2016 IEEE 18th Conference on Business Informatics

A Cross-Organizational Process Mining Framework for Obtaining Insights from Software Products: Accurate Comparison Challenges

Ü. Aksu*, D. M. M. Schunselaar*, H. A. Reijers*
*Vrije Universiteit Amsterdam, The Netherlands
Email: {u.a.aksu, d.m.m.schunselaar, h.a.reijers}@vu.nl

***Abstract*— Software vendors offer various software products to large numbers of *enterprises* to support their organization, in particular Enterprise Resource Planning (ERP) software. Each of these enterprises use the same product for similar goals, albeit with different processes and configurations. Therefore, software vendors want to obtain insights into how the enterprises use the software product, what the differences are in usage between enterprises, and the reasons behind these differences. Cross-organizational process mining is a possible solution to address these needs, as it aims at comparing enterprises based on their usage.**

In this paper, we present a novel Cross-Organizational Process Mining Framework which takes as input, besides event log, semantics (meaning of terms in an enterprise) and organizational context (characteristics of an enterprise). The framework provides reasoning capabilities to determine what to compare and how. Besides, the framework enables one to create a catalog of metrics by deducing diagnostics from the usage. By using this catalog, the framework can monitor the (positive) effects of changes on processes. An enterprise operating in a similar context might also benefit from the same changes. To accommodate these *improvement suggestions*, the framework creates an improvement catalog of observed changes. Later, we provide a set of challenges which have to be met in order to obtain the inputs from current products to show the feasibility of the framework. Next to this, we provide preliminary results showing they can be met and illustrate an example application of the framework in cooperation with an ERP software vendor.

***Index Terms*—Cross-Organizational Process Mining, Framework, ERP**

I. Introduction

Software vendors offer various enterprise software products (hereinafter product) to large numbers of *enterprises* (hereinafter organizations), to support them in their business processes. For instance, organizations use Enterprise Resource Planning (ERP) software for order management, delivery, invoicing, stock management, keeping customer records, and human resource management. However, each organization uses the same product for similar goals but different settings. Furthermore, organizations use the same terms and functions in different ways in the same product. For example, the term *loan* has different semantics within a bank compared to a library. In accordance with this information, cross-organizational process mining is a possible solution which aims at comparing multiple organizations using process mining.

Cross-organizational process mining enables one to compare organizations from different perspectives, e.g., the ordering of activities or subdivision of work amongst resources. Current cross-organizational mining approaches ([1],[2],[3]) mainly use event logs which are collection of events as the sole input. At the same time, the current approaches primarily use text matching techniques to relate terms between organizations to find out differences in software product usage. This might hinder accurate comparisons. For instance, in the aforementioned example, the current cross-organizational process mining approaches match the term *loan* inside a bank to the term *loan* inside a library. However, the semantics of the term *loan* is different for a bank and a library. The approaches do not take into account these semantics. Making comparison without semantics might generate inadequate and ambiguous insights which may trigger irrelevant improvement actions and changes. Eventually, these irrelevant improvement actions and changes may cause unexpected results. Next to the semantics, the approaches do not consider the organizational context, e.g., the term *loan* might bring different regulations or preconditions for a bank than for a library.

In order to do a more accurate comparison between organizations, we present a novel Cross-Organizational Process Mining Framework which takes semantics and organizational context, in addition to event logs, as inputs. With these inputs, the framework provides reasoning capabilities which determine what to compare and how. For instance, with the semantics and the organizational context, the framework can determine that the term *loan* has a different semantic for a bank than for a library. By using this semantic difference, the framework can determine that the loan process in the bank is not comparable to the loan process in the library. Moreover, by having clear semantics and the organizational context, we can monitor changes in the processes and the (positive) effects they may bring within a given context. An organization operating in a similar context might also benefit from the same changes. In order to accommodate these *improvement suggestions*, the framework will create a catalog of observed improvements.

Organizations may be sensitive in case of using their data explicitly for comparison. There may be security regulations across the organizations which may complicate the comparison. Although the privacy is an important concern, there are various approache [4] to ensure the anonymity of the inputs

This is an AMUSE paper. See amuse-project.org for more information.

