"The final publication is available at Springer via
https://doi.org/10.1007/978-3-319-48472-3."

Process Instance Similarity: Potentials, Metrics, Applications

Johannes Pflug and Stefanie Rinderle-Ma

University of Vienna, Austria,
Faculty of Computer Science
{johannes.pflug, stefanie.rinderle-ma}@univie.ac.at

Abstract. The analysis of process instance similarity offers valuable input for certain application fields including the evaluation of instance clusters, the identification of compliance abuses, and process optimization. In this paper, we discuss the topic of instance similarity in general: We show that similarity might be determined from different process perspectives such as control flow, time, and instance attributes. Each of these perspectives impose individual requirements on the similarity calculation concerning data and structure. Four metrics for process instance similarity are proposed covering different perspectives. The applicability and feasibility of the proposed metrics are evaluated based on a prototypical implementation and real-world process logs from the BPI challenges.

Keywords: Business process analysis, process instance similarity, similarity metrics

1 Introduction

Process model similarity has been intensively researched and various metrics for quantifying differences in the models have been defined (e.g., [8,6]). Process model similarity can support tasks such as process redesign or refactoring of process model repositories [8,21]. If two process models are determined as similar, the expected way of process instances that traverse these models might be similar as well [9]. Process model similarity is calculated during design time [22]. Process instance similarity, in contrast, covers the execution level of a business process as well; a process instance traverses the process model during runtime, hence building the temporal, data and resource aspects of the process execution. A set of properties including process instance attributes (e.g., the color or size of a print job) are associated to process instances [3]. Exploiting similarity of process instances bears the following potentials:

Identification of instance clusters: Some obvious classifications of instances are easy to identify. In a cooperative warehouse scenario where process instances represent customer orders, clusters of orders that contain a certain product, that have been placed in the same range of time or by customers from certain peer groups could be built. However, some coherences between instance attributes and resulting clusters might remain unknown. Knowledge about these kinds of

clusters can be valuable to the company's strategy and might be a competitive advantage.

Analysis of process execution: From a management point of view, basic numbers about process instances during and after process execution are typically considered. In the warehouse example from above, facts such as the total number of orders, their throughput time from order until delivery, and the total financial value would probably be of particular interest. Outliers could be considered as well. Analyzing the process execution of certain clusters of similar instances can provide further information about potentials and bottlenecks.

Compliance abuses: There are different ways of finding deviations from the defined process, including samples, human control, or the analysis of logs [23]. Analyzing the similarity of instances can provide additional information [6], for example, identifying clusters of instances that pass the same way through the process model, or have similar instance attributes or processing behavior. This can be advantageous in identifying compliance abuses early and providing a higher degree of transparency. Moreover, checks can become potentially more efficient as decisions can be based on instance clusters instead of deciding on each instance in a separated fashion (cf. checks on change correctness based on instance clusters described in [20]).

Resource optimization: Processing similar instances in a row is often more efficient than an instance order by chance (cf. batch processing as presented in [27]). The *Dynamic Instance Queuing* approach [26], e.g., is a way to optimize the resource behavior by classifying instances first and then processing similar items as a batch. In case studies from the medical [25] and industrial [26] domain, performance gains up to 19% in terms of throughput times, decrease in costs in an extend of 4% and a significant gain in quality of service could be achieved.

Based on the above reflections investigating process instance similarity seems to be promising. However, compared to process model similarity, not much effort has been spent on investigating the specificities of process instance similarity. A few techniques from the field of process mining exist (e.g., [24,32]) with a focus on trace comparison and activity similarity. In summary, analyzing instance similarity along the following two guiding research questions seems of interest: *RQ1: How can process instance similarity be defined?* and *RQ2: How to define meaningful metrics to measure process instance similarity?*.

In this paper, *RQ1* and *RQ2* are tackled by providing a first step towards the analysis, identification, quantification, and application of process instance similarity. We show potentials that arise out of the analysis of instance similarity and discuss views, measures, and techniques that form the basis for defining instance similarity ($\mapsto$ *RQ1*). Furthermore, we provide a first selection of metrics to assess the similarity between process instances ($\mapsto$ *RQ2*). The applicability and feasibility of the metrics are evaluated based on a proof-of-concept implementation as well as on a real-world dataset.

The paper is structured as follows: In Sect. 2.1, the identification of instance similarity from different process views is discussed. Section 3 provides four metrics that cover different aspects of process instance similarity. The applicability

and requirements for instance similarity identification are analyzed for five real-world datasets in Sect. 4. For a concrete real-world dataset from the health-care domain, we evaluate similarity scores for the process instances based on the described metrics, compute clusters of similar instances and evaluate the results against an algorithm from the *process mining* field. Results are discussed (Sect. 6) and put into research context (Sect. 7). Sect. 8 concludes with a summary.

2 Similarity views and identification

This section presents which views on process instance similarity can be identified.

2.1 Views on process instance similarity

Similarity of objects is a domain-specific property that heavily depends on the perspective of the viewer who perceives the environment [19]. In this section, we aim to identify these perspectives for *process instance similarity*. We approach this by analyzing the structure of a process instance. Based on [28], a process instance I is defined as follows:

Definition 1 (Process Instance, adapted from [28]). *A process instance I is defined as $I := (S, M^S, Val^S, EH, attr, attrV)$ where*

- *$S = (N, E, D)$ denotes the process schema the execution of I is based on. N denotes the set of tasks/activities, E the set of control and data edges, and D the set of process data elements. Note that we abstract from a concrete process meta model. Example meta models could be BPMN or EPCs.*
- *$M^S = (N^S, E^S)$ describes node markings N^S and edge markings E^S of I.*
  $N^S : N \mapsto$ {Activated, Completed, Skipped};
  $E^S : E \mapsto$ {Not, TRUE, FALSE}.
- *$Val^S : D \mapsto VAL \cup \{UNDEF\}$ reflects for each data element $d \in D$ either its current value $v \in VAL$ or value* UNDEF *(if d has not been written yet).*
- *$EH = \langle e_0, ..., e_k \rangle$ is the execution history of I. $e_0, ..., e_k$ denote events associated to starting / completing tasks in N. The events can be equipped with further information, e.g., on the data values read / written.*
- *$Attr$ denotes the set of instance attributes.*
- *$AttrV$ is a function on $attr$. It reflects for each attribute $a \in AttrV$ the value that is assigned to a when creating I.*

