Towards a Framework for KPI Evolution

Eladio Domínguez[1][a], Beatriz Pérez[2][b], Ángel L. Rubio[2][c] and María A. Zapata[1][d]
[1]Dpto. de Informática e Ingeniería de Sistemas, Univ. de Zaragoza, Zaragoza, Spain
[2]Dpto. de Matemáticas y Computación, Univ. de La Rioja, La Rioja, Spain

Keywords: KPIs, Performance Measurement, Evolution Framework.

Abstract: Key Performance Indicators (KPIs) are becoming essential elements for measuring business performance. In recent years, KPIs management has been the subject to sustained interests for researchers and practitioners alike, deriving into a large research corpus of approaches addressing aspects in matters as varied as modelling, maintenance or expressiveness of KPIs. In particular, since both businesses and processes have to be adapted to ever-changing requirements, the KPIs that measure their performance must evolve accordingly. However, based on a previous review of the literature, we found that little attention has been paid to the provision of mechanisms to manage KPIs evolution. Our long-term research goal is to provide a fully proposal for supporting KPIs evolution management. In this position paper, we present the first ideas of a conceptual framework for addressing this issue, proposing a pattern-driven KPI evolution specification and a KPI evolution meta-model made up of two interconnected views. Our proposal is general enough to be applied regardless of the specific KPIs management approach being used.

1 INTRODUCTION

Today's data society provides an opportunity for enterprises and institutions to access a wealth of information of many different types. Since continuous business improvement is a must-have goal for any organization, proper use of available information can lead to changes in procedures, processes and systems. Changes can be triggered by different causes, such as modifications of the business strategy, correction of detected errors, legislative updates or technological adaptations (Cognini et al., 2018).

In order to determine the aspects that must be improved, it is necessary to measure the performance of business activities. That is the reason why the definition and use of Key Performance Indicators (KPIs) within information systems has become widespread in any business area. In view of the relevance of KPIs, it is natural that research on this topic is intense and extensive. Despite this (or precisely because of it), there does not exist a standard model for KPIs definition and use. In (Domínguez et al., 2019) we addressed this problem, approaching our solution by performing a literature review and defining a *KPI management taxonomy*, which captures, in a unified way, the overall properties of KPIs gathered in the literature. This taxonomy is unfolded in five dimensions, each one answering a different question: *What is measured by a KPI?*, *What features are considered?*, *What is a KPI measured for?*, *What artifacts are used?*, and *What are the characteristics of each approach?* (for details, we refer to (Domínguez et al., 2019)).

One of the findings of this KPIs taxonomy is that very few works or approaches consider KPIs customization and evolution management as one of their characteristics. If, as explained, any area of business management needs to introduce changes to adapt business to new circumstances, KPIs that measure the performance (of the activities) of that area should also be modified accordingly.

Making use of the terminology that is presented in (Reichert and Weber, 2012), in the context of Process-Aware Information Systems we could speak of *KPI flexibility* to refer to any aspect related to modifications that can be considered in performance indicators. Reichert et al. (Reichert and Weber, 2012) differentiate between four different variants for the notion of process flexibility: *variability* (implies the existence of different process variants); *looseness*

[a] https://orcid.org/0000-0002-3270-899X
[b] https://orcid.org/0000-0001-9235-7311
[c] https://orcid.org/0000-0002-3052-6404
[d] https://orcid.org/0000-0002-9531-1586

Domínguez, E., Pérez, B., Rubio, Á. and Zapata, M.
Towards a Framework for KPI Evolution.
DOI: 10.5220/0009465604630469
In *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2020)*, pages 463-469
ISBN: 978-989-758-421-3

(processes are not fully pre-specified); *adaptation* (process instances are adapted to cope with emerging events); and *evolution* (the capacity of modifying a process permanently to accommodate evolving needs).

Although all these variants could be transferred to the KPIs domain, we focus on the specific issue of *KPIs evolution*. In view of the large number of different approaches to KPIs management, and that few of them consider KPI evolution aspects (Domínguez et al., 2019), our long-term research goal is to build on a *KPI evolution framework* that can be used with whatever particular KPIs approach. This is certainly a complex aim, as evolution can be considered on many levels. On the one hand, and from a business perspective, it is necessary to determine what specific changes in KPIs need to be considered in order to achieve business goals. On the other hand, from the software engineering prism, it is necessary to have an infrastructure that allows the transfer of those changes (fixed at the business level) to an implementable conceptual model. Therefore, we intend to deal with KPI evolution in a conceptual comprehensive manner but, at the same time, bearing in mind that the implementation must be achievable. In this position paper we present some aspects of a *framework* that allows the development of conceptual models of KPI evolution. To achieve this goal, we rely on a notion of *KPI evolution pattern* and on a *KPI evolution metamodel* which includes both a structural and an execution view.