2378-1971/16 $31.00 © 2016 IEEE
DOI 10.1109/CBI.2016.25

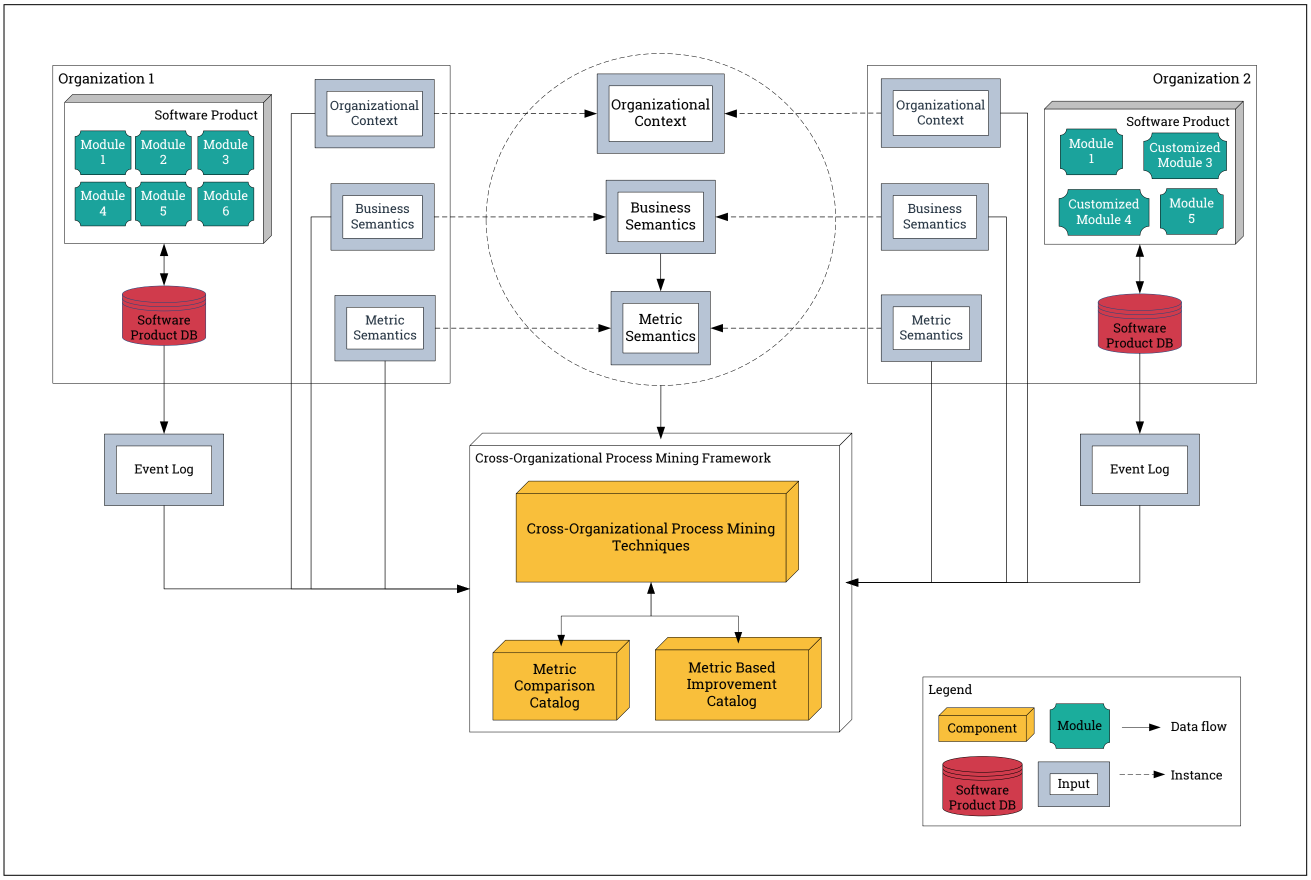

Figure 1. Architecture of the Cross-Organizational Process Mining Framework

and the results considering the privacy related concerns. Next to this, specific approaches can be developed to anonymize the semantics and organizaitonal context of an organization.

However, the majority of the software products do not provide semantics and an organizational context which can be used by the framework. In order to show the feasibility of the framework, we need to provide the inputs the framework expects. To this end, we provide a set of challenges which have to be met in order to obtain these inputs from current software products. Next to presenting these challenges, we sketched some solutions in meeting the challenges in cooperation with a Dutch ERP software vendor (our industrial partner) for showing these challenges can be met. Later, we give an example application of the framework.

The structure of the paper is as follows. In Section 2, we present the Cross-Organizational Process Mining Framework. In Section 3, we show the feasibility of the framework by using the ERP product which is developed by our industrial partner, and we discuss the challenges by providing the solutions to them in order to be able to show the feasibility of the framework. In Section 4, we illustrate an example application of the framework. We conclude the paper in Section 5 and list future work in Section 6.

II. Cross-Organizational Process Mining Framework

In this section, we introduce the Cross-Organizational Process Mining Framework and discuss its architecture. Figure 1 shows the architecture of the framework. The framework consists of three components, Cross-Organizational Process Mining Techniques, Metric Comparison Catalog, and Metric Based Improvement Catalog. The first one is focusing on giving insights into usage commonalities and differences between various organizations in the usage of software products, and execution of their business processes. The second one contains information on how to compare the metrics. The third one enables the framework to determine and store the possible changes which will have positive effects on a metric. This can be used to propose the same changes for other processes in the same context in another organization. In the remaining of this section, we give the details of the components.

A. *Cross-Organizational Process Mining Techniques*

The Cross-Organizational Process Mining Framework contains cross-organizational process mining techniques to compare organizations. However, each organization can have expertise on different subjects, e.g., depending on its goals and knowledge. For instance, while an ERP software vendor focuses on order management, human resources, product and stock management, and invoicing, an insurance company focuses on risk management. Beyond this, the number of employees of an organization can determine its operational scope, e.g. an organization with only five employees cannot handle the same amount of issues compared to an organization which has five hundred employees working in different locations. For this reason, the techniques need an input which defines the characteristics of an organization to make more accurate comparisons. We name this the *Organizational Context*. The Organizational Context contains information about an organization e.g., number of employees, specialties, locations, and operated area like space or underground in a country. The Organizational Context enables the techniques to determine the comparability of organizations and to do a fair comparison amongst them. An organization can operate a process differently under various circumstances. For example, the delivery process of an organization operating in a flat country can have a smaller delivery duration than an organization which operates in a highland area. Moreover, there can be local laws and regulations which affect organizations differently. As a result, while determining the comparability, the techniques need to consider organizational context.

In addition to the comparability of organizations, the framework needs to determine how to compare the organizations based on their usage of a software product. To do that, the framework contains cross-organizational process mining techniques. These techniques discover commonalities and differences at the usage amongst organizations. However, the terms inside the commonalities can carry different meanings across organizations. Therefore, the techniques need inputs defining semantics of the usage, *Business Semantics*, which are common for organizations.

The Business Semantics works like an *ontology* that defines the semantics of things which are common for all organizations. For example, Figure 2 shows how two organizations are handling an issue reported by the customer. On the one hand, the first organization checks if the issue is already known or not. In case the problem is known, the organization shares the solution for the known problem and closes it. If the problem is not known, then the organization checks whether it is reproducible or not. If it is not reproducible, then the organization rejects it. On the other hand, the second organization does these two checks in one step called *Evaluate*. The first organization defines *Evaluate* as a two steps activity which starts with *Check Known Problem* and ends with *Reproduce Problem*. Based on this information, the techniques can determine that the semantics are the same for different terms. In addition, in this example, the techniques determine that *Reject* has a different meaning in two organizations, i.e., in the first one *Reject* reflects a problem is not reproducible while in the second one it is reflecting the rejection of a problem due to being invalid. As a result, the Business Semantics enable the framework to determine comparable things more accurately.