Based on Def. 1, the following views on instance similarity can be identified:

All objects that describe the process model S, including schema, nodes and edges are covered by the *model view* at design time. The *execution view*, in contrast, describes the execution perspective of a process which is represented by the node and edge markings M^S. In this context, instance similarity can mean a similar execution path through the process model - both in the past and future. Note that the execution history EH might contain the information of

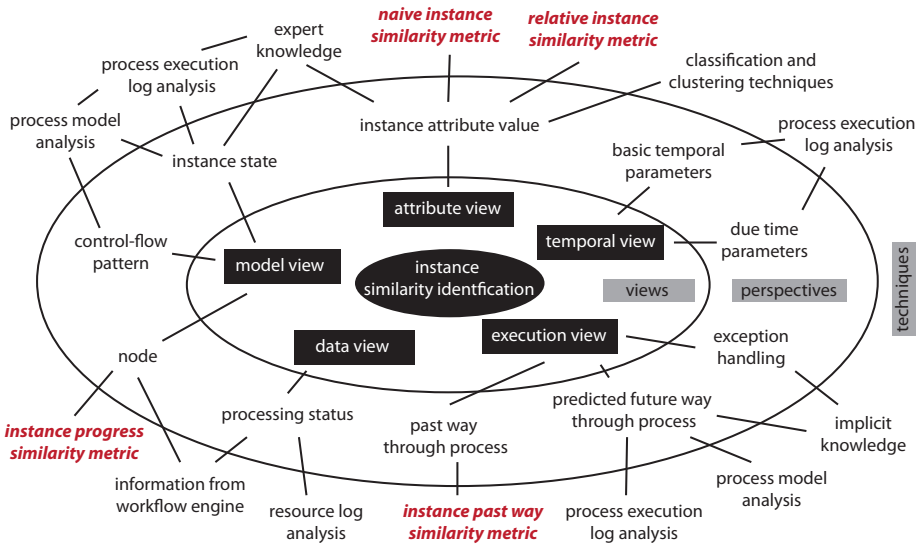

Fig. 1. Instance similarity identification

other views as well depending on which information in logged, e.g., resources and data values.

In the *data view*, elements D that are related to the process are represented. This can also include *resources*, which are central to the performance of the process execution. The execution of a process implies a *temporal* view which is logged in the execution history EH. Central aspects are the duration of the throughput time, processing time and waiting time as well as due times. Finally, instance attributes $Attr$ and its values $AttrV$ represent the *attribute* view, to which static properties of process instances are associated to.

2.2 Process instance similarity identification

For any of the five views described in Sect. 2, different techniques and means for similarity identification exist. The structure provided in the following covers both, techniques from existing literature, mainly from the domain of process mining as well as the proposed instance similarity metrics that are further described in Sect. 3. The decision which views are valuable to analyze strongly depends on the application scenario as well as on the data that is available. Fig. 1 shows the five similarity views in the inner circle, while the associated perspectives are shown in the middle circle. The techniques for the identification of clusters of similar instances are shown in the outer circle. The metrics in bold italic font are defined in Sect. 3.

Attribute view: Similarity on the attribute view means similarity of instance attributes and instance attribute values. Hence, clusters of instances with attributes that are considered as similar arise. Some clusters can be determined

by experts based on experience and an individual evaluation which attributes are valuable to consider. This technique typically focuses on numeric values, for which statistic evaluation techniques are easy to apply. Non-numeric values are usually being transformed into quantitative values by introducing codes. Two metrics operating on the attribute values of instances are suggested in Sect. 3.1. Artificial intelligence techniques, especially classification algorithms, are another promising possibility to identify clusters of instances without a priori knowledge (compare, for example, Dynamic Instance Queuing [25,26]).

Model view: The identification of the associations between instances and activities is typically provided by the workflow engine but can also be identified by analyzing the process log. In Sect. 3.2 we suggest a metric that evaluates the similarity of two instances based on their current position in the process model. Instances might be also identified by the connection to a certain control-flow pattern occurs as well. As control-flow patterns can be structured hierarchically, an instance can be associated to several patterns (cf., for example, tree edit distance for process model similarity [6]). To evaluate these relations, besides information from the process log the process model must be known as well. Finally, the state of instances can be considered in the process context: Instances might be dead-locked such that further processing is not possible, e.g. if dead-ends occur or some requirements for further processing are not fulfilled. The assessment of such error states often requires expert knowledge of the processing environment.

Execution view: The execution view describes the execution path of a process instance through the process model. The most obvious way to define similarity in terms of the execution view is to evaluate the previous execution path of a instance through the process model and compare it to the other instances' execution paths. Such techniques are known from the field of *process mining*, e.g., comparing the equivalence between processes based on observed behavior, i.e., by comparing traces [24]. Moreover, similarity of *activities* is provided by several existing techniques such as [33]. In Sect. 3.2 we suggest a metric that is able to evaluate a similarity score based on the execution path of two instances through the process model. Another approach would be to base the similarity analysis on the predicted future execution paths of process instances. Two factors seem to be promising for the prediction of the further execution paths; the past execution path through the process model and the instance data values which were evaluated by decision rules at gateways in the process.

Data view: During process execution, certain process-specific objects arise which are represented as data elements. We associate resources to the data view as well: Human and technical resources provide processing capability to activities. Instances associated to activities are first assigned to an appropriate resource by transferring it to a waiting queue and - when the resource offers available processing capability - the processing begins. Similarity in the resource context can be reflected by clusters of instances that have been processed by the same resource which can easily be analyzed by mining the resource log. At a certain time t, similarity might also represent the set of instances that share

the same processing status, i.e. *waiting in queue*, *processing as a single item* or *processing as a batch*.

