An Approach for Automatically Deriving Key Performance Indicators from Ontological Enterprise Models

Aksu, U.A.; Schunselaar, D.M.M.; Reijers, H.A.

published in
SIMPDA 2017 Data-driven Process Discovery and Analysis
2017

An Approach for Automatically Deriving Key Performance Indicators from Ontological Enterprise Models

Ünal Aksu, Dennis M.M. Schunselaar, and Hajo A. Reijers

Vrije Universiteit Amsterdam, The Netherlands
{u.a.aksu,d.m.m.schunselaar,h.a.reijers}@vu.nl

Abstract. Organizations use Key Performance Indicators (KPIs) to monitor whether they attain their goals. Software vendors that supply generic software provide predefined KPIs in their software products for these organizations. However, each organization wants KPIs to be tailored to its specific goals. Therefore, software vendors spend significant efforts on tailoring KPIs to organizations. That tailoring process is time-consuming and costly due to differences in the real-world phenomena of these organizations. In this context, we present our novel Automated KPI Derivation Approach. To automate the derivation of KPIs, our approach obtains the exact meaning of the terms in the real-world phenomena of an organization that is modeled in the form Ontological Enterprise Models (OEMs). As a proof-of-concept we implemented our approach. We demonstrate its use in a real-life setting and present preliminary results.

Keywords: Key Performance Indicators, Ontological Enterprise Modeling, Enterprise Resource Planning

1 Introduction

Organizations use generic software, e.g., Enterprise Resource Planning (ERP) software, to support their business processes. Within this software, measuring the performance of business processes is essential for organizations [8] while progressing towards their goals. To this end, organizations use Key Performance Indicators (KPIs). For instance, *average duration of product delivery* is a KPI that organizations use to monitor their product delivery processes. By tracking this KPI, organizations can predict how much staff must be assigned to their product delivery processes to keep the duration of a product delivery below a certain threshold, e.g., on average 3 days.

Software vendors that supply generic software products typically provide predefined KPIs for organizations. However, predefined KPIs will not work successfully in all organizations because organizations want KPIs tailored to their specific organizational goals [8]. For instance, while *average duration of product delivery* is a relevant KPI for an organization that operates in the retail domain, for a library it may not be relevant. Currently, to deal with providing tailored KPIs, software vendors either customize their software products for each organization based on the KPI definitions of organizations or they include Business Intelligence (BI) functionality into their software products to let organizations design custom KPIs and dashboards. However, both of these solutions require a significant effort of software vendors and organizations [8]. To this end, a large number of approaches have been proposed for defining, modeling, and customizing

⅄ This work is a result of the AMUSE project. See amuse-project.org for more information.

KPIs [4–6, 9–11]. These approaches either provide a set of KPIs with the assumption that they will be directly used by numerous organizations or introduce a structure for defining and modeling KPIs such that organizations can customize KPIs for themselves. However, if an organization wants to use the KPIs that are defined using one of these approaches, then, first, the organization needs to obtain the knowledge on the exact definitions of these KPIs, and secondly, the organization needs to identify the related activities in its business processes to calculate the value of these KPIs. For example, *The percentage of on-time product delivery* can be defined as a KPI via these approaches to monitor a product delivery process of various organizations e.g., retailers, rental agencies, and banks. On the one hand, organizations need to manually determine what are the related real-world business activities in their product delivery process to derive that KPI, e.g., packing, shipping, and physical delivery. On the other hand, organizations need to identify the operations that are required to calculate the value of that KPI. All but one [5] do not provide a solution for determining the related activities and operations in business processes to derive KPIs and calculate their values. To determine the related activities and operations in business processes for deriving a KPI and calculating its value, del Río-Ortega et al. use annotations [5] in business processes as a solution. But manually annotating the business processes in organizations requires a significant effort due to the required technical knowledge from organizations. Moreover, the term *delivery* in the same KPI, *The percentage of on-time product delivery* might indicate different concepts for a retailer and a rental agency. For a retailer, it might indicate a set of activities that end when the customer receives the product and becomes the owner of it. But the same term in a rental agency might indicate an activity that results with the delivery of the key of a house to a customer where the customer neither becomes the owner of the house nor the key of the house. However, these approaches do not consider the differences in the meanings of the real-world business activities in the business processes in organizations. Therefore, due to the deficiencies of these approaches that we exemplify above, tailoring KPIs to organizations using these approaches becomes time-consuming, costly, and error-prone ①.

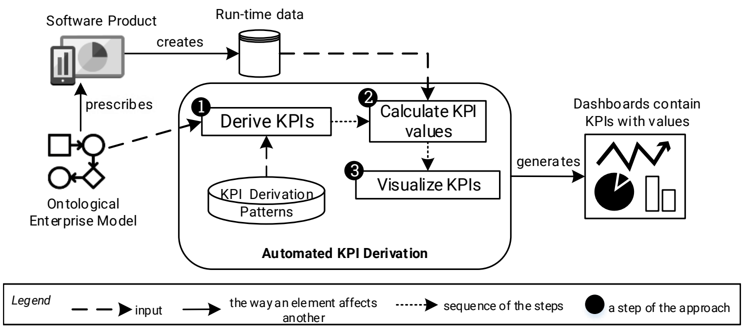

Fig. 1. Our Automated KPI Derivation Approach

With this paper, we present a novel approach to automatically derive tailored KPIs for organizations. In organizations, mostly end-users define the requirements for tailoring KPIs who often do not consider technical implementation details. ② End-users express these requirements in terms of real-world enterprise concepts such as business processes, the activities in business processes, products or services. ③ Although software vendors cope with tailoring KPIs by developing custom-made solutions for organizations, customization possibilities are limited and require technical knowledge from organizations. To meet the aforementioned challenges (①, ②,and ③), we develop a novel approach that derives tailored KPIs for organizations using the knowledge on real-world enterprise concepts in organizations. To automatically obtain that knowledge, our approach takes an *Ontological Enterprise Model (OEM)* as input. Using an OEM, our approach derives KPIs automatically by means of KPI derivation patterns (see ❶ in Fig. 1). A KPI derivation pattern denotes the criteria when and how KPIs will be derived. If our approach determines that a given OEM satisfies the criteria in a KPI derivation pattern, it will derive KPIs with respect to these criteria. Subsequently, our approach calculates the values of the derived KPIs (see ❷ in Fig. 1) using the *run-time data* for that given OEM. Finally, our approach visualizes the derived KPIs (see ❸ in Fig. 1) in the form of dashboards. In short, our approach provides tailored KPIs to organizations by automatically deriving the KPIs using the real-world enterprise concepts of these organizations without requiring any technical knowledge from them.To validate our approach, we apply it on a case study; we will present preliminary results in this context.