B. Metric Comparison Catalog

The Metric Comparison Catalog enables one to compare the metrics of the observed software product usage of organizations. For instance, throughput of two processes in the same organizational context can be compared. In order to know which metrics are comparable, the framework needs semantics for the metrics. We name this *Metric Semantics*. The Metric Semantics comprises of a metric's context, its definition, how and when to calculate it, and an optional threshold. On the one hand, the metric's context is a reference to the organizational context via the enterprise semantics. Thus, the framework can identify a metric's meaning. In order to compare the metrics, this component contains metric classifiers. A metric classifier gives information about the metric how to compare it, and information about the operations which must be applied to the metric before comparison. For instance, summation and division operations must be done for an average metric after an unit conversion. On the other hand, the threshold enables the framework to compare deviation boundaries for metrics. In particular, there can be a deviation from predefined boundaries for metrics arising from legal regulations or agreements. For instance, in a country there can be a legal upper limit for a process' completion time or average response time for a specific step in the process. Let's assume that in a country there is a legal upper limit for the loan process completion time for banks. By defining this upper limit as a threshold in the framework, both the software vendor developing the software for banks and the banks themselves can benefit from it. The framework can enable the software vendor to do benchmarking based on deviations from the threshold. From the bank's perspective, the framework can show the distance from the threshold amongst others and how it changes over the time. Eventually, the threshold can be a trigger for organizations to take improving actions.

Figure 2 shows an example scenario in which a software vendor wants to obtain insights from two organizations' problem resolution based on a throughput metric *(throughput-resolved metric is defined as the total number of resolved incidents in an hour)*. On the one hand, there is Process X in Context c1, which reflects an incident management process. The process starts with checking the incident whether it is a known problem or not. If it is a known problem, then the *Share Solution* branch is activated. Otherwise, the *Verifying Reproducibility* branch is executed. On the other hand, there is another process (Process Y), which reflects an incident management process in the same context as the previous. In Process Y, an incident is firstly evaluated and then it is solved or closed. As shown in the figure, there is a *Close* step in both processes, but the semantic is different. In Process X, *Close* reflects that the incident is *resolved*. Conversely, in Process Y it means it is *not resolved*. In addition, there is a throughput metric for both solve and close. During the cross-organizational process mining, the Metric Comparison Catalog checks the metrics' context and compares them. In our example, the Metric Comparison Catalog determines that

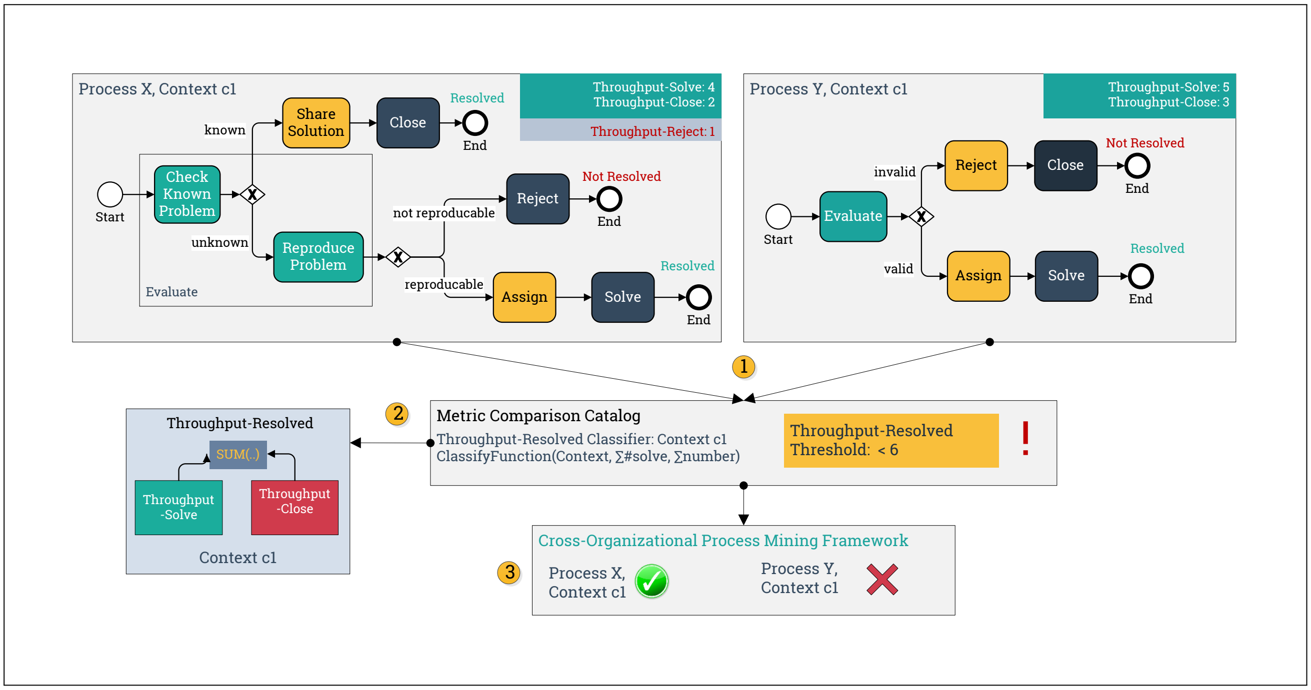

Figure 2. An example scenario showing the usage of Metric Comparison Catalog

in Process X the step *Close* has the same semantic from a throughput metric perspective as the step *Solve*. Therefore, the total throughput for resolved in Process X is the summation of throughput-solve and throughput-close ($(4+2) = 6$ incidents).

As shown in the figure, there is a threshold definition for the throughput metric, such that the framework can generate a warning using the threshold definition in order to trigger improvement in real-time. In the example, the threshold value is defined as a minimum number of resolved issues. Based on this, the framework checks if there is any throughput number value for problem resolution less than the threshold. If so, the framework generates a warning. With this, one can be notified in real-time to investigate the process execution. Next, the semantics and organizational context provide a base to create derived metrics. For instance, an organization can have different branches around the world in different cities, countries, and also on continents. Using this information, the framework can enable the Metric Comparison Catalog to generate derived metrics for each level in the organization and compare them with other branches which have a similar organizational context. As a result, this structure of the Metric Comparison Catalog enables one to do real-time process mining as a next step.

C. Metric Based Improvement Catalog

The framework can provide improvement proposals by obtaining insights from the Metric Based Improvement Catalog component. This component stores the information about the changes done in a process and its effects on the metrics. With this information, the framework can determine the possible changes that can be applied to another process which operates in the same context.