This paper is organized as follows. Next, we outline the related work of this research. In Section 3, we describe our approach for KPIs evolution, illustrating it by means of two examples of application. Finally, conclusions and further work are set out in Section 4.

2 RELATED WORK

We start by considering works that deal with KPIs evolution, and end up with works that address evolution in a general way, not linked to a specific scope.

One relevant proposal in the literature that highlights the need to consider evolution in performance measurement is (Kennerley and Neely, 2002). However, this work, on the one hand, focuses on the factors that affect that evolution and, on the other hand, raises the discussion at the level of measurement systems, and does not descend to the detail of specific indicators. In this sense, the notion of measure redefinition is considered in (Schütz and Schrefl, 2014). However, this approach is somewhat limited, as it considers that the different elements of a measure (dimensions and rule of calculation) are simply overridden when redefinitions are included. One step further, Diamantini et al. (Diamantini et al., 2016) consider the effects of classic create/update/delete operations of KPIs and, in particular, point to the need of consistency checking on the set of indicators. In a slightly different line, authors of (Friedenstab et al., 2012) present the abstract syntax of a specification language of measures that includes connections with other related elements (processes and dashboards). These connections enable the traceability of the modifications that have been made. As another example, del-Río-Ortega et al.'s work (del-Río-Ortega et al., 2013) defines indicators called Process Performance Indicators (PPIs), which are directly linked to business processes, connecting the evolution of the former to the changes made to the latter. The same authors in (Estrada-Torres et al., 2016) address the question of variability in PPIs.

The interest in considering the evolution of KPIs does not only occur in research articles, but also in concrete practical applications. For example, the Health Information and Quality Authority states in its report (HIQA, 2013) that "a plan to review the KPI at regular intervals with a view to refinement in response to stakeholder demands or improved data availability" is necessary. This report also asserts that, since "health services are continually evolving", it is fundamental that KPIs give answers to these changes. In an even more general context, IBM introduces the notion of *KPI history* (IBM Knowledge Center, 2011) in the description of its tool *Business Process Manager*. This is "an optional feature that you can use to track and analyze KPI changes over time".

With regard to works that address evolution in software engineering or information systems contexts in a general way, most of them are based on models and metamodels (Cicchetti et al., 2012; Domínguez et al., 2011; Pons et al., 2000; Ruiz et al., 2013). These strategies mainly focus on aspects such as structuring issues of evolution (Pons et al., 2000), tracing aspects (Domínguez et al., 2011; Ruiz et al., 2013), or comparing and merging of models' versioning (Cicchetti et al., 2012). In the case of KPIs evolution, conceptual modeling can also be used as a development strategy. But, as far as we know, there is no KPI evolution metamodel in the literature. We particularly claim that conceptual modeling in KPIs evolution could be considered manifold. For example, it can provide methods for gaining better understanding not only of the very complex relationships between the elements directly involved in the evolution of KPIs, but also of the implications such changes may carry (such as side effects a change in a KPI may have on other business elements or on related KPIs).

Another aspect considered when tackling business activities behaviour is to take advantage of the use of *patterns* (Ruiz et al., 2013). Patterns are recognized to be a very useful construct to capture recurring concepts within a particular domain. Aimed at being generic and reusable in a wide variety of scenarios, they are specified in a language- and technology-independent way (Russell et al., 2016). But again, we are not aware of any pattern devoted specifically to KPIs evolution.

3 SUPPORTING KPI EVOLUTION

Aimed at giving a proposal for supporting KPIs evolution, we have slightly inspired in the framework for business processes model evolution presented by (Ruiz et al., 2013). Specifically, in (Ruiz et al., 2013), authors propose a pattern-driven specification for models evolution by means of two artefacts: a pattern definition metamodel (defined based on the ideas from (Pfister et al., 2011)) and an evolution metamodel (for which they based on (Bernstein, 2003)). Starting from these ideas, in this paper, we contribute with a fine grained support for evolution aspects applied to the particular case of KPIs definition. More specifically, we have defined a *KPI evolution metamodel* which includes two complementary views (a *structural view* and an *execution view*), which are built upon concrete *KPI evolution patterns* providing a semantic description of different KPI changes. Next, we describe in detail these two main parts of our approach.

