

Identifying Variability in Process Performance Indicators

Bedilia Estrada-Torres^(✉), Adela del-Río-Ortega,
Manuel Resinas, and Antonio Ruiz-Cortés

Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Sevilla, Av. Reina Mercedes s/n, 41012 Seville, Spain
{iestrada,adeladelrio,resinas,aruz}@us.es

Abstract. The performance perspective of business processes is concerned with the definition of performance requirements usually specified as a set of Process Performance Indicators (PPIs). Like other business process perspectives such as control-flow or data, there are cases in which PPIs are subject to variability. However, although the modelling of business process variability (BPV) has evolved significantly, there are very few contributions addressing the variability in the performance perspective of business processes. Modelling PPI variants with tools and techniques non-suitable for variability may generate redundant models, thus making it difficult its maintenance and future adaptations, also increasing possibility of errors in its managing. In this paper we present different cases of PPI variability detected as result of the analysis of several processes where BPV is present. Based on an existent metamodel used for defining PPIs over BPs, we propose its formal extension that allows the definition of PPI variability according to the cases identified.

Keywords: Business process variability · Process performance indicators · Variability in PPIs

1 Introduction

A business process (BP) may vary according to its specific context [1,2], due to changes in original process requirements [3], by the evolution of its environment of application [4], to reflect new allocation of responsibilities, new strategic and business goals, or by changes in general inputs of the BP [5]. The modelling of business process variability (BPV) focuses on identifying variable and invariable parts of a BP (e.g., its control-flow, data or resources) with the aim of managing different versions of the same process together [6–8]. Managing BPV promotes reuse and reduce maintenance efforts and costs of change in BPs [9,10].

This work has received funding from the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2015-70560-R (BELI), P12-TIC-1867 (COPAS) and P10-TIC-5906 (THEOS)).

The performance perspective of BPs is concerned with the definition of performance requirements, usually as a set of Process Performance Indicators (PPIs) that address different dimensions like time, cost and quality [11]. PPIs provide valuable insights about the performance of processes and organizations, facilitate decision-making tasks and identify possible improvement areas [12]. Their management is part of the whole BP lifecycle, from the design and definition of PPIs together with BPs, to the configuration and implementation of both of them, the monitoring of PPIs after execution phase during which PPI values were gathered, and finally the evaluation of the values obtained [12].

Consequently, like other BP perspectives such as control-flow or data, there are cases in which PPIs are subject to variability. This variability can be related to variations that take place in other perspectives (e.g., if an activity measured by a PPI does not appear in a certain variant), but it can also be related to variations in PPIs themselves regardless of the other perspectives (e.g., the target value for a PPI in an incident management process may change depending on the criticality of the incident without this involving any changes in the control-flow).

Unfortunately, as far as we are concerned, there are no studies that deal with the modeling of variability in the performance perspective of BPs. This is undesirable because, like with other BP perspectives, the definition and modification of PPI variants can be a repetitive, laborious and error-prone task. In contrast, having an explicit model of the variability of PPIs together with the other perspectives of the BP helps to guarantee consistency and correctness across PPI variants and can reduce maintenance efforts and costs of change.

In this paper we analyse how variability affects the performance perspective of BPs from the definition of PPIs. To this end, processes to manage incidents in the *Andalusian Health Service (SAS)* and *SCOR processes* have been analysed to identify how PPIs change depending on the variability in the BP and by changes in the requirements for specifying its own attributes. As a result, we come up with several dimensions in which PPIs and their attributes (like measure definitions) can vary. Based on this analysis, we extend the PPINOT Metamodel [12], a metamodel for the definition of PPIs over BPs, to model the variability on PPIs together with the other perspectives of the BP. Furthermore, we define the syntactic validity of this variable PPIs model and we formalize how to obtain the PPI model for each business process variant (PV).

The remainder of this paper is structured as follows. Section 2 introduces background information about variability in BPs and PPIs. The motivating scenario of this approach is presented in Sect. 3. Section 4 identifies dimensions of change to explain how variability affects PPI definitions, and those are related to a real case in Sect. 5. Section 6 shows the PPINOT Metamodel and its extension to manage variability in PPIs. Finally, Sect. 7 draws conclusions and outlines our future work.

2 Related Work

This paper addresses three main areas: (i) the variability in business processes, (ii) PPIs, and (iii) the variability in performance indicators. Below we describe related work on those areas.

2.1 Variability in Business Process

Business processes may exist as a collection of different variants [9,13,14] that share a common base structure and some strategic and business goals. When this variability is not explicitly managed, each variation in the process is modelled as an independent process of each other. This ensures the representation of all information, but depending on the amount of PVs to be defined, a long amount of models could be generated, introducing redundancy and making future adaptations difficult. The lack of control over these multiple PVs usually causes each variant takes more time to be designed, configured and modified. It also may introduce errors from the definition of variants to the evaluation of its performance [2,6].

To solve this issue, many approaches to manage the variability in BPs have been proposed. Most of them focus on the *design and analysis* phase of the BP lifecycle [4], wherein new *Business Process Modeling Languages (BPML)* or expansion for existing ones are proposed. These languages are aimed at avoiding redundancy through *reuse* of some parts of BP flow, identifying common parts of the flow and modeling a BP block only once [15]. This favors reducing duplicated information, thus decreasing design-time and maintenance-time of models [16]. Provop [1,17], C-EPC [18], C-iEPC [8] and BPFM [14] are some examples of proposals for managing variability.

Although, most related work about BPV is focused on variability of control-flow [1,17,19], there are proposals that address variability in data or resources [8,19]. However, as far as we are concerned, there are no studies on the variability in the performance perspective of BPs.