The paper is structured as follows. We describe our research approach, which is an instance of the design-science paradigm, in Sect. 2. After that, in Sect. 3, we present our approach aimed at automatically deriving KPIs from an OEM by means of KPI derivation patterns. In Sect. 4, we describe the case study context where we validate our approach. In Sect. 5, we describe how we define KPI derivation patterns to derive KPIs in the real-world phenomena of organizations. Afterwards, in Sect. 6, we show the applicability of our approach. In Sect. 7, we provide an overview of related work on deriving KPIs and tailoring them to organizations. Finally, in Sect. 8, we present our conclusions and the potential directions for future work.

2 Research Method

In this research, we propose a novel approach to automatically derive tailored KPIs for organizations from an OEM by means of KPI derivation patterns. By applying the design-science methodology, we develop the design artifact of our approach. Then, we evaluate it on a case study at a Dutch ERP software vendor by applying the four case study techniques that are defined by Yin [13] as sources of case study evidence. In the following paragraph, we elaborate on how we applied these techniques.

Firstly, in the beginning of this research, a training session is conducted to get familiar with an OEM language (called NEXT OEM Language) that the case study company is developing to model an organization's business processes in the form of OEMs [12]. Secondly, to see how real-world phenomena, essential to an organization, are modeled in the form of OEMs, we examined a set of OEMs that are modeled by the case study company. During that examination, we modeled a sample OEM and observed the generated software from that OEM to see how real-world phenomena essential to an organization are realized. Thirdly, to elicit the knowledge on the exact definitions of KPIs, which are offered by the case study company in its current ERP software product, we reverse engineered and analyzed the documentation of that ERP software product. To verify the knowledge on KPIs that we obtained, we conducted unsupervised interviews with the dashboard team of the case study company who maintains the KPIs, which are offered in the current ERP software product of the case study company. Finally, based on that knowledge, we defined KPI derivation patterns via the NEXT OEM Language such that one should be able to derive KPIs when a given OEM matches the criteria expressed in these patterns. These KPI derivation patterns were reviewed and evaluated by the dashboard team of the case study company and also by the architects who work on defining the NEXT OEM Language at the case study company. In the following section, we explain how our approach derives KPIs automatically by means of KPI derivation patterns.

3 Automated KPI Derivation from an Ontological Enterprise Model

In this section, we explain the details of our Automated KPI Derivation Approach. Which inputs our approach takes and in which steps these inputs are used are depicted in Fig. 1. In short, our approach takes an OEM for deriving tailored KPIs for organizations using KPI derivation patterns. Then, it calculates the values of the derived KPIs using the run-time data for the given OEM and visualizes the KPIs in the form of dashboards. To determine *when* and *how* KPIs can be derived, our approach contains KPI derivation patterns. A KPI derivation patterns is composed of two parts: *when* and *how*. The *when* part of a KPI derivation pattern specifies the criteria that a given OEM needs to meet for deriving a set of KPIs. The actions that need to be applied while deriving a set of KPIs are expressed in the *how* part of a KPI derivation pattern. For the representation of KPI derivation patterns, we use Condition-Action (CA) rules [7]. A CA rule is activated when its condition becomes true, i.e., if the criteria expressed in the *when* part of a KPI derivation is fulfilled, then the *how* part of it will be executed.

Definition 1. *A KPI derivation pattern is encoded in terms of a Condition-Action rule (c, a), where **c** is an expression that returns a value of true or false for an element of a given OEM. **a**=<a1,...,an> is a list of actions (an action can be included multiple times) that will be applied while deriving a set of KPIs from a KPI derivation pattern. If **c** returns true for an element of a given OEM, a KPI will be derived from the given KPI derivation pattern. Then, the actions in the list **a** will be applied on the derived KPI to calculate its value using the run-time data of the given OEM. The last action returns the calculated value.*

Step 1-Derive KPIs: In this step, our approach derives a set of KPIs when a given OEM (the precondition for our approach) satisfies the criteria expressed in the *when* part of a KPI derivation pattern. Then, our approach names the derived KPIs with respect to a *naming rule*, which corresponds to a particular KPI derivation pattern. A naming rule prescribes how the KPIs that are derived from a particular KPI derivation pattern will be named, i.e., which elements in the given OEM will be used for naming and how. For example, a set of derived KPIs will be named by the concatenation of the following texts: "total value of" and the text in the name of a particular OEM element that is contained in the when part of a specific KPI derivation pattern. Below, in Algorithm 1, we show how our approach derives KPIs.

Algorithm 1: KPI Derivation. The algorithm derives KPIs from a given *OEM* OEM^G by means of *KPI derivation patterns* KPI^P and names the derived KPIs KPI^D with respect to *naming rules NR*.

forall the ***KPI pattern*** $kp \in KPI^P$ **do**
    $c \leftarrow$ getWhenPartOfPattern(kp)
    $r \leftarrow$ getNamingRule(kp, *NR*)
    forall the ***OEM element*** $e \in OEM^G$ ***that meets the critera expressed in*** c **do**
        $k \leftarrow$ createKPI(kp, e)
        applyNamingOnKPI(k, r)
        add created KPI k into KPI^D
    end
end

Step 2-Calculate KPI Values: The run-time data for a given OEM contain references to corresponding elements in an OEM. Using the references one can obtain the related run-time data for a particular OEM element. In this step, to calculate the value of a derived KPI, our approach gets the referenced OEM element in the derived KPI. After that, our approach gets the actions (defined in the *how* part of the referenced KPI derivation pattern) that are required to calculate the value of the derived KPI. Then, our approach obtains the related run-time data for the referenced OEM element using the function named *getInstancesOf*. This function returns the related run-time data for a given OEM element from the run-time data of the given OEM using the references from the run-time data to the corresponding elements in the given OEM. If there is no related run-time data for the referenced OEM element, our approach indicates that KPI by a *no-value* marker. The calculation of the KPI values is depicted in Algorithm 2.