3.1 KPI Evolution Patterns

Our main goal for defining *KPI evolution patterns* is to capture commonly occurring changes in KPIs, so that they can be applied to different KPIs management proposals. In order to determine our pattern-driven specification for KPIs evolution, we have considered, not only different change pattern specification proposals (Mulyar et al., 2008; Ruiz et al., 2013; Reichert and Weber, 2012), but also the taxonomy for KPIs management we proposed in (Domínguez et al., 2019) which includes KPIs evolution aspects (del-Río-Ortega et al., 2013; Diamantini et al., 2016; Schütz et al., 2016).

As for the former works (Mulyar et al., 2008; Ruiz et al., 2013; Reichert and Weber, 2012), all consider several pattern specification elements, which are a *name*, an *abbreviation*, a *description* of its functionality, and illustrating *examples* to clarify the pattern's use. Besides these aspects, as (Mulyar et al., 2008) does, we have considered it appropriate to include a *motivation* aimed at identifying the nature of the modifications in the environment which can trigger the change. As a way of example, in Table 1 we show a KPI evolution pattern, named *Update Calculation Rule (UCR)*, describing the change of the calculation rule of a KPI. Several circumstances can motivate the need to make this type of change, from a revision in the company's policies to a simple correction in the definition of a KPI. For example, in Table 1 two situations exemplifying the use of this pattern are mentioned. One of them is related to the educational context and gathers a modification in the evaluation criteria of an on-line course. The change is intended to penalise the number of wrong answers given by the students, so that KPI 'Session Learning Adequacy per Training Step' (SLATS) (Calabro et al., 2015) calculated as the sum of the 'Correctness Score per Training Step' (CSTS) and 'Reload Cardinality per Training Step' (RCTS), is changed applying a factor of 1.5 to the number of reloads (CSTS + 1.5*RCTS). The other example is associated to sustainability and environmental standards which are changed as new regulations are established and environmental research evolves. For instance, the G4 Sustainability Reporting Guidelines (Global Reporting Initiative, 2015) were superseded by the GRI Sustainability Reporting Standards (GRI Standards) (Global Reporting Initiative, 2016). Several changes are introduced by GRI standards, among them, as an illustrative example for this article, we will use the fact that instead of calculating only the total weight of waste, two indicators are introduced in GRI standards for calculating separately hazardous and non-hazardous waste. As a consequence, the calculation rule of the total waste indicator is changed to be transformed into the sum of the two new indicators (total waste = hazardous waste + non-hazardous waste).

As for aspects specifically related to KPIs evolution, we have based on the three different characteristics (*modification*, *traceability* and *change propagation*) involved in KPIs evolution established in the taxonomy we presented in (Domínguez et al., 2019). While the first one, *modification*, is intrinsically related with the KPIs modification itself, the other two, *traceability* and *propagation*, could be considered or not. Due to this optionality and based on an idea proposed in (Reichert and Weber, 2012), we have included in the patterns specification a *design choices* section (see Table 1). Different implementation options for the *traceability* and *propagation* aspects can be specified in this section.

Traceability helps to maintain information about the different business elements related with the KPIs

Table 1: Example of a KPI evolution pattern.

Pattern UCR: Update Calculation Rule	
Description	The calculation rule of a KPI is updated
Examples	- A modification in the evaluation criteria of an on-line course, penalizing the number of wrong answers, can imply a change in the KPI "Session Learning Adequacy per Training Step" (SLATS) (Calabro et al., 2015) adding, for example, a factor of 1.5 to the number of reloads. - A company supersedes the G4 Sustainability Guidelines (Global Reporting Initiative, 2015) for the GRI Standards (Global Reporting Initiative, 2016), so that the changes introduced in the GRI Standards must be considered in the calculation of KPIs.
Motivation	Several reasons can motivate a change in the calculation rule of a KPI, for example, a change in the goals of the company, a change in the business processes or a change in other KPIs involved in the calculation rule.
Design choices	*Traceability*: - To store or not the information related with the change that provokes the calculation rule modification (goal, process, KPI, …) - The previous calculation rule is deleted or stored in order to know the evolution over time. - The previous values of the KPI are deleted or stored together with the previous calculation rule. *Propagation*: - The relationships between KPIs must be updated according to the new calculation rule and the calculation rules of other related KPIs must be rewritten accordingly. - Other KPI attributes, such as, value type, target or status options, must be analyzed to decide if they should be modified
Consistency conditions	Properties such as identity, equivalence and consistency among the KPIs must be evaluated (Diamantini et al., 2016). If the consistency is not preserved, then the modification will not take place.
Related patterns	*Influenced by*: Create KPI (Pattern CK), Delete KPI (Pattern DK), Update Calculation Rule (Pattern UCR) *Influences*: Update Target (Pattern UT), Update Calculation Rule (Pattern UCR)

