

VISUAL PPINOT: A Graphical Notation for Process Performance Indicators

Adela del-Río-Ortega · Manuel Resinas · Amador Durán ·
Beatriz Bernárdez · Antonio Ruiz-Cortés · Miguel Toro

Received: 28 September 2015 / Accepted: 8 March 2017 / Published online: 20 June 2017
© The Author(s) 2017. This article is an open access publication

Abstract Process performance indicators (PPIs) allow the quantitative evaluation of business processes, providing essential information for decision making. It is common practice today that business processes and PPIs are usually modelled separately using graphical notations for the former and natural language for the latter. This approach makes PPI definitions simple to read and write, but it hinders maintenance consistency between business processes and PPIs. It also requires their manual translation into lower-level implementation languages for their operationalisation, which is a time-consuming, error-prone task because of the ambiguities inherent to natural language definitions. In this article, VISUAL PPINOT, a graphical notation for defining PPIs together with business process

models, is presented. Its underlying formal metamodel allows the automated processing of PPIs. Furthermore, it improves current state-of-the-art proposals in terms of expressiveness and in terms of providing an explicit visualisation of the link between PPIs and business processes, which avoids inconsistencies and promotes their co-evolution. The reference implementation, developed as a complete tool suite, has allowed its validation in a multiple-case study, in which five dimensions of VISUAL PPINOT were studied: expressiveness, precision, automation, understandability, and traceability.

Keywords Business process management · Business process modelling · Process performance measurement · Key performance indicators · Process performance indicators · Graphical notation · Visual PPINOT

Accepted after three revisions by Prof. Dr. Becker.

Electronic supplementary material The online version of this article (doi:[10.1007/s12599-017-0483-3](https://doi.org/10.1007/s12599-017-0483-3)) contains supplementary material, which is available to authorized users.

A. del-Río-Ortega (✉) · M. Resinas · A. Durán · B. Bernárdez ·
A. Ruiz-Cortés · M. Toro
Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Sevilla, Sevilla, Spain
e-mail: adeladelrio@us.es

M. Resinas
e-mail: resinas@us.es

A. Durán
e-mail: amador@us.es

B. Bernárdez
e-mail: beat@us.es

A. Ruiz-Cortés
e-mail: aruiz@us.es

M. Toro
e-mail: migueltoro@us.es

1 Introduction

Collecting and analysing process-related key performance indicators (KPIs) are the first prerequisites for holistic process management and form the basis for consistent and continuous process optimisation (Kronz 2006). These process-related KPIs are also known as *process performance indicators* (PPIs) and are a key asset in evaluating the performance of business processes (Andrikopoulos et al. 2008). PPIs are quantifiable metrics that allow an evaluation of the efficiency and effectiveness of business processes. They can be measured directly by data that are generated within the process flow and are aimed at process control and continuous optimisation (Chase et al. 2011).

However, PPI management is not only restricted to the evaluation phase of the business process management (BPM) lifecycle, but also includes a number of steps that

must be carried out throughout the whole lifecycle (Kronz 2006). i.e., PPIs need to be defined, the corresponding business processes must be instrumented, PPI values have to be computed. They can be monitored and analysed using techniques such as business activity monitoring (BAM) (Dresner 2003), business process intelligence (BPI) (Grigori et al. 2004), or process mining (van der Aalst et al. 2003, 2010), and finally, a PPI redefinition can be required in case of the evolution of either the associated business processes or the PPIs themselves.

Today it is common practice that process-oriented organisations usually define PPIs in natural language (Wetzstein et al. 2008). However, although PPI definitions in natural language are easy to read and write, they present some problems. First, since business processes are usually expressed in graphical notations (Kettinger and Teng 1997), the use of natural language could lead to serious consistency issues when, for instance, business processes evolve but PPI definitions are not consequently updated and become obsolete. Another major problem is the lack of automated processing, i.e., PPIs need to be redefined in a language amenable to automation in later stages of their lifecycle. This situation has two additional drawbacks. On the one hand, it takes time and resources, which increases the cost of deploying a performance management solution in the organisation and limits PPI evolution. On the other hand, PPI evolution may introduce errors because the gap between natural language and implementation languages is significant (Wetzstein et al. 2008; van der Aa et al. 2016). Furthermore, ambiguities introduced by natural language have to be manually detected and removed to automatically compute PPIs. This is a particularly error-prone task because the people implementing PPIs do not usually share the same context as the people who define them since – due to the nature of their work – the former are usually closer to technology, whereas the latter are closer to management.

The automated processing problem can be alleviated if the organisation uses a process-aware information system (PAIS) that supports the definition of PPIs, as many business process management systems (BPMSs) do. In this case, PPIs can be precisely defined using the mechanisms provided by the PAIS. However, the definition is platform-specific, i.e., it cannot be exported to other platforms, which is something desirable as shown by the current interest in BPMN and other standards. Furthermore, it is common that organisations use more than one information system, and being platform-specific prevents the definition of end-to-end PPIs. Finally, most PAIS define PPIs using a predefined set of application-specific forms that are not intended to provide an overall and customised view of the PPIs defined for a given process. An analysis of several PAIS is detailed in Sect. 9.3.

Finally, from an academic perspective, a number of research proposals for the definition of PPIs have been presented (Castellanos et al. 2005; Popova and Sharpan-skykh 2010; Saldivar et al. 2016; Pedrinaci et al. 2008; Wetzstein et al. 2008; Momm et al. 2007; Costello and Molloy 2009; González et al. 2009; Friedenstab et al. 2012; Delgado et al. 2014), but all of them fall short both of expressiveness to define most PPIs that can be found in real scenarios, and of visually representing the links between PPIs and business process models (see Sect. 9.1 for more details).

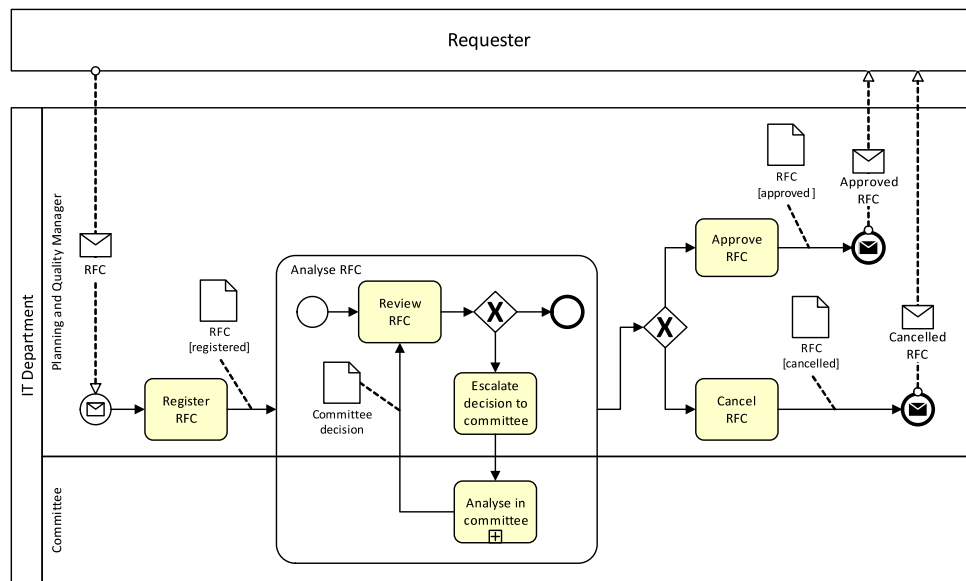
The goal of the presented research is to provide a mechanism to define PPIs that solves the aforementioned problems. To this end, we present VISUAL PPINOT, a graphical notation for PPI definition that is designed to be used together with business process models and is aimed at facilitating and automating PPI management. This is mainly achieved by means of the following features. First, VISUAL PPINOT is based on the PPINOT metamodel (del Río-Ortega et al. 2013), which provides a precise and unambiguous definition of PPIs, thus allowing their automated processing in the different activities of the lifecycle. Second, VISUAL PPINOT provides traceability by design between PPIs and business processes because PPIs must be explicitly connected to business process elements, thus avoiding inconsistencies and promoting their co-evolution. Finally, VISUAL PPINOT enables a definition of PPIs that is independent of the platforms used to support the PPIs in the business process lifecycle, which reduces vendor lock-in and allows definitions of PPIs encompassing several information systems. In addition, in comparison with other research proposals, VISUAL PPINOT improves them in terms of expressiveness and in terms of providing an explicit visualisation of the link between PPIs and business processes.

VISUAL PPINOT has been validated in two ways. On the one hand, the features of VISUAL PPINOT have enabled the development of software that supports the management of PPIs. The result is the PPINOT TOOL SUITE, a set of PPI management tools¹ for designing (del Río-Ortega et al. 2016), analysing (del Río-Ortega et al. 2013), computing (del-Río-Ortega et al. 2013a), and visualising PPIs. On the other hand, the usefulness of VISUAL PPINOT has been validated through a multiple-case study with three industrial cases and one academic one, in which five dimensions of VISUAL PPINOT were studied: expressiveness, precision, automation, understandability, and traceability.

The remainder of this article is organised as follows. In Sect. 2, a real case scenario that motivated our research work is presented. Section 3 describes our research question, which is followed by the research method that

¹ PPINOT TOOL SUITE is available at <http://www.isa.us.es/ppinot>.

Fig. 1 Request for change (RFC) management process (simplified version)



answers it. A brief introduction of the PPINOT metamodel is provided in Sect. 4. In Sect. 5, the notation and semantics of VISUAL PPINOT are described. Information regarding the notation design rationale is provided in Sect. 6. Some details about PPINOT TOOL SUITE are given in Sect. 7. In Sect. 8, we present the evaluation of our approach. In Sect. 9, the related work is discussed. Finally, Sect. 10 draws the conclusions from this research and outlines our future work.

2 Motivating Scenario

This section introduces a real scenario that motivated our research and where VISUAL PPINOT was applied. It deals with the management of PPIs in the context of the Request for Change (RFC) management process in the Information Technology (IT) Department of the Andalusian Health Service. The BPMN diagram in Fig. 1 describes a simplified version of this process.

The process starts when a *requester* submits an RFC. Then, the *planning and quality manager* registers the RFC and analyses it to make a decision. According to several factors such as availability of resources or the requirements affected by the requested changes, the manager either approves, cancels, or escalates the RFC to a committee for further analysis. The RFC document, represented as a BPMN *data object*, passes through several states such as *registered*, *cancelled*, or *approved*. The RFC document also contains information relevant for the process such as the *project* and the *information systems* affected by the RFC, the *type of change* requested (i.e., *adaptive*, *corrective*, or *perfective*), and the RFC *priority*.

The IT department also possessed a set of PPIs associated with the RFC management process. They were defined in a natural language and collected in tables. A simplified and refined version of these is shown in Table 1. To be computed, these PPIs needed to be translated to a machine-readable language. In this particular scenario, they were usually manually translated into SQL queries to gather the required information stored in different databases to compute their values. This required time and effort from a number of resources and led in many cases to wrong PPI values mainly due to both: misinterpretation of the original definitions or a lack of information in them. A derived drawback was the manual endeavour required whenever one of the two types of asset (business processes or PPIs) evolved and the other had to be properly updated, which frequently resulted in inconsistencies.

This scenario, which makes evident the problems mentioned above, will serve to illustrate our approach in the following sections.

3 Research Question and Methods

Taking into consideration all the information presented in the previous sections, we formulated the following research question:

How should PPIs be defined to improve the automated support for the PPI management lifecycle?

To address this research question, we followed design science principles as suggested by Hevner et al. (2004). In particular, we applied the design science research methodology (DSRM) (Peffer et al. 2007) as follows:

Table 1 PPIs defined for the request for change (RFC) management process

ID	Description	Target value	Scope
PPI1	Percentage of approved RFCs	95%	Weekly
PPI2	Percentage of cancelled RFCs	5%	Weekly
PPI3	Average time of committee decision	1 working day	Weekly
PPI4	Percentage of corrective approved RFCs	95%	Weekly
PPI5	Percentage of perfective and adaptive approved RFCs	5%	Weekly
PPI6	Percentage of RFC analysis time	50%	Weekly
PPI7	Average time of RFC analysis	2 working days	Weekly
PPI8	Number of RFCs under analysis	2 RFCs	Weekly
PPI9	Number of RFCs per type of change	20 for corrective RFCs 30 for adaptive RFCs 20 for perfective RFCs	Monthly
PPI10	Number of RFCs per project	50 for project HR 60 for project Diraya 1 for project Pharma	Monthly
PPI11	Average RFC lifetime	3 working days	Monthly

Problem identification and motivation phase We approached this phase from two different angles. On the one hand, we carried out a systematic literature review to collect existing proposals related to our research question, i.e., PPI definition. On the other hand, we analysed several real scenarios in which PPIs were defined to understand their requirements and to identify points of improvement. The result of this phase has been partially described in del Río-Ortega et al. (2013), and the conclusion we reached is that current approaches to define PPIs involve either using natural language, or mechanisms specifically provided by PAIS, or research proposals. However, all these approaches fall short of providing an expressive and precise notation that is traceable to the business process and amenable to automated processing while, at the same time, all stakeholders can understand it. This conclusion is extensively discussed in Sects. 1 and 9 of this article.