Temporal view: Instances receive timestamps at certain events during the process execution. These timestamps include the trigger times of start events, the arrival times at gateways and activities, the processing start and end times of resources as well as the time when the instance has reached the end event. Out of these timestamps, basic temporal parameters can be evaluated, e.g. the average throughput and processing times at resources which constitute a basis for similarity analysis, i.e., by determining clusters of instances with similar throughput, waiting and processing times at certain resources as well as similar average durations. An interesting similarity approach arises when due times are involved. Different states can be determined for the instances at a certain time, e.g. *no deadline defined*, *deadline secure*, *deadline at risk*, *deadline kept* or *dead line not kept*. Having criteria defined for these states, clusters of instances with similar states can be evaluated by analyzing *xes*-standardized[1] process logs.

The described views on process instance similarity are not completely orthogonal, i.e. certain aspects from the views overlap. The *execution view*, e.g., covers aspects from the *model view* as well, since the execution of a workflow represents the instantiation of a process model: Process instances are associated to activities, which also occur as nodes in the model. Furthermore, views arise both from design time and runtime. Referring to the *attribute view*, some instance attribute values are predefined at design time, but during run time, values can either vary or be amended to the instance attribute set.

3 Metrics

For many applications, it is essential to compare process instances based on a numeric similarity score. The following metrics result in a value between 0 and 1, where 0 indicates no similarity and 1 indicates identical elements. This corresponds to the process model metrics described by [8]. The four proposed metrics cover three of the five views described in Sect. 2.1. They represent a first approach to quantify process instance similarity. Future work will cover more complex metrics for comparing instances.

3.1 Instance attribute metrics

The two metrics presented in the following exploit the set of instance attributes *Attr* and the corresponding values *AttrV* (cf. Def. 1).

Naive Instance Attribute Similarity Metric: Let I_1 and I_2 be two process instances with instance attribute sets $Attr_{I_1}$ and $Attr_{I_2}$ and corresponding values $AttrV_{I_1}$ and $AttrV_{I_2}$ respectively.

The *Naive Instance Attribute Similarity (NIAS)* of I_1 and I_2 is defined as follows:

[1] http://www.xes-standard.org/

$$NIAS(I_1, I_2) := \begin{cases} \frac{|AttrV_{I_1} \cap AttrV_{I_2}|}{max(|Attr_{I_1}|, |Attr_{I_2}|)}, & \text{if } max(|Attr_{I_1}|, |Attr_{I_2}|) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

For example, the *NIAS* between two instances I_1 and I_2, with

- $Attr_{I_1}$ = {"layout", "numberOfPages", "color"}
- $AttrV_{I_1}$ = {"portrait"; 8; "green"}
- $Attr_{I_2}$ = {"layout", "numberOfPages", "color", "margin"}
- $AttrV_{I_2}$ = {"landscape"; 8; "green"; 1.5}

turns out as $NIAS(I_1, I_2) = \frac{2}{4} = 0.5$.

The *Naive Instance Attribute Similarity Metric* focuses on identical attribute values and is therefore simple to evaluate. Values that are not identical are neglected, which is suitable especially for attributes with qualitative attributes (nominal scale). It lacks significance when attributes with a high variance of numeric values occur, as only same numbers will be counted. In this case, we recommend applying the relative instance attribute similarity metric as described in the following.

Relative Instance Attribute Similarity Metric: Let I_1 and I_2 be two process instances. Furthermore, let $Attr_{I_1}$ ($Attr_{I_2}$) be the attribute set of process instance I_1 (I_2). Assume that $|Attr_{I_1}| = |Attr_{I_2}|$ holds or dummy values are used to meet this requirement. Let further $AttrV_{I_1}^Q := \{v \in AttrV_{I_1} \mid \text{v is numeric}\}$ ($AttrV_{I_2}^Q := \{v \in AttrV_{I_2} \mid \text{v is numeric}\}$) denote the numeric values of $AttrV_{I_1}$ ($AttrV_{I_2}$). Non-numeric values are either to be left out or mapped onto a numeric values (e.g., by using scales). The *Relative Instance Attribute Similarity (RIAS)* of I_1 and I_2 is defined as follows, based on the Pearson product-moment correlation coefficient:

$$RIAS(I_1, I_2) := \begin{cases} \left| \frac{COV(AttrV_{I_1}^Q, AttrV_{I_2}^Q)}{\sigma_{AttrV_{I_1}^Q} \cdot \sigma_{AttrV_{I_2}^Q}} \right|, & \text{if } \sigma_{AttrV_{I_1}^Q}, \sigma_{AttrV_{I_2}^Q} \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

As an example for the evaluation of *RIAS*, we consider two instances I_1 and I_2, with

- $Attr_{I_1} = Attr_{I_2}$ = {"numberOfPages"; "pixel_height"; "pixel_width"}
- $AttrV_{I_1}$ = {8; 800; 300} and $AttrV_{I_2}$ = {5; 5000; 100}

This results in a *RIAS* of 0.93.

The *Relative Instance Attribute Similarity Metric* incorporates all numeric attributes for the evaluation of the similarity by measuring the degree of linear dependence between the two value sets of the instances. Therefore it describes the similarity on a more comprehensive basis than *NIAS*. The *RIAS* is not robust, so its value can be misleading if outliers are present.

3.2 Instance progress metrics

The two metrics presented in the following are based on the execution state of process instances.

Instance Progress Similarity Metric: For an instance $I = (S, \ldots)$ with $S = (N, \ldots)$, let Act(I, t):= { n ∈ N | $N^S(n)$ = ACTIVATED at time t} denote the set of nodes that are activated at time t. Let further $N_{START}(I,t) := \{n \in N \mid$ n is on the shortest path from the start node of I to one of the nodes in Act(I, t)$\}$ denote the set of nodes located on one of the shortest paths from the start node to the currently activated nodes. Moreover, let $N_{END}(I,t)$ be the set of all nodes on a shortest path from a node in Act(I, t) to the end node in S_I[2].