(goals, processes or the KPIs themselves) and to keep a historical trace between the different evolution versions. For example, the relationship between Process Performance Indicators (PPIs) and business process elements is considered in (del-Río-Ortega et al., 2013). As for *Pattern UCR*, this type of relationship can allow us to trace the processes whose changes provoke a modification in the calculation rule of KPIs.

Propagation, for its part, is related with the incremental update mechanisms necessary for maintaining coherence among the different involved elements of KPIs. For example, redefinition of calculated measures is proposed in (Schütz et al., 2016) as consequence of customization processes. As another example, a change in the calculation rule of a KPI may involve the modification of its target value.

Other pattern specification aspects considered by (Mulyar et al., 2008) are the issues related with problems potentially encountered when using the pattern, and solutions for overcoming these problems. In the case of KPIs evolution, we consider it necessary to specify the *consistency conditions* that need to be verified and that can be compromised by an inadequate change, as well as, the actions to be performed when a change does not preserve these conditions. For example, conditions of identity, equivalence and coherence among calculation rules are considered in (Diamantini et al., 2016) together with reasoning tools to facilitate KPIs management.

Finally, to make it explicit the relations of propagation between patterns, a *related patterns* aspect is specified (also included in (Reichert and Weber, 2012)), indicating the patterns influenced by the pattern that is being defined (*influences*) and the patterns by which it is influenced (*influenced by*). For example, as we show in Table 1, the *Pattern UCR* influences the *Update Target* pattern (or *Pattern UT*) since, as said before, a change in the calculation rule of a KPI may involve the modification of its target value. Besides, the *Pattern UCR* is influenced by the *Create KPI* pattern (or *Pattern CK*) and the *Delete KPI* pattern (or *Pattern DK*) since the creation or deletion of KPIs can imply the redefinition of a calculation rule, as shown with the waste indicator example. These other patterns are described in the supplementary material (Supplementary material, 2020). Additionally, as presented in Table 1, the application of *Pattern UCR* can itself influence the application of the same pattern to another KPIs (thus, the pattern can be influenced by itself).

3.2 KPI Evolution Metamodel

As previously stated, our *KPI evolution metamodel* includes two complementary views (a *structural view* and an *execution view*), which are built upon our *KPI*