Figure 3 shows an example scenario. In the figure, there is Process X in Context c1 which starts with Task A. Task A is followed by B, C and D. Afterwards, Process X ends. This process has metric *m1* with value *100*. After a while, the organization using this process changes its model without changing its goal and context. The organization decides to parallelize the tasks B and C (Process XE). After the change, the metric's value improves by *20%* percent (from 100 to 80). The Metric Based Improvement Catalog stores this information. The framework knows there is another process (Process Y) which is used by a different organization in the same context. Furthermore, the organization uses the same metric for the process but the process has a different model. With this knowledge, the framework provides the improvement proposal to *parallelize B and C for Process Y*.

In this section, we discussed the Cross-Organizational Process Mining Framework and its components by explaining the opportunities which they bring with the example scenarios. In the following section, we list a set of challenges at transforming a software product's data into the inputs required by the framework. Next, we sketch solutions to meet these challenges and illustrate an example application.

III. Challenges and Solutions

Preliminary results section merged with this section. Figure 5-6 are newly added As shown by the architecture of the

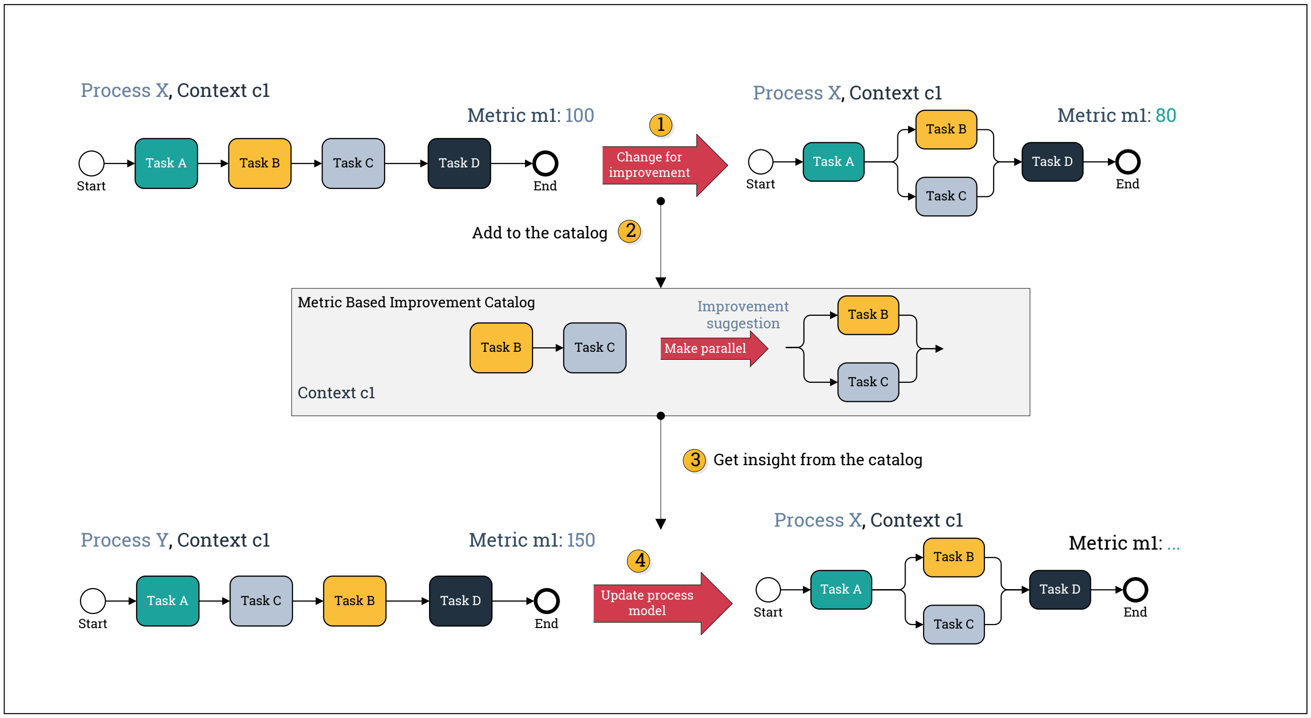

Figure 3. An example scenario shows the usage of Metric Based Improvement Catalog

framework in Figure 1, the framework needs four inputs; organizational context, business semantics, metric semantics, and event logs. For this reason, we describe challenges which need to be met in order to make the inputs available for the framework. Next tot this, most of the software products do not provide these inputs in general. Therefore, we sketch solutions for some of the challenges to turn a software product's data into the inputs expected by the framework. Next, to show the applicability of our solutions, we have applied our solutions to the ERP product developed by our industrial partner and used by more than *10.000* customers. In the ERP product, the inputs which are required by the framework are not readily available. Therefore, we follow the steps shown in figure 4 to develop solutions for transforming the ERP product's data into the required inputs. In order to show the feasibility of the sketched solutions, we use two sample organizations' data which is provided by our industrial partner.

A. Business Semantics

The framework needs clean and unambiguous business semantics of the product and each of its elements. With the semantics, the framework can determine which data is comparable and how it can be compared.

- Challenge 1.1: Determine business semantics of the current product.

Having clean and unambiguous semantics is the most important challenge for us, because, it is the precondition to gain information and create knowledge. Without understanding the meaning of the data, we cannot use it properly. However, the business logic and semantics behind the elements are currently hard-coded in the application. Next, the current product's database contains partial information related to this. In order to use the semantics, first we need to uncover them. At the company, we worked with architects to uncover the semantics. Besides this, we talked with product owners and database administrators to discover the business logic inside the current product. As a result of these discussions, we created a *rained class model* [5]. The model shows the elements of the current product's business logic. With this, we were able to get the general semantics for organizations who use the current product. Later, we created a *class model* for each organization to get organization specific semantics. We mapped one class model to the other in order to determine which elements are comparable. We did the mapping by following three steps; selection of an element from a model, checking its meaning inside the organization, and matching an element from another model. With this, we were able to give the proper semantics as an input to the framework. For now, we ignored non-matching elements, but we will provide solutions in the future.