Let $I_1 = (S_{I_1}, \ldots)$ and $I_2 = (S_{I_2}, \ldots)$ be two process instances. The *Instance Progress Similarity (IPS)* of I_1 and I_2 at time t is defined as follows:

$$\mathsf{IPS}(I_1, I_2, t) := 1 - \left| \frac{|N_{START}(I_1,t)|}{|N_{START}(I_1,t)| + |N_{END}(I_1,t)|} - \frac{|N_{START}(I_2,t)|}{|N_{START}(I_2,t)| + |N_{END}(I_2,t)|} \right| \tag{3}$$

Consider as an example two instances I_1 and I_2 running on the same schema S that constitutes a sequence of eight activities (nodes), a start node, and an end node. At time t, let the second activity be activated for I_1 and the sixth activity for I_2. This means $N_{START}(I_1,t) = 1$, $N_{END}(I_1,t) = 6$ for instance I_1 and $N_{START}(I_2,t) = 5$, $N_{END}(I_2,t) = 2$ for instance I_2. The IPS of I_1 and I_2 at time t therefore is $1 - |\frac{1}{7} - \frac{5}{7}| = 1 - \frac{4}{7} = \frac{3}{7} = 0.43$.

Instance Past Way Similarity Metric: Let $I_1 = (S_{I_1}, \ldots)$ and $I_2 = (S_{I_2}, \ldots)$ be two process instances. Furthermore, let
$N_{PAST}(I_1,t)$:= {n ∈ S_{I_1} | $N^S(n)$ = COMPLETED at time t }
$N_{PAST}(I_2,t)$:= {n ∈ S_{I_2} | $N^S(n)$ = COMPLETED at time t } be the set of nodes I_1 (I_2) has passed until time t. $N_{PAST}(I_1,t) \cap N_{PAST}(I_2,t)$ then describes the set of nodes that both I_1 and I_2 have passed and $N_{PAST}(I_1,t) \cup N_{PAST}(I_2,t)$ the set of nodes that at least one of the instances has passed. We define the similarity between two instances' *past way through the process (IPW)* at time t by

$$\mathsf{IPW}(I_1, I_2, t) := \begin{cases} \frac{|N_{PAST}(I_1,t) \cap N_{PAST}(I_2,t)|}{|N_{PAST}(I_1,t) \cup N_{PAST}(I_2,t)|}, & \text{if } |N_{PAST}(I_1,t) \cup N_{PAST}(I_2,t)| > 0 \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

For example, two instances I_1 and I_2 with $N_{PAST}(I_1,t)$ ={A, B, D, F, G, H} and $N_{PAST}(I_2,t)$ = {A, B, C, E, G, H} would have a similarity score of 0.5, as $N_{PAST}(I_1,t) \cap N_{PAST}(I_2,t)$ = {A, B, G, H} and $N_{PAST}(I_1,t) \cup N_{PAST}(I_2,t)$ = {A, B, C, D, E, F, G, H}.

[2] If several end nodes are allowed by the respective meta model, the closest one is selected.

4 General Applicability

The identification of clusters of similar instances on the different views and perspectives as described in Sect. 2.2 requires the availability of different data about the process execution and the process instances. Although a basic process execution log probably exists for any process execution, the different similarity views require certain specific information: For the *attribute view*, instance attributes and instance attribute values need to be available. For a more sophisticated analysis, the instance attribute values need to be logged not only once at the beginning or the end of the process execution, but for each activity. This way, the dynamic character of the attribute values can be incorporated in the instance analysis. From the *model view* and *execution view*, data about the association of instances to activities at any time needs to be available. This is the case for most traces, as at least timestamps about the arrival times at activities are logged. If the process model is available, associations to control-flow patterns can be evaluated as well. More sophisticated investigations require expert knowledge, as even the combination of execution log and process model covers only explicit knowledge and lacks patterns that are implicit. The *temporal view*, however, requires more information from the execution log than for the model and execution view: To determine the waiting time and processing time at activities and resources, both the arrival time at the node of the process model and the processing start need to be logged. In a *xes*-standardized log, this is represented by the *lifecycle extension* tag [16]. Probably the most difficult part in similarity analysis is the *data view* as the resource behavior has mostly been neglected by research in favor of the process perspective [29]. To assess the similarity between instances based on their processing behavior, a resource log is required that includes information about the processes instances as well as relevant timestamps and environment data for each resource.

We investigated logs from five real-world process executions to determine which of the similarity views and perspectives can be identified by the given data. The processes cover the building permit application from several Dutch municipalities [13], the service desk and IT operations process from the Rabobank Group ICT [12], and the incident and problem management from Volvo IT Belgium [30]. Furthermore, we investigated real-world data about loan/overdraft applications of customers from another Dutch financial institute [11] and about the Gynecology department treatment process from a Dutch Academic Hospital [10].

The analysis has been performed by analyzing the logs both by hand and with tool support to evaluate mathematical indicators. The results are shown in Table 1, where "++" means that sufficient data is available to apply the respective instance similarity measure without any limitation and "+" means that the perspective can be applied, but assumptions have to be made, e.g. concerning timestamps in the process history log: In an ideal scenario (labeled with "++"), for any instance at any activity, the arrival time, the processing start time as well as the processing end time are individually known. However, the arrival timestamp is often missing. One could assume that the arrival time

at an activity equals the processing end time of the preceding activity. This scenario would be labeled as *applicable with assumptions* ("+"). "-" means that no data is available to assess the corresponding instance similarity measure.