Algorithm 2: KPI Value Calculation. This algorithm calculates the values of the derived KPIs KPI^{D} using the run-time data OEM^{R} of OEM^{G} that is used at deriving KPIs.

forall the ***Derived KPI*** $k \in KPI^{D}$ **do**
- $e \leftarrow$ getReferencedOEMElement(k)
- $kp \leftarrow$ getReferencedKPIDerivationPattern(k)
- $Acts \leftarrow$ getHowPartOfPattern(kp)
- $eIns \leftarrow$ getInstancesOf(e, OEM^{R})
- $v \leftarrow$ calculateKPIValue$(eIns, Acts)$
- setKPIValue(k, v)
- **add** KPI k into KPI^{DV}

end

Step 3-Visualize KPIs: Derived KPIs with their values will be visualized using *visualization rules* and *filtering rules*. *A visualization rule* prescribes the presentation style for derived KPIs, i.e., which visual element placement scheme, chart type, or coloring scheme needs to be used. Based on which dimensions derived KPIs can be filtered is prescribed by *Filtering rules*. To show a set of derived KPIs from different perspectives, KPI dimensions can be combined with these KPIs. Related to that, our approach shows derived KPIs from a time perspective.

In this paper, we present our Automated KPI Derivation Approach that is a part of our Cross-Organizational Process Mining Framework, which we introduced in [2]. We considered the implementation of our approach[1] as a plug-in of *ProM* [1], which is an extensible process mining framework that supports a large number of process mining techniques by means of plug-ins. Our considerations for *ProM* are two-fold: (1) *ProM* handles the import and export of run-time data, (2) we aim to benefit from a wide variety of process mining techniques in the future while developing our framework. We have implemented our approach (except some filtering functionality). We can load an OEM with its run-time data, and show the dashboards for the automatically derived KPIs. In the following section, we define the context of the case study that we conduct to evaluate our approach.

4 Case Study Context

As we introduced in Sect. 1, our Automated KPI Derivation Approach automatically derives KPIs from OEM. In the case study that we conduct, an OEM was provided by the case study company. The case study company is developing a novel model-driven software generation approach [12] to aid automatically generating ERP software from a model. As part of NEXT,

[1] The implementation of our Automated KPI Derivation Approach is available at https://www.amuse-project.org/portal-amuse/kpiderivation

a declarative modeling language, namely the NEXT OEM Language is being developed by the case study company. This language will contain modeling elements such that they will reveal and reflect the meaning of the real-world enterprise concepts of an organization (e.g., business processes, the activities in business processes, products or services) and the interaction between these concepts. The NEXT OEM Language provides a holistic perspective (i.e., not separate perspectives for data, interactions, time, and conditions as in the UML) for modeling these concepts in an organization. Thus, the information inside an organization, which is required to generate the tailored ERP software for that organization can be captured by means of the model, which is created via the NEXT OEM Language. Moreover, the language is aimed at modeling that required information in the form of OEMs. Since an OEM is the precondition for our approach, we use the NEXT OEM Language to define KPI derivation patterns. Thus, we can determine whether the given OEM meets the criteria expressed in these KPI derivation patterns. Although the NEXT OEM Language is under development, it does not bring any limitations to our approach. Because the NEXT OEM Language provides sufficient elements for deriving KPIs from the given OEM by applying our approach in the case study. In this context, we introduce the case study company that provides the OEM for our approach. Then, we describe the relevant aspects of the NEXT OEM Language for our approach while expressing KPI derivation patterns in terms of the elements of the given OEM, which is created using that language. **Company Identification:** Our case study company is an ERP software vendor that develops and distributes its ERP software product (called MyERPSuite). MyERPSuite is used by more than 1.2 million end-users of 10,000 customers from various domains, e.g., retail, production, and accountancy. Furthermore, to provide relevant insights for these customers, the dashboard team of the case study company created a rich set of dashboards in MyERPSuite based on customer experience and expert knowledge. These dashboards consist of numerous KPIs for various areas such as sales, purchase, finance, and product development.

4.1 NEXT OEM Language

We start with a simple OEM to familiarize the reader with the NEXT OEM Language. We suppose there is an organization, called NRetailCorp, which operates in the retail domain. Its main business is selling products to its customers. Figure 2 depicts a simple OEM that NRetailCorp created to model its business. As shown in Fig. 2, NRetailCorp sometimes makes sales offers ❶ to its customers. It is also possible that a customer may order ❷ products without a sales offer being issued. After a sales order has been placed by a customer, NRetailCorp delivers ❸ the products to the customer. Subsequently, NRetailCorp creates a sales invoice ❹ and sends it to the customer to inform the customer.

To whom NRetailCorp delivers are denoted with the *person* and *organization* blocks in the OEM (see Fig. 2). From the perspective of NRetailCorp, these blocks perform the role of *customer*, which can be either people or organizations in the real-world. In addition, the products that NRetailCorp delivers are denoted by the block that is *good*. The type of the good, person, and organization blocks is **entity**. An entity represents a specific thing that exists independently in the real world, e.g., a person, an organization, a service, time, or goods. In the NEXT OEM Language, the difference between entities is captured by their type, which can have a value of organization, person, service, time, or good. The customer (a view on the *person* or *organization* block) and product blocks (a view on the *good* block) are both of type **role**. Roles are views on entities that describe in what way NRetailCorp considers the entities. For example, a person might be both a customer and a supplier of NRetailCorp.