B. Organizational Context

To eliminate mis-comparison of organizations having different characteristics, the framework needs the organizational context as an input. The organizations adjust their processes depending on their characteristics and circumstances arising from these characteristics to reach their goals. In particular, the

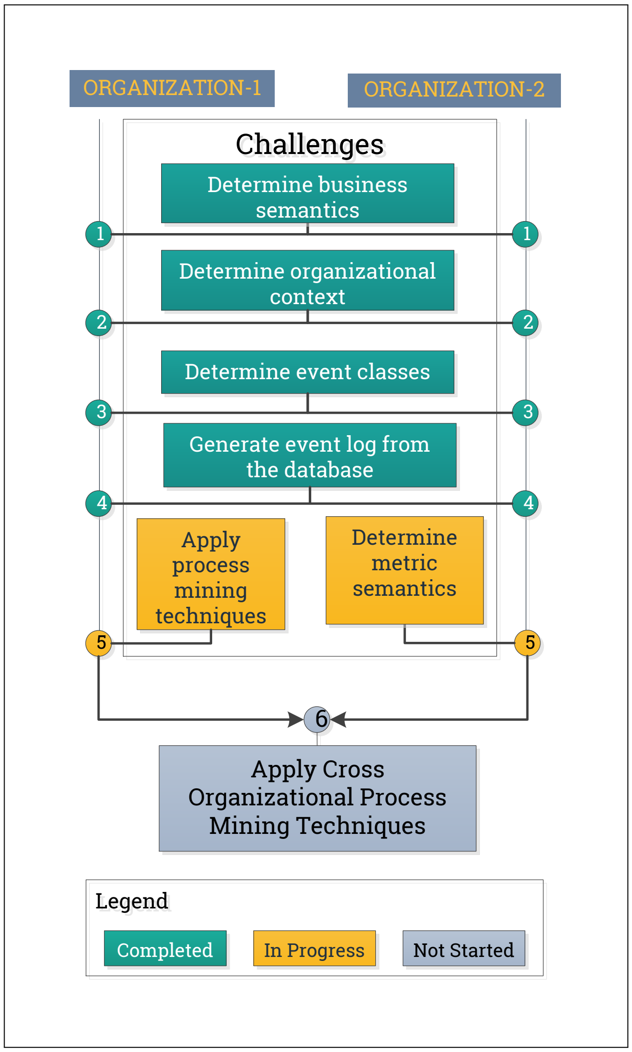

Figure 4. Followed steps to transform the current product's data into the inputs required by the Cross-Organizational Process Mining Framework

number of employees, the working domain, legal regulations, and the working environment can be the main reasons for adjusting the processes. Next to this, these adjustments affect the process metrics. For instance, there can be a throughput metric which has not in the same value range in two organizations. In addition, there can be time dependent behavior among the organizations. With the organizational context input, the framework can determine how to compare the organizations showing time dependent behavior.

- Challenge 2.1: Determine organizational contexts.

In order to make a fair comparison between organizations, we also need to identify whether they are in the same context. Our industrial partner provided us data that shows which organization uses which modules of the current product, how many entities are stored in specific tables for each organization, and characteristic attributes (e.g., name, location, customer target audience, number of employees) of each organization. This data was adequate as a starting point to identify how to compare the organizations. Based on this information, we discussed with the product owners to select the relevant parts of the data which can help us to compare organizations. In the end, we selected target customer audience and location as characteristic attributes. Target customer audience attribute value can have one of these values; B2B (Business-to-Business), B2C (Business-to-Customer) or B2G (Business-to-Government). Later, we filtered the organizations based on these attributes to select two of them as sample for an example application.

C. Event log

The event log, a footprint of a process' execution, is the essential input for process mining. An event log is a collection of recorded actions for a particular process instance. Each action belongs to an event class, i.e., event type. For example, a process comprising of create, update, save, and archive steps may have ⟨create, update, save, update, save, update, save, archive⟩ as a possible trace. In this trace, *create*, *update*, *save*, and *archive* represent event classes. However, most software products do not record these actions explicitly. Therefore, it is required to derive an event log from software products' data.

- Challenge 3.1: Determine event classes.

The constrained class model that we created while determining business semantics contains relations between objects. These relations represent the interaction between objects in the current product. Next to this, the current product's help document contains the actions one can do while using the product. By using the constrained class model and the help document, we can determine the event classes. The event classes enable one to determine which part of the product's data can be used for event log generation. To this end, we listed possible event classes by using the constrained class model and the help document. Later, we discussed with the product owners to select suitable event classes in the list.

- Challenge 3.2: Generate event log from the database.

The current product records particular actions that are done by the user as operations in the database, e.g., item creation, item update, item packing, and item delivery. Furthermore, there are some date columns in the database which store the time that a specific action was executed. For example, the update action time is stored in a *modifiedDate* column, the delivery start is stored in *deliveryStartDate* column, and the delivery complete time is stored in *deliveryEndDate* column. Using these date columns, we can create events. However, to do process mining, we need to build the traces which depict the sequence of events. In order to build the traces, we chose an approach that uses Redo logs as our reference to generate event log [6]. Database schema usage and process instance identification parts of the approach helped us to determine the process identifiers in order to build traces. However, relationships for the current product are not stored at DBMS (Database Management System) level but, they are stored