Table 1. Applicability of instance similarity views in certain real-world scenarios (++ applicable without limitation; + applicable with assumptions; - not applicable)

Similarity category/ Similarity perspective	BPI 2015 Municipal	BPI 2014 Rabobank	BPI 2013 Volvo	BPI 2012 Financial	BPI 2011 Hospital
Attribute view					
Non-numeric attr. value similarity	++	-	-	-	++
Numeric attr. value similarity	++	++	-	+	++
Model view					
Node	++	++	++	++	++
Control-flow pattern	+	+	-	++	-
Instance state	+	+	+	+	+
Data view					
Processing status (waiting)	++	-	++	++	-
Processing status (processing)	-	-	++	++	-
Execution view					
Past way through process	++	++	+	++	++
Predicted future way	++	++	+	++	++
Exception handling	-	+	-	-	-
Temporal view					
Basic temp. parameters (tt)	+	+	++	++	+
Basic temp. parameters (pt)	-	-	++	++	-
Due time parameters	++	-	-	-	-

The analysis of the five real world scenarios shows that a broad variety of structurally different datasets fulfill the requirements for instance similarity identification, so that at least some of the views described in Sect. 2 can be analyzed. The *model* and *execution view* are supported best, which is due to the reason that information about passed activities are part of basically any process trace log. Furthermore, a suitable amount of instance attributes is provided by most of the datasets. The *data* and *temporal view* incorporate the highest requirements in terms of scope of the process traces.

5 Application in a real-world health-care scenario

In the following, the feasibility of the similarity metrics is investigated based on a proof-of-concept implementation, the application to the real-world hospital log [10], and in comparison to another similarity metric based on this log ("gold standard").

5.1 Computation of similarity scores

We applied the described metrics in a real-world dataset derived from a Dutch academic hospital [10]. The *xes*-based logfile contains a complete set of instances (i.e., patients), certain instance attributes such as diagnosis, treatment, and age of the patient as well as process traces. To reduce noise and infrequent behavior from the process event log, we filtered the dataset using a simple heuristics filter. The result was a dataset containing 430 instances and a total of appr. 600 different events. For any pair of the process instances, we evaluated the similarity score based on the four metrics, which results in four symmetric matrices of a length and width of 430 entries containing 184900 similarity scores.

For the instance attribute based metrics $NIAS$ and $RIAS$ (cf. Sect. 3.1), 15 instance attributes were considered. Most of these attributes represent *codes*, e.g. a *treatment id*. However, not only identical codes represent similarity, as adjacent codes belong to groups of codes with similar treatments. This means similar codes represent similar treatments, hence process instances might be similar even if codes are not identical. For this reason, the application of both $NIAS$ (which refers to *identical* attribute values) and $RIAS$ (which refers to *similar* attribute values) is promising. For the model and execution based metrics IPS and IPW (cf. Sect. 3.2), the process traces of the instances are analyzed. In total, the dataset includes 21198 different process traces with an average of 49 traces per process instance. IPW is expected to evaluate lower similarity scores, as identical process traces are expected to be less probable in a setting with such a large number of traces per instance. For IPS the number of traces per instance is irrelevant as it represents a process-based metric, where the evaluation of the score depends on the position in the process model.

Table 2 shows some basic parameters from the similarity matrices for any of the four similarity metrics. As described in Sect. 3, the metrics result in a similarity score between 0 and 1, where 0 indicates no similarity and 1 indicates identical elements. The results show that all metrics are quite robust, meaning the evaluated scores distribute over a large scale from non similar (~ 0) to very similar (~ 1). However, while the instance attribute based metrics $NIAS$ and $RIAS$ evaluate large numbers of totally unequal instances with a similarity score of 0, the model and execution based metrics IPW and IPS evaluate only little numbers of those unequal instances. The reason is that the variety of possible outcomes of the instance attribute values might be wide spread, while the range of possibilities to traverse a process model with 600 events seems to be limited. Any of the metrics except for $NIAS$ evaluate very equal instances with a similarity score of 1.0. $NIAS$ assigns the score of 1.0 only to the 430 scores where identical instances are compared. This is due to the fact that $NIAS$ identifies similarity by *identical* instance attribute values, while $RIAS$ includes *similar* values as well. For all metrics, the average similarity score and the median are relatively similar, which means the scores are resilient to extremely large and small values. This is also the case for the *filtered* average and median score, where scores with a value of 0 and 1 were excluded.

The basic parameters indicate an adequate level of homogeneity for any of the four matrices evaluated based on the different metrics. The evaluated scores allow to compare process instances in terms of their similarity for the scope of the respective metric. However, comparability between different metrics is limited so far. In future work, a *meta*-metric could be developed that allows to define inter-perspective similarity scores.

Qualitatively, the results are interesting as well. The parameters captured by the information systems of the hospital indicate that around one third of the patients have a personal and medical background that does not allow to reference them to other patients in any way. Another quarter of patients has a very similar background, i.e. diagnosis, treatment and other attributes are very similar to each other. For the rest of the patients, only certain attributes are similar. In regards of process, it becomes obvious that almost half of all patients share a similar history for this hospital, meaning that the treatments operated by the hospital have been equal. This fits to the observation that around 20% of all patients are expected to be in the same stadium of treatment, around 30% more patients share at least a similar position in the overall treatment process.

Table 2. Basic parameters about the similarity score matrices for data set [10])

	NIAS	RIAS	IPW	IPS
Average similarity score	0.14	0.65	0.54	0.93
Median	0.00	0.98	0.50	0.95
Average filtered similarity score	0.33	0.98	0.39	0.93
Filtered median	0.33	1.00	0.22	0.95
No. of identical instance scores (score = 1.0)	430	67582	45306	36126
No. of unsimilar instances scores (score = 0.0)	104738	62730	150	212
Min value (instances with score = 0 excluded)	0.08	0.53	0.08	0.00
Max value (only scores with different instances included)	0.89	1.00	0.92	1.00

5.2 Identification of similarity clusters

The similarity scores evaluated in Sect. 5.1 enable the comparison of process instances in terms of similarity in the context of the different similarity views. For further analysis such as the identification of patterns (cf. Sect. 1), it is interesting to evaluate clusters of similar process instances based on the similarity scores. For this purpose, we applied the *clustering for large applications (clara)* algorithm [18]. *Clara* supports distance matrices, i.e. it is able to identify groups based on similarity scores.