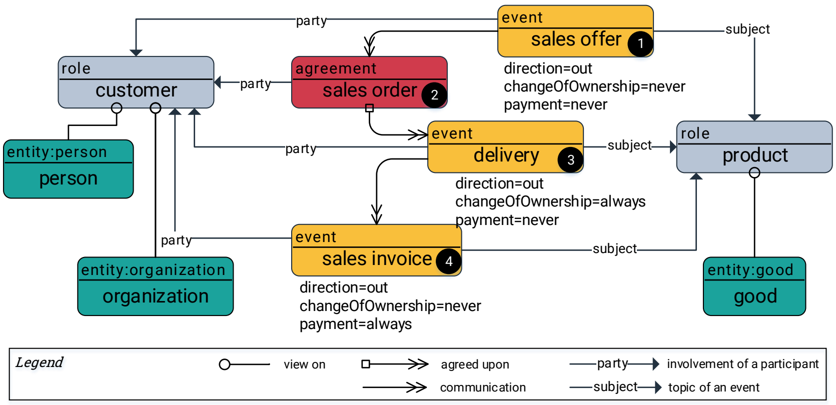

Fig. 2. A simple OEM created by NRetailCorp to model its business

Real-world business activities (e.g., making a sales offer, delivering products, or invoicing) that are relevant for NRetailCorp to administer are denoted by **event** blocks (see ❶, ❸, and ❹ in Fig. 2). Since not a single organization is exactly identical to another, these activities vary from one organization to another. Therefore, to capture the variability between organizations, each event block has a set of *characteristics*. For instance, the interactions between NRetailCorp and external entities, and what the interactions are about, are encoded in the *party* and *subject* characteristics respectively. Something of value that leaves NRetailCorp is indicated by setting the *direction* characteristic to *out*. Using the *changeOfOwnership* characteristic, the delivery event indicates whether the good is sold or rented out (see ❸ in Fig. 2). The necessity of a payment at an event is encoded in the *payment* characteristic: as the delivery of products are not free of charge, the *payment* characteristic of an invoicing event is set to *always* (see ❹ in Fig. 2).

As mentioned above, events are used to abstract real-world business activities in an organization. Although an event is stateless and timeless while modeling, i.e, at design-time, it will be executed at a specific moment in time and will move into a state. Therefore, the state of an event is encoded in the *lifecycle* characteristic, which will get a value (*done* or *todo*) at its execution. Moreover, time related information of an event, which will get values during execution, are encoded in the following characteristics: *plannedStartDate*, *plannedEndDate*, *actualStartDate*, and *actualEndDate*.

An **agreement** (see ❷ in Fig. 2) expresses an obligation between NRetailCorp and a customer to make certain things happen for the sale of products: goods have to be delivered to the customer. The edges between agreements and events indicate the events which can be performed based on an agreement. Moreover, the edges between events indicate the communication between real-world business activities. Using edges, one can traverse between events and agreements both at design time and run-time. To this end, the event has the following characteristics: *incomingConnectedEvents*, *outgoingConnectedEvents*, *incomingConnectedAgreements*, and *outgoingConnectedAgreements*. The contents of the agreement are represented by the events that can be performed based on this agreement, e.g., delivery of the products for NRetailCorp. The monetary value of the contents of the agreement is encoded in the *value* characteristic of the agreement. The *value* of an agreement will be available at run-time.

Modelers and end-users in organizations are not necessarily familiar with technical concepts such as database management systems, programming languages, or development environments. By using minimal but sufficient elements with their characteristics the NEXT OEM Language enables modelers and end-users to reason and model real-world phenomena, essential to organizations. Thus, a modeler or an end-user can easily comprehend OEMs independently from technical details and limitations. How could it be generalized to any process of the same organizations. Moreover, by being independent from technical details OEM becomes reusable for various organizations, e.g., the OEM that NRetailCorp created (see Fig. 2) can be applicable also for other organizations in the retail domain, real-estate agencies, and construction companies. Since the language is under development, over time more and more concepts will be added to the language by the case study company to capture much more real-world phenomena in OEMs, e.g., planning, production, accounting, and resource management. In the following section, we define KPI derivation patterns using the NEXT OEM Language.

5 KPI Derivation Patterns

In Sect. 3, we presented our approach that is aimed at automatically deriving KPIs from an OEM by means of KPI derivation patterns. To enable our approach to determine whether a given OEM meets any criteria for deriving KPIs, we select a set of KPIs in real-world phenomena, essential to organizations, and define KPI derivation patterns in terms of OEM elements based on the similarities of the selected KPIs. To do so, first, we obtain the meanings of the selected KPIs and the operations required to calculate the values of these KPIs. Secondly, with this knowledge we formulate the selected KPIs with using the NEXT OEM Language. Finally, we analyze the similarities and differences between these formalizations. Accordingly, in the following first subsection, we describe how we obtain the meanings and calculation logics of the selected KPIs and formulate these KPIs using the NEXT OEM Language. In the second subsection, we explain how we define KPI derivation patterns using these formalizations.

5.1 Obtaining the Knowledge on KPIs

As a set of KPIs in real-world phenomena, essential to organizations, we took the KPIs that are offered by the case study company to its customers in its current ERP product (MyERPSuite). The majority of the KPIs that are frequently used by a significant amount of customers are located in the dashboards for Sales, Purchase, and HRM. Therefore, together with the dashboard team, we selected a set of KPIs from these areas. Afterwards, we acquired the knowledge on the exact definition of these KPIs and calculation of their values. With this knowledge, we formalized each KPI using the NEXT OEM Language. Below, we list the formalizations of the KPIs mostly used from the dashboard for the Sales area.

