PPINOT: A Tool for the Definition and Analysis of Process Performance Indicators[*]

Adela del-Río-Ortega, Cristina Cabanillas, Manuel Resinas, and Antonio Ruiz-Cortés

Universidad de Sevilla, Spain
{adeladelrio,cristinacabanillas,resinas,aruiz}@us.es

Abstract. A key aspect in any process-oriented organisation is the evaluation of process performance for the achievement of its strategic and operational goals. Process Performance Indicators (PPIs) are a key asset to carry out this evaluation, and, therefore, having an appropriate definition of these PPIs is crucial. After a careful review of the literature related and a study of the current picture in different real organisations, we conclude that there not exists any proposal that allows to define PPIs in a way that is unambiguous and highly expressive, understandable by technical and non-technical users and traceable with the business process (BP). Furthermore, it is also increasingly important to provide these PPI definitions with support to automated analysis allowing to extract implicit information from them and their relationships with the BP. In this work we present PPINOT, a tool that allows the graphical definition of PPIs together with their corresponding business processes, and their subsequent automated analysis.

1 Introduction

It is increasingly important to evaluate the performance of business processes (BPs), since it helps organisation to define and measure progress towards their goals. Performance requirements on BPs can be specified by means of Process Performance Indicators (PPIs).

According to Franceschini *et al.* [1] and based on the conclusions drawn in our previous works [2, 3], four critical elements for indicators can be identified: (1) their *definition*, that should be unambiguous and complete; (2) *understandability*, PPIs should be understood and accepted by process managers and employees; (3) *traceability with the BP*, enabling to maintain coherence between both assets, BP models and PPIs; and (4) *possibility to be automatically analysed*, allowing thus not only to gather the information required to calculate PPI

[*] This work has been partially supported by the European Commission (FEDER), Spanish Government under the CICYT project SETI (TIN2009-07366); and projects THEOS (TIC-5906) and ISABEL (P07-TIC-2533) funded by the Andalusian local Government.

M.V. de Castro, J.M. Gómez, L. Iribarne (Eds.): JCIS'2012, pp. 125-128, ISBN:978-84-15487-26-5.
Jornadas Sistedes'2012, Almería 17-19 sept. 2012, Universidad de Almería.

values, but also to infer knowledge to answer questions like *what are the business process elements related to PPI P?* (a set of 3 analysis operation families amenable to be performed on PPI definitions can be found in [3]).

We address these issues by providing PPINOT tool, that allows the graphical definition of PPIs together with their corresponding BPs, and their subsequent automated analysis. To the best of our knowledge, there not exists any similar tool for the definition of PPIs. Concretely, we can highlight the following PPINOT features that give the novelty to our proposal:

BPMN 2.0 compliant. PPIs can be defined over BP diagrams (BPDs) previously modelled using the de facto standard BPMN 2.0.

Graphical definition of PPIs. PPINOT supports the graphical definition of PPIs using a graph-based graphical notation that is easily understandable by non-technical users, at the same time that it is supported by a metamodel that assures the precise and complete definition of PPIs.

Automated analysis of PPIs. The aforementioned metamodel support allows to automatically formalise PPI definitions using Description Logics, enabling to obtain information about the way PPIs and BP elements influence each other. Concretely, two kinds of analysis operations are supported in the current version of PPINOT: (I) *BPElements involved*, that allows to answer the question *Given a PPI P, Which are the process model's elements involved?*, this information is very useful in many scenarios, like for instance when a PPI must be replaced with others (maybe because it is very costly to obtain its value) and it is necessary to assure that every element of the BP that was measured before is measured in the new case; and (II) *PPIs associated to BPElement*, that allows to answer the question *Given a BPElement E, Which are the PPIs associated or applied to them?*, this information can assist during the evolution of BPs, e.g. if a part of the BP has evolved and is modified, for instance if an activity is deleted, this analysis allows to identify which PPIs will be affected and should be updated.

Figure 1 shows a screenshot of PPINOT tool in use.

2 PPINOT Tool: Definition and Structure

The structure of PPINOT is depicted in the component model of Figure 2. This tool has been implemented as an extension of the ORYX platform. Concretely we have provided a new stencil set with the shapes and connectors of the PPI graphical notation (PPINOT Oryx stencilset), that extends the existing one of BPMN. In addition, a new plugin called *PPINOT analyser* has been developed. It supports the formalisation of PPI graphical definitions to DL, and the subsequent analysis. Furthermore, it has been designed as a reusable component so that it can be integrated into other environments than Oryx without any change. In the following we describe the way PPINOT works.

