On Similarity of Object-Aware Workflows

Mohammad Javad Amiri, Mahnaz Koupaee, Divyakant Agrawal
Department of Computer Science, University of California Santa Barbara
Santa Barbara, California
{amiri, koupaee, agrawal}@cs.ucsb.edu

***Abstract*—Business processes (workflows) are typically the compositions of services (activities and tasks) and play a key role in every enterprise. Finding similar processes in process repositories helps enterprises to reduce their cost and increase their performance. The similarity of different business processes has been measured based on activity labels and structural factors. However, inaccurate and incomplete labels and the existence of multiple labels for similar activities affect the accuracy of the existing methods. Furthermore, with recent advances in business process management and developing innovative paradigms like artifact-centric and decision-aware process modeling, data has become an inseparable part of the process modeling. While data objects and the way they are accessed are recently used to measure the similarity of activities, this approach does not address activities with different granularities. In this paper, we present an approach to measure the similarity of business processes based on the similarity of the life cycles of their objects. The experiments show the effectiveness of the approach to improve the accuracy of the processes similarity task.**

***Index Terms*—Workflow, Process Similarity, Object Life Cycle**

I. Introduction

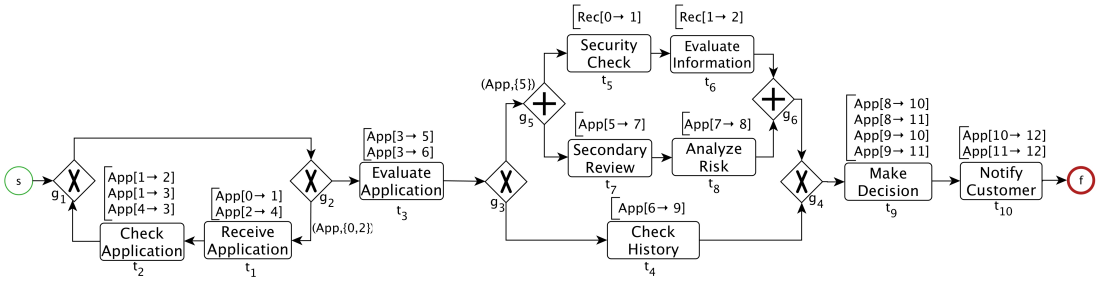
A business process consists of a set of activities performed in coordination in an organizational environment to accomplish a business goal. Business process management (BPM) includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes [19]. An important problem in BPM is to determine whether two process models exhibit similar behaviors [18]. Many large enterprises require hundreds of processes to fulfill their duties. For example, the total number of business process models in the SAP reference model or the repository of Dutch Local Governments exceeds 600 [1] and in OA System of CMCC this number has been over 8,000 [7] of which many are similar or even identical. While most existing approaches rely on the structure of the process and label of activities to find similar processes, in this paper, we study the role of data objects in process similarity measurement.

Process similarity measurement is a significant task in business process management for multiple reasons; first, the identification and merging of identical or similar processes being executed in different parts of an organization prevent the duplication of activities. Similarity measurement can also be used by multinational enterprises to identify national branch processes that no longer comply with the enterprise reference model [18]. In brief, scenarios such as organization merging, user requirements changing, and model repository management are some applications of process similarity measurement [7]. Furthermore, businesses are constantly being changed and extended. In many cases, separate small businesses unite with each other and form a single business; identifying similar business processes can be used to reduce the cost of expanding businesses [6].

Most of similarity measurement approaches depend on the activity labels. To calculate the similarity of activity labels, schema matching [15] and ontology matching [8] techniques are used. Labels can be compared either syntactically or semantically. To compare labels on the syntactic level, string edit distance is used in some approaches [11]. Some other methods tokenize strings into words and then compare the similarity [17]. Natural language processing techniques [12] [14] are used to compare labels on semantic level. There are several drawbacks of using labels in similarity measurement. First of all, labels may be chosen in an inexact and incomplete way that does not reflect what really a task does. Besides, there might be meaningless labels which do not convey useful information about the activities. Moreover, different words or synonyms can be used to describe the same activity [5], thus the same activity might have multiple labels. Although there have been a lot of attempts to overcome the challenges caused by labels such as ontology-based techniques and NLP, which help reduce the adverse effects of mentioned problems, yet the accuracy of those methods is not satisfactory.