***KPI-Total value of sales orders*:** In this KPI, sales orders are *agreements* between an organization and its customers for the delivery of products. To derive this KPI, we need to determine the *delivery events* (see ❸ in Fig. 2) that are in the context of agreements (see ❷ in Fig. 2) and not yet executed, i.e., the *lifecycle* characteristic has the value *todo*. Then, using the edges between the deliveries and agreements we can determine the sales orders. Afterwards, we can derive this KPI and calculate its value by summing up the values of the determined agreements. In the formalizations, we use **EVENTS**, **AGREEMENTS**, **ROLES**, and **ENTITIES** to denote the set of run-time instances of *Events*, *Agreements*, *Roles*, and *Entities* respectively.

$$\textit{Goods} = \{g \mid g \in \textsc{Entities} \wedge g.\text{type} = \text{good}\}$$

$$\begin{aligned}\textit{DeliveriesToDo} = \{d \mid d \in \textsc{Events} \wedge d.\text{changeOfOwnership} = \text{always} \wedge d.\text{direction} = \text{out} \\ \wedge\, d.\text{payment} = \text{never} \wedge d.\text{lifecycle} = \text{todo} \wedge d.\text{subject} \in \textit{Goods}\}\end{aligned}$$

$$\textit{Orders} = \{o \mid o \in \textsc{Agreements} \wedge o.\text{outgoingConnectedEvents} \cap \textit{DeliveriesToDo} \neq \emptyset\}$$

$$\text{KPI SalesOrdersValue} = \sum_{o \in \textit{Orders}} o.\text{value}$$

***KPI-Average delivery duration*:** In this KPI, the delivery duration of a product is defined as the difference between the end and start time of a delivery event. In order to derive this KPI, we need to determine the *delivery events* (see ❸ in Fig. 2) that are completed, i.e., the *lifecycle* characteristic has the value *done*. Thereafter, we can calculate the value of this KPI by applying a function that calculates the difference between the *actualEndDate* and *actualStartDate* of an event. To this end, we use a function named *ActualDuration*, which is formalized below.

$$\begin{aligned}\textit{DeliveriesDone} = \{d \mid d \in \textsc{Events} \wedge d.\text{changeOfOwnership} = \text{always} \wedge d.\text{direction} = \text{out} \\ \wedge\, d.\text{payment} = \text{never} \wedge d.\text{lifecycle} = \text{done} \wedge d.\text{subject} \in \textit{Goods}\}\end{aligned}$$

$$\text{ActualDuration}(e) = e.\text{actualEndDate} - e.\text{actualStartDate} \text{ where } e \in \textsc{Events}$$

$$\text{KPI DeliveryAverageDuration} = \frac{\sum_{d \in \textit{DeliveriesDone}} \text{ActualDuration}(d)}{|\textit{DeliveriesDone}|}$$

***KPI-Percentage from offer to order*:** In this KPI, acceptance means that a sales offer is accepted by a customer and a sales order is created for that offer. Therefore, we need to determine the sales offers that are followed by a sales order to derive this KPI. We formalized sales orders in the KPI-*Total value of sales orders*. Now, we need to obtain sales offers (see ❶ in Fig. 2) that precede the determined sales orders. To this end, we need to check the events that can precede an agreement. Then, we can calculate the value of this KPI.

$$\begin{aligned}\textit{Offers} = \{e \mid e \in \textsc{Events} \wedge e.\text{subject} \in \textit{Goods} \wedge e.\text{direction} = \text{out} \wedge e.\text{payment} = \text{never} \\ \wedge\, e.\text{changeOfOwnership} = \text{never}\}\end{aligned}$$

$$\textit{Offer2Order} = \{o \mid o \in \textit{Offers} \wedge o.\text{outgoingConnectedAgreements} \cap \textit{Orders} \neq \emptyset\}$$

$$\text{KPI Offer2OrderPercentage} = \frac{|\textit{Offer2Order}|}{|\textit{Offers}|} \times 100$$

The formalized KPIs above are not exclusive for the Sales area. Table 1 lists some of the KPIs from the dashboards for Sales, Purchase, and HRM that show resemblances with each other with respect to OEM elements that are used for formulating them. For example, both *Average delivery duration* and *Average receive duration* are the KPIs that show the average duration of an event, which will be performed as the result of an agreement. While in *Average delivery duration* the event is a delivery and the agreement is a sales order, in *Average receive duration* the event is a receive and the agreement is a purchase order. Furthermore, both delivery and receive events change the ownership of goods.

Table 1: Some of the KPIs from the Sales, Purchase, and HRM areas that show resemblances with each other with respect to the real-world phenomena that are expressed using the NEXT OEM Language elements in their formula

Area	**KPI**	**Formula**
Sales	*Total value of sales orders*	$\sum_{s \in Orders} s.\text{value}$
Purchase	*Total value of purchase orders*	$\sum_{p \in PurchaseOrders} p.\text{value}$
HRM	*Total value of the personnel costs at employment*	$\sum_{e \in Employments} e.\text{value}$
Sales	*Average delivery duration*	$\frac{\sum_{d \in DeliveriesDone} \text{ActualDuration(d)}}{\lvert DeliveriesDone \rvert}$
Purchase	*Average receive duration*	$\frac{\sum_{g \in GoodsReceiptsDone} \text{ActualDuration(g)}}{\lvert GoodsReceiptsDone \rvert}$
Sales	*Percentage from offer to order*	$\frac{\lvert Offer2Order \rvert}{\lvert Offers \rvert} \times 100$
HRM	*Percentage from vacancy to employment*	$\frac{\lvert Vacancy2Employment \rvert}{\lvert Vacancies \rvert} \times 100$

5.2 Defining KPI Derivation Patterns

By analyzing the formalizations of the KPIs that are depicted in Table 1, we express KPI derivation patterns using Condition-Action (CA) rules that we explained in Sect. 3. Then, together with the dashboard team of our case study company, we check whether new KPIs can be derived using these KPI derivation patterns. For the cases that new KPIs can be derived using a KPI derivation pattern, together with the same team, we have evaluated that whether these new KPIs are relevant for organizations. The first row of Table 1 contains the following KPIs: *Total value of sales orders*, *Total value of purchase orders*, and *Total value of the personnel costs at employment*. By analyzing the formalizations of these KPIs, we note that there is a pattern: each KPI is related to an *agreement*, e.g., sales order in Sales, purchase order in Purchase, and employment in HRM. Therefore, we define a KPI derivation pattern, namely *Agreement Totality* and express it as a CA rule according to the definition of a KPI derivation pattern (see Definition 1 in Sect. 3). The existence of *agreement* elements is the *c* part of the CA rule that corresponds to the when part of KPI Pattern-1. In the a part of the CA rule there are two actions. The first action (a1) gets the related run-time data for an agreement from the run-time data of an OEM (OEM^R) using the function named *getInstancesOf* (see Step- 2 in Sect. 3). The second one (a2) sums the values in the related data and calculates the value of a KPI derived from *Agreement Totality*.