Objective of the solution phase The objective defined in this phase was the development of a graphical notation for PPI definition that should be designed to be used together with the business process model and aimed at facilitating and automating PPI management. Furthermore, according to the results of the previous phase, the notation should be expressive, traceable to the business process, amenable to automation, precise, platform-independent, and comprehensible for all stakeholders.

Design and development phase This phase involved the design and development of two novel artefacts, namely, (1) an all-in-one graphical notation for a definition of PPIs that overcomes the identified problems, i.e., VISUAL PPINOT, and (2) the PPINOT TOOL SUITE, the tool to support such a definition and to automate parts of the PPI management lifecycle.

Demonstration phase This phase involved the development of a software prototype of VISUAL PPINOT and PPINOT

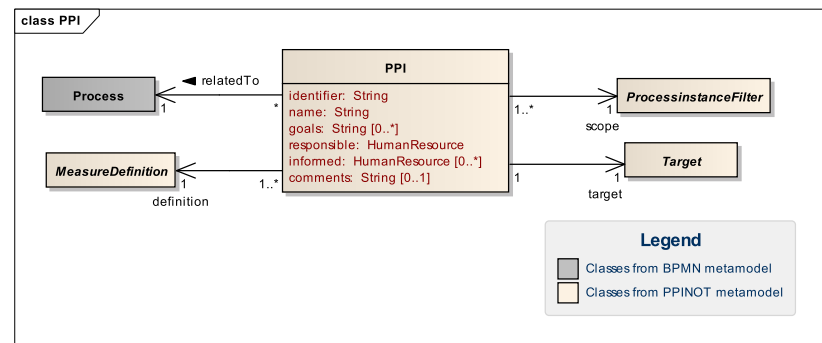
TOOL SUITE. This prototype effectively showed that the models defined in VISUAL PPINOT are amenable to automation and remove – or at least reduce drastically – the need to redefine PPIs to compute them. Furthermore, it also showed that the solution is platform-independent since it was used to compute PPIs from different sources.

Evaluation phase We carried out a multiple-case study with four different cases. This enabled the researchers to conduct an empirical evaluation of VISUAL PPINOT in terms of the five aforementioned dimensions: expressiveness, precision, automation, understandability and traceability. A fact that reinforces the positive feedback obtained from our evaluation is that we are currently working on a project whose goal is to deploy VISUAL PPINOT and PPINOT TOOL SUITE in production to define and compute the PPIs used in dozens of service level agreements (SLAs) the Andalusian Health Service has with its providers. Our approach was chosen from a number of possible solutions because of the ability provided by VISUAL PPINOT in defining PPIs at a higher level of abstraction and still to be able to automate their computation.

4 Background: Defining PPIs with PPINOT

The PPINOT metamodel was first introduced in del Río-Ortega et al. (2013) and serves as a foundation for VISUAL PPINOT. It was developed following an iterative and incremental process that included the following three steps (Brambilla et al. 2012): *modelling domain analysis*, which involved defining the metamodel's purpose and identifying the modelling concepts and their properties; *modelling language design*, which involved formalising these models; and *modelling language validation*, which involved

Fig. 2 PPINOT metamodel overview



instantiating the metamodel with more examples to validate its completeness and correctness. In particular, its purpose is to identify “how” PPIs are measured, i.e., how the information required for their computation can be obtained from business processes. The modelling concepts were identified on the basis of an exhaustive analysis of the literature and using examples from several scenarios, as suggested in López-Fernández et al. (2015). Furthermore, a set of competency questions derived from the Specific, Measurable, Attainable, Realistic, Time-sensitive (SMART) criteria (Shahin and Mahbod 2007) were also considered. The modelling languages used to formalise the models were the Unified Modeling Language (UML) and the Web Ontology Language (OWL) (del Río-Ortega et al. 2013). Finally, the validation involved its application to a number of real scenarios.

Figure 2 shows an overview of the PPINOT metamodel. A PPI is referred to by means of an *identifier*, described by means of its *name* and related to a process (*relatedTo*). It is also possible to establish the strategic or operational *goals* that the PPI is related to. PPIs are defined (*definition*) by means of a *MeasureDefinition*. In addition, a PPI has a *target* which must be reached, indicating the consecution of the previously defined goals, and a *scope* which specifies the subset of process instances that must be considered to compute the PPI value. The *responsible*, *accountable*, and *informed* attributes of the PPI can also be defined. Finally, other information can be added as *comments*.

As described in del Río-Ortega et al. (2013), the types of measure that can be used to define a PPI are classified according to two dimensions: number of process instances and nature of the measure. As a result, the different types of measures depicted in Fig. 3, and described below, are possible.²

Base measure It is a measure obtained directly from a single-process instance and does not require any other measure to be computed. It has four subclasses:

- **Time measure** It measures the duration between two time instants. It can be subdivided into *linear time measure* and *cyclic time measure*. This distinction makes sense if the *time measure* is calculated based on elements located within a loop.
- **Count measure** It measures the number of times something happens.
- **State condition measure** It is a Boolean value that measures the fulfilment of a certain condition in either running or finished process instances. This condition refers to the state of a business process element.
- **Data measure** It measures the value of a specific attribute of a data object.

The definition of this type of measures also includes certain conditions which are applied to the corresponding business process elements, as depicted in Fig. 3.

Aggregated measure It is defined by using an aggregation function such as sum or average over one of the previous measures defined over multi-process instances. Furthermore, when aggregating measures, it is possible to group them by the content of a certain data object.

Derived measure It is defined as a mathematical function over one or more measure definitions. There are two types of derived measures depending on whether they are defined over *single-* or *multi-*instance measures (*derived single-instance measure* and *derived multi-instance measure*, respectively).

5 VISUAL PPINOT: Notation and Semantics

VISUAL PPINOT, our graphical notation for the definition of PPIs over BPMN diagrams, has its foundations on the PPINOT metamodel introduced in the previous section. As BPMN itself, it is a graph-based notation in which each element has a set of attributes corresponding to its underlying metamodel element. The Online Appendix A (available online via <http://springerlink.com>) includes an overview of VISUAL PPINOT, inspired by the widely known *BPMN Poster* available at <http://www.bpmb.de>.

² The version presented here is an evolution of the metamodel presented in del Río-Ortega et al. (2013), resulting from its application and refinement. For a more detailed description of the PPINOT metamodel, we refer the reader to del Río-Ortega et al. (2013).

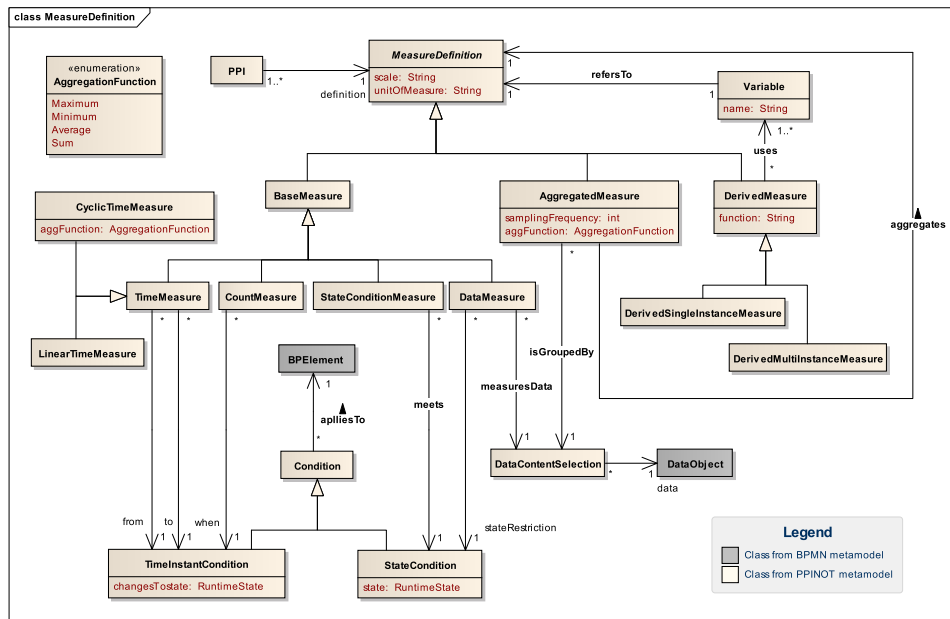


Fig. 3 Definition of measures in the PPINOT metamodel

5.1 PPIs and Measure Categories in VISUAL PPINOT

In VISUAL PPINOT, PPIs are depicted as a rectangle decorated with a gauge icon on its upper left corner, its ID centred at the top, and the measure defining the PPI displayed inside the rectangle. The target value and the temporal scope are displayed together with their corresponding icons in an optional grey bottom compartment as shown in Fig. 4a.

On the other hand, measures can be classified into the three main measure categories present in the PPINOT metamodel: *base* measures, *aggregated* measures, and *derived* measures.

5.1.1 Base Measures in VISUAL PPINOT

Base measures are represented as short rulers with their name underneath as depicted in Fig. 4b. A small icon is added on the upper left corner depending on the measure type: *time*, *count*, *state condition*, or *data*.

5.1.2 Aggregated Measures in VISUAL PPINOT

Base measures generate one value for each instance of the process they are defined for. Sometimes, it is interesting to know not only the value of a measure for a single-process instance, but also an aggregation of the values corresponding to the multi-process instances in the scope of the corresponding PPI. These situations are modelled in VISUAL PPINOT using *aggregated measures*, which are displayed as three stacked base measure icons (representing their *multi-instance* nature) with an aggregation function inside: **AVG** for *average*, **MAX** for *maximum*, **MIN** for *minimum*, **SUM** for *summation*, etc. (see Fig. 4c). They are connected to the *single-instance* measure being aggregated using *aggregates* connectors, depicted as solid lines starting with a white diamond and labelled with “aggregates” (in boldface in Fig. 4c to distinguish them from placeholders). In the case of base-measure aggregation, both icons can be combined into one, as shown in Fig. 5.

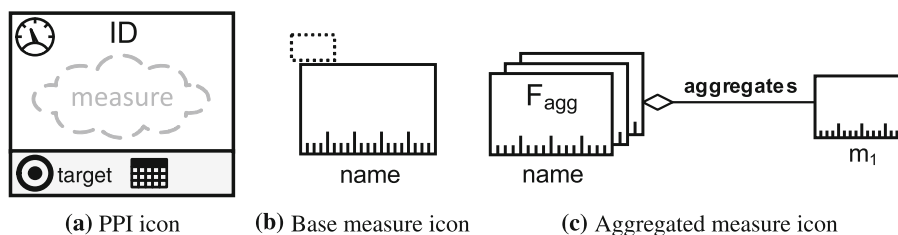


Fig. 4 VISUAL PPINOT icons for PPIs, base measures, and aggregated measures

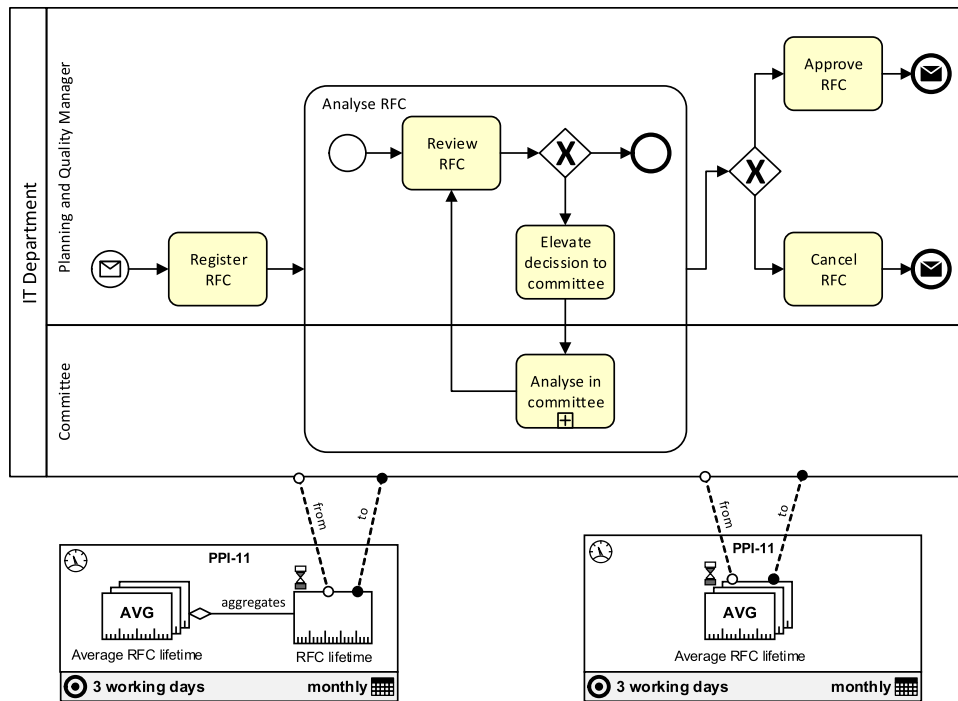


Fig. 5 Equivalent aggregated measures for the Average RFC lifetime PPI (PPI-11 in Table 1)

5.1.3 Derived Measures in VISUAL PPINOT

Derived measures are visually distinguished by a function symbol (f_x) on the upper left corner and by the expression of their derivation function inside the ruler icon. Function variables are connected to derived measures by uses connectors labelled with the corresponding variable names as depicted in Fig. 6. Depending on whether the derivation function is defined over single- or multi-instance measures, derived measures are classified accordingly and their icons are simple or three-stacked as shown in Fig. 6a and b, respectively. Notice that all the measures involved in a derived single- or multi-instance measure must also be

single- or multi-instance according to the derived measure being defined. Figure 7 shows the example of the derived measure “Percentage of RFC analysis time”.