2.2 Process Performance Indicators (PPIs)

A *Process Performance Indicator (PPI)* can be defined as a quantifiable metric focused on evaluating the performance of a BP in terms of efficiency and effectiveness. They are measured directly by data generated within the process flow and are used for process controlling and continuous optimization [20]. These PPIs are managed together with the BP lifecycle [12]. In design and analysis phase, PPIs are modelled together with the BP. During the configuration phase, the instrumentation of the processes that are necessary to take the measures must be defined. During BP enactment, PPIs should be monitored taking into account the PPI values obtained from execution data. Finally, during the evaluation, monitoring information obtained in the enactment phase will help to identify correlations and predict future behaviors.

Different approaches have been proposed for measuring the performance of BPs using PPIs. Some of them include domain-specific languages, metamodels, rules, techniques and notations, to address different phases in the PPI lifecycle. *MetricM* [21] and *PPINOT* [12] are examples of these approaches.

Regardless of the notation used, a *PPI* is defined by means of a set of attributes that specifies relevant aspects to establish what and how to measure [12,22]. The most relevant and recurrent attributes, besides the attributes required to identify the PPI (name, id, description, etc.) are: a *Process* in which the *PPI* is defined, a set of *Goals* indicating the relevance of the PPI, a *Measure definition* that specifies how to calculate the *PPI*, *Target* values to be reached indicating the consecution of the previously defined goals, the *Scope* that is used to define the subset of instances to be considered to calculate the *PPI* value, and the *human resources* involved.

2.3 Variability in Performance Indicators

As far as we know, there are no approaches addressing the variability of PPIs. However, in [23] some concepts about variability and indicators are treated. In this paper the variability is managed using design patterns (composite pattern), defining entities to gather goals, categories, indicators for individual, units for sets of indicators or single indicators, associated to different persons or academical units. The model proposed is based on [24], where each entity is modeled by decorator patterns, to add many features and functions dynamically.

However, unlike in our proposal, the authors do not deal with the traceability between PPIs and BPs and how they can vary together. In addition, they do not detail how the variability model is configured for a specific variant. Finally, the variability in KPIs are described just at a high level of abstraction and it is hardly applicable in different scenarios.

3 Motivating Scenario

The *Supply Chain Operation Reference* model (SCOR) [25] is a process reference model for supply chain management. It enables users to address, improve, and communicate supply chain management practices within and between all interested parties in the enterprise. We focus on two elements of its structure: *processes* and *measure definitions* (called *metrics* in *SCOR*).

SCOR processes identify a set of unique activities within a supply chain. These activities are described at a high level of abstraction because implementation of processes requires internal and specific definitions of activities of each organization, which are out of the scope of SCOR. *SCOR measure definitions* are defined as a standard for measuring the process performance.

Due to its structure and the definition of its components, *SCOR processes* have variability. *Deliver process (D)*, for instance, is defined as the processes associated with performing customer-facing order management and order fulfillment activities. It can be implemented in four different ways depending on the selected

strategy: *D1-Stocked Product*, *D2-Make to Order Product*, *D3-Engineering to Order Product* and *D4-Retail Product*. Each of them is a PV of Deliver. An excerpt of those PVs are shown in Fig. 1. They have a set of common tasks among them, but also have differences depending on the strategy selected. PV-2 varies in 13% with regard to activities defined for PV-1 (PV-1 has 15 activities), PV-3 and PV-4 differ in 33% and 100% respectively. For simplicity, we only focus on the three first PVs, because *D-4* is totally different from the other PVs.

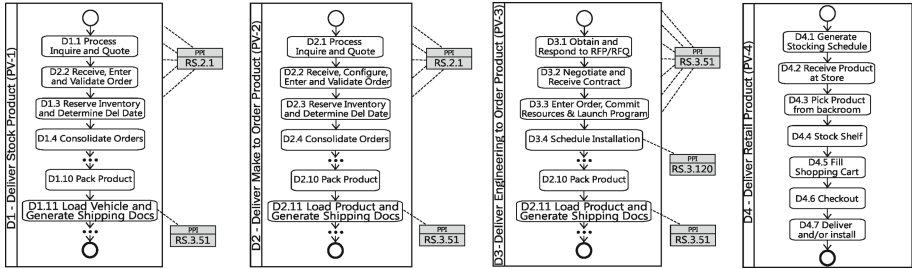


Fig. 1. Four variants of *Deliver Process*

Variability is also reflected in SCOR through its *measure definitions*, due (i) to their dependence on the BP flow in which they are defined or (ii) by specific requirements of the measures defined for each variant. Measures like *RS.3.120 Schedule Installation Cycle Time* reflect the first case. The measure is defined only in a PV, because it is connected to the task *D3.4 Schedule Installation* that only appears in PV-3. The second case is manifested in measures that vary regarding the required components to calculate its value. For example, in PV-1 and PV-2 the *RS.2.1 Source Cycle Time* measure requires 5 different time values from 5 process tasks, while in PV-3 this measure requires 7 different time values.

Currently, although there are BPMLs that allow us to model BPV, there do not exist tools and techniques to model variability in PPIs. In SCOR, for example, Deliver process defines 100, 96 and 96 measures for PV-1, PV-2 and PV-3 respectively, and almost half of them are repeated for all or some PVs. If we want to model them, it would be necessary to model independently the PPIs of each variant, making it a laborious and time-consuming task. Furthermore, if in the future, a PPI changes, we must modify one by one each variant involved, which does not ensure the PPI integrity through all variants, because we could forget to make some changes. If these errors are not detected, they may be carried throughout the whole lifecycle process leading to new problems like monitoring poorly defined PPIs and collecting inaccurate information that will be used in decision-making, to name a few.

In summary, modeling the variability in PPIs brings similar advantages than modeling the variability in the other perspectives of the BPs. Consequently, PPIs should be defined by means of tools and techniques that allow us to represent

variability aspects in the BP performance perspective, taking into consideration all dimensions that affect their variability.

