluation is going to be the analysis by specialists in a qualitative manner and also in a statistic manner by metrics calculating the hits and misses of the method proposed.

8. REFERENCES

[1] Azevedo, L.G. Santoro, F. Baiao, F., Souza, J., Revoredo, K., Pereira, V., Herlain, I. 2009. "A Method for Service Identification from Business Process Models in a SOA Approach". In T. Halpin et al., eds. Enterprise, Business-Process and Information Systems Modeling. pp. 99-112.

[2] Azevedo, L. G., Baiao, F., Santoro, F. Souza, J. F. 2011. "A Business Aware Service Identification and Analysis Approach". In: IADIS International Conference Information Systems 2011, March, 11-13, Avila, Spain.

[3] Barbosa, F. S. "Comparing Three Aspect Mining Techniques." 2008. Doctoral Symposium in Informatics Engineering (DSIE'08). Portugal.

[4] Buijs, J. C. A. M., van Dongen, B. F., van der Aalst, W. M. P. 2013. "Mining Configurable Process Models from Collections of Event Logs". BPM 11th Conference, China.

[5] Campos J. P., Braga J. L., Resende A. M. P., Silva C. H. O. 2010. "Identification of aspect candidates by inspecting use cases descriptions". SIGSOFT, vol 35, pp 1-9.

[6] Cappelli, C., Leite, J., Batista, T; Silva, L. 2009. "An Aspect-Oriented Approach to Business Process Modeling.". EA-AOSD, USA.

[7] Cappelli, C., Santoro, F. M., Leite, J. C. S. P., Medeiros, A. L., Batista, T., Romeiro, C. S. C. 2010. "Reflections on the modularity of business process models. The case for introducing the aspect-oriented paradigm". BPM Journal Vol. 16.

[8] Casachi, R. A., Camolesi, A. R. 2012. "Uso de Programação Orientada a Aspecto no Desenvolvimento de Aplicações que utilizam conceitos de Tecnologia Adaptativa". Adaptive Technology Workshop.

[9] Charfi, A., Mezini, M. 2004. "Aspect-oriented web service composition with AO4BPEL". In Web Services (pp. 168-182). Springer Berlin Heidelberg.

[10] Charfi A., Müller H., Mezini M. 2010. "Aspect-Oriented Business Process Modeling with AO4BPMN". In T. K. et al., editor, Modelling Foundations and Applications, volume 6138 of LNCS, pages 48-61. Springer.

[11] Collell, D. C. 2012. “Aspect-oriented modeling of business processes”. Master's thesis, der Technischen Universitat Darmstadt, Darmstadt.

[12] Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A. 2013. “Fundamentals of Business Process Management”. Springer.

[13] Garcia, R. 2010. "O que é Programação Orientada a Aspectos?". Java Framework Portal. Available at <http://www.javaframework.org/portal/2010/04/14/o-que-programao-orientada-a-aspectos>. Accessed in May, 2014.

[14] Günther W. C., Verbeek E. 2014. "OpenXES - Developer Guide 2.0". Available at: http://www.xes-standard.org/openxes/developerguide.Accessed on February, 2015.

[15] Kiczales, G., Hugunin, J., Hilsdale, E., Kersten, M., Palm, J., Lopes, C., Griswold, B., Isberg, W. 2003. "Aspect Oriented Programming". Final Technical Report from Air Force Research Laboratory from Palo Alto Research Center. Rome, New York. July.

[16] Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes C., Loingtier J., Irwin, J. 1997. “Aspect-Oriented Programming”. European Conference on Object-Oriented Programming (ECOOP), Finland. Springer-Verlag LNCS 1241. June.

[17] Jalali, A. 2014. "Assessing aspect oriented approaches in business process management." Perspectives in Business Informatics Research. Springer International Publishing, p231-245.

[18] Jalali, A. 2014. "Aspect Mining in Business Process Management." Perspectives in Business Informatics Research. Springer International Publishing, p246-260.

[19] Josuttis, N. 2007. “SOA in practice: The Art of Distributed System Design”. Beijing; Cambridge. O’Reilly, 324p.

[20] OMG. 2011. “Business Process Modeling Notation Specification”. January.