Recently, there has been significant attention toward using data objects in business process management. In order to fulfill a business process, various data objects are required and the result of process execution can be observed in the creation or updating of data. Hence, data is considered as the core of a business process in different frameworks and paradigms like data-aware [10], artifact-centric [13] [9], and decision-aware [4] process modeling. Modeling data accesses as a part of a process is also supported in modeling standards such as BPMN and organizations attempt to model data while modeling the process in order to show their processes more precisely.

To deal with different, multilingual or meaningless activity labels, and due to the importance of using data objects, in our prior work [3], data access patterns are used to find similar processes. In that approach, first, the similarity of different activities based on their data objects is measured, and then these similarities are used to measure the similarity of a pair of business processes. While considering data objects makes the process similarity task more accurate, that approach still cannot address the existence of activities with different granularities. Business process designers have different opinions



App: Application, Rec: LoanRecord. States of Rec: 0(RecCreated), 1(SecurityChecked), 2(InfoEvaluated). States of App: 0(Initiated), 1(Received), 2(Incomplete), 3(Complete), 4(Resubmitted), 5(MoreInfoNeeded), 6(Evaluated), 7(Reviewed), 8(RiskAnalyzed), 9(HistoryChecked), 10(LoanApproved), 11(LoanDenied), 12(Archived).

Fig. 1. The *LoanApproval* Process P_1

on the granularity of activities. While some of them define activities as fine-grained atomic tasks that might perform a single read/write operation, some others define activities as independent coarse-grained components that fulfill a service. As a result, an activity in a process model might perform the same task as what a collection of activities in another process does. Therefore, activity-based process similarity approaches might fail even when data objects are taken into account.

To tackle the problem of handling activities with different granularities, *object life cycles* can be used to measure the similarity of different processes. Each process uses different data objects where the state of each object evolves during the execution of a process. The behavior of data objects in terms of state changes is usually modeled using a variant of a state machine called an object life cycle [16]. Since object life cycles can be shown independent of the underlying business processes, the granularity of activities does not play any role in object life cycles.

This paper focuses on processes with stateful objects and studies the business process similarity problem. Note that while data is not modeled, "finite domain" data can be captured using object states. We present an approach to measure the similarity between two different business processes using object life cycles. For each object within a business process, the object life cycle in the form of a directed graph is extracted. Then the life cycle graph of an object in two different processes is compared using graph similarity techniques. Finally, the overall similarity of all the objects within a pair of processes is calculated and considered as the similarity of those two processes.

This paper is organized as follows. Section II illustrates the complexity issue with a concrete example. Process modeling is presented in Section III. Section IV measures the similarity of business processes using their data objects. In Section V, an evaluation of both approaches is presented, and Section VI concludes the paper.

II. Motivations

Consider a *LoanApproval* process in a bank, where customers can apply for a loan by submitting an application. A bank employee reviews it to check the completeness of the application. A complete application is then evaluated. Depending on the requested loan amount and customer qualifications, different paths can be chosen. For a small loan or a low-risk customer, the bank only checks the customer's history, but if the loan amount is large or this is the first loan of the customer, more checking is required. For this group of applications, the bank checks and validates some security information and at the same time does a secondary review on the application and a thorough credit risk analysis. The bank then makes a decision to approve or reject the loan request and the application is archived after notifying the customer.

Fig. 1 and Fig. 2 show two different BPMN process models for the *LoanApproval* workflow. The *LoanApproval* process uses App(lication) and (Loan)Rec(ord) objects and changes their states as the process proceeds.

Let's assume that these two models belong to two different branches of the same bank and the bank has decided to merge (unify) these two branches; thus it checks the processes to see whether they are similar or not. Indeed, different branches of the bank access the same database, however, their workflow for the same business process might be different. *LoanApproval* processes P_1 and P_2 (Fig. 1 and Fig. 2) seem to be different at first glance; from the structural point of view, the number of activities is different. They have different gateways, and while the first model has a loop block and a parallel block within a conditional fragment, the second model has only a nested conditional block. Furthermore, the labels of the activities diverge entirely and it is difficult to find a relation between activity nodes of one process with the activity nodes of the other one.