5.2 Time Measures in VISUAL PPINOT

Time measures, visually identified by an hourglass (⌚), are used to measure the duration between the occurrence of two events, considering as events not only BPMN events, but also state transitions of BPNM elements such as activities, pools, or data objects. Notice that a time measure has an undefined value until both events have happened, something that is relevant for aggregated measures.

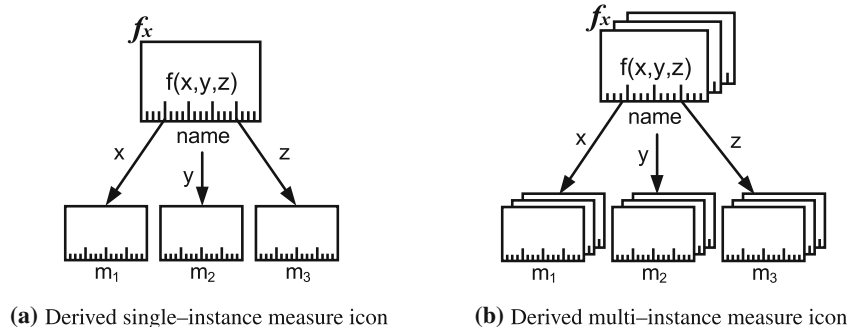


Fig. 6 VISUAL PPINOT icons for derived measures

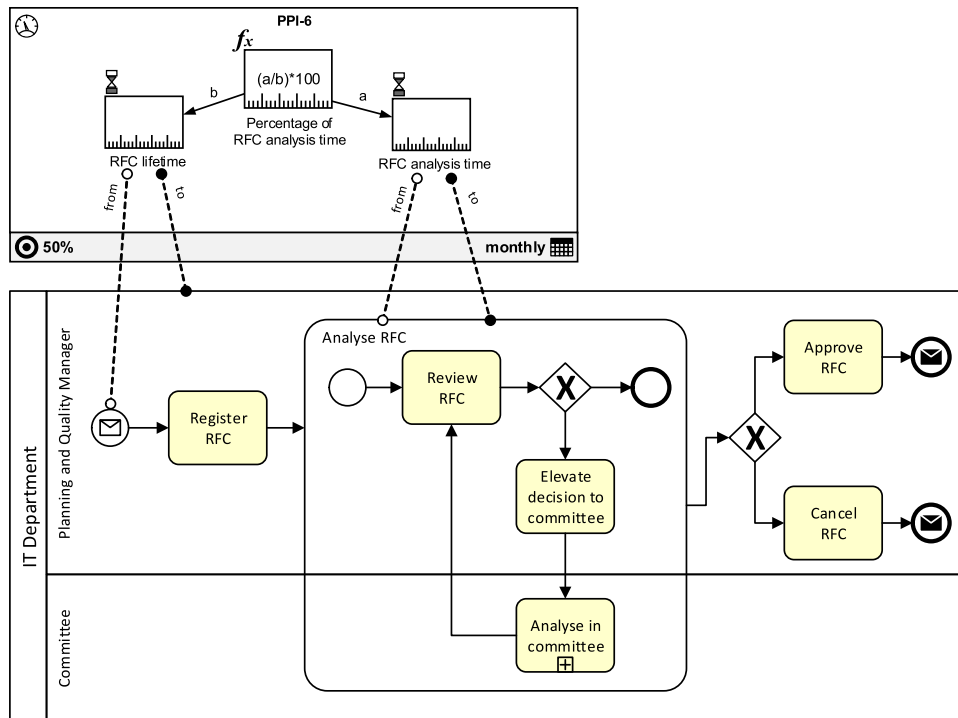


Fig. 7 Derived measure corresponding to the Percentage of RFC analysis time (PPI-6 in Table 1)

To indicate the two events, time measures use *time* connectors, represented as dashed lines. As shown in Fig. 8a, the connector for the first event is labelled with “from” and decorated with an empty circle on the end close to the measure icon (\circ -----), whereas the connector for the second event is labelled with “to” and decorated with a filled circle (\bullet -----). Because the *start* and *end* of activities and pools are by far the most relevant events for defining time measures, they have their own graphical representation: the *start* event is depicted as an empty circle (\circ -----), whereas the *end* event is represented as a filled circle (\bullet -----). In both cases, the name label is left as

optional and it is usually omitted. The semantics of these two events are *token based* (OMG 2011), i.e., we consider that a *start* event happens when a token arrives at a BPMN element and that the *end* event happens when the token leaves an element. These two events are usually used with pools and activities, but they can also be used with BPMN events (see Sect. 5.6.1 for details). If an event is related to a state transition, the corresponding time connector must be labelled with the *target state*, i.e., the state to which the BPMN element must change to consider the event as triggered. Any state defined in the BPMN specification (OMG 2011) can be used with pools and activities, as well

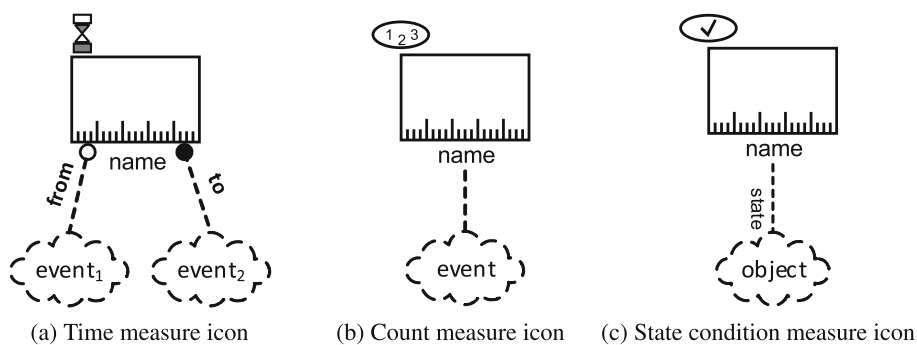





Fig. 8 VISUAL PPINOT icons for time, count, and state condition measures

Table 2 Time connector rules, also applicable to applies-to connectors in count measures

	-----○ start	----- state	-----● end	-----
 activity/pool	Token arrives	BPMN-defined states	Token leaves	N.A.
 event	Token arrives	N.A.	Token leaves	Token leaves
 data object	N.A.	User-defined states	N.A.	N.A.

as any user-defined state can be used with data objects. A summary of the applicability of time connectors is displayed in Table 2. Figures 5 and 7 present several examples of time measures.

5.3 Count Measures in VISUAL PPINOT

Count measures, identified by an ellipse with the numbers 1, 2, and 3 inside ($\langle 1,2,3 \rangle$), are used to count how many times a given event happens. Events are linked to count measures using applies-to connectors as shown in Fig. 8b, and their applicability rules are the same as for time connectors, summarised in Table 2. Figure 14 contains two examples of count measures.

5.4 State Condition Measures in VISUAL PPINOT

State condition measures, decorated with an ellipse containing a checkmark ($\langle \checkmark \rangle$), generate Boolean values depending on the current state of activities, pools, or data objects. As depicted in Fig. 8c, these BPMN elements are linked to state condition measures using applies-to connectors, which must be labelled with the target state name

($\langle \checkmark \text{state} \rangle$). Notice that the start and end event notations cannot be used with this type of measures because they are not actual states but events (see Table 2).

In the case of state condition measures aggregation, Boolean values are mapped to integers, i.e., $false \mapsto 0$, $true \mapsto 1$. Because of this mapping, the aggregation functions are not the same as those commented on in Sect. 5.1.2, but the following ones are summarised in Fig. 9: (#) number of process instances in which the state condition holds, equivalent to the summation function; (%) percentage of process instances in which the condition holds, equivalent to the average function; (\exists) true if there exists at least one process instance in which the condition holds, i.e., when the values of the minimum and maximum aggregation functions are 0 and 1 respectively, as shown on the right side of Fig. 9; (\forall) true if the condition holds in all the process instances in scope, i.e., if the minimum and maximum functions values are both 1; (\nexists): true if there does not exist any process instance in which the condition holds, i.e., if the values of the minimum and maximum functions are both 0.

Some examples of aggregated state condition measures are shown in Fig. 10.

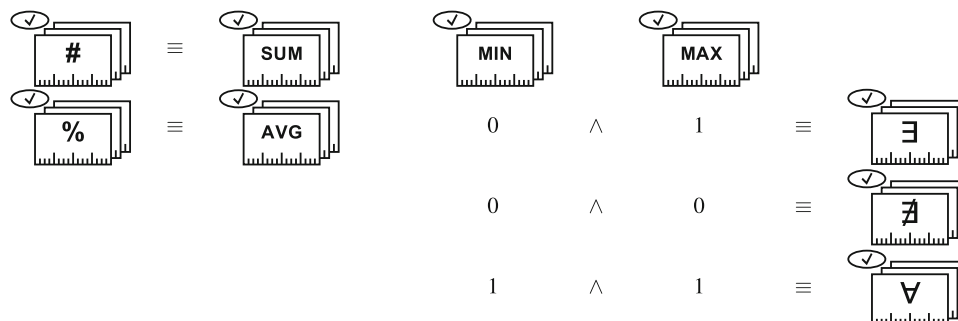


Fig. 9 Equivalent semantics of aggregated state condition measures, in which the combination of $\text{MIN} = 1$ and $\text{MAX} = 0$ is not considered because it is a contradictory situation (\perp) and, thus, it cannot happen

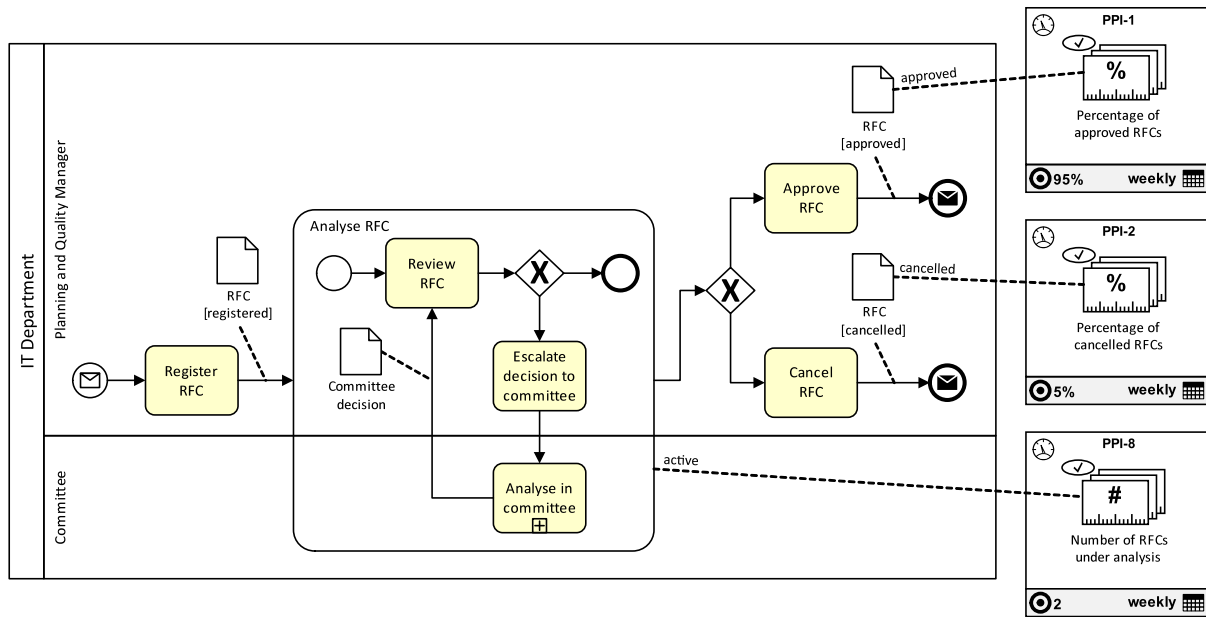


Fig. 10 Aggregated state condition measures corresponding to PPI-1, PPI-2, and PPI-8 in Table 1

5.5 Data Measures in VISUAL PPINOT

Data measures, identified by a small data object icon (📄), are used to obtain the value of a specific attribute of a data object. The applies-to connector linking the measure icon with the data object must specify the attribute reference to be measured and, optionally, the state the data object must be in to actually obtain the value, as depicted in Fig. 11a). If the state were specified and the data object were in a different state, the value of the measure would be *undefined*. Notice that, to aggregate data measures, the measured attribute must belong to a data type with at least the \leq , $+$, and \div operators defined to properly apply the usual aggregation functions.