A BPD is defined in *Oryx* [4]. Then, the set of PPIs is defined over such BPD using the *PPINOT Oryx Stencilset* . An xml file containing all this information (BPD + PPIs) is obtained from Oryx (through the *PPINOT Service*) and

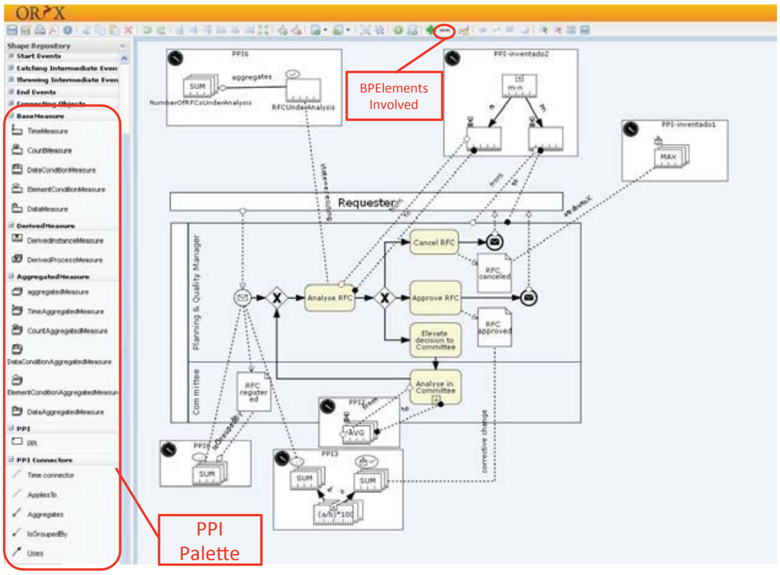

Fig. 1. PPINOT screenshot

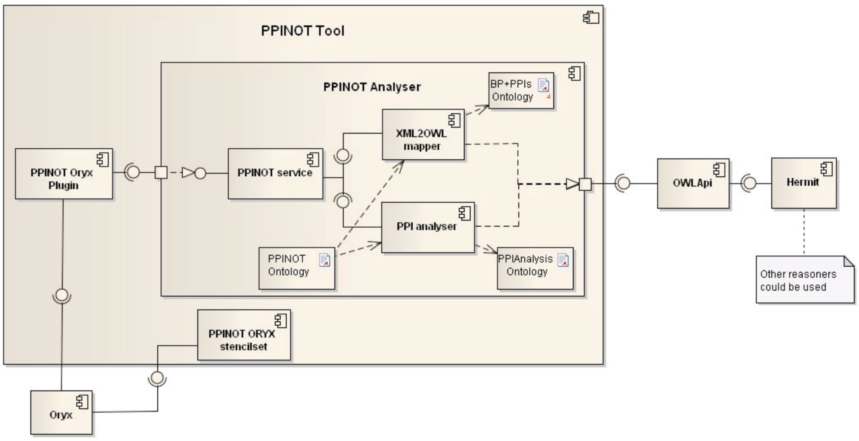

Fig. 2. PPINOT component model

mapped to OWL using the PPINOT ontology described in [2]. This is done by the *XML2OWL Mapper* and produces the OWL file BP+PPIs Ontology, which is the target file of the DL reasoner (in this case *HermiT*) used by the *PPI Analyser*, so the proper DL operations are executed on the PPI definitions of this OWL file to infer the information required. Finally, an OWL file containing the information required (elements involved in a PPI definition or PPIs associated to a given BP element) is automatically generated (PPIAnalysisOntology).

3 PPINOT Structure

In this section we provide the steps we will follow in order to try PPINOT Tool.

1. In oder to access the tool a firefox window must be opened and the url `http://labs.isa.us.es:8081/backend/poem/repository` accessed.
2. There are several BPDs available or a new on can be created using the BPMN 2.0 stencil set.
3. Once the BP is open, the PPI extension must be added in order to be able to define the set of PPIs corresponding to that process.
4. Using the PPI palette, these PPIs must be modelled.
5. The following step is elated to the automated analysis of PPIs. In this case, we will try the *BPElements involved* operation. In order to try it, the *PPINOT query* plugin must be selected.
6. A measureDefinition from the list of all measureDefinitions shown by the system (corresponding to the PPIs defined) has to be selected.
7. The system provides a list with all the BP elements involved in the selected measureDefinition, and hence, involved in the corresponding PPI.

4 Conclusions

In this work we present PPINOT, an easy-to-use tool for the definition and automated analysis of PPIs. PPINOT satisfies the necessity of a tool that, on the one hand, fills the visual gap between BPs and their corresponding PPIs by allowing the modelling of such PPIs together with the corresponding BP; and on the other hand, automates the error-prone and tedious task of analyse PPIs. We plan to extend PPINOT in order to support the whole set of analysis operations identified in [3]. Another direction we are working on is to use PPINOT in order to obtain the PPIs' values from the execution of BPs in a BPMS.

References

1. Franceschini, F., Galetto, M., Maisano, D.: Management by Measurement: Designing Key Indicators and Performance Measurement Systems. Springer Verlag (2007)
2. del Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: Defining process performance indicators: An ontological approach. In: Proceedings of the 18th International Conference on Cooperative Information Systems (CoopIS). OTM 2010, Part I. (October, 2010) 555–572
3. del Río-Ortega, A., Resinas, M., Ruiz-Cortés, A.: PPI definition and automated design-time analysis. Technical report, Applied Software Engineering Research Group (2012)
4. Decker, G., Overdick, H., Weske, M.: Oryx - an open modeling platform for the bpm community. In: BPM. (2008) 382–385