However, if we check the states of the objects, i.e., App and Rec, and how these states change along the process, a high percentage of similarity between these two models can be observed. The state of Rec object in both models changes for 0 to 1, and then from 1 to 2. We call the sequence of these states the *life cycle* of the object, e.g., the life cycle of Rec object is $[0 \rightarrow 1 \rightarrow 2]$. The life cycle of App object for *LoanApproval* processes P_1 and P_2 is shown in Fig. 3 and Fig. 4 respectively. As can be seen, the only difference is that in process P_1, if the received application (request) is incomplete, the application is returned to the customer, so that the customer has one more chance to complete and resubmit it. However, in process P_2, if the received application is incomplete, the bank instantly

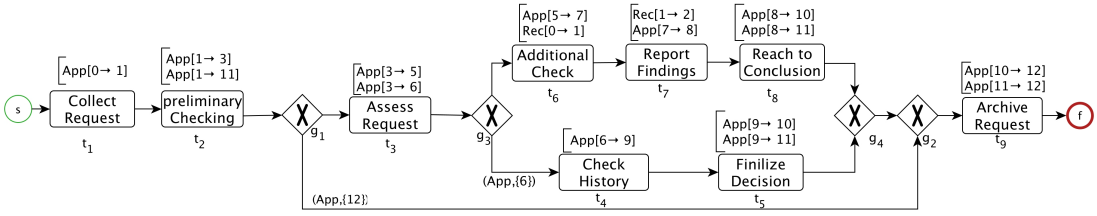


Fig. 2. The `LoanApproval` Process P_2

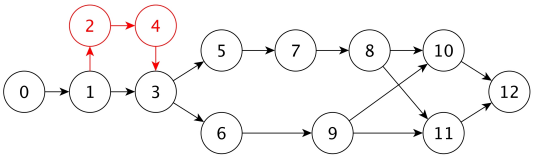


Fig. 3. The Life cycle of `App` Object in `LoanApproval` Process P_1

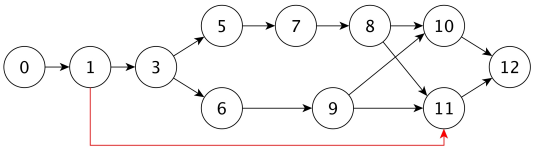


Fig. 4. The Life cycle of `App` Object in `LoanApproval` Process P_2

rejects the application.

From this simple process, we observe that two processes even with different structures in terms of the number of activities and the activity labels might have similar behavior.

III. Process Modeling

In this section, we first introduce a model for business processes and then define the life cycle of objects which are used by business processes.

We assume the existence of a countably infinite set $\mathcal{I}$ of (object) *identifiers* (or IDs). Let S be a finite set of states. An *object* is a pair (o, q) where $o \in \mathcal{I}$ is a (unique) ID and $q \in S$ is a state.

In this paper, we adopt the formal definition of a process from [2]. We focus on one type of edges corresponding to "sequence flow" in BPMN, and three types of nodes: "event", "activity", and "gateway". Furthermore, we consider two classes of *events*: *start* and *final* events.

An *activity* represents a unit of work. In our model an activity has a name (label) and works on a set of objects where the activity might change the states of its objects.

Definition 1. An *activity* is a triple (α, O, τ) where α is a unique activity name, O is a set of object IDs, and $\tau \subseteq S^{|O|} \times S^{|O|}$ is a set of transitions with $|O|$ incoming and $|O|$ outgoing states.

We denote an activity (α, O, τ) simply by α and the set of all activities by $\mathcal{A}$.

Note that a *silent* activity, i.e., an activity that does nothing, can be presented with the empty sets of objects and transitions. Activities here can also model decision nodes in decision-aware process modeling [4].

A *gateway* controls the divergence and convergence of sequence (execution) flows. There are four kinds of gateways: (exclusive) *choice* and *merge*, and (parallel) *split* and *join*. A choice gateway simulates an *if-else* statement: exactly one of the outgoing flows will be chosen. A merge gateway continues an incoming flow, a split gateway forwards a flow to every outgoing edge, and finally, a join gateway synchronizes flows from all incoming edges and combines them into one outgoing edge. We use symbols `C`, `M`, `S`, and `J` to denote choice, merge, split, and join gateways, respectively.