4 Variability in Definitions of PPIs

In order to identify variability in PPIs, we studied several BPV cases and analysed different models to represent PPIs. First, we modeled the SCOR processes with their PVs. Then, we selected those with more similar activities in the control-flow of their PVs: *Deliver* and *Make*. After, we modeled, compared and classified the measures defined for those PVs in the SCOR model. Finally, we compare all PPI attributes among PVs, to identify cases of variation on PPIs. Similar study was made for PPIs of the SAS processes. As a result, we identified two dimensions of change in PPI definitions, namely:

Dim-1: *A PPI varies depending on whether it is defined for all process variants or not.*

Dim-2: *A PPI varies depending on attributes required to define it, which may change depending on the variant in which it is defined.*

Suppose a BP family that has more than one PV. If a PPI is defined for all those PVs and all its attributes do not change, there is no variability. Instead, if a PPI is defined in only one or some of its PVs, regardless of whether their attributes change or not, we are representing the variability expressed by **Dim-1**.

In addition, a PPI, regardless of the behavior derived from *Dim-1*, may vary depending on the changes applied over the value of one or more of its attributes. In Sect. 2.2 we mentioned a set of attributes that conform a PPI, and here we list some cases where the PPI variability is reflected, considering that a PPI varies if at least one of the following attributes changes:

Target (*T*) changes when the target value to be reached changes. For example, the *Andalusian Health Service* defines a PPI for measuring the percentage of resolved incidents in a period of time and in which its target values depend on the priority established for the measured service. If priority is very high, the target value is very high (resolved incidents $\geq 95\%$); if priority is high, the target value changes (resolved incidents $\geq 90\%$) and if priority is normal, the target value also changes (resolved incidents $\geq 82,5\%$).

Scope (*S*) changes when the set of instances to be evaluated changes. For example, if we have one PV that applies during weekdays and another one that applies in weekends (e.g., due to limited availability of resources available on weekends), we might define two variants of the same PPI, one that evaluates instances that take place on weekdays, and another one that evaluates those that take place on weekends.

Human resources (*HR*) may change by two attributes: **responsible** and **informed**. For example, taking up the previous example, depending on the priority of an incident, the person responsible for the PPI or the person informed about its value might change, e.g., because high priority incidents are resolved by a different team.

Measure definition (M) is through which a PPI is calculated. In this case, there are two dimensions of change, one related to the measure definition itself and another one related to the relationship with the BP:

Dim-2.M1: *A measure definition maintains its structure, but may vary depending only on the business process elements to which it is connected.*

Dim-2.M2: *A measure definition changes its structure and may vary depending on the requirements of the process variant.*

Dim-2.M1 might occur when a PPI is connected to a task that is not available for all PV where the PPI is defined, or because the definition requirements change and the PPI is assigned to a different task depending on the PV where the PPI is defined. An application example of **Dim-2.M1** is the PPI defined over the SCOR measure *RS.3.51 - Load Product & Generate Shipping Documentation Cycle Time*, which is defined in the Deliver process over the task 11. In *PV-1* this PPI is computed over the task *D1.11 Load Vehicle & Generate Shipping Documents*, but in *PV-2* and *PV-3* this task is not available (See Fig. 1). For this reason the same PPI is defined over an equivalent task, (*D2.11, D3.11*) *Load Product & Generate Shipping Docs*.

Dim-2.M2 might occur when a PPI is defined in two PVs (or more) as the sum of some measures, but in *PV-1* needs to explicitly use a set of measures that differs from the set of measures defined for *PV-2*. An example is the *Source Cycle Time* measure definition described in Sect. 3.

5 PPI Variability in Two Case Studies

Considering the dimensions of change introduced in Sect. 4, we have analysed case studies to confirm that variability of PPIs is covered by the dimensions proposed. Tables 1 and 2 summarize and classify according to the dimensions proposed, the variability of two SCOR processes (*Deliver* and *Make*) and on the PPIs to manage incidents in the *SAS processes*, respectively.

Both tables include: a column indicating the *Dimension* of change; on its right, the \times mark and \checkmark mark indicate whether or not there is variability in that dimension. Numbers under marks indicate sub-dimensions fulfilled in each case. The next column *describes* the dimensions and the last one shows the *number of measures* detected according each pair of possible dimensions.

For *SCOR processes*, six scenarios were identified: the first and the most common, indicates that there is no variability in 69 measures; in the following five, variability is reflected in one dimension or in both. In the last case, variability is reflected in **Dim-2** by both sub-dimensions. In these processes, for **Dim-2**, only sub-dimensions of measures are considered ($M1$, $M2$), because *SCOR* does not specify attributes like target, scope or human resources, since these depend on specific requirements of each organization.

Instead, in our second example the variability of other PPI attributes is evidenced. Specifically, values of targets (T) or other attributes of the PPI change frequently depending on the priority of the incident (very high, high or normal)

Table 1. Classification of SCOR measures according to dimensions of change.

Dimension	Description	Total
Dim-1 ✗	- Measure is defined in all process variants	69
Dim-2 ✗	- Measure is defined in the same way in all process variants	
Dim-1 ✗	- Measure is defined in all process variants	8
Dim-2 ✓ ₁	- <i>Dim-2.M1</i> : The PPI is connected to different BPElements	
Dim-1 ✗	- Measure is defined in all process variants	4
Dim-2 ✓ ₂	- <i>Dim-2.M2</i> : The PPI is calculated using different values	
Dim-1 ✓	- Measure is defined in some process variants	8
Dim-2 ✓ ₁	- <i>Dim-2.M1</i> : The PPI is connected to different BPElements	
Dim-1 ✓	- Measure is defined in some process variants	24
Dim-2 ✗	- Measure is equal defined in all PV where the PPI appears	
Dim-1 ✗	- Measure is defined in all process variants	7
Dim-2 ✓	- <i>Dim-2.M1, Dim-2.M2</i> : The PPI is connected to different BP Elements and is calculated using different values	
Total of measures		120

Table 2. Classification of SAS PPIs according to dimensions of change.