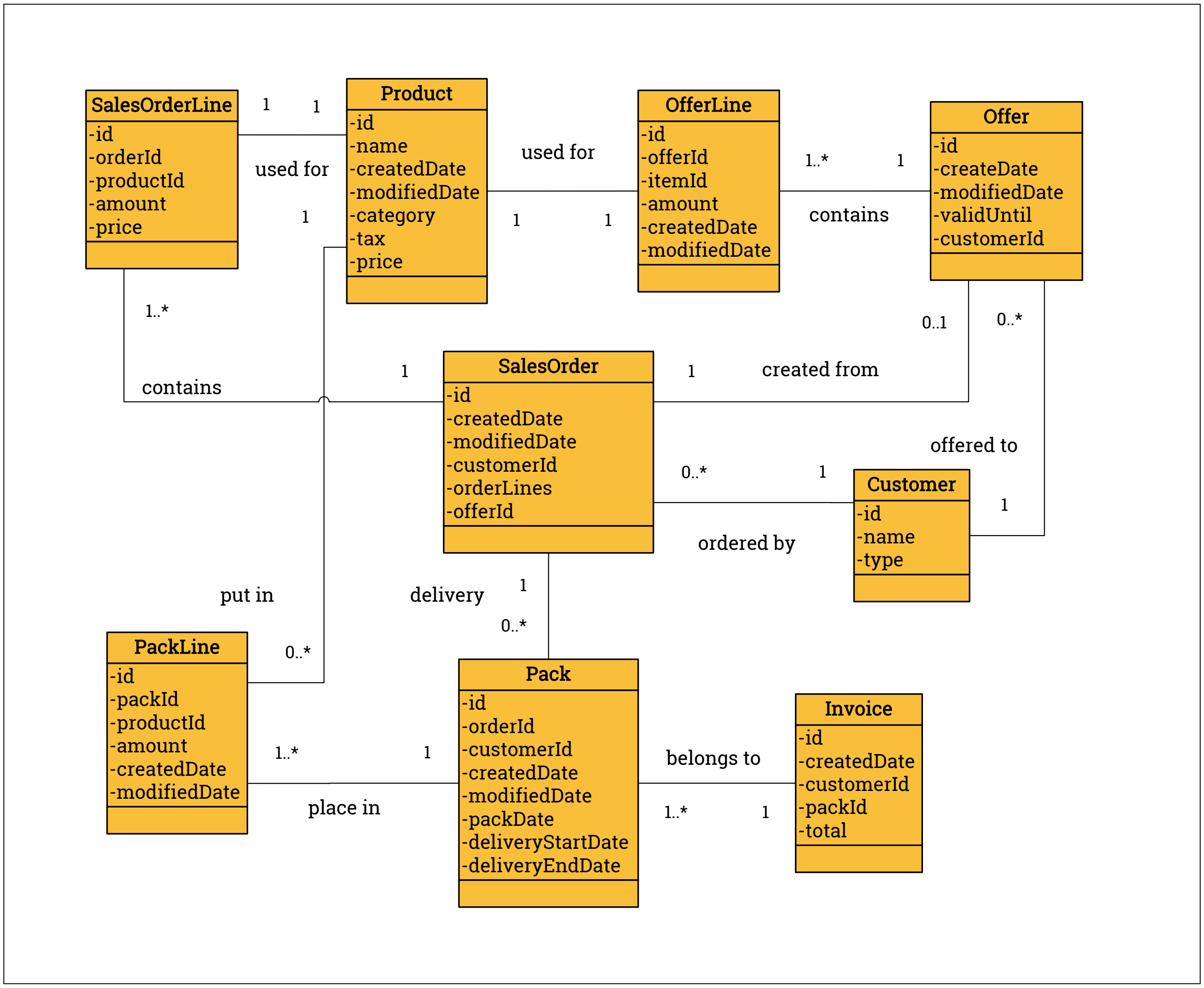

Figure 5. The ERP product’s constrained class model and the elements in the model

in the database at particular tables. Therefore, we developed a software module which reads the relationship information stored at particular tables in the database and creates the database schema. Then, we matched the tables in the schema with the event classes that we created in the previous step. After that, we prepared custom database queries and extracted the event log.

D. Metric Semantics

In order to know which metrics are comparable and how to compare them, the framework needs semantics for the metrics.

- Challenge 4.1: Determine metric semantics.

The current product has business intelligence features for tracking Key Performance Indicators (KPI) which are defined as a standard set. As a starting point, we discussed about these KPIs with product owners in order to determine whether they are related to the processes or the operational data e.g., disk usage and database size. After listing process related metrics, we will determine their contexts and comparison methods in order to compare them in an accurate way.

IV. An Example Application of the Cross-Organizational Process Mining Framework

In this section, we give an example application of the cross-organizational process mining framework. We use the inputs that we transform from an ERP software product’s data in the previous section. In the example, we illustrate how the framework generates insights both from a software vendor’s (in this example it is our industrial partner) and an organization’s perspectives. Figure 7 illustrates the interaction between the user and our framework. To obtain insights, first, one needs to have process mining questions. The questions help to determine which cross-organizational process mining

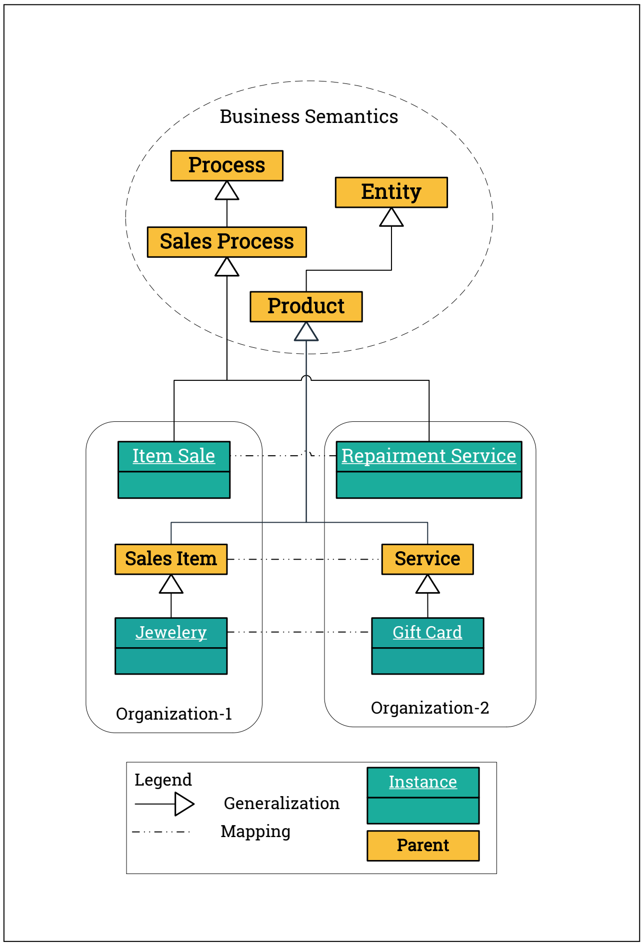

Figure 6. Mapping of two sample organizations' constrained class model elements and element relations with Business Semantics