Definition 2. A *Business Process schema* is a tuple $P = (N, s, f, L, E, O)$ where

- N is a finite non-empty set of nodes,
- s and f are the start and final events,
- L is a labeling function that assigns either an activity in $\mathcal{A}$ or a gateway (`C`, `M`, `S`, or `J`) to each node in N,
- $E \subseteq (N - \{f\}) \times (N - \{s\})$ is a finite set of control flow edges such that
  1) s has one outgoing and no incoming edge,
  2) f has one incoming and no outgoing edge,
  3) each activity node has one incoming and one outgoing edge,
  4) each choice or split gateway node has one incoming and at least two outgoing edges, and
  5) each merge or join gateway node has at least two incoming and one outgoing edges,

  and
- $O \subseteq \mathcal{I}$ is a finite set of data objects.

The size of a process schema P (denoted as $|P|$) is the number of activity nodes within the process.

Example 3. Continuing with the process shown in Fig. 1, nodes s and f are the start and end nodes, t_i's ($1 \leqslant i \leqslant 10$) and g_i's ($1 \leqslant i \leqslant 6$) are the activity and gateway nodes in N, and (s, g_1), (g_1, g_2), ..., (t_9, t_{10}), and (t_{10}, f) are the set of edges in E. Function L is used to assign labels to nodes. For t_i, $1 \leq i \leq 10$, $L(t_i) = \alpha_i$ which is an activity, e.g., $L(t_1) = (\alpha_1, \{\texttt{App}\}, ((\texttt{App}_0, \texttt{App}_1), (\texttt{App}_2, \texttt{App}_4))$. For the gateway nodes, $L(g_1)$=$L(g_4)$=`M`, $L(g_2)$=$L(g_3)$=`C`, $L(g_5)$=`S`, and $L(g_6)$=`J`. Finally, $\{\texttt{App}, \texttt{Rec}\}$ is the set of objects. Similarly, different elements of the process P_2 (Fig. 2) can be explained. □

We now define *object life cycle*. For each object, its life cycle shows the possible state changes during the period of the execution of a business process.

Definition 4. Given a process schema $P=(N, s, f, L, E, O)$, for each object $o \in O$, $G_P(o) = (Q, T)$ is the *object life cycle* of object o in process P where

- $Q \subseteq S$ is a set of states, and
- $T \subseteq Q \times Q$ is a set of state transitions such that for each transition $(q, q') \in T$, there is a set of activity nodes $A \subseteq N$ where for each $t \in A$ such that $L(t) = (\alpha, O, \tau)$, $(q, q') \in \Pi_o(\tau)$.

Note that $\Pi_o(\tau)$ is the *projection* of relation τ over object o resulting in a pair of in- and out-states.

The size of an object life cycle G_P (denoted as $|G_P|$) is the number of states (nodes) within the graph.

Example 5. Consider the `LoanApproval` process P_1 (Fig. 1), the life cycle of object `Rec` is $G_{P_1}(\texttt{Rec}) = (\{0, 1, 2, \}, \{(0, 1), (1, 2)\})$. The life cycle of object `App`, as can be seen in Fig. 3, is $G_{P_2}(\texttt{App}) = (\{0, 1, ..., 12\}, \{(0, 1), (1, 2), ..., (10, 12), (11, 12)\})$.

The life cycle of object `App` in the `LoanApproval` process P_2 (Fig. 2) is shown in Fig. 4. □

IV. Object-based Similarity of Business Processes

In this section, we present a method to measure the similarity of business processes based on the similarity of their data objects. We first define the object life cycle similarity for a pair of data objects, and then present a method to measure the similarity of two business processes using the similarities of their data objects.

The similarity of a pair of object life cycles is measured by comparing the position of states in both graphs. For each state in an object life cycle, we define two sets of predecessor and successor states.