5.6 Advanced Topics in VISUAL PPINOT

There are some features of VISUAL PPINOT with slightly more complex semantics than the ones described in

previous sections. They are not strictly needed to understand the main concepts of the notation, but they are included in this article to provide a thorough overview of our proposal.

5.6.1 Duration of BPMN Events in VISUAL PPINOT

Most of the different types of BPMN events are considered to consume no time, i.e., they just *happen* during the course of a process (OMG 2011). Nevertheless, there are some intermediate *catching* events in which the process can wait for a significant amount of time. If the duration of this process waiting were interesting for some PPI, it could be measured using a time measure together with the *start* and *end* events applied to the same BPMN event, as shown in Fig. 12. The token-based semantics of these two events would measure the duration between token arrival and leaving, i.e., the duration of the process waiting for the BPMN event to happen.

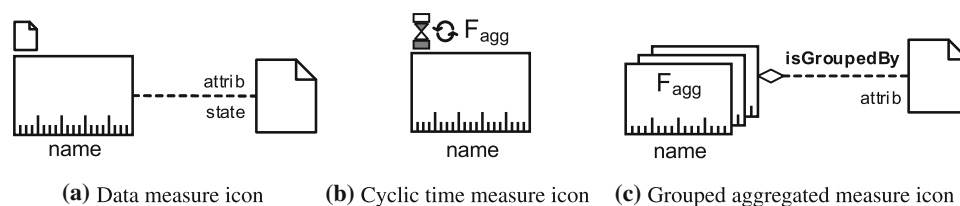


Fig. 11 VISUAL PPINOT ICONS FOR DATA, CYCLIC TIME, AND GROUPED AGGREGATED MEASURES

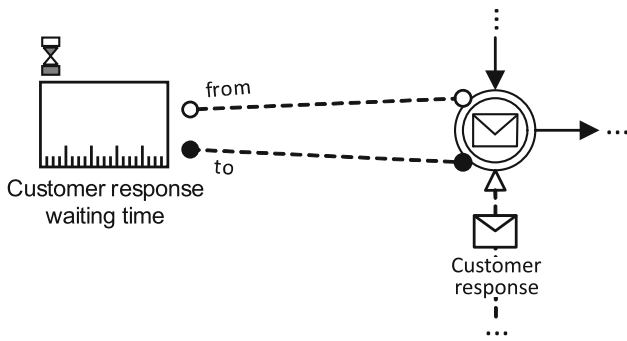


Fig. 12 Example of a measure of the duration of a BPMN event

5.6.2 Cyclic Time Measures in VISUAL PPINOT

In certain circumstances, the two events associated with a time measure can happen more than once during the execution of an instance of a business process, usually in execution loops like the one in the Analyse RFC subprocess in Fig. 13. In those circumstances, the linear time measure described in Sect. 5.2 would measure the duration between the first occurrence of the “from” event and the last occurrence of the “to” event, as depicted at the top right of Fig. 13.

In the case those semantics were not appropriate for the measure at hand, VISUAL PPINOT allows the use of cyclic time measures, which aggregate the durations of the generated (from, to) event pairs. Visually, as Fig. 11b shows, the only differences with linear time measures are the cycle symbol added to the hourglass icon and the aggregation function. The difference between both types of time measures is graphically displayed in Fig. 13. Notice that, when a cyclic time measure is used to measure the duration

between two events that cannot happen more than once in the same instance of a given business process, the result would be the same as if a linear time measure were used, regardless of the aggregation function applied.

5.6.3 Grouping Aggregated Results in VISUAL PPINOT

In the motivating scenario described in Sect. 2, PPI-9 and PPI-10 in Table 1 describe their target values depending on the type of change in an RFC and on the project that the RFC affects, respectively. In these situations, the value of the aggregated measure – number of RFCs – must be grouped by some data object attributes – typeOfChange and project of the RFC data object. To model this type of measures, VISUAL PPINOT introduces the isGroupedBy connector, depicted, as shown in Fig. 11c, as a dashed line starting with a white diamond and labelled with “isGroupedBy” and the name of the data object attribute used to group the measure values. Figure 14 contains examples of grouped aggregated measures.

5.6.4 Partial Percentages in VISUAL PPINOT

Percentages are commonly used in PPI definitions. For example, in Table 1, 5 out of 11 PPIs are defined as percentages. In the case of percentages defined over all the process instances in scope, an aggregated state condition measure with the percentage of process instances aggregation function (%) can be used, as shown in Fig. 10. In other cases in which the percentage is defined over a subset of the process instances, the measure definition becomes more complex. As an example, consider the PPI Percentage of corrective approved RFCs (PPI-4 in

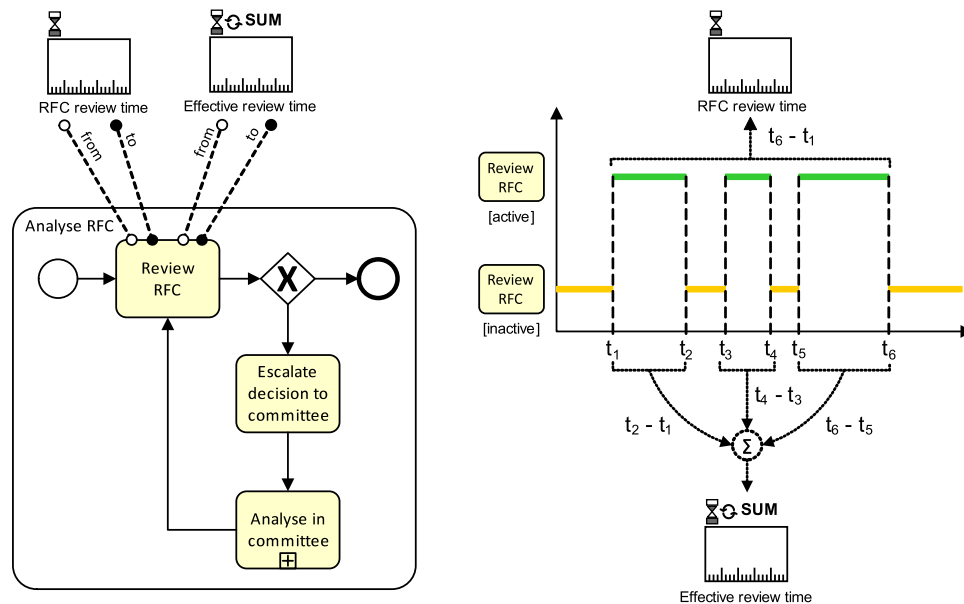


Fig. 13 Linear and cyclic time measure examples and semantics

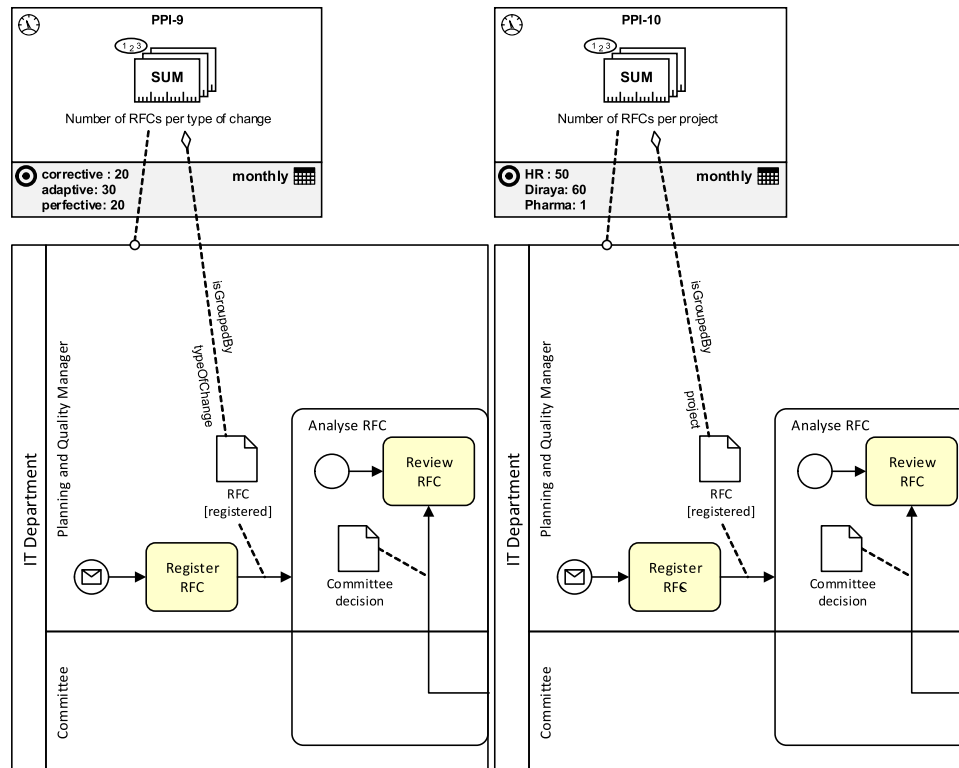


Fig. 14 Grouped aggregated measures corresponding to PPI-9 and PPI-10 in Table 1

Table 1). In this PPI, the percentage denominator is not the number of all RFCs but the number of all *approved* RFCs, something that makes the measure more difficult to be described, especially when compared with the PPI Percentage of approved RFCs (see Figs. 10 and 15 for a comparison of both percentage measures).

6 VISUAL PPINOT: Design Rationale

In the design of a new visual notation, the two main decisions are (1) the choice of those semantic constructs with a graphical representation, and (2) how to use visual variables to encode information graphically. In VISUAL PPINOT, these two decisions have been made following the BPMN 2.0 design guidelines and the principles of the *Physics of Notation* (Moody 2009). On the one hand, since VISUAL PPINOT is intended to be used together with BPMN 2.0 diagrams, it seemed reasonable to follow BPMN 2.0 design guidelines. On the other hand, the *Physics of Notation* has been specifically developed as a theory of visual notation design, including nine principles synthesised not only theoretically, but also from empirical evidence. The rationale behind the two aforementioned decisions is described in the next two sections.

6.1 Choice of Semantic Constructs with Graphical Representation

Most elements of the PPINOT metamodel have a 1:1 correspondence with the graphical symbols in VISUAL PPINOT, as suggested by the *Physics of Notation* principle of *semiotic clarity*. However, some *symbol deficit*, i.e., leaving some semantic constructs without graphical representation, was introduced to limit the *graphic complexity*, as suggested by the *Physics of Notation* principle of *graphic economy*, which states that the number of graphical symbols should be cognitively manageable.

The decision of which semantic constructs do not have a graphical representation was made according to (1) the frequency they appear in the related literature and the scenarios in which VISUAL PPINOT has been applied and (2) the type of information they convey. Concerning the former, we excluded those semantic constructs that appear with lower frequency. Specifically, the PPI temporal scope and target can be graphically depicted only when they are simple (e.g., monthly, lower than 7, or between 10 and 15) but not when they have more complex semantics (e.g., working days or Christmas holidays). As for the latter, we excluded some attributes of PPIs and measure definitions whose information is provided by means of a free text field such as *goals*, *informed*, or *comments*.

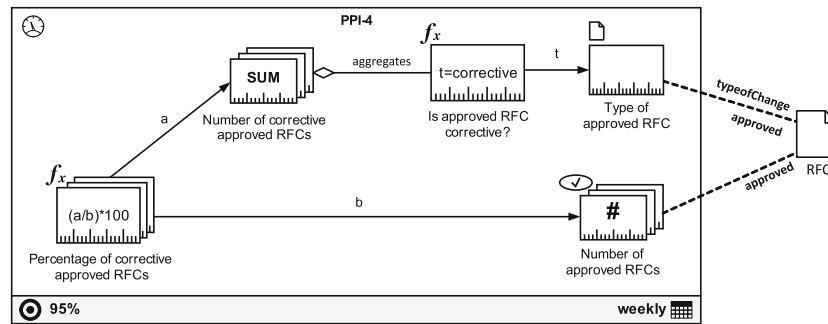


Fig. 15 Example of partial percentages corresponding to PPI-4 in Table 1

Some *symbol redundancy*, i.e., having more than one symbol for a single semantic construct, was also introduced to allow the modelling of aggregated base measures in their expanded or abbreviated form (see Fig. 5), thus providing an explicit mechanism for dealing with *diagrammatic complexity*, as suggested by the Physics of Notation principle of *complexity management*.

The complete correspondence between the PPINOT metamodel and the VISUAL PPINOT graphical symbols is included as Online Appendix B.

6.2 Use of Visual Variables to Graphically Encode Information

The Physics of Notation defines eight dimensions or *visual variables* of the graphic design space, which can be used to graphically encode information (Moody 2009). They are divided into *planar* (horizontal and vertical position) and *retinal* variables (shape, size, colour, brightness, orientation, and texture). In VISUAL PPINOT, only *shape*, *brightness*, *texture*, and *position* are used, although the last one is only used for enclosing measures inside PPI icons. The unused visual variables can be freely used by the user to emphasise concepts from the business domain. This decision is not aligned with the Physics of Notation principle of *visual expressiveness*, which pursues the use of the full range and capacities of visual variables, but it maintains consistency with BPMN 2.0.