Dimension	Description	Total
Dim-1 ✗	- Does not vary with regard to the PV	9
Dim-2 ✗	- (<i>Dim-2.T</i>) Target does not vary	
Dim-1 ✗	- Does not vary with regard to the PV	8
Dim-2 ✓	- (<i>Dim-2.T</i>) Target varies depending on the priority value	
Dim-1 ✗	- Does not vary with regard to the PV	1
Dim-2 ✓ _o	- (<i>Dim-2.T</i>) Target does not vary. Other attributes vary (priority)	
Total of PPIs		18

that is being handled by the process. Table 2 classifies those PPIs in accordance with our dimensions of change.

6 Defining Variability of PPIs in PPINOT

As mentioned in Sect. 2, there are no proposals that allow PPIs to be associated with more than one PV or with various types of measures. To overcome this problem, following the same approach that has been followed in other proposals focused on control-flow such as C-EPC, one can extend an existing model to define PPIs in order to support the dimensions of change identified in the previous section. In this paper, we extend the PPINOT Metamodel¹, which is a

¹

More details available at <http://www.isa.us.es/ppinot/>.

metamodel for the definition of PPIs first introduced in [12]. However, the same ideas can be applied to any other PPI metamodel.

Before introducing our proposal, we define and present a formal definition for the original PPINOT Metamodel. Next, on the base of those definitions, we built a set of definitions that introduces the dimensions of change (Sect. 4).

6.1 The PPINOT Metamodel

PPINOT has been developed on the basis of the *PPINOT Metamodel* [12], which is depicted in Fig. 2. The metamodel allows the definition of a performance model composed of a set of *PPIs*. A PPI is linked with a measure definition and the list of attributes described in Sect. 2.2 can be specified for each PPI. *PPINOT* allows the definition of a wide variety of measures, namely: *base measures*, which represent a single-instance measure that measures values of time, count, conditions or data; *aggregated measures*, which are defined by aggregating one of the base measures that measures several process instances; and *derived measures*, which represent either a single-instance or a multi-instance measure whose value is obtained by calculating a mathematical function over other measures. The traceability with a BP model is kept by means of *conditions* that link measures with the elements of a BP (i.e., activities, events, data objects). Figure 3 provides more details about elements the of the metamodel.

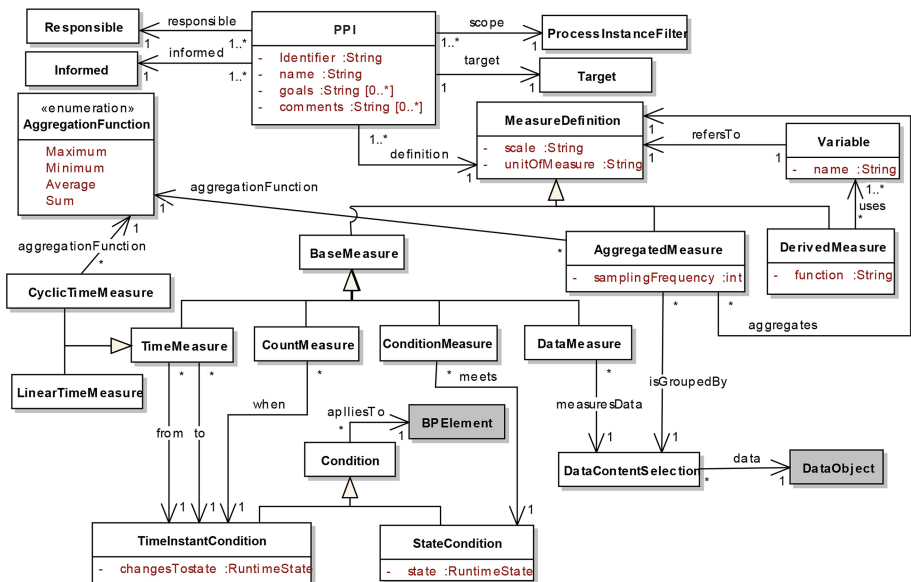


Fig. 2. Excerpt of the *PPINOT Metamodel*





PPI	Containing the measure that defines the PPI. A scope, a target, measures and human resources like responsible and informed, are attributes associated to it.		
	Base Measure: It measures each process separately.		
Measures	Aggregated Measure: It measures several process instances by aggregating them through an aggregation function: MIN, MAX, AVG, SUM.		
	Derived Measure: It performs a mathematical function over several process instances or process measures. <i>Uses</i> condition indicates measure values required in the derived function.		
Type of measures			
 Time measure: measures the duration between two instant conditions. If the time measure is taken between elements located within a loop, the measure may be <i>linear</i> or <i>cyclic</i> , depending on whether only the first occurrence is considered, or values are aggregated for more than one process instance. Time measure requires a pair of time instant conditions: <i>from</i> and <i>to</i> , to indicate the <i>start</i> and <i>end</i> point of the measuring.	 Count measure: indicates the number of times that certain instant condition is met. It requires a <i>time instant condition: when</i> , that indicates the point when something happens and should be measured (start, end and other state).	 Condition measure: Checks whether an activity, a pool, a data object or certain events are in a given state (ready, cancelled, completed, etc.) It is related with the BP Elements by a state condition: <i>meets</i> , that indicates the condition whose fulfillment is being measured.	 Data measure: obtains the value of certain part of a data object. <i>measureData</i> selects the part of the data object that is being measured and data establish the connection with the BP element.

Fig. 3. Description of *PPINOT* elements

In order to formally define a PPINOT performance model, we first need to formalise the concept of *Condition*, which is the link between the performance model and the other elements of the business process.