Definition 6. Given a process schema $P=(N, s, f, L, E, O)$, Let $G_P(o) = (Q, T)$ be the object life cycle of object $o \in O$, For each state $q \in Q$:

- $\sigma_P(q) = \{r \mid (q, r) \in T\}$ is the set of *successor* state nodes of q, and
- $\pi_P(q) = \{r \mid (r, q) \in T\}$ is a set of *predecessor* state nodes of q.

Example 7. Consider the life cycle of `App` object in `LoanApproval` process P_1 which is shown in Fig. 3. The predecessor and successor sets for some of the states are computed as follows. $\sigma_{P_1}(1)=\{2, 3\}$, $\pi_{P_1}(1)=\{0\}$, $\sigma_{P_1}(3)=\{5, 6\}$, $\pi_{P_1}(3)=\{1, 4\}$, $\sigma_{P_1}(12)=\varnothing$, and $\pi_{P_1}(12)=\{10, 11\}$. □

When the predecessor and successor states of each state are determined, the *state similarity* of two states can be computed by comparing the corresponding predecessor and successor states.

Definition 8. Given process schemas $P = (N, s, f, L, E, O)$ and $P' = (N', s', f', L', E', O')$, let $G_P(o) = (Q, T)$ and $G_{P'}(o) = (Q', T')$ be the life cycles of object $o \in (O \cap O')$ in P and P' respectively. For each state $q \in (Q \cap Q')$ the *state similarity* of q is

$$sim^s_{(P,P')}(q)=\left(\frac{|\sigma_P(q) \cap \sigma_{P'}(q)|}{|\sigma_P(q) \cup \sigma_{P'}(q)|}+\frac{|\pi_P(q) \cap \pi_{P'}(q)|}{|\pi_P(q) \cup \pi_{P'}(q)|}\right)/2 \quad (1)$$

Note that if the state has no predecessor (successor) in both process schemas, the corresponding part of the equation 1 is considered as 0.

Example 9. Consider the life cycles of `App` object in `LoanApproval` Processes P_1 and P_2, the state similarity of different states are computed as follow:

- $sim^s_{(P_1,P_2)}(1) = 0.5 * \left(\frac{|\{2,3\} \cap \{3,11\}|}{|\{2,3\} \cup \{3,11\}|} + \frac{|\{0\} \cap \{0\}|}{|\{0\} \cup \{0\}|}\right) = 0.66$
- $sim^s_{(P_1,P_2)}(9) = 0.5 * \left(\frac{|\{10,11\} \cap \{10,11\}|}{|\{10,11\} \cup \{10,11\}|} + \frac{|\{6\} \cap \{6\}|}{|\{6\} \cup \{6\}|}\right) = 1$

Similarly, for all the states of the life cycle of `App` object in both `LoanApproval` Processes P_1 and P_2, the state similarity can be computed as follows. $sim^s_{(P_1,P_2)}(i) = 1$ for $i \in \{0, 5, 6, 7, 8, 9, 10, 12\}$, $sim^s_{(P_1,P_2)}(1) = 0.66$, $sim^s_{(P_1,P_2)}(3) = 0.75$, and $sim^s_{(P_1,P_2)}(11) = 0.83$.

□

When the similarity of two object life cycles regarding a state is computed, we can compute the similarity of two object life cycles, called *object similarity* by computing the average of the similarity of their states.

Definition 10. Given process schemas $P = (N, s, f, L, E, O)$ and $P' = (N', s', f', L', E', O')$, let $G_P(o)=(Q, T)$ and $G_{P'}(o)=(Q', T')$ be the life cycles of object $o \in (O \cap O')$ in P and P' respectively. The *object similarity* of o is

$$sim^o(G_P(o), G_{P'}(o)) = \frac{\Sigma_{q \in (Q \cap Q')} sim^s_{(P,P')}(q)}{|Q \cup Q'|} \quad (2)$$

Note that we compute state similarity for the states which are shared between both object life cycles, however, to compute the object similarity, all the states are taken into account, i.e., the denominator of sim^o relation in Equation 2 is the set of all states in both object life cycles $(Q \cup Q')$.

Example 11. Consider the life cycle of `App` and `Rec` objects in `LoanApproval` Processes P_1 and P_2, $sim^o(G_{P_1}(\texttt{App}), G_{P_2}(\texttt{App})) = \frac{10.24}{13}$=0.79 and $sim^o(G_{P_1}(\texttt{Rec}), G_{P_2}(\texttt{Rec})$=1. □

Finally, we can compute the similarity of two business processes using the similarity of their objects. Since object life cycles have a different number of nodes, we take the size of object life cycles into account.