Shape is the main visual variable of VISUAL PPINOT nodes because of its privileged role in perceptual discrimination (Moody et al. 2009). Therefore, different shapes have been used for different constructs. These shapes are graphic metaphors commonly used for the semantic concepts they represent, following the Physics of Notation principle of *semantic transparency*.³ Thus, a ruler is used as a metaphor of a measure, a gauge as a metaphor of an indicator, an

hourglass as a metaphor of time, and so on. Furthermore, considering also that similar shapes should be used to represent similar constructs (Moody et al. 2009), similar symbols were designed for derived single-instance and derived multi-instance measures.

Shape is also used to distinguish between the main groups of connectors. To distinguish between subgroups, we used *brightness* for *from* and *to*, and *texture* for *aggregates* and *isGroupedBy*. In these cases, we decided to use text to reinforce graphical differences as suggested by the Physics of Notation principle of *dual coding*. The specific way in which visual variables have been used has been inspired by BPMN 2.0 and other similar notations such as UML because most users of VISUAL PPINOT are expected to be familiar with these notations.

Finally, the use of shading, line thickness, colours, or any other distinction that does not fax/copy well or make symbols difficult to draw by hand was intentionally avoided. This decision has been supported by the experience gained in the evaluation scenarios, in which all the symbols were easily drawn by hand, something greatly appreciated in visual notations (Rumbaugh 1996).

7 Tool Support: The PPINOT TOOL SUITE

VISUAL PPINOT diagrams can be developed using an Oryx-based editor available at <http://www.isa.us.es/ppinot>. Oryx (Decker et al. 2008) is an open-source platform to build web-based diagram editors providing native support for BPMN and allowing the definition of new graphical notations by means of so-called *stencil sets*, which have been used for the VISUAL PPINOT editor. Furthermore, VISUAL PPINOT is part of the PPINOT TOOL SUITE (see Fig. 16), a set of tools aimed at facilitating and automating PPI management that are built around the PPINOT metamodel and described in the following paragraphs according to their purpose.

PPI Definition The PPINOT TOOL SUITE offers two ways to define PPIs. They can be graphically defined together with BPMN diagrams using the VISUAL PPINOT editor or using a

³ All of them, except *state* and *data*, appear amongst the first results in a search in Google Images. *Data* does not appear because it was chosen to match the BPMN symbol for data objects. *State* does not appear because it is a polysemic word and we refer to a very abstract meaning of the word.

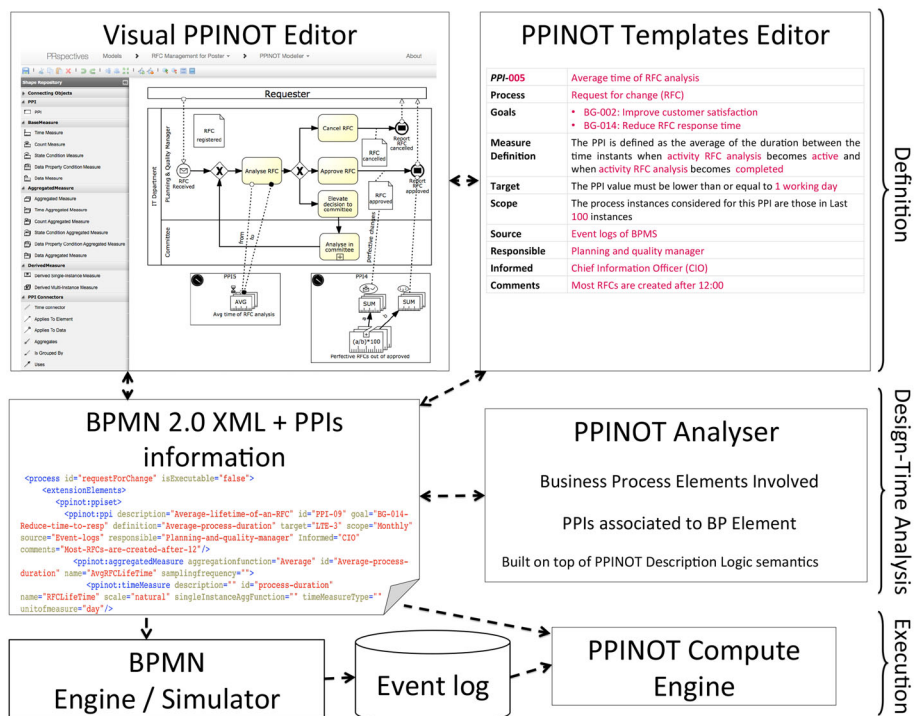


Fig. 16 Overview of the PPINOT TOOL SUITE (<http://www.isa.us.es/ppinot>)

template-based textual notation (del Río-Ortega et al. 2016) implemented in the PPINOT TEMPLATES editor, which guides the user by providing linguistic patterns in the different template fields and allowing to change seamlessly from one notation to the other. In both cases, a standard BPMN 2.0 XML document extended with PPI-related information is obtained as output.

PPI Design-time analysis In PPINOT ANALYSER, two traceability analysis operations are currently supported: (1) *business process-elements involved*, which returns the business process model's elements directly or indirectly involved in certain PPI, and (2) *PPIs associated with business process-element*, which returns the PPIs associated with or applied to a given business process model element. These two operations can assist during the evolution of business processes and their PPIs, and were first introduced in del Río-Ortega et al. (2013).

PPI Computation The PPINOT COMPUTE ENGINE computes PPI values using the event log of the associated business process. It has been designed to use many different types of event logs, which shows how the system is independent of the platform used to enforce the business process. In particular, the current implementation supports logs from a business process simulator (BIMP⁴), a service-desk

manager solution, and an issue management system, amongst others.

8 VISUAL PPINOT Evaluation

To assess the applicability of VISUAL PPINOT and its features we carried out a multiple-case study with four cases, that is presented in the following sections.

8.1 Case Study Research Process

Our research method in this evaluation has been based on case study research. Specifically, we have followed the case study research process proposed in Runeson and Höst (2009) and Runeson et al. (2012), as described below and summarised in Fig. 17.

1. Case study design. The main reason for conducting this study was the need to evaluate VISUAL PPINOT. Considering this, together with the theoretical framework provided by the previously conducted literature review, the objective of this case study was established as the empirical revision of VISUAL PPINOT in terms of expressiveness, precision, automation, understandability and traceability. This objective was further refined into the five research questions that are presented and addressed in following subsections.

⁴ <http://bimp.cs.ut.ee/>.

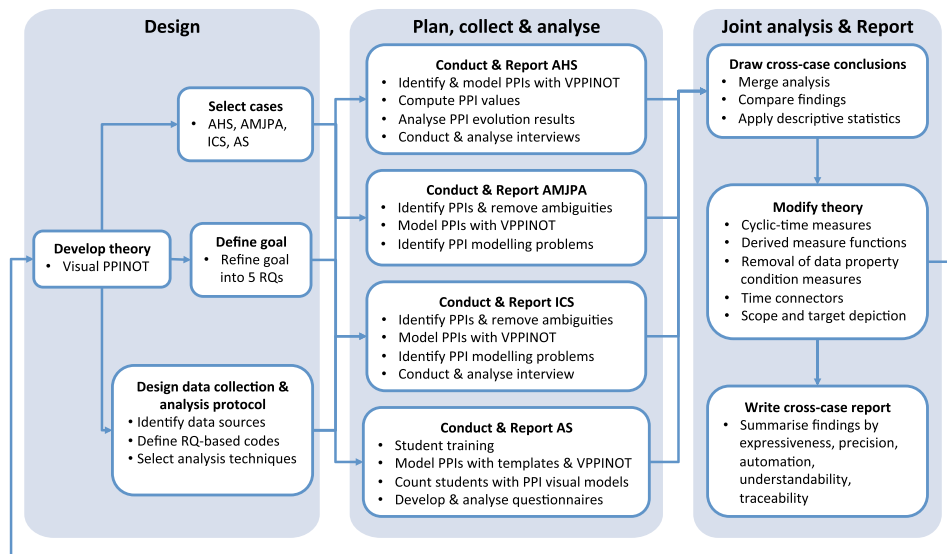


Fig. 17 Process of our multiple-case study inspired by Runeson et al. (2012)

As for the case selection, initially, three cases were selected: the IT Department of the Andalusian Health Service (AHS), the Information and Communication Service of the University of Seville (ICS), and the Andalusian Ministry of Justice and Public Administration (AMJPA). They were selected for three main reasons: (1) the evaluation of VISUAL PPINOT in different domains; (2) the interest of the organisations in improving their processes because of their involvement in a quality certification process or in adopting widely acknowledge good practices (Office of Government Commerce 2007; (3) the availability of data sources and subjects' willingness to cooperate.

Afterwards, and with the aim of broadening the number of domains and user profiles, a new case with technical graduate and undergraduate students of the University of Seville (Academic Scenarios – AS) was added. This led to a multiple-case study because of the different contexts of each case. Finally, we designed the data collection protocol, defining the desired data to be collected as well as the type of analysis to be performed, establishing a plan to address the following steps of our multiple-case study.

2. Plan, collect and analyse. This activity was performed for each of the case studies, including the following three steps:

2.1. *Preparation for data collection* This step involved different activities depending on the case, including the identification of the archival data available that could be provided to the researchers for analysis, the preparation of

material and subsequent training of one of the cases' participants for the definition of PPIs, the graphical modelling and, in some cases, computation of PPIs with VISUAL PPINOT, the definition of interview questions for two of the cases and the design of a questionnaire for another of the cases.

2.2. *Collecting evidence* In order to collect evidences we followed the principles for data collection in Verner et al. (2009). We used as many sources of data as were available. This included the use of data collection techniques from the three degrees defined in Lethbridge et al. (2005): (1) direct methods consisting of interviews and questionnaires; (2) indirect methods applied in our academical case; (3) independent analysis of already available work artefacts. In addition, during the course of some of the cases, unexpected opportunities for collecting data emerged as reported in the following sections. To store the data collected, we combined the use of repositories for text documents together with spreadsheets that ease their posterior analysis.

2.3. *Analysis of collected data* The analysis performed in the study is of the deductive type, implying that categories of analysis are imposed prior to the data collection. The activities performed involved different types of qualitative analyses, inspired by the process for qualitative data analysis introduced in Runeson et al. (2012). They were complemented by some quantitative analyses based on descriptive

statistics. The specific analysis was chosen depending on the type of data available and our goal in the case study. The analysis was conducted by multiple researchers, specifically three, in order to reduce bias by individual researchers. First, two individual researchers analysed the data, and then their results were merged and discussed by both researchers together with an additional one.

3. Joint analysis and report: Once all cases were analysed, we drew cross-case conclusions, leading in some cases to the confirmation of our hypothesis, and in others to the modification of our theory, i.e., our graphical notation. As a last step, we wrote the current report, including the results found and the specification of the identified limitations.

8.2 Case Study Design

The established objective for our multiple-case study was refined into five research questions, namely:

- RQ₁ – Expressiveness To what degree is VISUAL PPINOT capable of expressing what needs to be expressed? Were there some cases that could not be expressed by VISUAL PPINOT?
- RQ₂ – Precision How far is VISUAL PPINOT better in helping to arrive at a precise specification than, e.g., text? What are the experiences and user feedback?
- RQ₃ – Automation How does VISUAL PPINOT help to automatically obtain PPI values without redefining them for computation?
- RQ₄ – Understandability How well or easy can users understand VISUAL PPINOT? How easy is it to read and write?
- RQ₅ – Traceability What are the benefits and challenges of integrating the PPIs model with the BPMN model?

Regarding the four selected cases, Table 3 summarises their information in terms of processes, activities, PPIs, and people involved. In the case of the IT Department of the Andalusian Health Service, VISUAL PPINOT was applied to

the RFC management process (with 27 activities in the real model) and its 11 associated PPIs. Apart from two VISUAL PPINOT experts, the two managerial roles involved in this case were the ones responsible for processes and continuous improvement, and for SLAs and performance management respectively, both belonging to the quality group of the department.

In the Information and Communication Service of the University of Seville, VISUAL PPINOT was applied to four business processes, with a number of activities between 6 and 23, framed in the context of the IT support to the university staff (email incident management, for instance), and their 16 associated PPIs. In this case, the roles involved were the section manager, the technical manager of the teaching and research support area, and the software development manager together with two researchers and one master's student who worked as a VISUAL PPINOT assistant for the ICS staff.

In the Andalusian Ministry of Justice and Public Administration, there were 29 PPIs described in 5 processes with a number of activities between 8 and 36, ranging from social and health benefits management to suggestions, complaints, and claims management. In this case, three researchers amongst the authors of this article were the roles involved.

Finally, VISUAL PPINOT was evaluated in two master's courses of the University of Seville belonging to the *Master of Information and Communication Technology Management*, and the *Master of Software Engineering and Technology*, and in one bachelor course (*Processes and Services Management*) of the same university. In total, 112 students were trained in VISUAL PPINOT and modelled at least two PPIs in the master's courses, and at least eight PPIs in the bachelor course, for a real business process specified in BPMN. The processes belonged to many different domains: health, justice, university (scholarships, enrolment, research project management, etc.), software development or politics, to name a few. In total, 374 PPIs were modelled in these processes.