For $NIAS$ and $RIAS$, the separation of the 430 process instances into three clusters is promising due to the silhouette width of the clusters, for the remaining metrics four clusters are evaluated. The best result is evaluated for $RIAS$ with a silhouette width for the three clusters of 0.97, which means the clusters are very well separated. Fig. 2 shows a plot for the cluster analysis for $RIAS$ for

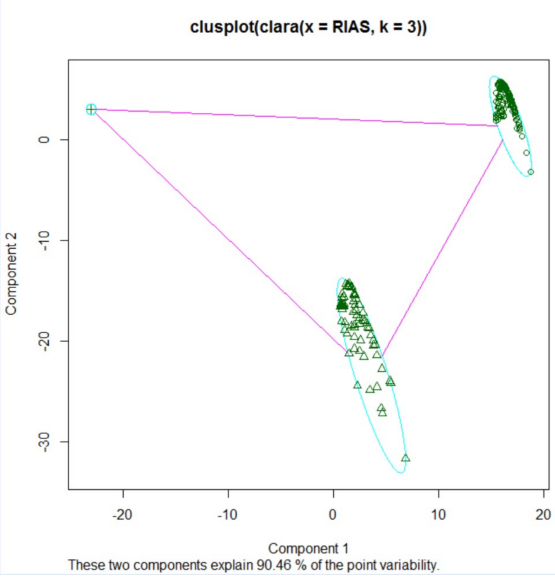

Fig. 2. Cluster plot for $RIAS$

the principal components, which already explain most of the point variability. The clusters are homogeneously distributed over the three clusters (205/72/153 instances for cluster 1/2/3). Process instances with a very low similarity score as well as process instances with a similarity score of 0 are represented by the top left cluster from Fig. 2.

The clusters derived from the $NIAS$ metric cannot be separated by the *clara* algorithm as well as the ones from the $RIAS$ metric (average silhouette width of 0.54) which is due to the fact that the bandwidth of scores is higher. The average silhouette widths of IPW (0.88) and IPS (0.74) indicate a robust result as well.

5.3 Evaluation

We evaluated the results derived from any of the four metrics. We used a different approach for the instance attribute based metrics $NIAS$ and $RIAS$ and for the model and execution based metrics IPW and IPS: To the best of our knowledge, no technique for evaluating instance attribute based similarity exists at the moment. However, in this particular dataset, the attribute *org:group* represents a parameter for instance similarity, as patients treated in the same group of the hospital are expected to have a similar medical background. To prevent side effects, this parameter has been excluded from the similarity evaluation by the two metrics before. For IPW and IPS, we verified the result from the metrics against the activity clusters derived from the *discover clusters* algorithm by Verbeek [33]. This algorithm evaluates activity clusters based on the pro-

cess traces, which we compared to the current activity of the process instances clustered by the similarity metrics.

Table 3 shows the results of the evaluation. For the instance attribute based metrics, half of the patients for a certain group were assigned correctly. Over 80% of the patients evaluated to be in a certain group of the hospital were correctly identified, which shows that the recognition works stable both for $NIAS$ and $RIAS$. For IPW and IPS, the precision is slightly higher, but the corresponding recall drops. However these results need to be put in perspective: The *discover clusters* algorithm targets at identifying clusters of similar *activities*, while the metrics described in this section aim at identifying clusters of similar *instances*. Although these two targets are correlated to each other and represent a resilient mean for verification, we expect different outcomes in a certain extent.

Table 3. Evaluation results

	Precision	Recall
Naive instance attribute similarity (NIAS)	0.56	0.86
Relative instance attribute similarity (RIAS)	0.52	0.84
Instance past way similarity (IPW)	0.60	0.75
Instance progress similarity (IPS)	0.60	0.49

5.4 Implementation

For the computation of the similarity scores of the four metrics as well as for the evaluation of the results, various implementations and techniques have been applied. The initialization of the *xes*-based dataset from the health-care domain has been applied by an existing workflow engine [26] that strongly orients on the generic structure of a classic workflow engine [4,5]. This engine is implemented in *JAVA* and allows to initialize *xes*-based instance files, *BPMN 2.0* process models and additional resource information. To evaluate the similarity scores, we developed plugin processors for any of the four metrics. This allowed us to compute a symmetric matrix of similarity scores that covers any combination of two process instances.

The analysis of instance clusters was evaluated using the *clara* clustering algorithm. We used the implementation of the package *cluster*, part of the R[3] software environment for statistical computing and graphics. Besides the plots shown in this work, R also provided a mapping of instances on cluster ids.

The evaluation of the model and execution based metrics IPW and IPS was applied against the *discover clusters* algorithm provided by Verbeek [33]. To compute the clusters of similar activities, the process mining framework *ProM*[4] was used. *ProM* was also applied to reduce the noise in the input logfile. Techniques

[3] www.r-project.org
[4] http://www.promtools.org

for the evaluation of the metrics were implemented as another plugin for the workflow engine described above.

6 Discussion

This work represents a first step towards the investigation of process instance similarity. Hence it has some limitations: The metrics provided in this work cover only three of the five similarity views (cf. Sect. 2). At the moment, they do not allow to evaluate an integrated similarity score for several views. The proposed metrics cover different fields of application: $NIAS$ and $RIAS$ should be applied when similarity shall be evaluated based on instance attributes. $NIAS$ is more strict than $RIAS$, hence being suitable for scenarios in which the relevant attributes have a nominal scale or if similarity is referred to as identical attribute values. $RIAS$, in contrast, evaluates scores based on the relative similarity of attribute values, which makes it suitable for most scenarios with numeric attribute values. IPS evaluates similarity scores based on the position of the instances in the process model, while IPW uses the instances' processing history.