Definition 12. Given process schemas $P = (N, s, f, L, E, O)$ and $P' = (N', s', f', L', E', O')$, for each object $o \in (O \cap O')$, let $G_P(o) = (Q_o, T_o)$ and $G'_P(o) = (Q'_o, T'_o)$ be the object life cycles of o in P and P' respectively. The *process similarity* of P and P is

$$sim^D(P, P')=\frac{\Sigma_{o \in (O \cap O')}(sim^o(G_P(o), G_{P'}(o)) * |\{Q_o \cup Q'_o\}|)}{\Sigma_{o \in (O \cup O')}|\{Q_o \cup Q'_o\}|)} \quad (3)$$

TABLE I
MEASURING THE SIMILARITY OF DIFFERENT PROCESSES USING BOTH ACTIVITY-BASED AND OBJECT-BASED APPROACHES

Process	P_{i1}		P_{i2}		P_{i3}		P_{i4}		Score(10)	
	Sim^A	Sim^D	Sim^A	Sim^D	Sim^A	Sim^D	Sim^A	Sim^D	Sim^A	Sim^D
P_0	0.93	0.94	0.84	0.75	0.65	0.77	0.61	0.64	9	8
P_1	0.88	0.76	0.84	0.88	0.82	0.81	0.45	0.70	1	10
P_2	1	1	0.81	0.92	0.68	0.77	0.62	0.81	10	9
P_3	0.94	0.78	0.91	0.82	0.72	0.75	0.64	0.72	7	10
P_4	0.73	0.85	0.54	0.73	0.48	0.61	0.44	0.79	9	8
P_5	0.91	0.88	0.84	0.78	0.70	0.83	0.54	0.67	10	8
P_6	0.79	0.85	0.75	0.69	0.72	0.76	0.57	0.71	8	9
P_7	0.96	0.96	0.93	0.91	0.87	0.82	0.87	0.84	9	10
P_8	0.98	0.96	0.91	0.89	0.66	0.71	0.63	0.75	9	10
P_9	0.66	0.74	0.58	0.71	0.47	0.63	0.44	0.66	10	10

Example 13. Continuing with `LoanApproval` Processes P_1 and P_2, $sim^D(P_1, P_2) = \frac{(0.79*13)+(1*3)}{13+3} = \mathbf{82.75}\%$

□

V. EVALUATIONS

In this section, two sets of experiments are conducted to compare the performance of the object-based (Sim^D) similarity measurement approach with activity-based similarity (Sim^A) measurement approach (our prior work [3]). We create a repository of 10 real processes where each process has five different instances (a base process model and four variants). These processes are taken from four different enterprises and most of the variants of each process are obtained from papers, reference models, and real businesses and some others are designed manually. Basically, the repository contains 50 process schemas from 10 different processes where different variants of a process access the similar data objects. Since the origins of these processes are different, we have modified some of the data objects.

We measured the similarity of a (base) process model and its four variants using two approaches: activity-based similarity (Sim^A) and object-based similarity (Sim^D). We then asked the domain experts of each enterprise to rank the variants based on the similarity with the base model. For each process, three experts are asked to rank variants separately. In the case of conflicting ranking, we ask them to discuss and agree on rankings. From these ten processes, in two cases experts did not agree on the ranking even after negotiation, so we just consider the average of all three rankings and rank the variants. For each approach the variants are also ranked based on their similarity scores; the highest score is ranked as 1 and the lowest one is ranked as 4.

Next, we quantify the conformance of each of the two approaches with experts' opinions using the resulting ranks. Given a base process, if the approach identifies the variant with the maximum similarity score correctly (based on the experts' opinions), the approach gains 4 points. This point for rank 2, rank 3 and rank 4 variants are 3, 2, and 1 respectively. Also if we just need to substitute rank 1 and rank 2 variants, the approach gains 4 points (from 7 possible points), for the substitution of rank 2 and rank 3, the approach gains 3 points (instead of 5 possible points), and it gains 2 points if we need to substitute rank 3 and rank 4 variants. Since there are four variants (other than the base process) for each process, the maximum possible gained point is 10 per process.