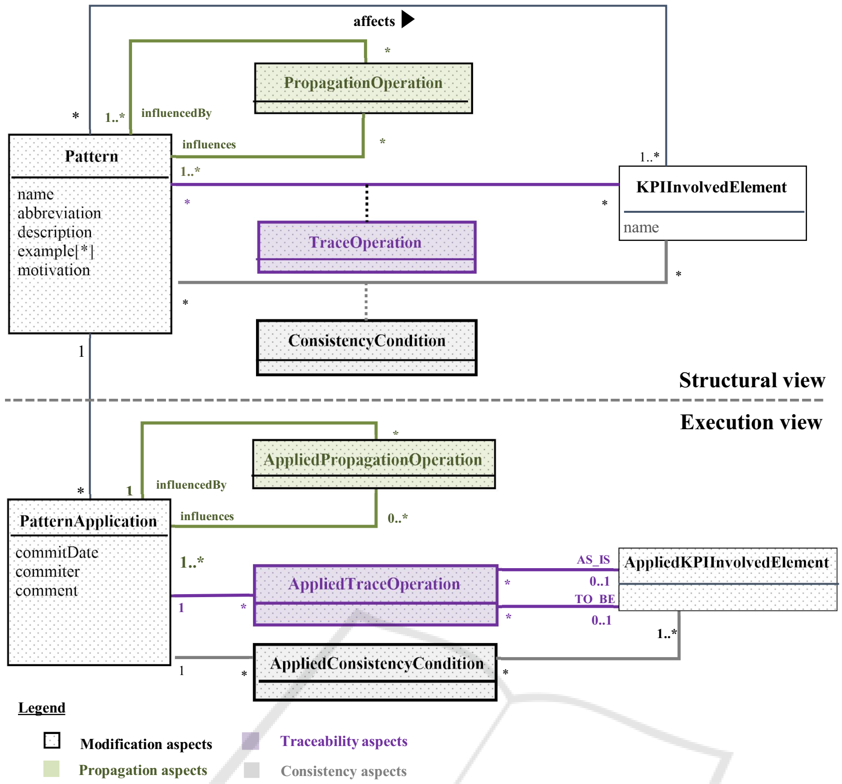

Figure 1: KPI evolution metamodel.

evolution patterns proposal. In Figure 1, we show these two views and the way they are interconnected. While the *structural view* deals with the structural aspects of the evolution as defined by the KPI evolution patterns (see top of Figure 1), the *execution view* concerns the application of such patterns, registering mainly the way in which concrete KPI instances undergo modifications as patterns are instantiated (see bottom of Figure 1). Both views reflect the different key elements identified by our *KPI evolution patterns* proposal, but from different perspectives: structural definition and applications' instances, respectively. More specifically, in addition to specify KPIs changes themselves (*modification* aspects), the main idea behind both views is to cope with the remainder key evolution aspects as identified by our patterns' structure (that is, not only *traceability* and *change propagation* issues, but also *consistency conditions*, and *influences/influenced by* patterns' associations).

Before going on to describe each view, we would like to remark that, trying to be as general as possible, we will explain our evolution metamodel using the terminology of the taxonomy we presented in (Domínguez et al., 2019). However, it must be noted that the evolution metamodel can be used with whatever external KPIs management approach (such as the *PPINOT* metamodel (del-Río-Ortega et al., 2013) or the *KPIOnto* ontology (Diamantini et al., 2016)). For example, in the case of the *Pattern UCR* of Table 1, the `KPIInvolvedElement` would correspond to 'calculation rule' according to (Domínguez et al., 2019), but it would be referred as 'measureDefinition', if the *PPINOT* metamodel is considered, or as 'formula', if the *KPIOnto* ontology is chosen. That is, the entities involved in the evolution of a KPI would correspond to concrete elements of the chosen KPI management approach (a metamodel, an ontology, a taxonomy, etc.).

Structural View. As for the *structural view* (see top of Figure 1), it includes a `Pattern` metaclass with the basic definition information of each KPI evolution pattern (such as the *name*, *abbreviation*, etc.). The entities concerned by a KPI modification are represented by the `KPIInvolvedElement` metaclass. Thus, the KPI entities involved in a KPI evolution pattern are represented in the metamodel by the association between the `Pattern` and `KPIInvolvedElement` metaclasses. At the same time, these entities can be elements *directly* or *indirectly* involved in the KPI evolution pattern (Ralyté and Léonard, 2017). On the one hand, the entities *directly* concerned by a KPI *modification* are those elements explicitly involved in the

KPI pattern (for example, the calculation rule in the *Pattern UCR*). On the other hand, we refer to the entities *indirectly* concerned by a KPI *modification* as any other element not explicitly involved in the KPI evolution pattern (for example, the target of a KPI in the *Pattern UCR*).

Traceability or *consistency* aspects regarding either direct or indirect entities, are represented by the association metaclasses `TraceOperation` and `ConsistencyCondition`, respectively, between the `Pattern` and `KPIInvolvedElement` metaclasses (see Figure 1). For example, if we have decided to apply the *Pattern UCR* so that it traces previous values of modified calculation rules, an instance of the `TraceOperation` is created, linking the `Pattern` instance corresponding to the *Pattern UCR* and the `KPIInvolvedElement` instance with name 'calculation rule'. This allows us to trace any evolution aspect of a `KPIInvolvedElement` as consequence of the application of a pattern. Similarly, linked with a `KPIInvolvedElement` instance named 'KPI', a `ConsistencyCondition` instance would be created to show consistency requirements with other KPIs, as defined by the *Pattern UCR*. *Propagation* aspects are represented by the `PropagationOperation` metaclass which registers the side effects of the evolution pattern on other related entities. This fact is represented in Figure 1 by the `influencedBy` and `influences` association roles.