Definition 1 (Condition). *Let bp be a business process, \mathcal{A} be a not empty set of activities for bp , $\mathcal{S}_{\mathcal{A}}$ be a set of activity states of \mathcal{A} , \mathcal{D} be a finite set of data objects for the bp , $\mathcal{S}_{\mathcal{D}}$ be a finite set of data object states of \mathcal{D} , $\mathcal{A}_{\mathcal{D}}$ be a non-empty set of data object attributes of \mathcal{D} , \mathcal{E} be a non-empty set of events for the bp , $\mathcal{S}_{\mathcal{E}}$ be a set of event states of \mathcal{E} . $\mathcal{C}_{bp} = \mathcal{A} \times \mathcal{S}_{\mathcal{A}} \cup \mathcal{D} \times \mathcal{S}_{\mathcal{D}} \cup \mathcal{E} \times \mathcal{S}_{\mathcal{E}}$ is the set of all possible Conditions that can be defined over bp .*

For example, a condition $\mathcal{C} = (D1.1, active)$ represents the moment when activity *D1.1* becomes *active* in a given running instance.

Now, a PPINOT performance model can be defined as follows.

Definition 2 (PPINOT Performance Model). *Let bp be a business process, \mathcal{C}_{bp} be the set of all possible conditions defined over bp , \mathcal{S} be the set of scopes that can be defined for a PPI, \mathcal{T} be the set of targets that can be defined for a PPI, \mathcal{HR} be the set of human resources that can be related to the PPI, $\mathcal{F}_{agg} = \{MIN, MAX, AVG, SUM, \dots\}$ be a set of aggregation functions. A performance model PM over \mathcal{S} , \mathcal{T} , \mathcal{HR} , \mathcal{C}_{bp} and \mathcal{F}_{agg} is a tuple $PM = (P, M, L_P, L_M)$, where:*

- P is the set of process performance indicators of a bp ;
- $M = BM \cup AggM \cup DerM$ is a set of measure definitions, where:
 - $BM = TimeM \cup CountM \cup StateM \cup DataM$ is a finite set of base measures, where: *TimeM*, *CountM*, *StateM*, *DataM*, are the set of time, count, state condition and data measures defined by PM , respectively.
 - $AggM$ is the set of aggregated measures defined by PM ;
 - $DerM$ is the set of derived measures defined by PM ;
- $L_P = sco \cup tar \cup res \cup inf \cup mes$ is the set of links between a PPI $p \in P$ and its attributes, where:

- $sco \subseteq P \times \mathcal{S}$ is the set of scope links assigned to each PPI;
- $tar \subseteq P \times \mathcal{T}$ is the set of target links assigned to each PPI;
- $res \subseteq P \times \mathcal{HR}$ is the set of human resource links to indicate the person responsible of the PPI;
- $inf \subseteq P \times \mathcal{HR}$ is the set of human resource links to indicate the people informed about the PPI;
- $mes \subseteq P \times M$ is the set of links with the measure that defines each PPI;
- $L_M = cond \cup data \cup agg \cup cyclic \cup uses \cup derfun$ is the set of links between measure definitions and its attributes, where:
 - $cond = from \cup to \cup when \cup meets$ is a set of links among measures and conditions, where:
 - ◊ $from \subseteq TimeM \times \mathcal{C}$ is the set of links to time conditions, from;
 - ◊ $to \subseteq TimeM \times \mathcal{C}$ is the set of links to time conditions of to type;
 - ◊ $when \subseteq CountM \times \mathcal{C}$ is the set of links to time condition, when;
 - ◊ $meets \subseteq StateM \times \mathcal{C}$ is the set of links to state conditions, meets;
 - $data \subseteq DataM \times \mathcal{D} \times \mathcal{S}_{\mathcal{D}} \times \mathcal{A}_{\mathcal{D}}$ is the set of links to data conditions;
 - $cyclic \subseteq TimeM \times \mathcal{F}_{agg}$;
 - $agg \subseteq AggM \times (BM \cup DerM) \times \mathcal{F}_{agg}$ is the set of functions to measure a set of process instances when an aggregated measure is used;
 - $uses \subseteq DerM \times M \times \mathbb{N}$ is the set of links between a derived measure and the set of measures involved with it;
 - $derfun \subseteq DerM \times F$ is the set of links between derived measures and its functions, where: F is the set of all possible functions that could be resolved using derived measures;

Given a connector link $lm \in L_M$, $\Pi_M(lm)$ represents the measure involved in lm and $type_M(lm) \in T_M$, where $T_M \in \{from, to, when, meets, cyclic, data, agg, uses, derfun\}$ represents the type of the link. For instance, let $lm = (m_1, c_1) \in from$, $\Pi_M(lm) = m_1$ and $type_M(lm) = from$.

Similarly, given a connector link $lp \in L_P$, $\Pi_P(lp)$ represents the PPI where the attribute has been assigned and $type_P(lp) \in T_P \in \{sco, tar, res, inf, mes\}$ represents the type of the link. We also define $L_P[p, t]$ as the subset of L_P whose PPI is p and whose type is t , i.e., $L_P[p, t] = \{lp \in L_P \mid \Pi_P(lp) = p \wedge type_P(lp) = t\}$. Likewise, $L_M[m, t]$ is the subset of L_M whose measure definition is m and type is t , i.e., $L_M[m, t] = \{lm \in L_M \mid \Pi_M(lm) = m \wedge type_M(lm) = t\}$.

We can now define a syntactically correct PPINOT performance model PM . This is based on the metamodel specification introduced in [12] and displayed in Fig. 2. We mainly specify restrictions about relationships of measuring elements and define link constraints between PPIs and its attributes and between measures and its connectors.