Table I shows the results of this step. Each row belongs to a process P_i and its variants P_{i1}, P_{i2}, P_{i3}, and P_{i4}, ($0 \leq i \leq 9$). For each variant, we specify the similarity score resulted by each approach. Then we compare the rankings resulted by each approach (based on their similarity scores) by the rankings of experts' opinions and give each approach a score out of 10. For example for process P_1, similarity scores of the activity-based approach for variants P_{11}, P_{12}, P_{13}, and P_{14}, are 0.93%, 0.84%, 0.65%, and 0.61% respectively. Since P_{11} has the maximum score, its rank is 1. Similarly, P_{12}, P_{13}, and P_{14} are ranked as 2, 3, and 4. From the experts point of view also, since variant P_{11} has the most similarity to the base model, its rank is 1, P_{12} is ranked as 2, P_{14} ranked as 3 and finally P_{13} ranked as 4. As a result, process P_1 gets 9 point, since it ranks variant 1 and 2 correctly and we just need to substitute variant 3 and 4. Similarly, other processes and the object-based approach are scored.

By computing the total scores, the activity-based approach gets totally 82 points out of 100, and the object-based approach gets totally 92 points, which shows that the object-based approach has better performance. As can be seen, even the activity-based similarity measurement has a satisfactory result because to measure the similarity of processes in that approach both structural and behavioral metrics are taken into account.

It should also be noted that for the most cases where an approach ranks a process incorrectly, the similarity scores of variants are very close to each other. For example in Process P_2, the difference between the similarity score of the first and the second instances resulted by the activity-based approach is only 0.03.

In the next set of experiments, we measure *precision* and *recall* of both approaches. We defined a threshold for similar processes and found the variants with the similarity score more than the threshold. The threshold is defined based on the experts' opinions and depends on the processes (it is around 0.8 for most of the processes). The experts are asked to find the variants of each process with the similarity more than the defined threshold. We then measure the precision and recall of both approaches based on these results. Table II shows the

TABLE II
SIMILAR PROCESSES' VARIANTS DETECTED BY DIFFERENT APPROACHES

	Activity-based	Object-based	Experts
P_0	P_{01}, P_{02}	P_{01}	P_{01}
P_1	P_{11}, P_{12}, P_{13}	P_{11}, P_{12}	P_{11}, P_{12}
P_2	P_{21}, P_{22}	P_{21}	P_{21}
P_3	P_{31}, P_{32}	P_{31}	P_{31}, P_{32}
P_4	$\varnothing$	P_{41}	P_{41}
P_5	P_{51}, P_{52}	P_{51}, P_{53}	P_{51}
P_6	$\varnothing$	P_{61}	P_{61}
P_7	P_{71},P_{72},P_{73},P_{74}	P_{71},P_{72},P_{73},P_{74}	P_{71},P_{72},P_{73},P_{74}
P_8	P_{81}, P_{82}	P_{81}, P_{82}	P_{81}, P_{82}
P_9	P_{91}	P_{91}, P_{92}	P_{91}

results of both approaches and experts' opinions.

We then categorize these 40 variants into four groups to measure the *precision*, *recall*, and *accuracy* of both approaches, i.e., activity-based similarity (Sim^A) and object-based similarity (Sim^D). The four groups are: (1) Similar variants that are detected as similar (*True Positive*), (2) Similar variants that are detected as non-similar (*False Negative*), (3) Non-similar variants that are detected as similar (*False Positive*), and (4) Non-similar variants that are detected as non-similar (*True Negative*). Table III shows the results of evaluating both approaches by counting the number of variants that are categorized in each group.

We now use the results from Table III to compute the precision, recall, and accuracy of both approaches. These three metrics are defined based on the values of True Positive(TP), False Negative(FN), False Positive(FP), and True Negative(TN) where *precision* is defined as $\frac{TP}{TP+FP}$, *recall* is equal to $\frac{TP}{TP+FN}$, and *accuracy* is $\frac{TP+TN}{TP+TN+FP+FN}$.