Execution View. Regarding the execution view (see bottom of Figure 1), as described previously, it represents aspects regarding the application of evolution patterns as established in the structural view. In particular, this view is linked with the previous one by means of the relationship between the `Pattern` metaclass and the `PatternApplication` metaclass. Each time a pattern is applied, a `PatternApplication` instance is created which is linked with the corresponding `Pattern` instance. The `PatternApplication` metaclass includes concrete attributes (such as the *commitDate*, with the moment in which the pattern's application takes place, the *committer*, and any remarkable *comment* she/he considers necessary to make). More specifically, as authors suggest in (Cicchetti et al., 2012), we advocate to represent not only "who" performs the change (in contrast to (Ruiz et al., 2013) which does not consider change's responsibilities), but also "when" such changes are performed. This view mainly defines entities representing application instances of the ones defined in the *structural view* (`AppliedPropagationOperation`, `AppliedTraceOperation` and `AppliedConsistencyCondition`, together with the `AppliedKPIInvolvedElement`). In particular, as an evolution pattern is applied, this view registers concrete evolution traces from a source element version (`AS_IS`) to another target version (`TO_BE`) of such an element (both instances of the external `AppliedKPIInvolvedElement`).

As a way of example, we use the *SLATS* KPI (Calabro et al., 2015), included in Table 1 to exemplify our patterns approach. As we have said before, lets suppose that we want to change the calculation rule of this KPI, currently given by 'Correctness Score per Training Step' (CSTS) plus 'Reload Cardinality per Training Step' (RCTS), and we proceed to apply the *Pattern UCR*. Before executing this pattern, the corresponding consistency conditions are checked to make sure that the change will not contradict any previously defined KPI. If the result is affirmative, an instance of the `AppliedConsistencyCondition` is registered per any checked KPI. Thus, the pattern's application takes place, evolving the KPI's calculation rule from the current calculation rule instance (`AS_IS`), given by `CSTS + RCTS`, to the evolved calculation rule instance (`TO_BE`) represented by `CSTS + 1.5*RCTS`. As for propagation aspects, they are present in the waste indicator example where, in particular, the redefinition of the calculation rule of the waste indicator is *influenced by* the creation of the two new indicators ('hazardous_waste' and 'non-hazardous_waste'). In this case, the propagation task would result in two instances of the `AppliedPropagationOperation` metaclass, each one linked with an instance of the `PatternApplication` metaclass corresponding to an application of the *Pattern CK* (*influences* role), and on the other hand, with the instance of the `PatternApplication` metaclass related to the application of the *Pattern UCR* (*influencedBy* role).

4 CONCLUSIONS AND FUTURE WORK

This paper presents our research proposal to define and develop mechanisms for supporting KPIs evolution management. In particular, we propose to support KPIs evolution by means of a *KPI evolution pattern* notion and a *KPI evolution metamodel*, made up of two interconnected views. The goal of the long-term research we are developing is to provide a general framework for dealing with KPIs evolution.

In order to develop this general framework, several lines for further research must be addressed. Firstly, the conceptual proposal presented in this paper must be applied to real scenarios analyzing its advantages and detecting improvements that may be incorporated. In addition, the integration of our proposal within other KPIs management approaches (such as the *PPINOT* metamodel (del-Río-Ortega et al., 2013) or the *KPIOnto* ontology (Diamantini et al., 2016)) must be studied, determining the way in which the elements of our proposal are embedded properly in such proposals. Finally, mechanisms for giving automatic support to our overall approach must be proposed, based on the possibility of improving tools already developed for the management of KPIs.

ACKNOWLEDGEMENTS

This research was funded by the Spanish Ministry of Economy and Competitiveness, project number EDU2016-79838-P.

REFERENCES

Bernstein, P. A. (2003). Applying Model Management to Classical Meta Data Problems. In *Proc. of the 1st Biennial Conference on Innovative Data Systems Research (CIDR)*, pages 209–220.

Calabro, A., Lonetti, F., and Marchetti, E. (2015). KPI evaluation of the business process execution through event monitoring activity. In *Proc. of the International Conference on Enterprise Systems (ES)*, pages 169–176.

Cicchetti, A., Ciccozzi, F., and Leveque, T. (2012). A solution for concurrent versioning of metamodels and models. *J. object technol.*, 11:1:1–32.