[21] Pahlsson, N. 2012. "Aspect-Oriented Programming - An Introduction to Aspect-Oriented Programming and AspectJ". Department of Technology from University of Kalmar. Topic Report for Software Engineering. Novembro.

[22] Rago, A., Abait E., Marcos C., Diaz-Pace A. 2009. “Early Aspect Identification From Use Cases Using Nlp And Wsd Techniques”. 15th Workshop on Early aspects from International Conference on Aspect-Oriented Software Development.

[23] Richetti, P. H. P., Baião F. A., Santoro F. M. 2014. "Declarative Process Mining: Reducing Discovered Models Complexity by Pre-Processing Event Logs." Business Process Management. Springer International Publishing, p400-407.

[24] Sampaio A., Chitchyan R., Rashid A., Rayson P. 2005. “EA-Miner: a tool for automating aspect-oriented requirements identification”. 20th IEEE/ACM International Conference on Automated software engineering.

[25] Santos, F. J. N., Leite, J. C. S. P., Cappelli, C., Batista, T. V., Santoro, F. M. 2011. "Using goals to identify aspects in business process models." Proceedings of the 2011 international workshop on Early aspects. ACM.

[26] Signavio. 2014. "BPM Academic Initiative".Available at < http://www.signavio.com/bpm-academic-initiative/>. Accessed in December, 2014.

[27] Soares, A. H. V., Rocha, A. de R., Alves, F. L., Alves, J. C. 2012. "Programação Orientada a aspectos - uma visão geral". Department of Computer Science of the Federal University of Lavras.

[28] Souza A., Cappelli C., Santoro F., Azevedo L. G., Leite J. C. S. do P. 2011. ”Service Identification in Aspect-Oriented Business Process Models”. SOSE 6th International Symposium.

[29] Tavares F., Marinho L. 2014. “AO-BPM 2.0: Modelagem de Processos Orientada a Aspectos”. Final graduation work. Available at: http://bsi.uniriotec.br/tcc/publicacoes.html.

[30] Van der Aalst W.M.P, Weijters A.J.M.M. 2004. “Process mining: a research agenda”. Journal Computers in Industry – Special Issue: Process/Workflow mining Vol. 53 Issue 3, pp. 231 – 244. Elsevier Science Publishers B. V. Amsterdam. Abril.

[31] Van der Aalst, W.M.P., Adriansyah, A., Alves De Medeiros, A.K., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., Brand, P.C.W. van den, Brandtjen, R., Buijs, J.C.A.M., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., Leoni, M. de, Delias, P., Dongen, B.F. van, Dumas, M., Dustdar, S., Fahland, D., Ferreira, D.R., Gaaloul, W., Geffen, F. van, Goel, S., Gunther, C.W., Guzzo, A., Harmon, P., Hofstede, A.H.M. ter, Hoogland, J., Ingvaldsen, J.E., Kato, K., Kuhn, R., Kumar, A., La Rosa, M., Maggi, F.M., Malerba, D., Mans, R.S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Motahari Nezhad, H., Muehlen, M. zur, Munoz-Gama, J., Pontieri, L., Ribeiro, J.T.S., Rozinat, A., Seguel Peréz, H., Seguel Pérez, R.E., Sepúlveda, M., Sinur, J., Soffer, P., Song, M.S., Sperduti, A., Stilo, G., Stoel, C., Swenson, K., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, H.M.W., Verdonk, M.C., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, A.J.M.M., Wen, L., Westergaard, M. & Wynn, M.T. 2011. “Process mining manifesto”. In F. Daniel, K. Barkaoui & S. Dustdar (Eds.), Business Process Management Workshops (BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I), (Lecture Notes in Business Information Processing, 99, pp. 169-194). Berlin: Springer.

[32] Verbeek, H.M.W. 2010. ”ProM 6 Getting Started”. ProM documentation. Available at < http://www.promtools.org/prom6/downloads/prom-6.0-getting-started.pdf >. September.

[33] Wang, J., Zhu, J., Liang, H., Xu, K. 2005. “Aspect-Oriented Business Process Modeling”. Research Report, IBM – China.