(c) if a $\in$ OEM $\wedge$ a is an agreement
then (a1) $E1 =$ getInstancesOf(a, OEM^R)
(a2) $\sum_{el \in \text{E1}} el.value$

Fig. 3. *Agreement Totality* is expressed using CA rules

In the second row of Table 1, there are two different KPIs: *Average delivery duration* and *Average receive duration*. By analyzing the formalizations of these KPIs, we note that these KPIs show the average duration for an *event* on which an organization and an external entity agree, e.g., the delivery event on which an organization and its customer agree at a sales order. Based on that, we specified a new pattern, namely *average duration of an event that is executed as a result of an agreement*. While checking the new pattern together with the dashboard team, we noticed that the new pattern enables one to derive a new KPI for the HRM area. Because there is an *agreement* in the HRM area, namely *employment* that can be used with the pattern to derive a new KPI that shows average work duration. We discussed the relevance (i.e., whether a valuable insight is provided for organizations via new KPI) of that new KPI with the dashboard team. The team mentioned that from an organization's perspective only tracking the average duration of something of value that enters or leaves an organization is important and relevant for the organization, e.g., goods come into or leave the organization. However, in the KPI, *Average work duration*, there is no change of ownership because in that KPI, *time* is the entity and it is not owned by anyone in the real-world phenomena of organizations. Therefore, we need to update the new pattern such that it addresses the events that change the ownership of *goods*. The second KPI derivation pattern that we defined, namely *Ownership Duration* is shown below.

(c) if e $\in$ OEM $\wedge$ e is an event $\wedge$ e.changeofOwnership=always $\wedge$ e.payment=never $\wedge$ e.subject $\in$ Goods
then (a1) $E1 =$ getInstancesOf(e, OEM^R)
(a2) $E2 = \{el \mid el \in E1 \wedge el.$lifecycle=done $\}$
(a3) $\frac{\sum_{el \in E2} \text{ActualDuration}(el)}{|E2|}$

Fig. 4. *Ownership Duration* is expressed using CA rules

By checking the third row of Table 1, we note that the two KPIs in that row show the percentage of the *events that preceded an agreement* to the *events that can precede an agreement*. The resemblance in the formalizations of these two KPIs seems to be a pattern, i.e., in both KPIs we see an event that precedes an agreement. Based on this resemblance we specified a new pattern, namely *Continuation Percentage*. In the condition part of that pattern, we determine whether a given event can precede an agreement using its *outgoingConnectedAgreements* characteristic at design time. After getting the related run-time data for the given event, we need to filter when the given event preceded an agreement in the run-time. For that, we need to check the *lifecycle* characteristic of an event. If an event's *lifecycle* characteristic has the *done* value, this means that event preceded an agreement in the run-time.

(c) if e $\in$ OEM $\wedge$ e is an event $\wedge$ e.changeofOwnership=never $\wedge$ e.payment=never $\wedge$ e.subject $\in$ Goods $\wedge$ e.outgoingConnectedAgreements $\cap$ Agreements $\neq \emptyset$
then (a1) $E1 =$ getInstancesOf(e, OEM^R)
(a2) $E2 = \{el \mid el \in E1 \wedge el.$lifecycle=done $\}$
(a3) $\frac{|E2|}{|E1|} \times 100$

Fig. 5. *Continuation Percentage* is expressed using CA rules

After defining this new pattern, we started verifying it together with the dashboard team. We found that there is an *event*, namely *purchase offer* that can precede an *agreement*, namely *purchase order* in the Purchase area. This means that a new KPI, *Percentage from purchase offer to purchase order*, can be derived using the new pattern. With the dashboard team, we discussed whether the new KPI forms a basis for decisions to improve the purchasing process in organizations. The team stated that by tracking the new KPI, organizations can determine the actions to increase the chance that purchase offers lead to purchase orders. Furthermore, the team mentioned that the new KPI might be used as an indicator by organizations to compare their suppliers. This means that the new KPI seems quite relevant, and it provides specific and valuable insights for the Purchase area.

Since the defined KPI derivation patterns are expressed using the NEXT OEM Language, the changes in the NEXT OEM Language will require modifying the KPI derivation patterns.

For example, if the event element in the NEXT OEM Language becomes insufficient to capture particular real-world business activities, then it will be updated by the case study company and subsequently, we will modify the defined KPI derivation patterns based on that update. Due to the changes in the NEXT OEM Language, maintenance of the OEMs that were created with it will require some effort of the case study company. In addition, over time more concepts of the ERP applications, for example finance, accounting, project management, and customer relationship management, will be able to be modeled in the form of OEMs by developing new elements for the NEXT OEM Language. In accordance with that, we will be defining new KPI derivation patterns for deriving the KPIs related to these concepts. In the following section, we demonstrate the use of our approach in a real-life setting and present preliminary results.

6 Validation

In this section, we show the applicability of our approach to a case study. As we described in Sect. 3, our approach takes two inputs: an OEM and the run-time data for that OEM. In the case study, an OEM will be provided by the case study company. The OEM that the case study company provided us is an extended version of the simple OEM that we presented in Sect. 4. In the extended OEM, there is a purchasing part in addition to the sales part in the simple OEM. Unfortunately, the run-time data for the extended OEM is not operational yet because the language that the company develops for modeling real-world phenomena in organizations in the form of OEMs is still under development. However, this does not prevent us to evaluate our approach because the business processes that are modeled in the extended OEM are offered by the case study company to its customers in its current ERP product (recall MyERPSuite in Sect. 4). Therefore, we use MyERPSuite as a source for obtaining the run-time data, which captures the same information. In the following subsections, we describe how to obtain the run-time data and elaborate on the application of our approach, respectively.