With respect to the data collection protocol, interviews, questionnaires and some available archival data were the initially defined data sources, according to the information available at that moment. Later on, they were refined and

Table 3 Summary of the characteristics of the cases

	AHS	ICS	AMJPA	AS	Σ
# Processes	1	4	5	93	103
# Activities	27	6–23	8–36	5–38	5–38
# PPIs	11	16	29	374	430
People involved	2 MR + 2 VPE	2 MR + 2 TR + 2VPE + 1 VPA	3 VPE	112 St + 2 VPE	123

MR managerial role (technical background), *VPE* visual PPINOT expert, *TR* technical role, *VPA* visual PPINOT assistant, *St* graduate and undergraduate students

extended during the course of the study, as detailed in the following section.

8.3 Preparation and Data Collection

The preparation of data collection and its actual collection were performed by means of iterative cycles. We applied data collection techniques from the three degrees defined in Lethbridge et al. (2005). In the following, we describe them, organised by data type. The degree is specified in parenthesis.

Process models and previously defined PPIs (third degree) The three studied organisations had previously undergone a BPM initiative, and, as a result, all processes under study were already modelled either in BPMN (AHS) or in another non-standard notation like flow diagrams (ICS, AMJPA). For each process, PPIs were defined using *ad hoc* table-based notations written in a natural language. All these available data were collected, and, in the required cases, the process models were translated to BPMN by some of the researchers involved to allow the definition of the corresponding PPIs in PPINOT TOOL SUITE. In the academic scenarios, the process models were modelled in BPMN by the students using the documentation provided to them from different sources.

PPIs defined in VISUAL PPINOT (second and third degree) The definition of the provided indicators using VISUAL PPINOT involved different roles depending on the unit. In the AHS, the 11 PPIs associated with the RFC management process were modelled with the supervision and support of the one responsible for processes and continuous improvement (the result is presented in Online Appendix C). In the ICS, a preprocessing was needed, since some of the indicator definitions provided were not actually referred to the business processes but to other aspects of the organisation, i.e., they were not actual PPIs. Once filtered out those indicators, the VISUAL PPINOT assistant modelled the 16 PPIs under the supervision of all the other roles involved from the ICS staff, throughout 11 meetings (read Sánchez-Jerez 2012 for more details, including business process and PPI models). In the AMJPA, two VISUAL PPINOT experts were in charge of modelling the 29 PPIs. Finally, in the AS, students were required to define PPIs textually using some templates and patterns provided, whereas their graphical definition using VISUAL PPINOT was optional (just for improving their grades).

Event logs and PPIs computation (third degree) In the AHS unit, a set of event logs containing the information of the business process execution on a period of 24 months was also available. This information was used by two VISUAL PPINOT experts to compute the PPI values in the provided period. This was carried out through the PPINOT

TOOL SUITE, in particular the PPINOT Compute Engine, using as input both the PPI definitions in VISUAL PPINOT and the provided event logs. The result was the set of raw values for the input set of PPIs, which were stored in our data repositories.

Evolution information (third degree) As part of an ongoing collaboration that some of the authors maintain with the AHS, we also had the opportunity of working on an evolution of the RFC management process, serving as a consultant for its re-modelling in BPMN. This scenario provided data to check how the VISUAL PPINOT definition helps in the case of the associated business process evolution. Specifically, we gathered the process models and PPIs defined before and after the evolution.

Interviews and questionnaires (first degree) Direct methods for collecting data were used in three of the cases. In particular, in the AHS, the final graphical PPI definitions together with their computation results were presented by two researchers to the person responsible for SLAs and performance management during an interview. This was a semi-structured interview, more in the form of a discussion. We used the interview questions prepared as a guide of important information to be gathered. Furthermore, the conclusions of the evolution of the process model were also discussed with him in that interview. Most interesting facts and important answers were written down by one of the researchers in the form of notes that were later sent to the interviewee for validation. In the ICS case, the final graphical PPI definitions were presented to the software development manager during a structured interview by the aforementioned VISUAL PPINOT assistant, who also wrote down the answers for their later validation by the interviewee. Finally, questionnaires were conducted with students at the end of their courses in order to gather information regarding identified expressiveness-related limitations and understandability-related weaknesses in VISUAL PPINOT.

8.4 Analysis of Collected Data

The analysis performed in the study is of the deductive type, implying that categories of analysis are imposed prior to the data collection. The main categories of analysis in our multiple-case study correspond with the five research questions and are: expressiveness, precision, automation, understandability and traceability. The specific actions accomplished during this analysis are detailed next.

1. The whole set of PPIs of the different cases was reviewed to check whether all of them could be defined with VISUAL PPINOT or not. For those PPIs that could not be defined with VISUAL PPINOT, the reasons that prevented their definition were identified and

- categorised according to: (1) limitation of PPINOT, (2) missing or ambiguous information, (3) indicators not related to the process (i.e., not actual PPIs). When possible, clarification and provision of missing information were required to the roles involved in PPI modelling.
2. Descriptive statistics were applied in order to obtain information related to the different VISUAL PPINOT constructs used, first in the set of PPIs defined with VISUAL PPINOT in each of the cases, and then in the whole set of PPIs defined within our multiple-case study. Table 4 summarises this information.
 3. In the academic case, the number of students that decided not only to define their PPIs with the provided templates, but also to graphically model them, were count. Furthermore, we analysed the templates provided by the students and identified and marked those that included erroneous or ambiguous definitions.
 4. The modelling mistakes introduced by students in their assignments were reviewed. Specifically, we annotated them with the VISUAL PPINOT constructs that were incorrectly used in the model. This information together with the results of their questionnaires were used to identify the notation constructs that presented more problems in relation to their understandability and correct use.
 5. The changes suffered by the process and their impact in the graphically defined PPIs were analysed by using the evolution information obtained in the AHS case. Specifically, we checked whether the change had an influence on any of the PPIs of the process and, if so, we checked in which manner the PPI was affected.
 6. The data collected during the interviews and questionnaires was analysed to draw conclusions regarding the automation and the understandability. Specifically, two researchers analysed the interviews and questionnaires and coded them according to these categories. The coded data was stored in tables together with references

to the data source in order to ensure full traceability and the maintenance of a chain of evidence. These tables were then used to identify results across data sources and cases, merging the results obtained by the two researchers and discussing them with the third researcher. In addition, regarding the validation, in the cases of the AHS and the ICS, the preliminary results from the study, including the interview, were presented back to the interviewees and other people involved during a meeting, and their opinions were collected and any misinterpretation corrected. Final conclusions were based on all the gathered information.

8.5 Results

This section presents the results identified after the analysis performed on the collected data. We structure the findings according to the five categories corresponding to the research questions posed in Sect. 8.2.

RQ₁ – Expressiveness After the analysis of the collected data, some limitations were detected in VISUAL PPINOT and the notation was consequently improved. The first improvement was related to the distinction between linear and cyclic time measures, identified during the definition of PPI-3 in the AHS, which implied measuring an average duration located within a loop. The second was the inclusion of the *isGroupedBy* connector, used to define different target values according to a certain attribute value of a data object, as required by PPI-9 and PPI-10 from the AHS with respect to RFC objects (see Fig. 14). The third improvement was the optional inclusion of the target values and temporal scopes in the PPI icon, provided they are simple enough to be displayed. Finally, the range of function types used within derived measures was expanded to include Boolean and relational functions in addition to arithmetic ones, to meet the requirements of some PPIs.

After these improvements, more than 400 real-world PPIs from different domains were defined in VISUAL PPINOT.

Table 4 Summary of the PPIs characteristics modelled within the case study

	AHS	ICS	AMJPA	AS	Σ
Time aggregated measure	3	2	9	192	206
Count aggregated measure	2	17	31	234	284
State condition aggregated measure	5	0	0	28	33
Data aggregated measure	2	0	1	43	46
Aggregated measure	2	0	5	9	17
<i>isGroupedBy</i>	2	0	0	4	6
Derived single-instance measure	3	0	5	10	18
Derived multiple-instance measure	2	3	3	133	141
Simple (numeric) targets/other targets	11/0	16/0	29/0	187/4	243/4
Temporal scopes/other scopes	11/0	16/0	29/0	72/19	128/19

As detailed in Table 4, all VISUAL PPINOT symbols were used at least once, although some of them (e.g., Time or Count Aggregated Measure) were used much more frequently than others (e.g., State Condition Aggregated Measure). Furthermore, most of the PPIs modelled in the case study had simple numeric targets (98.4 %) and temporal scopes (87 %), thus allowing their complete graphical representation.

RQ₂ – Precision The application of VISUAL PPINOT to the PPIs previously defined in other formats revealed three major limitations: (1) the indicator definitions were not clear because they used ambiguous language, making their interpretation and computation difficult; (2) some of the indicators lacked a clear relation to the business processes; actually some of them were not related to any business process and could not be computed on the basis of any business process execution values; and (3) for those indicator definitions related to business processes, many lacked the explicit relationship to the different business process elements, i.e., it was not straightforward to instrument the corresponding business process to obtain the PPI values.

Further results in this regard were also obtained from the analysis of the material collected with students in our fourth case. Specifically, from the 17.4% who opted to directly provide the textual PPI definitions without the graphical notation, in half of the cases, erroneous or ambiguous definitions were provided. Other definitions were not erroneous but did not explicitly refer to specific elements of the business process; therefore, preprocessing was required to identify them and instrument the corresponding processes. These results reinforce our claim that defining PPIs with VISUAL PPINOT forces the user to precisely define them and explicitly link the PPI with the elements of the process model related to its definition.

RQ₃ – Automation The computation results obtained via PPINOT TOOL SUITE from the PPI definitions in VISUAL PPINOT and the collected event logs were presented to the person responsible for SLAs and performance management during the interview. He was asked to validate them and he compared them with the reports they had for that time period and reported them to be correct. During this interview, he recognised the usefulness of VISUAL PPINOT in this scenario because of two reasons: first, it was not necessary to spend time in implementing PPIs using SQL queries to compute their values, and, second, the interpretation that was given to a PPI using VISUAL PPINOT was much easier to understand than finding out the interpretation by going through a set of SQL queries. In addition, VISUAL PPINOT proved to be platform independent since PPINOT TOOL SUITE was developed without knowledge of the information system that provided the logs for computing PPI values.

RQ₄ – Understandability The first applications of VISUAL PPINOT revealed a couple of limitations related to the

labelling of time connectors and the use of data property condition measures. This led us to a twofold improvement of the notation. On the one hand, the labels “start” and “end” of time connectors were changed to “from” and “to”, respectively, since the first labelling seemed to refer to the start and end of activities and pools instead of the events that allow the duration to be measured. On the other hand, we removed the graphical construct for data property condition measures because VISUAL PPINOT users did not clearly distinguish between them and data measures, and the construct was no longer necessary since they can be modelled as a derived single-instance measure with a Boolean function on a data measure.

After these improvements, the *users*, i.e., organisations’ employees, students and researchers, were able to read and validate PPI graphical definitions as reported on during the interviews. Furthermore, the users from the AS unit (students) were also able to generate them after a proper training, and even more important, more than 80% preferred to define their PPIs graphically rather than textually, even when it was not compulsory in their assignments. These results are encouraging and coincide with the results obtained in the experiment performed in Mora et al. (2011), where graphical and textual models of software measurement were compared, demonstrating the graphical model to be more understandable and modifiable than the textual one.

RQ₅ – Traceability Findings related to this aspect were found in the context of the AHS case, during the evolution scenario described above. In this case, the business process update was meant to change the position of certain activities, to add some new ones to depict some unrepresented exceptions to the normal flow, and to refine some other aspects like the data flow. These changes barely affected the previously defined PPIs, except for the cases in which the relocated activities had connections to PPIs. In these cases, the graphical editor maintained those connections between the PPIs and the relocated activities, which allowed an immediate and automatic update of the PPIs.

8.6 Limitations

Regarding expressiveness, despite the considerable number of PPIs defined in different domains, we cannot state that all possible PPIs can be defined with VISUAL PPINOT. However, extension points were already defined in its underpinning metamodel (del Río-Ortega et al. 2013), and corresponding extensions could also be performed in the visual notation when identified.

Concerning understandability, there is a limitation to generalisation since all the VISUAL PPINOT users in our multiple-case study had technical backgrounds (most of them were engineers, but there were also mathematicians

Table 5 Feature coverage of the related work analysed

	Expressiveness	Precision/ automation	Visual representation	Traceability
Castellanos et al.	State condition and data measure not clearly definable	Yes	Yes	~
Popova et al.	Not applicable for count, state condition and data measures	Yes	Yes	Yes
Saldivar et al.	Spreadsheet-based, equivalent expressiveness to VISUAL PPINOT	Yes	No	No
Pedrinaci et al.	State condition and data measures not clearly definable	Yes	No	N.A
Wetzstein et al.	Data measures not definable	Yes	No	Yes
Momm et al.	State condition and data measures not definable	Yes	No	Yes
Costello et al.	Only cycle time measure is definable	Yes	No	Yes
González et al.	State condition measure not clearly definable	Yes	No	Yes
Korherr et al.	Not applicable for count, state condition and data measures	~	~	~
Friedenstab et al.	State condition measure not clearly definable and data measure not definable	Yes	Yes	Yes
Delgado et al.	Predefined set, not applicable for data measures	Yes	~	~

and computer scientists). It might be possible that the results regarding the ability to read and write PPI definitions in VISUAL PPINOT are different if their profiles are non-technical (from social sciences, for instance).