The metrics described in this work are kept simple, however more complex similarity metrics can be defined that incorporate additional parameters of the process environment. Our approach therefore leaves room for further work, both in the field of identification and application of process instance similarity:

Fields of application: Process instance similarity evaluation can be beneficial to a lot of fields in business process and workflow systems research (cf. Sect. 1): Business process analysis, process mining, process redesign, compliance insurance or processing strategy optimization. Instance similarity is already implemented as part of the *Dynamic Instance Queuing*, a lightweight approach for the optimization of the business process execution during runtime that has proven to be effective in several case studies [25,26]. We argue that similarity analysis of instances offers great potential for process optimization. For process execution analysis, future work needs to cope mechanisms to automatically analyze process execution logs and present the results.

Cooperative scenarios: In this work, we evaluated the similarity of instances from the same process model. However, the techniques for instance similarity identification can also be applied to instances from different processes. This is especially promising for cooperative scenarios, where instances from various process models interact with each other. One could even include existing techniques from the similarity evaluation of process models into process instance similarity metrics and vice versa.

Integration with process mining techniques: Instance similarity and process mining techniques share the fact that event logs represent a mean for their analysis. Existing algorithms from the process mining area [14,17,15] cover the identification of activity clusters. The presented IPW metric relates with behavioral similarity measures such as [24]. More complex metrics will be implemented that might be complementary to existing process mining technique, hence further the analysis capabilities for process execution data.

7 Related Work

Understanding how objects are partitioned into useful groups to form concepts is important to most disciplines [19]. Recent literature has covered the analysis of similar objects concerning process models and process activities. However, the topic of analyzing process instance similarity has not yet been discussed in general yet. The topic has been striven by certain *process mining* techniques; grouping similar items also represents an explicit or implicit task that is fulfilled by human experts during process execution. Furthermore, the association of instances to clusters with certain (shared) properties is featured as part of the execution log analysis by some workflow engines and process mining tools. All these scenarios share the fact that instance similarity is applied both *non-formalized* and *implicitly*. In this work, we argue a formal background for instance similarity analysis is required, including a common definition and provide a general view on the topic. The following research topics are related to the task of identifying similar instances in a process:

Process model similarity: Process model similarity calculation is hindered by multiple inherent sources of heterogeneity [9]. Even if two process models define exactly the same behavior at the same level of granularity and with the same projection on the real-world process, the process models might still look quite different [9]. The identification of similar models requires indexing techniques. The indexing task is mostly based on metrics, which, e.g., evaluate label similarity, process traces, subsequent tasks or process patterns [8]. MTree [34] and B+ tree [31] techniques are being applied as well. Process model similarity supports tasks including process model redesign and refactoring of process model repositories [21].

Process and activity mining techniques: Process mining is about the extraction of information about processes from transaction logs recorded by information systems [1]. Discovery algorithms produce process models out of the input event log, typically without any further information. For conformance checking, the reference process model needs to be available. With conformance checking algorithms, it can be verified if certain events fit the process model and vice versa. The third type of process mining is enhancement. Here, the idea is to extend or improve an existing process model using information about the actual process recorded in some event log [2]. *Activity mining* is a specialized discipline in the area of the discovery techniques. Activity mining aims to discover impact-targeted activity patterns in huge volumes of unbalanced activity transactions [7]. Techniques for the identification of similar activities (e.g. [33]) are associated to the area of activity mining as well. Behavioral equivalence of processes based on trace comparison has been investigated as well, e.g., [24]. This can be estimated as related to the instance progress metrics suggested above. However, to the best of our knowledge, no further analysis of instance similarity exists in the area of process mining.

8 Summary

The identification of similar process instances offers many potentials, including means for process analysis and optimization as well as the evaluation of instance clusters and compliance abuses. However, while similarity of process models and process traces (in the context of process mining) have been intensively investigated in existing literature, process instance similarity has drawn little attention so far. In this work, we explore the fundamentals of process instance similarity, including first steps for formalization: We show that process instances are not similar *in general*, but they are similar concerning certain *perspectives*: The model view, execution view, attribute view, data view and temporal view. For each of these perspectives, different measures and techniques for instance similarity identification exist. We implemented four metrics to identify similar instances. These metrics allow to evaluate a numeric similarity score that makes it possible to compare process instances based on their similarity characteristics. Besides an analysis of the general applicability, these metrics have been applied in a real-world dataset from the health-care domain and clusters of similar instances have been computed. The results were evaluated against techniques from the process mining area.

Future work will cope with more complex metrics and cover the temporal and data view as well. So far, similarity has been evaluated for each view separately. It would be desirable to develop a metric that integrates several views. Especially an integration of the execution, data and temporal view could be promising to identify bottlenecks in business processes.

Acknowledgments

This work has been funded by the Vienna Science and Technology Fund (WWTF) through project ICT15-072.

References

1. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: A survey of issues and approaches. Data Knowl. Eng. 47(2), 237–267 (Nov 2003), http://dx.doi.org/10.1016/S0169-023X(03)00066-1
2. van der Aalst, W., et al.: Process Mining Manifesto, pp. 169–194. Springer Berlin Heidelberg, Berlin, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-28108-2_19
3. van der Aalst, W.M.P., van Hee, K.M.: Workflow Management: Models, Methods, and Systems. MIT Press (2002)
4. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer, Berlin (2004)
5. Anderson, K.: Web services and related technologies (2006)
6. Becker, M., Laue, R.: A comparative survey of business process similarity measures. Computers in Industry 63(2), 148–167 (2012), http://dx.doi.org/10.1016/j.compind.2011.11.003