6.1 Obtaining Run-Time OEM Instances

Together with an expert in the case study company, we selected two real organizations (Organization A and Organization B) from the customers of the case study company. The selected organizations operate in the retail domain and both are Business-to-Business (B2B) organizations. Moreover, these organizations execute sales and purchasing processes in MyERPSuite with respect to the extended OEM. In order to obtain the run-time data for that OEM, together with three domain experts in the case study company, we created mappings from the database schema of MyERPSuite to the elements inside the given OEM. These mappings are embedded in a set of database queries to extract the execution history from the database of MyERPSuite for each element in the OEM. Each mapping specifies which table in the database and which fields in a table are related to a mapped OEM element. We executed these database queries for the selected organizations and extracted the run-time data as an Event Log for each organization. Both of these event logs contain 6 months of data (January 2014 to June 2014). The event log for Organization A contains 108K cases and 159K events. The event log for Organization B contains 9K cases and 27K events. In both of the event logs, there are 5 types of events: *Sales Order*, *Delivery*, *Sales Invoice*, *Purchase Order*, and *Goods Receipt*.

6.2 Applying the Automated KPI Derivation Approach

In this subsection, we discuss the results that we obtained by the application of our Automated KPI Derivation Approach on the case study, which we explained its context in Sect. 4. In Fig. 6, the derived KPIs for the two organizations in the case study are shown. As we mentioned while elaborating on the steps of our approach in Sect. 3, our approach shows the derived KPIs from a time perspective, i.e., the values of the derived KPIs are shown using a particular time dimension in a time-line. Accordingly, in the same figure (in Fig. 6), time series charts are used to show the values of the derived KPIs for each month in a time-line.

Fig. 6. The KPIs that are derived in the case study using the KPI derivation patterns and the related fragments of the given OEM with these KPIs

The KPIs that are derived using the KPI derivation pattern, *Agreement Totality* are depicted in the first chart (see ❶ in Fig. 6). The second chart (see ❷ in Fig. 6) shows these KPIs from a monthly basis. Our approach determined two different agreements by applying KPI Pattern-1: sales order and purchase order. This means that there are two concrete instances of KPI Pattern-1 in the given OEM. Based on that, our approach derived two KPIs, namely *Total value of sales orders* and *Total value of purchase orders*. These two KPIs consist of the total value of agreements for an organization. By analyzing the values of the derived KPIs, one can note that the total value of the agreements of Organization A is higher than Organization B.

The third chart (see ❸ in Fig. 6) shows the derived KPIs based on the KPI derivation pattern, *Ownership Duration*. The same KPIs are shown in the fourth chart (see ❹ in Fig. 6) on a monthly basis. In these charts, we expected to see two KPIs: *Percentage from offer to order* and *Percentage from purchase offer to purchase order* with respect to the given OEM. However, the charts are empty. This means that there are references from the run-time data to the OEM elements specified in KPI Pattern-2. Regarding that, we discussed about the empty chart with two product managers from the case study company. The two product managers noted that the selected two organizations are not using sales offer and purchase offer events, which are not mandatory to execute in MyERPSuite.

The KPIs that are derived from the KPI derivation pattern, *Continuation Percentage* are shown in the fifth (see ❺ in Fig. 6). In addition, the derived KPIs are depicted on a monthly basis in the sixth chart (see ❻ in Fig. 6). By applying KPI Pattern-3 to the given OEM, our approach determined two events that change the ownership of goods and follows an agreement, namely delivery and goods receipt. This means that there are two concrete instances of KPI Pattern-3 in the given OEM. In accordance with that, our approach derived two KPIs using KPI Pattern-3, namely *Average delivery duration* and *Average receive duration.* By analyzing the fifth chart, one can note that the average delivery duration KPI for the two organizations are nearly the same; however, the average duration of goods receipts for Organization A is different than for Organization B.

Above, we presented the results that we obtained by applying our approach on a real-life case study. The results indicate that our approach automatically derives and captures relevant KPIs for organizations. Moreover, we showed that using the derived KPIs relevant insights can be provided to organizations.

7 Related Work

Much work has been conducted on measuring, tracking, and evaluating the performance of business processes using KPIs. Moreover, various approaches have been proposed for defining, modeling, and customizing KPIs. In this section, we list some of the approaches, which touch upon the aspects that we are interested in: deriving KPIs and tailoring them to organizations.

Jansen-Vullers et al. present a set of KPIs [6] for each dimension of the Devil's Quadrangle [3] to evaluate the effectiveness of a redesign. Despite the fact that some of the KPIs are formalized, most of them are manually defined in natural language. However, on the one hand, it is required to obtain the meaning of each KPI to determine whether they are relevant for numerous organizations. On the other hand, organizations have to implement custom solutions to calculate the values of these KPIs. Therefore, a significant effort is required to customize the provided KPIs for organizations.

A framework for modeling KPIs within organizations has been presented by Popova and Sharpanskykh [11]. In this framework, the authors formulate KPIs and relationships between them. However, all these formalizations are in natural language and specified manually. In addition, while defining KPIs the meanings of the terms have not been taken into account; only the values assigned to the set of attributes of the KPIs have been taken into account. For example, PI20-efficiency of the planning process is defined as a KPI, but it is not explained when this KPI can be derived and what is the meaning of efficiency in this KPI.

Pedrinaci and Domingue present a Metrics Ontology [9] that is aimed at specifying domain-independent KPIs and computing their values. Subsequently, by using semantic technologies op top of that Metrics Ontology to support automated reasoning for computing KPIs, Pedrinaci et al. present a tool called SENTINEL [10]. On the one hand, the authors manually define a set of KPIs. On the other hand, it is not clear whether there are explicit relations between the computed KPIs and the activities in business processes. Unfortunately, it requires technical knowledge of ontological query building to define KPIs by end-users. As a result, it becomes a highly technical challenge to specify when and how these KPIs can be derived for organizations. Unlike that, in our approach, we aim to meet this challenge by defining KPI derivation patterns to automatically derive KPIs from an OEM such that it requires minimal technical knowledge from end users to reason and model.