With respect to the automatic computation, a limitation can be found in the information required from the logs. In order to compute most common PPIs, a log with the activities carried out, with the time when they were carried out, and the process instance to which they belonged is necessary. This is the typical information provided by most process-aware information systems in the form of process event logs. However, if the information systems are not process-aware, this information might be harder to obtain. Nevertheless, we do not consider this a serious issue since, according to our experience, many information systems used in organisations nowadays are process-aware. Furthermore, new techniques are being developed to gather this information from non-process-aware information systems (Rodríguez et al. 2012; van der Aalst 2015).

Finally, though the model integration brings some benefits like evolution traceability and the possibility to see business process models together with their associated PPIs, it can also involve some challenges regarding readability. When the number of PPIs increases, the readability of business process models including PPIs decreases. This can be alleviated using technological tools to decide, from the whole set of PPIs defined for a business process, which PPIs to show and which to hide.

9 Related Work

We have distinguished between the related work with a focus and a scope similar to the work presented in this article (i.e., PPI definition), and other works that can be

considered as complementary approaches. In addition, the support for PPI definition provided by current BPMSs is also analysed in this section.

9.1 PPI Definition Approaches

The measurement of business process performance has triggered many research efforts, yielding a variety of different approaches. Many of them propose languages and architectures for describing and monitoring PPIs, some from a general point of view such as Castellanos et al. (2005), Popova and Sharpanskykh (2010) or Saldivar et al. (2016), whereas others are specific to certain contexts such as semantic business processes (Pedrinaci et al. 2008; Wetzstein et al. 2008) or service-oriented architectures (Momm et al. 2007).

In general, VISUAL PPINOT improves not only the expressiveness of those works, but also the visual representation of business process-PPI links. Regarding expressiveness, VISUAL PPINOT, i.e., the PPINOT metamodel, allows PPI definitions which are not possible to express in other approaches, especially those related to states or data, as analysed in del Río-Ortega et al. (2013) and summarised in Table 5.

Regarding the representation of business process-PPI links, the aforementioned approaches mainly focus on semantics and hardly on syntax details that could ease the understanding of PPI definitions. Costello and Molloy (2009) and González et al. (2009) are some of the authors who have already identified this problem and have made some proposals to improve the comprehension of PPI definitions and bring them closer to non-technical stakeholders. Their proposals include a PPI visual model and “a language for high-level monitoring, measurement data collection and control of business processes”, although

they actually present textual (e.g., XML-based) mechanisms that require a certain level of technical knowledge and, in the case of Costello and Molloy (2009), are only focused on time measures.

On the other hand, Korherr and List (2007) have extended the BPMN and EPC metamodels in order to define business process goals and performance measures, including cost, quality, and cycle time measures, although only cycle time measures are explicitly connected to business process elements and visually modelled. In contrast, VISUAL PPINOT provides a visual representation and an explicit connection with the business process for all of the allowed measures and, in addition, considers all the information required to define and calculate them.

With a level of expressiveness similar to VISUAL PPINOT, Friedenstab et al. (2012) have proposed a graphical notation for BAM. When compared to it, VISUAL PPINOT presents some improvements such as the definition of PPIs related to data, the explicit and visual representation of connectors to the BPMN elements, the set of principles for obtaining cognitively effective visual notations taken into account in its development, and the supporting tool and subsequent validation of our proposal.

Finally, another very closely related work is the one presented in Delgado et al. (2014), where an execution measurement model for business processes realised by services is proposed based on an existing software measurement ontology (García et al. 2009). This model provides a predefined set of generic execution measures organised according to the four dimensions of the *Devil's quadrangle* (Jansen-Vullers et al. 2008; Dumas et al. 2013), i.e., time, cost, quality, and flexibility, together with measures for lean and service executions. It also proposes a method and a tool to guide and support execution measurement and the subsequent business process improvement.

In contrast, VISUAL PPINOT allows for the definition of domain-specific, user-defined PPIs, apart from the predefined ones proposed in Delgado et al. (2014); these PPIs are visually modelled together with the business process; and VISUAL PPINOT is intended for defining measures on any type of business process, including those realised partially or exclusively by humans. Actually, VISUAL PPINOT can complement the work in Delgado et al. (2014) by broadening the spectrum not only of the business processes to be measured, but also of the measures themselves.

Table 5 summarises this analysis of the related literature. In particular, we have evaluated to what extent the related approaches that are directly comparable to VISUAL PPINOT fulfil or cover the different features identified as desirable and evaluated in our approach. A \sim sign in a cell indicates that that particular approach addresses that feature partially.

9.2 Other Complementary Approaches

In the context of frameworks for measurement dimensions, a number of works have been proposed, such as Cross and Lynch (2007), Keegan et al. (1989), Brignall et al. (1991), Kaplan and Norton (1992), Brand and Kolk (1995), or Adams and Neely (2002), but the aforementioned *Devil's quadrangle* and its four dimensions (time, cost, quality, and flexibility) has proven to be the most suitable for business processes (Jansen-Vullers et al. 2008; Dumas et al. 2013). The main difference between these frameworks and VISUAL PPINOT (and other similar approaches such as the ones discussed in the previous section) is that while VISUAL PPINOT focuses on “*how*” the indicators are measured – i.e., how the information required for their computation can be obtained from the process – the frameworks focus more on “*what*” is measured by the indicators. One of the consequences of this difference is the need of using *proxies* (which can be defined in VISUAL PPINOT) for the operationalization of dimensions that cannot be directly measured such as quality or flexibility, e.g., using the number of complaints received or the number of items returned as proxies of the quality of a purchase process (Jansen-Vullers et al. 2008). This is the reason why VISUAL PPINOT does not include specifically quality or flexibility measures.

Other works are focused on one particular dimension of the *Devil's quadrangle*. With respect to the *time* dimension, there are some timed-BPMN proposals such as Lanz and Reichert (2014), Cheikhrouhou et al. (2013), or Mendoza et al. (2011). However, these approaches are focused on modelling temporal constraints in the process flow instead of defining measures, i.e., they restrict the process behaviour according to certain time constraints. Therefore, they could be seen as an extension of the time event that BPMN and other similar notations include, and the definition of measures in VISUAL PPINOT could be reused as a mechanism to specify those restrictions. In fact, after the analysis of the time patterns presented in Lanz et al. (2014), VISUAL PPINOT covers 8 out of the 10 patterns. The two uncovered patterns are TP4 (*Fixed Date Element*) and TP5 (*Schedule Restricted Element*). To cover TP4, we would need an extension to the PPINOT metamodel already identified in del-Río-Ortega et al. (2015). As for TP5, it is not covered straightforwardly since schedules are not artefacts present as part of the business process model; however, if this information was provided within a data object, it could then be expressed in VISUAL PPINOT.

Regarding the approaches related to the *cost* dimension (Magnani and Montesi 2007; Sampath and Wirsing 2009, 2011), their focus is on obtaining cost estimations of processes based on past executions, but they are not intended to compute actual values of the instances that are currently running, which is the goal of VISUAL PPINOT.

Table 6 PPI definition support by current BPMs

	Expressiveness	Visual representation	Traceability	Automatic computation	Reports
IBM BPM	Predefined and user defined	N.A.	N.A.	Yes	User defined
ARIS PPM	Predefined and user defined	Yes	N.A.	Yes	User defined
BizAgi Modeler	Predefined	No	–	Yes	Predefined
Bonita Open Solution	Predefined	No	–	Yes	Predefined
Adonis Comm. Edition	Predefined	No	–	Yes	Predefined
Oracle BPM Suite 12c	Predefined	No	–	Yes	Predefined
TIBCO Business Studio	Predefined	No	–	Yes	Predefined
Camunda	Predefined (duration)	No	–	Yes	Predefined

Other approaches that are somehow related to the context of this article are those that try to integrate risks with business processes. Risk-aware business process management seeks to reason about the likelihood and the impacts of the occurrence of various types of risks, such as security or regulatory non-compliance (Suriadi et al. 2014). Obtaining undesired values for certain PPIs can be understood as a type of risk, therefore, PPI definitions in VISUAL PPINOT can be used as the input to many of the existing approaches to define risks, such as Jakoubi et al. (2010) or Rosemann and zur Muehlen (2005).

It is also worth mentioning the recently released DMN standard for decision modelling (OMG 2016). Although it serves a different purpose, it is also concerned with calculating process-related measures. VISUAL PPINOT can complement DMN, especially in the context of decision logic modelling. Since decision rules in DMN are defined through a number of expressions that are evaluated using input variables (that can also be themselves expressions), VISUAL PPINOT can be used to define those process-related input variables.

9.3 Tools

The evaluation of the tool support for PPI definition in current BPMSs is based on the analysis performed and presented in Saldivar et al. (2016) and on further analysis we have carried out. The most representative commercial tools at the time of writing as well as open-source solutions have been considered for the evaluation. In particular, we have considered IBM Business Process Manager (IBM 2009), ARIS Process Performance Manager (Scheer et al. 2006), BizAgi Modeler (BizAgi 2015), Bonita Open Solution (Bonitasoft 2011), Adonis Community Edition (BOC Group 2015, Oracle Business Process Management Suite 12c (Oracle 2014), TIBCO Business Studio (TIBCO 2014), and Camunda (Camunda 2014).

Because not all the tools could be installed for their study, we have also based our analysis on the official documentation published by each solution. Sometimes, the

documentation provided insufficient information to draw a conclusion about a particular feature. The results of the analysis are summarised in Table 6 and described in the following paragraphs. Most of the analysed tools provide predefined standard measures such as duration, idle time, cost, throughput, or resource utilisation. IBM BPM and ARIS PPM are the exception, since they allow the user to define their own measures although with some restrictions. Regarding the former, it is only possible to define measures using arithmetic operations on process variables. As for the latter, it is possible to define a wide range of measures except for data measures, as long as it can be deduced from the documentation.

With respect to visual representation, only ARIS PPM offers a partial graphical mechanism for PPI definition through measurement points defined over EPC models. However, a comprehensive graphical definition of PPIs is not possible according to the available documentation, since only measurement points can be graphically depicted, whereas the rest of the elements involved in a PPI have to be described textually using some forms provided by its user interface.

It makes sense to take traceability between business processes and PPIs into consideration only in tools where user-defined measures are allowed, i.e., ARIS PPM and IBM BPM, but as far as their documentations describe, it is not clear whether they support it. Regarding the possibility to automatically compute PPI values, all of the analysed tools offer this feature. In addition, most of them allow the generation of reports, either predefined or user-defined, depending on whether predefined or user-defined measures are available, respectively.

10 Conclusions

The work presented in this article is a contribution to the process performance management field. The graphical notation proposed, VISUAL PPINOT, together with the supporting tool described, PPINOT TOOL SUITE, can be considered

very novel artefacts since prior work regarding this topic is not abundant and has not holistically addressed the issues that have driven VISUAL PPINOT development, namely, expressiveness, precision, amenability to automation, platform independence, understandability by all stakeholders, and traceability to the business process.

Expressiveness: VISUAL PPINOT has proved to be more expressive than current research proposals and industrial tools in terms of the PPIs that can be defined. Furthermore, it is sufficiently expressive to define more than 400 PPIs during its evaluation in the multiple-case study.

Precision: VISUAL PPINOT is precise by design since it is based on the PPINOT metamodel. Furthermore, the application of VISUAL PPINOT to the definition of a number of PPIs showed how a significant number of PPIs defined in a natural language were ambiguous and required clarification. However, defining PPIs with VISUAL PPINOT forced the user to precisely define them and explicitly link them with the elements of the related process model.

Automation and platform independence: The implementation of the PPINOT TOOL SUITE together with the results of the case study showed that PPIs defined in VISUAL PPINOT are amenable to automation and remove (or at least reduce drastically) the need to redefine them for computation. Moreover, VISUAL PPINOT is platform-independent, as shown in its application in the case study and in the implementation of the PPINOT TOOL SUITE, thus enabling PPI computation for different platforms.

Understandability: VISUAL PPINOT is based on a number of principles aimed at designing cognitively effective visual notations. Furthermore, the users in our multiple-case study were able to read, validate, and define PPIs. This lets us conclude that understandability is not a problem that could hinder its use.

Traceability: VISUAL PPINOT provides an inherent traceability between PPI definitions and business process models, promoting their coherence during maintenance, as shown during the case study.