7. Cao, L.: Activity Mining: Challenges and Prospects, pp. 582–593. Springer Berlin Heidelberg, Berlin, Heidelberg (2006), http://dx.doi.org/10.1007/11811305_65
8. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (April 2011), http://dx.doi.org/10.1016/j.is.2010.09.006
9. Dijkman, R.M.e.a.: Seminal Contributions to Information Systems Engineering: 25 Years of CAiSE, chap. A Short Survey on Process Model Similarity, pp. 421–427. Springer Berlin Heidelberg, Berlin, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-36926-1_34
10. van Dongen, B.: Real-life event logs - hospital log (2011), http://dx.doi.org/10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54
11. van Dongen, B.: Bpi challenge 2012 (2012), http://dx.doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
12. van Dongen, B.: Bpi challenge 2014 (2014), http://dx.doi.org/10.4121/uuid:c3e5d162-0cfd-4bb0-bd82-af5268819c35
13. van Dongen, B.: Bpi challenge 2015 (2015), http://dx.doi.org/10.4121/uuid:31a308ef-c844-48da-948c-305d167a0ec1
14. Guenther, C.W.: Mining activity clusters from low-level event logs. In: Eindhoven University of Technology (2006)
15. Guenther, C.W., Rozinat, A., van der Aalst, W.M.P.: Activity Mining by Global Trace Segmentation, pp. 128–139. Springer Berlin Heidelberg, Berlin, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-12186-9_13
16. Guenther, C.W., Verbeek, E.: Xes standard definition. Tech. Rep. 2.0, Eindhoven University of Technology (March 2014)
17. Hompes, B.F.A., Verbeek, H.M.W., van der Aalst, W.M.P.: Finding Suitable Activity Clusters for Decomposed Process Discovery, pp. 32–57. Springer International Publishing, Cham (2015), http://dx.doi.org/10.1007/978-3-319-27243-6_2
18. Kaufman, L., Rousseeuw, P.J.: Clustering Large Applications (Program CLARA), pp. 126–163. John Wiley & Sons, Inc. (2008), http://dx.doi.org/10.1002/9780470316801.ch3
19. Keane, M.T., Smyth, B., O'Sullivan, J.: Similarity and Categorization, chap. Dynamic similarity: a processing perspective on similarity. Oxford University Press, Oxford (2001)
20. Kreher, U., Reichert, M., Rinderle-Ma, S., Dadam, P.: Effiziente repraesentation von vorlagen- und instanzdaten in prozess-management-systemen. Tech. Rep. 2009-08, Ulm University (2009), (in German)
21. La Rosa, M., Dumas, M., Ekanayake, C.C., García-Bañuelos, L., Recker, J., ter Hofstede, A.H.M.: Detecting approximate clones in business process model repositories. Inf. Syst. 49, 102–125 (2015), http://dx.doi.org/10.1016/j.is.2014.11.010
22. Lu, R., Sadiq, S.: On the Discovery of Preferred Work Practice Through Business ProcessVariants, pp. 165–180. Springer Berlin Heidelberg, Berlin, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-75563-0_13
23. Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: Compliance monitoring in business processes: Functionalities, application,and tool-support. Inf. Syst. 54, 209–234 (2015)
24. de Medeiros, A.K.A., van der Aalst, W.M.P., Weijters, A.J.M.M.: Quantifying process equivalence based on observed behavior. Data Knowl. Eng. 64(1), 55–74 (2008), http://dx.doi.org/10.1016/j.datak.2007.06.010

25. Pflug, J., Rinderle-Ma, S.: Dynamic instance queuing in process-aware information systems. In: Proc. 28th Annual ACM Symposium on Applied Computing (SAC '13). pp. 1426–1433 (2013)
26. Pflug, J., Rinderle-Ma, S.: Application of dynamic instance queuing to activity sequences incooperative business process scenarios. International Journal of Cooperative Information Systems 25(1), 1650002 (2016)
27. Pufahl, L., Bazhenova, E., Weske, M.: Evaluating the performance of a batch activity in process models. In: Business Process Management Workshops - BPM 2014 InternationalWorkshops, Eindhoven, The Netherlands, September 7-8, 2014. pp. 277–290 (2014)
28. Rinderle, S., Reichert, M., Dadam, P.: Flexible support of team processes by adaptive workflow systems. Distributed and Parallel Databases 16(1), 91–116 (2004), http://dx.doi.org/10.1023/B:DAPD.0000026270.78463.77
29. Russell, N., van der Aalst, W., ter Hofstede, A., Edmond, D.: CAiSE 2005, Porto, Portugal, June 13-17, 2005., chap. Workflow Resource Patterns: Identification, Representation and Tool Support, pp. 216–232. Springer Berlin Heidelberg, Berlin, Heidelberg (2005), http://dx.doi.org/10.1007/11431855_16
30. Steeman, W.: Bpi challenge 2013 (2013), http://dx.doi.org/10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07
31. Tao, J., Wang, J., Nianhua, W., La Rosa, M., ter Hofstede, A.H.M.: Efficient and accurate retrieval of business process models through indexing. Lecture Notes in Computer Science [On the Move to Meaningful Internet Systems: OTM 2010] 6426, 402–409 (March 2010), http://eprints.qut.edu.au/31996/
32. Thaler, T., Ternis, S.F., Fettke, P., Loos, P.: A comparative analysis of process instance cluster techniques. In: Thomas, O., Teuteberg, F. (eds.) Wirtschaftsinformatik. pp. 423–437 (2015), http://dblp.uni-trier.de/db/conf/wirtschaftsinformatik/wi2015.html#ThalerTFL15
33. Verbeek, H.M.W., van der Aalst, W.M.P.: Decomposed process mining: The ILP case. In: Business Process Management Workshops - BPM 2014 International-Workshops, Eindhoven, The Netherlands, September 7-8, 2014, RevisedPapers. pp. 264–276 (2014), http://dx.doi.org/10.1007/978-3-319-15895-2_23
34. Yan, Z., Dijkman, R., Grefen, P.: Fast business process similarity search. Distributed and Parallel Databases 30(2), 105–144 (2012), http://dx.doi.org/10.1007/s10619-012-7089-z