To enable one to define KPIs while modeling business processes, del Río-Ortega et al. present a meta-model [5]. With this meta-model one can define various KPIs; however, the KPIs need to be manually defined and associated with the model elements inside process models for each organization. Furthermore, the meta-model does not consider the meaning of the activities in the process models. Therefore, the KPIs defined for an organization using this meta-model cannot be applied in another organization to derive the KPIs automatically, i.e., the meta-model is a reference model for identifying the KPIs in the scope of an organization.

For comparing organizations using cross-organizational process mining, Buijs et al. present an approach [4]. Within this approach, a set of metrics used to compare organizations. For example, precision, cost-based fitness, and behavioral appropriateness. However, these metrics do not serve the same goals with KPIs as they are not directly related with the business processes performed by organizations.

The aforementioned approaches define KPIs either in natural language or in a way such that is not possible to automatically derive and tailor them to organizations. Although the approach [4] automatically derives a limited number of KPIs, only some of them are directly related with business processes.

8 Conclusion and Future Work

In this paper, we presented our Automated KPI Derivation Approach aimed at automatically deriving tailored KPIs for organizations from *Ontological Enterprise Models* (OEMs). An OEM gives the opportunity to know which aspect of an organization a modeler tries to model. In other words, we can gain the meaning of each modeling element in an OEM in terms of the business of an organization using the characteristics specific to each modeling element. For example, with an agreement element we can obtain the meaning of an agreement between a customer and an organization in the real-world phenomena of that organization. Moreover, different from other approaches, by automatically obtaining the meanings of the elements in an OEM our approach can deduce the knowledge for automatically deriving KPIs that are related to the modeled aspect of an organization and relevant for the organization.

We believe that our approach proposes a better way than current approaches by having the following advantages to meet the challenges of the process of deriving tailored KPIs such as time-consuming, costly, and being error-proneness. ① By obtaining the meanings of the elements in an OEM, our approach automatically copes with ambiguity in the terms in business processes and reduces the complexity at the process of deriving tailored KPIs. ② Moreover, by automatically obtaining the exact meaning of the terms in business processes in various organizations that are required for customizing KPIs our approach lowers the efforts of software vendors and organizations on customizing KPIs. ③ In addition, we provide a uniform view for KPIs such that the KPIs that are derived from the same KPI derivation pattern will be automatically named and visualized accordingly. Although the number of KPI derivation patterns that are currently supported in our approach is limited, we are still analyzing the KPIs related to various concepts in organizations (for example, finance and customer relationship management) to define new patterns for deriving more KPIs.

We have validated our approach by means of two steps. Firstly, as a proof-of-concept, we implemented our approach, which shows its feasibility. Secondly, we applied our approach on a case study and derived KPIs for two selected organizations whose business processes are modeled in the OEM that we illustrated in this paper, and discussed how these organizations can be compared using these derived KPIs. Other software vendors, who focus on deriving KPIs automatically and work in the ERP domain, can apply our approach. To this end, software vendors need to make the required inputs available for our approach. First, they need to model the business processes that they offered in their products in the form of an OEM using the NEXT OEM Language. Second, they need to obtain the run-time data for that OEM and define references from the run-time data to the corresponding elements in the OEM.

In future work, we will focus on the aspects that make KPIs relevant for organizations. For instance, a set of KPIs in the HRM area that are relevant for an organization might not be relevant for another that has outsourced the business processes for that area. This means that there can be factors that determine the relevance of KPIs for organizations, such domain, location, targeted customer audience, or number of employees. Therefore, we will focus on identifying the factors that can affect the relevance of KPIs for organizations. Moreover, we can do benchmarking between organizations using KPIs, so that they can see how they perform in comparison to each other.

Acknowledgments. 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. The NEXT Platform is developed and maintained by AFAS Software.

References

[1] van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Rozinat, A., Verbeek, H.M.W., Weijters, T.: ProM: The process mining toolkit. In: Proceedings of the Business Process Management Demonstration Track (BPMDemos 2009) (2009)

[2] Aksu, Ü., Schunselaar, D.M.M., Reijers, H.A.: A cross-organizational process mining framework for obtaining insights from software products: Accurate comparison challenges. In: 18th IEEE Conference on Business Informatics, CBI 2016 (2016)

[3] Brand, N., van der Kolk, H.: Workflow analysis and design. Deventer: Kluwer Bedrijfswetenschappen (1995)

[4] Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Towards cross-organizational process mining in collections of process models and their executions. In: Business Process Management Workshops - BPM 2011 International Workshops (2011)

[5] del-Río-Ortega, A., Resinas, M., Cabanillas, C., Cortés, A.R.: On the definition and design-time analysis of process performance indicators. Inf. Syst. 38(4) (2013)

[6] Jansen-Vullers, M., Loosschilder, M., Kleingeld, P., Reijers, H.A.: Performance measures to evaluate the impact of best practices. In: Proceedings of Workshops and Doctoral Consortium of the 19th International Conference on Advanced Information Systems Engineering (BPMDS workshop) (2007)

[7] Liu, L., Özsu, M.T. (eds.): Encyclopedia of Database Systems. Springer US (2009)

[8] Parmenter, D.: Key performance indicators: developing, implementing, and using winning KPIs. John Wiley & Sons (2015)

[9] Pedrinaci, C., Domingue, J.: Ontology-based metrics computation for business process analysis. In: Proceedings of the 4th International Workshop on Semantic Business Process Management (2009)

[10] Pedrinaci, C., Lambert, D., Wetzstein, B., van Lessen, T., Cekov, L., Dimitrov, M.: SENTINEL: a semantic business process monitoring tool. In: Proceedings of the First International Workshop on Ontology-supported Business Intelligence, OBI 2008 (2008)

[11] Popova, V., Sharpanskykh, A.: Modeling organizational performance indicators. Inf. Syst. 35(4) (2010)

[12] van der Schuur, H., van de Ven, E., de Jong, R., Schunselaar, D.M.M., Reijers, H.A., Overeem, M., de Graaf, M., Jansen, S., Brinkkemper, S.: Next: Generating tailored erp applications from ontological enterprise models. In: The Practice of Enterprise Modeling Conference, PoEM 2017 (2017)

[13] Yin, R.K.: Case study research: Design and methods. Sage publications (2013)