For the activity-based similarity, precision is $\frac{14}{14+4} = 0.78$, recall is $\frac{14}{14+2} = 0.87$, and accuracy is equal to $\frac{14+20}{14+20+4+2} = 0.85$ and for the object-based similarity, precision is $\frac{15}{15+2} = 0.88$, recall is $\frac{15}{15+1} = 0.94$, and accuracy is $\frac{15+22}{15+1+2+22} = 0.93$. As can be seen, the object-based approach has a better result than the activity-based one due to considering the life cycle of data objects.

VI. CONCLUSION

Measuring the similarity of business processes is significant in the business process management domain for various reasons. In this paper, an approach to measure the similarity of business processes considering the life cycle of objects is proposed. Each process uses different data objects where the state of each object evolves during the execution of a process. Object life cycles are mainly used to address the problem of handling activities with different granularities.

Many interesting questions remain. While in this paper data is modeled as a set of stateful objects, other attributes of data objects and relations between them (in terms of primary key, foreign key) can also be taken into account. In addition, the proposed method measures the similarity of a pair of processes, thus to find the most similar pair of processes within a repository, we need to check every pair of processes separately. An efficient algorithm might be able to find the most similar pair without comparing all the existing pairs. Furthermore, although the life cycle of different objects in the presented approach is considered independently, there might be some dependency between their life cycles, therefore considering those dependencies could improve the accuracy of the process similarity approach.

TABLE III
MEASURING PRECISION AND RECALL IN SAMPLE PROCESSES

Metrics	# of variants for different approach	
	Activity-based	Object-based
True Positive	14	15
False Negative	2	1
False Positive	4	2
True Negative	20	22

REFERENCES

[1] Documentair structuurplan. http://www.model-dsp.nl/.

[2] M. J. Amiri and D. Agrawal. View: An incremental approach to verify evolving workflows. In *ACM/SIGAPP Symposium on Applied Computing (SAC)*. ACM, 2019.

[3] M. J. Amiri and M. Koupaee. Data-driven business process similarity. *IET Software*, 11(6):309–318, 2017.

[4] K. Batoulis and M. Weske. Soundness of decision-aware business processes. In *Int. conf. on Business Process Management (BPM)*, pages 106–124. Springer, 2017.

[5] M. H. Baumann, M. Baumann, S. Schönig, and S. Jablonski. Towards multi-perspective process model similarity matching. In *Enterprise and Organizational Modeling and Simulation*, pages 21–37. Springer, 2014.

[6] R. Dijkman, M. Dumas, B. Van Dongen, R. Käärik, and J. Mendling. Similarity of business process models: Metrics and evaluation. *Information Systems*, 36(2):498–516, 2011.

[7] Z. Dong, L. Wen, H. Huang, and J. Wang. Cfs: a behavioral similarity algorithm for process models based on complete firing sequences. In *OTM*, pages 202–219. Springer, 2014.

[8] J. Euzenat and P. Shvaiko. *Ontology matching*, volume 18. Springer, 2007.

[9] R. Hull, J. Su, and R. Vaculin. Data management perspectives on business process management: tutorial overview. In *Int. conf. on Management of Data (SIGMOD)*, pages 943–948. ACM, 2013.

[10] V. Künzle and M. Reichert. Philharmonicflows: towards a framework for object-aware process management. *Journal of Software Maintenance and Evolution: Research and Practice*, 23(4):205–244, 2011.

[11] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10-8, pages 707–710, 1966.

[12] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[13] A. Meyer, L. Pufahl, D. Fahland, and M. Weske. Modeling and enacting complex data dependencies in business processes. In *Business process management*, pages 171–186. Springer, 2013.

[14] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[15] E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.

[16] K. Ryndina, J. M. Kster, and H. Gall. Consistency of business process models and object life cycles. In *Int. conf. on Model Driven Engineering Languages and Systems*, pages 80–90. Springer, 2006.

[17] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[18] B. van Dongen, R. Dijkman, and J. Mendling. Measuring similarity between business process models. In *Int. conf. on Advanced Information Systems Engineering*, pages 450–464. Springer, 2008.

[19] M. Weske. *Business Process Management: Concepts, Languages, Architectures (2nd Ed.)*. Springer, 2012.