Definition 3 (Syntactically correct PPINOT performance model). *Let $PM = (P, M, L_P, L_M)$ be a performance model, PM is syntactically correct if it fulfills the following requirements:*

- (1) *There is at least one PPI p in the performance model $|P| > 0$.*
- (2) *Each PPI attribute can only have exactly one single value linked to the PPI, except for the informed attribute. $\forall p \in P, t \in T_P \setminus \{inf\} (|L_P[p, t]| = 1)$*
- (3) *Measures have at most one link for each possible type of link in L_M except for uses: $\forall m \in M, t \in T_M \setminus \{uses\} (|L_M[m, t]| \leq 1)$*
- (4) *Depending on its type, measures have at least one element of their links:*
 - $\forall tm \in TimeM (\exists (tm, c_i) \in from \wedge \exists (tm, c_j) \in to)$
 - $\forall cm \in CountM (\exists (cm, c) \in when)$
 - $\forall sm \in StateM (\exists (sm, c) \in meets)$
 - $\forall dm \in DataM (\exists (dm, d, s, a) \in data)$
 - $\forall am \in AggM (\exists (am, m) \in agg)$
 - $\forall dm \in DerM (\exists (dm, f) \in derfun)$
 - $\forall dm \in DerM (\exists (d, m, x) \in uses)$
- (5) *A derived measure cannot be related to more than one measure with the same identifier: $\forall (d, m_i, x) \in uses \neg \exists (d, m_j, y) \in uses (x = y \wedge m_i \neq m_j)$*
- (6) *The identifiers used for a derived measure should be sequential, which is ensured if the highest identifier is equal to the number of uses links for such derived measure: $\forall (dm, m_i, x) \in uses (x \leq |L_M[dm, uses]|)$.*
- (7) *For all $(d, f) \in derfun$, $f \in F$ must be a function defined over the Cartesian product of the set of all possible values of the set of measures linked to d ($\{m \in M \mid (d, m, x) \in uses\}$), ordered according to x*

6.2 Extending the PPINOT Metamodel

The PPINOT performance model cannot model the variability identified in Sect. 4. To solve it, we introduce a *variable performance model* as an extension of a PPINOT performance model PM where PPIs, measures and connectors for linking measuring elements with bp elements or amongst them vary depending on the process variant to which they are applied. However, we need first to formally define what we understand as a process family and process variant.

Definition 4 (Process family). *A process family $PF = \{bp_1, \dots, bp_n\}$ is a set of business processes that share some common elements. Each $bp_i \in PF$ is called a process variant.*

This definition do not intend to be complete, but it just focuses on the elements that are relevant for variable performance models.

With this definition of process family, a variable performance model can be defined as follows.

Definition 5 (Variable performance model). *Let $PF = \{bp_1, \dots, bp_n\}$ be a process family, $\mathcal{PF} = \mathcal{P}(PF) \setminus \emptyset$ be the power set of PF without the empty set, and $\mathcal{C}_{PF} = \mathcal{C}_{bp_1} \cup \dots \cup \mathcal{C}_{bp_n}$ be the set of possible conditions defined over any process in the process family, a variable performance model is a tuple $PM^V = (P, M, L_P, L_M, P^V, L_P^V, L_M^V)$, where:*

- P, M, L_P, L_M refer to elements of a performance model defined over \mathcal{C}_{PF} .
- $P^V : P \rightarrow \mathcal{PF}$ defines the process variants to which each PPI applies.
- $L_P^V : L_P \rightarrow \mathcal{PF}$ defines the process variants to which each link between a PPI and its attributes applies.
- $L_M^V : L_M \rightarrow \mathcal{PF}$ defines the process variants to which each link between measures or between a measure and a process element applies.

Functions P^V , L_P^V and L_M^V introduce the modelling of the variability dimensions described in Sect. 4 as follows:

- P^V allows expressing **Dim-1** by providing a mechanism to specify which are the process variants to which a PPI applies.
- L_P^V allows expressing **Dim-2** by providing a mechanism to specify which are the process variants to which the alternative attributes for a PPI apply. This includes target, scope, human resources and measure definition, which are the links included in L_P
- L_M^V allows expressing **Dim-2.M1** and **Dim-2.M2** by providing a mechanism to specify which are the process variants to which the links between measure definitions and process elements (**Dim-2.M1**) or to which a certain structure of a measure definition (**Dim-2.M2**) apply. The former includes *cond* and *data* links, whereas the latter includes *cyclic*, *agg*, *uses* and *derfun* links.

Note that these variability functions can also be defined intensionally, i.e., by defining properties that all process variants to which a certain model element apply must fulfill (e.g., the presence of a certain activity in the variant).

A function that represents the process variants to which each measure applies (M^V) is not necessary because it can be derived from the variability functions of the PPIs (L_P^V) and measures (L_M^V) linked to it as follows:

$$M^V(m) = \bigcup_{(p_i, m) \in mes} L_P^V(p_i, m) \cup \bigcup_{(m_i, m) \in agg} L_M^V(m_i, m) \cup \bigcup_{(d_i, m, x) \in uses} L_M^V(d_i, m, x)$$

Based on these definitions, the concept of a syntactically correct variable performance model can be defined. In short, a syntactically correct variable performance model adds the necessary requirements to PM^V that ensure that each process variant has a syntactically correct performance model.