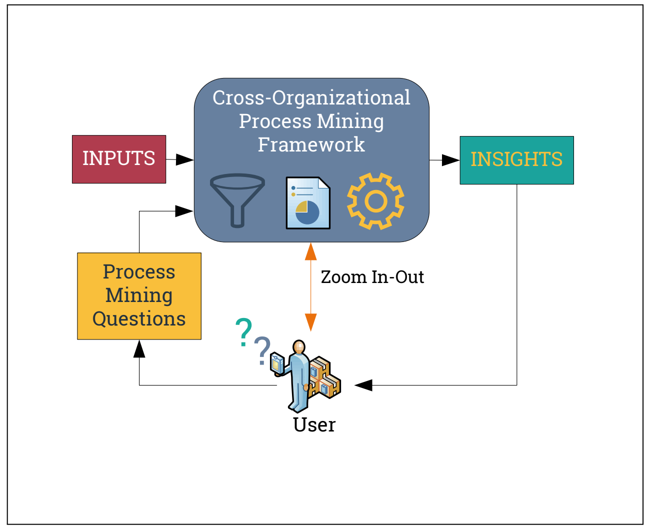

Figure 7. An illustration showing the interaction between the user and our framework

techniques can be applied. In this example, we use order management data of two sample organizations (Organization-1 and Organization-2). They are located in different countries in Europe. Both organizations are B2B. While Organization-1 is selling office materials, Organization-2 offers prepaid phone repairment service. Both use the same software product, developed by our software vendor, for their sales processes. In the current product, a sales process can start with or without an *offer*. The former continues the same as the latter if the customer accepts the offer.

A. *From Industrial Partner's Perspective*

Here, we explain the interaction between our industrial partner and our framework. As a software vendor, our industrial partner wants to see how organizations use its software product. Regarding to this, the software vendor has the following questions.

Process Mining Question 1 (PMQ1): *Does the sales process start with an offer or without an offer in organizations based on an organization's location?*
To generate insights for this question, the framework firstly determines the semantic for the term *sales process*. The framework looks into the Business Semantics and determines what is a sales process and what are the characteristics of a sales process. Then, it checks the event logs if organizations execute a sales process. Furthermore, the framework checks how a sales process can start. After determining the starting points, the framework searches starting points which are matching with the defined criteria in the question. After the search, the framework shows the generated insight. In this example the framework generates: "All the companies which are located in Europe start the sales process without an offer.". Regarding to the generated insight the software vendor may provide new process mining questions to zoom-in and obtain more specific insights. Let's assume that the software vendor created a new process mining question namely PMQ2.

PMQ2: *What is the most frequent start activity for the sales process in organizations which are located in Europe?*
During generation of the insights for this question, the framework determines the meaning of *most frequent* in addition to the steps in the previous question. As a metric, *most frequent* is defined in metric semantics. Therefore, the framework checks the metric semantics and the information about the term, *most frequent*, how and when to calculate it. In this example, the framework determines the meaning of the "most frequent" as *case frequency metric* and how to calculate it. At the end of the metric calculation the framework generates the insight: "The most frequent start activity is CreateOrder. 90% of the cases in Organization-1 and 89% in Organization-2 start with CreateOrder activity.".

Again, using the obtained insight the software vendor can create new process mining questions. For instance, the next new question can be "What is the most frequent start and end activities for completed sales processes in organizations which are located in Europe?". For this question, the framework needs to determine the meaning of the term *completed* for a sales process. Then, the framework can generate insights.

In addition, the software vendor can define a metric using the metric semantics catalog. Let's assume that the software vendor wants to compare the most frequent path in sales processes across organizations. The software vendor wants to see the result as a similarity metric (is defined by the vendor as the ratio of total common activity types inside the most frequent path of a sales process to the total activity types inside the most frequent path's of all sales processes). Based on the similarity metric, the framework checks activity types which have the same semantic. The framework determines that only *NewOrder* and *CreateOrder* activities have the same semantic. Based on this, the framework generates the insight: "The similarity is $1/5$ for the most frequent variant of the sales processes. The most frequent variant for Organization-1 is: 58%, [CreateOrder, PreparePack, Deliver] and the most frequent variant for Organization-2 is: 66%, [NewOrder, CheckStock, CreateInvoice]".

B. From An Organization's Perspective

An organization also can obtain insights and benefit from cross-organizational process mining. For instance, an organization who wants to compare itself with others can obtain insights which can help to determine how to improve its processes. Let's assume that an organization wants to compare itself with others in terms of process completion time, in particular, the mean case duration. To this end, the organization defines a process mining question, *What is the difference between our organization and the others based on the average duration of performing a sales process instance?*
To generate insights for this question, the framework needs to determine the meaning of *average*. Let's assume that average is by default defined in the framework as *the difference from the mean case duration metric*. With this metric definition the framework generates an insight, "A sales process instance is performed on average 40.25 hours in all organizations. Your organization performs a sales process by spending 70% more time than the average.". As we discussed before, based on this insight one can create more specific process mining questions. For example, the organization may also want to compare itself with others based on location, number of employees, or based on another characteristic.

The examples from a software vendor's perspective and an organization's perspective indicate that one can obtain more specific insights by executing more interaction cycles with our framework. In each cycle, one can define more granular process mining questions using organizational contexts, metric semantics, or business semantics.

V. Conclusion

In this paper, we present a generic Cross-Organizational Process Mining Framework which is aimed at comparing organizations based on the usage of a software product. The framework uses organizational context, business semantics, and metric semantics, apart from current approaches ([1],[2],[3]), in order to compare the organizations more accurately. Next, the inputs allow the framework to monitor the *concept drift*[1], i.e., the same process variant may operate differently under various circumstances possibly depending on the season. For example, there may be seasonal or environmental circumstances affecting the features of a process. The same process may have longer execution times in summer than winter. Also, the same process may have less delivery duration at in flat area than in a highland area. Furthermore, the framework has a Metric Comparison Catalog component that enables the framework to determine how to compare the metrics. Moreover, by having clear semantics and the organizational context, the framework can track changes in the processes and its (positive) effects. An organization operating in a similar context might benefit from the same changes. In order to accommodate these *improvement suggestions*, the framework creates a Metric Based Improvement Catalog of observed suggestions.

In order to be able to show the feasibility of the framework, we used the ERP product developed by our industrial partner. We first checked the availability of the inputs which our framework needs. In the product, the inputs were not readily available. Therefore, we listed challenges how to transform the product's data into the inputs required by the framework. Then, to meet these challenges, we sketched solutions which can be applied for any other products used by other organizations. On the one hand, the challenges related to the semantics can be resolved with the help of experts who are directly involved in product development. On the other hand, the challenges related to the event log extraction from a database can be met by extending different approaches ([6],[7],[8],[9],[10],[11]). However, there is still not a generic solution which can be applied to any kind of database which has no redo logs or just reflecting the current state of the data.