Finally, several directions have been identified for our ongoing and future work. An ongoing work on the automation of PPI management is the application of machine learning and natural language processing techniques for the automatic transformation of natural language PPI definitions into VISUAL PPINOT models that are directly amenable to automated computation (van der Aa et al. 2016). We are also working on extending VISUAL PPINOT for managing variability on PPIs (Estrada-Torres et al. 2016), and, in the near future, we intend to extend VISUAL PPINOT to allow the definition of resource-aware PPIs (del-Río-Ortega et al. 2013b).

Acknowledgements We would like to thank the IT Department of the AHS and the IC Service of the University of Seville for kindly

providing us their internal business process models, PPIs, and time; we also would like to thank Prof. M. Weske, and Dr. J.A. Parejo for their helpful comments in earlier versions of this article. This work has been partially supported by the European Commission (FEDER), Spanish Government, under the CICYT project BELI (TIN2015-70560-R), and Andalusian Government under the project COPAS (P12-TIC-1867).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Adams C, Neely A (2002) Prism reform. *Financ Manag* 5:28–31
- Andrikopoulos V, Benbernou S, Bitsaki M, Danylyevych O, Hacid M, van den Heuvel W, Karastoyanova D, Kratz B, Leymann F, Mancipopi M, Mokhtari K, Nikolaou C, Papazoglou M, Wetstein B (2008) Survey on business process management. Deliverable of S-cube. <http://www.s-cube-network.eu>. Accessed 21 March 2017
- BizAgi (2015) Bizagi process modeler user's guide. http://download.bizagi.com/docs/modeler/2904/en/Modeler_user_Guide.pdf. Accessed 21 March 2017
- BOC Group (2015) Adonis community edition getting started series. <http://en.adonis-community.com/welcome/webinars-and-tutorials/adonisce-getting-started-series>. Accessed 21 March 2017
- Bonitasoft (2011) Bonita open solution simulation guide. <http://www.bonitasoft.com/system/files/download/bos-5.6-simulation-guide.pdf>. Accessed 21 March 2017
- Brambilla M, Cabot J, Wimmer M (2012) Model-driven software engineering in practice, 1st edn. Morgan & Claypool Publishers, San Rafael
- Brand N, Kolk H (1995) Workflow analysis and design. *Bedrijfswetenschappen*. Kluwer, Dordrecht
- Brignall T, Fitzgerald L, Johnston R, Silvestro R (1991) Performance measurement in service businesses. *Manag Account* 69(10):34–36
- Camunda (2014) Process instance history view and instance duration report. <https://docs.camunda.org/manual/7.5/webapps/cockpit/bpmn/process-history-views/#process-instance-history-view>. Accessed 21 March 2017
- Castellanos M, Casati F, Shan MC, Dayal U (2005) ibom: a platform for intelligent business operation management. In: Proceedings of the 21st international conference on data engineering, pp 1084–1095
- Chase G, Rosenberg A, Omar R, Taylor J, Rosing M (2011) Applying real-world BPM in an SAP environment. Galileo Press, Baltimore
- Cheikhrouhou S, Kallel S, Guermouche N, Jmaiel M (2013) Toward a time-centric modeling of business processes in bpmn 2.0. In: Proceedings of IIWAS 2013, pp 154–163
- Costello C, Molloy O (2009) Building a process performance model for business activity monitoring. In: Wojtkowski W, Wojtkowski G, Lang M, Conboy B, Barry C (eds) Information systems development. Springer, Heidelberg, pp 237–248
- Cross KF, Lynch RL (2007) The “smart” way to define and sustain success. *Natl Product Rev* 8(1):23–33
- Decker G, Overdick H, Weske M (2008) Oryx—an open modeling platform for the bpm community. In: BPM 2008, pp 382–385

- del-Río-Ortega A, Cabanillas C, Resinas M, Ruiz-Cortés A (2013a) PPINOT tool suite: a performance management solution for process-oriented organisations. In: Proceedings of ICSOC 2013, pp 675–678
- del-Río-Ortega A, Resinas M, Cabanillas C, Cortés AR (2013b) Defining and analysing resource-aware process performance indicators. In: Proceedings of CAiSE'13 Forum, pp 57–64
- del-Río-Ortega A, Gutiérrez AM, Durán A, Resinas M, Ruiz-Cortés A (2015) Modelling service level agreements for business process outsourcing services. In: Proceedings of CAiSE 2015, pp 485–500
- del Río-Ortega A, Resinas M, Cabanillas C, Ruiz-Cortés A (2013) On the definition and design-time analysis of process performance indicators. *Inf Syst* 38(4):470–490
- del Río-Ortega A, Resinas M, Durán A, Ruiz-Cortés A (2016) Using templates and linguistic patterns to define process performance indicators. *Enterp Inf Syst* 10(2):159–192
- Delgado A, Weber B, Ruiz F, de Guzmán IGR, Piattini M (2014) An integrated approach based on execution measures for the continuous improvement of business processes realized by services. *Inf Softw Technol* 56(2):134–162
- Dresner H (2003) Business activity monitoring: BAM architecture. In: Gartner Symposium ITXpo. Ed. Gartner Group, Cannes
- Dumas M, Rosa ML, Mendling J, Reijers HA (2013) Fundamentals of business process management. Springer, Heidelberg
- Estrada-Torres B, Del-Río-Ortega A, Resinas M, Ruiz-Cortés A (2016) Identifying variability in process performance indicators. In: Proceedings of BPM Forum 2016, pp 91–107
- Friedenstab JP, Janiesch C, Matzner M, Muller O (2012) Extending BPMN for business activity monitoring. In: Hawaii international conference on system sciences 2012, pp 4158–4167
- García F, Ruiz F, Calero C, Bertoa MF, Vallecillo A, Mora B, Piattini M (2009) Effective use of ontologies in software measurement. *Knowl Eng Rev* 24(1):23–40
- González O, Casallas R, Deridder D (2009) MMC-BPM: a domain-specific language for business processes analysis. *Bus Inf Syst* 21:157–168
- Grigori D, Casati F, Castellanos M, Dayal U, Sayal M, Shan MC (2004) Business process intelligence. *Comput Ind* 53(3):321–343. doi:10.1016/j.compind.2003.10.007
- Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. *MIS Q* 28(1):75–105
- IBM (2009) Tutorials and samples for websphere business modeler version 6.2. <http://www-01.ibm.com/support/docview.wss?uid=swg27013902>. Accessed 21 March 2017
- Jakoubi S, Tjoa S, Goluch S, Kitzler G (2010) Risk-aware business process management. Establishing the link between business and security. In: Proceedings of complex intelligent systems and their application 2010, pp 109–135
- Jansen-Vullers MH, Kleingeld PAM, Netjes M (2008) Quantifying the performance of workflows. *Inf Syst Manag* 25(4):332–343
- Kaplan RS, Norton DP (1992) The balanced scorecard: measures that drive performance. *Harv Bus Rev* 70(1):71–79
- Keegan D, Eiler R, Jones C (1989) Are your performance measures obsolete? *Manag Account* 70(12):45–50
- Kettinger WJ, Teng JTC (1997) Business process change: a study of methodologies, techniques, and tools. *MIS Q* 21(1):55–80
- Korherr B, List B (2007) Extending the epc and the bpmn with business process goals and performance measures. In: Proceedings of ICEIS 2007, pp 287–294
- Kronz A (2006) Managing of process key performance indicators as part of the aris methodology. Corporate performance management: aris in practice. Springer, Heidelberg, pp 31–44
- Lanz A, Reichert M (2014) Dealing with changes of time-aware processes. In: Proceedings of BPM 2014, pp 217–233
- Lanz A, Weber B, Reichert M (2014) Time patterns for process aware information systems. *Requir Eng* 19(2):113–141
- Lethbridge TC, Sim SE, Singer J (2005) Studying software engineers: data collection techniques for software field studies. *Empir Softw Eng* 10(3):311–341
- López-Fernández JJ, Cuadrado JS, Guerra E, de Lara J (2015) Example-driven meta-model development. *Softw Syst Model* 14(4):1323–1347
- Magnani M, Montesi D (2007) Bpmn: how much does it cost? An incremental approach. In: Proceedings of BPM 2007, pp 80–87
- Mendoza LE, Capel I, Pérez MA (2011) A formalization proposal of timed bpmn for compositional verification of business processes. In: Proceedings of ICEIS 2013, pp 388–403
- Momm C, Malec R, Abeck S (2007) Towards a model-driven development of monitored processes. In: Proceedings of Tagung Wirtschaftsinformatik 2007, pp 319–336
- Moody DL (2009) The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans Softw Eng* 35(6):756–779
- Moody DL, Heymans P, Matulevicius R (2009) Improving the effectiveness of visual representations in requirements engineering: an evaluation of i* visual syntax. In: Proceedings of RE 2009, pp 171–180
- Mora B, García F, Ruiz F, Piattini M (2011) Graphical versus textual software measurement modelling: an empirical study. *Softw Qual J* 19:201–233
- Office of Government Commerce (2007) Information technology infrastructure library (ITIL) v3
- OMG (2011) Business process model and notation (BPMN) v2.0. <http://www.omg.org/spec/BPMN/2.0/PDF>. Accessed 21 March 2017
- OMG (2016) Decision model and notation (DMN)v1.1. <http://www.omg.org/spec/DMN/1.1/PDF>. Accessed 21 March 2017
- Oracle (2014) What's new in oracle bpm suite 12c. <http://www.oracle.com/technetwork/middleware/bpm/overview/bpm-12c-new-features-wp-2235510.pdf>. Accessed 21 March 2017
- Pedrinaci C, Lambert D, Wetzstein B, van Lessen T, Cekov L, Dimitrov M (2008) Sentinel: a semantic business process monitoring tool. In: International workshop on ontology-supported business intelligence, pp 26–30
- Peffer K, Tuunanen T, Rothenberger M, Chatterjee S (2007) A design science research methodology for information systems research. *J Manag Inf Syst* 24(3):45–77
- Popova V, Sharpankykh A (2010) Modeling organizational performance indicators. *Inf Syst* 35(4):505–527
- Rodríguez C, Engel R, Kostoska G, Daniel F, Casati F, Aimar M (2012) Eventifier: extracting process execution logs from operational databases. In: Proceedings of the demonstration track of BPM 2012, vol 940, pp 17–22
- Rosemann M, zur Muehlen M (2005) Integrating risks in business process models. In: Proceedings of ACIS 2005, p Article 50
- Rumbaugh JE (1996) Notation notes: principles for choosing notation. *J Object Oriented Program* 9(2):11–14
- Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. *Empir Softw Eng* 14(2):131–164
- Runeson P, Höst M, Rainer A, Regnell B (2012) Case study research in software engineering—guidelines and examples. Wiley, Hoboken
- Saldívar J, Vairetti C, Rodríguez C, Daniel F, Casati F, Alarcón R (2016) Analysis and improvement of business process models using spreadsheets. *Inf Syst* 57:1–19
- Sampath P, Wirsing M (2009) Computing the cost of business processes. In: Proceedings of UNISCON 2007, pp 178–183
- Sampath P, Wirsing M (2011) Evaluation of cost based best practices in business processes. In: Proceedings of EMMSAD 2011, pp 61–74

- Sánchez-Jerez AB (2012) Analysis and representation of key performance indicators applied to a real organization. Master's thesis, Univ. Sevilla
- Scheer A, Jost W, Heß H, Kronz A (2006) Corporate performance management: aris in practice. Springer, Heidelberg
- Shahin A, Mahbod MA (2007) Prioritization of key performance indicators: an integration of analytical hierarchy process and goal setting. *Int J Product Perform Manag* 56:226–240
- Suriadi S, Weiss B, Winkelmann A, ter Hofstede A, Adams M, Conforti R, Fidge C, La Rosa M, Ouyang C, Rosemann M, Pika A, Wynn M (2014) Current research in risk-aware business process management—overview, comparison, and gap analysis. *Commun Assoc Inf Syst* 34(52):933–984
- TIBCO (2014) Tibco business studio. <https://docs.tibco.com/pub/business-studio-bpm-edition/3.9.0/doc/html/index.html>. Accessed 21 March 2017
- van der Aa H, del-Río-Ortega A, Resinas M, Leopold H, Ruiz-Cortés A, Mendling J, Reijers HA (2016) Narrowing the business-IT gap in process performance measurement. In: Proceedings of CAiSE 2016, pp 543–557
- van der Aalst WMP, van Dongen BF, Herbst J, Maruster L, Schimm G, Weijters AJMM (2003) Workflow mining: a survey of issues and approaches. *Data Knowl Eng* 47(2):237–267. doi:10.1016/S0169-023X(03)00066-1
- van der Aalst WMP, Pesic M, Song M (2010) Beyond process mining: from the past to present and future. In: Proceedings of CAiSE 2010, pp 38–52
- van der Aalst WMP (2015) Extracting event data from databases to unleash process mining. In: BPM—driving innovation in a digital world. Management for Professionals, Springer, Heidelberg, pp 105–128
- Verner JM, Sampson J, Tosic V, Bakar NAA, Kitchenham B (2009) Guidelines for industrially-based multiple case studies in software engineering. In: Proceedings of RCIS 2009, pp 313–324
- Wetzstein B, Ma Z, Leymann F (2008) Towards measuring key performance indicators of semantic business processes. *Bus Inf Syst* 7:227–238