Definition 6 (Syntactically correct variable performance model). *Let PF be a process family, $\mathcal{PF} = \mathcal{P}(PF) \setminus \emptyset$ the power set of PF , $\mathcal{C}_{PF} = \mathcal{C}_{bp_1} \cup \dots \cup \mathcal{C}_{bp_n}$ be the set of possible conditions defined over any process in the process family, $PM^V = (P, M, L_P, L_M, P^V, L_P^V, L_M^V)$ is syntactically correct if it fulfills the following requirements:*

- (1) *There is at least one PPI for each process variant: $\forall bp_i \in PF (\exists p_i \in P (bp_i \in P^V(p_i)))$*

- (2) Each PPI attribute can only have exactly one single value linked to a PPI p in each variant in which the PPI applies $P^V(p)$, except for the informed attribute: $\forall p \in P, t \in T_P \setminus \{inf\} (\bigcup_{lp \in L_P[p,t]} L_P^V(lp) = P^V(p) \wedge \forall lp_i, lp_j \in L_P[p,t] (lp_i \neq lp_j \Rightarrow L_P^V(lp_i) \cap L_P^V(lp_j) = \emptyset)$
- (3) Measures have at most one link for each possible type of link in L_M except for uses in each variant: $\forall m \in M, t \in T_M \setminus \{uses\} (\forall lm_i, lm_j \in L_M[m,t] (lm_i \neq lm_j \Rightarrow L_M^V(lm_i) \cap L_M^V(lm_j) = \emptyset))$
- (4) Depending on its type, measures require at least one element of their links in each variant:
- $\forall tm \in TimeM (\bigcup_{lm \in L_M[tm,from]} L_M^V(lm) = M^V(m) \wedge \bigcup_{lm \in L_M[tm,to]} L_M^V(lm) = M^V(m))$
 - $\forall cm \in CountM (\bigcup_{lm \in L_M[cm,when]} L_M^V(lm) = M^V(m))$
 - $\forall sm \in StateM (\bigcup_{lm \in L_M[sm,meets]} L_M^V(lm) = M^V(m))$
 - $\forall dm \in DataM (\bigcup_{lm \in L_M[dm,data]} L_M^V(lm) = M^V(m))$
 - $\forall am \in AggM (\bigcup_{lm \in L_M[am,agg]} L_M^V(lm) = M^V(m))$
 - $\forall dm \in DerM (\bigcup_{lm \in L_M[dm,derfun]} L_M^V(lm) = M^V(m))$
 - $\forall dm \in DerM (\bigcup_{lm \in L_M[dm,uses]} L_M^V(lm) = M^V(m))$
- (5) Measures must not be applied to variants that do not contain the elements of the process they are linked to: $\forall (m, c) \in cond (\forall bp_i \in L_M^V(m, c) (c \in \mathcal{C}_{bp_i}))$ and $\forall (m, d, s, a) \in data (\forall bp_i \in L_M^V(m, d, s, a) ((d, s) \in \mathcal{C}_{bp_i}))$
- (6) A derived measure cannot be related in each variant to more than one measure with the same identifier, which means that if they have the same identifier, the intersection of their variants must be empty: $\forall (d, m_i, x) \in uses \neg \exists (d, m_j, y) \in uses (x = y \wedge m_i \neq m_j \wedge L_M^V(d, m_i, x) \cap L_M^V(d, m_j, y) \neq \emptyset)$
- (7) The identifiers used for a derived measure in each variant must be sequential: $\forall (d, m_i, x) \in uses (\forall bp_i \in L_M^V(d, m_i, x) (x \leq |\{u \in L_M[d, uses] | L_M^V(u) = bp_i\}|))$.
- (8) For all $(d, fn) \in derfun$, $fn \in F$ must be a function defined over the Cartesian product of the set of all possible values of the set of measures linked to d that apply for each variant bp_i to which (d, fn) applies $(\{m \in M | (d, m, x) \in uses \wedge bp_i \in L_M^V(d, m, x)\})$, ordered according to x .

Finally, using these definitions, it is easy to obtain a performance model $PM_i = (P_i, M_i, L_{P_i}, L_{M_i})$ for a specific process variant bp_i . For P_i , L_{P_i} and L_{M_i} , it just includes the elements of the variable performance model that apply to the process variant at hand. For M_i , it includes the measures that are used in the links of L_{M_i} . This can be formalised as follows.

Definition 7 (Performance model of a process variant). Let $PF = \{bp_1, \dots, bp_n\}$ be a process family, and $PM^V = (P, M, L_P, L_M, P^V, L_P^V, L_M^V)$ be a variable performance model of PF , the performance model of a variant bp_i of the process family is a tuple $PM_i = (P_i, M_i, L_{P_i}, L_{M_i})$, where:

- $P_i = \{p \in P | bp_i \in P^V(p)\}$
- $L_{P_i} = \{lp \in L_P | bp_i \in L_P^V(lp)\}$

- $L_{M_i} = \{lm \in L_M \mid bp_i \in L_M^V(lm)\}$
- $M_i = \{m \in M \mid \exists lm \in L_{M_i} (\Pi_M(lm) = m)\}$

A measure that varies for three PVs was modeled using the formal definition of PPINOT and we have also modeled three PVs of the *Deliver* process to graphically represent the dimensions of change (see <http://www.isa.us.es/ppinot/variability-bpm2016/>). To represent the elements of the metamodel in a visual way, we have used an extension of the graphical notation of PPINOT to specify the variants of each PPI together with a C-EPC model of the PVs.

7 Conclusions and Future Work

From this paper, we can conclude that the performance perspective of BPs is subject to variation like other perspectives and, as such, it is convenient to develop models and tools that manage this variability, favor reuse and reduce design and maintenance time.

This conclusion is the result of an analysis of several BPV cases and different models to represent PPIs that have allowed us to identify two dimensions of change in the definition of PPIs and another two dimensions of change in the definition of measure definitions. Some of these dimensions (**Dim-2.M1**) are related to variations in other perspectives like control-flow, but other dimensions show that PPIs can also be subject of their own variations regardless of the other perspectives such as changes in the target value of the PPI. Furthermore, the cases that we have analyzed show that the variability of PPIs is quite common, affecting almost half of the PPIs defined in each case.

In addition, based on this analysis, we provide a model to extend the modelling of BPV to the performance perspective of BPs. To this end, we extend the PPINOT metamodel with the concept of variable performance model and formalize the requirements of a syntactically correct variable performance model that ensures that each PV has a syntactically correct performance model.