In addition, we gave an example application of the framework both from a software vendor and an organization perspective. The example shows how a software vendor and an organization can benefit from the framework in order to obtain more insights. In the example, we also discussed the user interaction with our framework.

Other software vendors, who focus on comparing processes within different organizations, can apply our framework to their products by meeting the challenges that we presented. Next to this, organizations can benefit from by comparing themselves with other organizations. In particular, the metric based process improvement catalog provides improvement suggestions to the organizations by capturing changes inside other organizations which have positive effects on the metrics.

[1]In machine learning, concept drift means the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. This causes problems because the predictions become less accurate as time passes. In the context of process mining instead of a variable, the complete process is investigated. This makes it complicating to define the notion.

VI. Future Work

Cross-Organizational Process Mining is a novel and emerging area. However, there are related approaches to cross-organizational process mining from which we can benefit. These approaches are mainly focusing on process model similarity and comparison based on syntax. The approach in [12] presents process equivalence including *fitness*[2] and *precision*[3] notions. This approach can help us to develop a generic method to determine the similarity of process models amongst different organizations.

In addition, the approaches listed in [2] and [14] are beneficial to sketch a solution to make accurate comparisons. Moreover, the meta-model presented in [15] uses a semantic mapping that allows the design-time analysis of process performance indicators. Based on this, one can develop enhanced analysis techniques. And also, developed techniques can be integrated with machine learning techniques in order to propose better improvement suggestions.

Furthermore, the case study explained in [14] uses the Process Tree approach, presented in [16], to illustrate the process model comparison from a control-flow perspective. We can benefit from the Process Tree approach in order to visualize and emphasize the differences and commonalities between processes. Furthermore, we can extend this approach to reflect performance metrics which we are going to discover from the event log.

Acknowledgment This research was supported by the NWO AMUSE project (628.006.001): a collaboration between Vrije Universiteit Amsterdam, Utrecht University, and AFAS Software in the Netherlands.

References

[1] W. M. P. van der Aalst, "Configurable services in the cloud: Supporting variability while enabling cross-organizational process mining," in *On the Move to Meaningful Internet Systems: OTM 2010 - Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25-29, 2010, Proceedings, Part I*, 2010, pp. 8–25. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16934-2_5

[2] J. C. A. M. Buijs and H. A. Reijers, "Comparing business process variants using models and event logs," in *Enterprise, Business-Process and Information Systems Modeling - 15th International Conference, BPMDS 2014, 19th International Conference, EMMSAD 2014, Held at CAiSE 2014, Thessaloniki, Greece, June 16-17, 2014. Proceedings*, 2014, pp. 154–168. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-43745-2_11

[3] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "Towards cross-organizational process mining in collections of process models and their executions," in *Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part II*, 2011, pp. 2–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28115-0_2

[4] A. Burattin, M. Conti, and D. Turato, "Toward an anonymous process mining," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, Aug 2015, pp. 58–63.

[5] W. M. P. van der Aalst, "Extracting event data from databases to unleash process mining," in *BPM-Driving innovation in a digital world*. Springer, 2015, pp. 105–128.

[6] E. González López de Murillas, W. M. P. van der Aalst, and H. A. Reijers, *Process Mining on Databases: Unearthing Historical Data from Redo Logs*. Cham: Springer International Publishing, 2015, pp. 367–385. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23063-4_25

[7] H. M. W. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "Xes, xesame, and prom 6," in *Information Systems Evolution: CAiSE Forum 2010, Hammamet, Tunisia, June 7-9, 2010, Selected Extended Papers*, P. Soffer and E. Proper, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 60–75. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-17722-4_5

[8] S. Goel, J. M. Bhat, and B. Weber, "End-to-end process extraction in process unaware systems," in *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 2012, pp. 162–173. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36285-9_17

[9] D. Calvanese, M. Montali, A. Syamsiyah, and W. M. P. van der Aalst, "Ontology-driven extraction of event logs from relational databases."

[10] J. Li, H. J. Wang, and X. Bai, "An intelligent approach to data extraction and task identification for process mining," *Information Systems Frontiers*, vol. 17, no. 6, pp. 1195–1208, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10796-015-9564-3

[11] S. Pospiech, R. Mertens, S. Mielke, M. Städler, and P. Söhlke, "Creating event logs from heterogeneous, unstructured business data," in *Multidimensional Views on Enterprise Information Systems, Proceedings of ERP Future 2014, Dornbirn, Austria, November 17-18, 2014*, 2014, pp. 85–93. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-27043-2_7

[12] W. M. P. van der Aalst, A. K. A. de Medeiros, and A. J. M. M. Weijters, "Process equivalence: Comparing two process models based on observed behavior," in *Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings*, 2006, pp. 129–144. [Online]. Available: http://dx.doi.org/10.1007/11841760_10

[13] W. M. P. van der Aalst, *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19345-3

[14] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "Mining configurable process models from collections of event logs," in *Business Process Management - 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*, 2013, pp. 33–48. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40176-3_5

[15] A. del-Río-Ortega, M. Resinas, C. Cabanillas, and A. R. Cortés, "On the definition and design-time analysis of process performance indicators," *Inf. Syst.*, vol. 38, no. 4, pp. 470–490, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.is.2012.11.004

[16] D. M. M. Schunselaar, H. M. W. Verbeek, W. M. P. van der Aalst, and H. A. Reijers, "Petra: A tool for analysing a process family," in *Proceedings of the International Workshop on Petri Nets and Software Engineering, co-located with 35th International Conference on Application and Theory of Petri Nets and Concurrency (PetriNets 2014) and 14th International Conference on Application of Concurrency to System Design (ACSD 2014), Tunis, Tunisia, June 23-24, 2014.*, 2014, pp. 269–288. [Online]. Available: http://ceur-ws.org/Vol-1160/paper16.pdf

[2] The discovered model should allow for the behavior seen in the event log [13] i.e., a measure reflects that how much the model can reproduce the traces in the event log.

[3] The discovered model should not allow for behavior which is completely unrelated to what was seen in the event log [13].