Cognini, R., Corradini, F., Gnesi, S., Polini, A., and Re, B. (2018). Business process flexibility-a systematic literature review with a software systems perspective. *Inform. Syst. Front.*, 20(2):343–371.

del-Río-Ortega, A., Resinas, M., Cabanillas, C., and Cortés, A. R. (2013). On the definition and design-time analysis of process performance indicators. *Inf. Syst.*, 38(4):470–490.

Diamantini, C., Potena, D., and Storti, E. (2016). SemPI: A semantic framework for the collaborative construction and maintenance of a shared dictionary of performance indicators. *Future Gener. Comp. Sy.*, 54:352 – 365.

Domínguez, E., Lloret, J., Pérez, B., Rodríguez, A., Rubio, A. L., and Zapata, M. A. (2011). Evolution of XML schemas and documents from stereotyped UML class models: A traceable approach. *Inf. Softw. Technol.*, 53(1):34–50.

Domínguez, E., Pérez, B., Rubio, Á. L., and Zapata, M. A. (2019). A taxonomy for key performance indicators management. *Comput. Stand. Inter.*, 64:24–40.

Estrada-Torres, B., del Río-Ortega, A., Resinas, M., and Ruiz-Cortés, A. (2016). Identifying variability in process performance indicators. In *Proc. of the Business Process Management (BPM) Forum*, pages 91–107.

Friedenstab, J.-P., Janiesch, C., Matzner, M., and Muller, O. (2012). Extending BPMN for Business Activity Monitoring. In *Proc. of the 45th Hawaii International Conference on System Sciences (HICSS)*, pages 4158–4167.

Global Reporting Initiative (2015). G4 Sustainability Reporting Guidelines. Online at https://www2.globalreporting.org/standards/g4/Pages/default.aspx. Accessed on January 2020.

Global Reporting Initiative (2016). GRI Sustainability Reporting Standards. Online at https://www.globalreporting.org. Accessed on January 2020.

HIQA (2013). Guidance on developing key performance indicators (KPIs) and minimum data sets to monitor healthcare quality version 1.1. Technical report, Health Information and Quality Authority.

IBM Knowledge Center (2011). KPI history. Online at https://www.ibm.com/support/knowledgecenter/SSFTDH_7.5.0/com.ibm.wbpm.mon.widget.doc/topics/kpihist_ov.html. Accessed on January 2020.

Kennerley, M. and Neely, A. (2002). A framework of the factors affecting the evolution of performance measurement systems. *Int. J. Oper. Prod. Man.*, 22(11):1222–1245.

Mulyar, N., van der Aalst, W. M., and Russell, N. (2008). Process flexibility patterns. *Technische Universiteit Eindhoven*.

Pfister, F., Chapurlat, V., Huchard, M., and Nebut, C. (2011). A design pattern meta model for systems engineering. In *Proc. of the 18th World Congress of the International Federation of Automatic Control (IFAC)*.

Pons, C., Baum, G., and Kutsche, R.-D. (2000). Model evolution and system evolution. *J. Ccomput. Sci. Tech-Ch.*, 1(3):18.

Ralyté, J. and Léonard, M. (2017). Evolution Models for Information Systems Evolution Steering. In *Proc. of the 10th IFIP Working Conference on The Practice of Enterprise Modeling (PoEM)*, pages 221–235.

Reichert, M. and Weber, B. (2012). *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*. Springer.

Ruiz, M., España, S., Pastor, O., and González, A. (2013). Supporting organisational evolution by means of model-driven reengineering frameworks. In *Proc. of the 7th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–10.

Russell, N., van der Aalst, W. M., and ter Hofstede, A. H. (2016). *Workflow patterns: the definitive guide*. MIT Press.

Schütz, C. and Schrefl, M. (2014). Customization of Domain-Specific Reference Models for Data Warehouses. In *Proc. of the IEEE 18th International Enterprise Distributed Object Computing Conference (EDOC)*, pages 61–70.

Schütz, C. G., Neumayr, B., Schrefl, M., and Neuböck, T. (2016). Reference Modeling for Data Analysis: The BIRD Approach. *Int. J. Coop. Inf. Syst.*, 25(02):1650006:1–1650006:46.

Supplementary material (2020). KPI evolution patterns. Online at https://zenodo.org/record/3628668. Accessed on January 2020.