Our formal extension of the PPINOT metamodel is a first step to develop techniques and tools that facilitate the design and analysis of variability in PPIs, to ensure their correct definition and to reduce errors in the performance measurement.

As a direction for future work, we want to describe in detail and assess the graphical notation for the modelling of PPIs taking into account the BPV and all the PPI variability cases detected. To do this, we also need to develop tools that meet definitions and restrictions defined for PPI variability, and that will facilitate their complete managing until evaluation phase.

References

1. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and management of process variants. In: Handbook on Business Process Management 1. International Handbooks on Information Systems, pp. 237–255. Springer, Berlin Heidelberg (2010)

2. Reichert, M., Hallerbach, A., Bauer, T.: Lifecycle management of business process variants. In: *Handbook on Business Process Management 1*, pp. 251–278. Springer, Berlin Heidelberg (2015)
3. Hallerbach, A., Bauer, T., Reichert, M.: Guaranteeing soundness of configurable process variants in Provop. In: *2009 IEEE Conference on Commerce and Enterprise Computing*, pp. 98–105, July 2009
4. da Mota Silveira Neto, P.A., do Carmo Machado, I., McGregor, J.D., de Almeida, E.S., de Lemos Meira, S.R.: A systematic mapping study of software product linestesting. *Inf. Softw. Technol.* **53**(5), 407–423 (2011)
5. Milani, F., Dumas, M., Ahmed, N., Matulevicius, R.: Modelling families of business process variants: A decomposition driven method. *Inf. Syst.* **56**, 55–72 (2016)
6. Aiello, M., Bulanov, P., Groefsema, H.: Requirements and tools for variability management. In: *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, pp. 245–250 (July 2010)
7. Saidani, O., Nurcan, S.: Business process modeling: a multi-perspective approach integrating variability. In: Bider, I., Gaaloul, K., Krogstie, J., Nurcan, S., Proper, H.A., Schmidt, R., Soffer, P. (eds.) *BPMS 2014 and EMMSAD 2014*. LNBIP, vol. 175, pp. 169–183. Springer, Heidelberg (2014)
8. Rosa, M.L., Dumas, M., ter Hofstede, A.H., Mendling, J.: Configurable multi-perspective business process models. *Inf. Syst.* **36**(2), 313–340 (2011)
9. La Rosa, M., van der Aalst, W.M.P., Dumas, M., Milani, F.P.: Business process variability modeling: A survey. Report, ACM Digital Library (2013)
10. Torres, V., Zugal, S., Weber, B., Reichert, M., Ayora, C., Pelechano, V.: A qualitative comparison of approaches supporting business process variability. In: La Rosa, M., Soffer, P. (eds.) *Business Process Management Workshops*. LNBIP, vol. 132, pp. 560–572. Springer, Heidelberg (2012)
11. Lodhi, A., Koppen, V., Wind, S., Saake, G., Turowski, K.: Business process modeling language for performance evaluation. In: *47th Hawaii International Conference on System Sciences (HICSS)*, pp. 3768–3777 Jan 2014
12. del Río-Ortega, A., Resinas, M., Cabanillas, C., Ruiz-Cortés, A.: On the definition and design-time analysis of process performance indicators. *Inf. Syst.* **38**(4), 470–490 (2013)
13. Milani, F., Dumas, M., Matulevičius, R.: Identifying and classifying variations in business processes. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) *EMMSAD 2012 and BPMS 2012*. LNBIP, vol. 113, pp. 136–150. Springer, Heidelberg (2012)
14. Cognini, R., Corradini, F., Polini, A., Re, B.: Extending feature models to express variability in business process models. In: Persson, A., Stirna, J. (eds.) *CAiSE 2015 Workshops*. LNBIP, vol. 215, pp. 245–256. Springer, Heidelberg (2015)
15. Rolland, C., Nurcan, S.: Business process lines to deal with the variability. In: *43rd Hawaii International Conference on System Sciences (HICSS)*, pp. 1–10 (Jan 2010)
16. Machado, I., Bonifácio, R., Alves, V., Turnes, L., Machado, G.: Managing variability in business processes: An aspect-oriented approach. In: *Proceedings of the 2011 I Workshop on Early Aspects*. EA 11, pp. 25–30. ACM, New York, NY, USA (2011)
17. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. *J. Softw. Maintenance Evol. Res. Pract.* **22**(6–7), 519–546 (2010)
18. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modelling language. *Inf. Syst.* **32**(1), 1–23 (2007)

19. Razavian, M., Khosravi, R.: Modeling variability in business process models using UML. In: Fifth International Conference on Information Technology: New Generations, ITNG 2008, pp. 82–87 (April 2008)
20. del-Río-Ortega, A., Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: PPINOT tool suite: a performance management solution for process-oriented organisations. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSOC 2013. LNCS, vol. 8274, pp. 675–678. Springer, Heidelberg (2013)
21. Strecker, S., Frank, U., Heise, D., Kattenstroth, H.: MetricM: a modeling method in support of the reflective design and use of performance measurement systems. *Inf. Syst. e-Bus. Manage.* **10**(2), 241–276 (2011)
22. Popova, V., Sharpanskykh, A.: Modeling organizational performance indicators. *Inf. Syst.* **35**(4), 505–527 (2010)
23. Suhartono, D.: Variability model implementation on key performance indicator application. *Int. J. Innov. Manage. Technol.* **6**(1), 77–80 (2015)
24. Vianden, M., Lichter, H.: Variability model towards a metric specification process. In: Proceedings of the International Conference on Computer Science and Information Technology, pp. 76–79 (2011)
25. Apics, S.C.C.: Supply Chain Operations Reference Model: SCOR Version 11.0. Supply Chain Council APICS, CCOR, CPIM, CSCP, DCOR, SCOR, and SCOR-mark are all registered trademarks of APICS. All rights reserved (2015)