Bus Inf Syst Eng 60(2):95–114 (2018)
https://doi.org/10.1007/s12599-017-0490-4

RESEARCH PAPER

Prioritization of Interconnected Processes

Martin Lehnert · Maximilian Röglinger · Johannes Seyfried

Received: 15 December 2016 / Accepted: 8 February 2017 / Published online: 29 August 2017

Abstract Deciding which business processes to improve is a challenge for all organizations. The literature on business process management (BPM) offers several approaches that support process prioritization. As many approaches share the individual process as unit of analysis, they determine the processes' need for improvement mostly based on performance indicators, but neglect how processes are interconnected. So far, the interconnections of processes are only captured for descriptive purposes in process model repositories or business process architectures (BPAs). Prioritizing processes without catering for their interconnectedness, however, biases prioritization decisions and causes a misallocation of corporate funds. What is missing are process prioritization approaches that consider the processes' individual need for improvement and their interconnectedness. To address this research problem, the authors propose the *ProcessPageRank* (*PPR*) as their main contribution. The *PPR* prioritizes processes of a given BPA by ranking them according to their network-adjusted need for improvement. The *PPR* builds on knowledge from process performance management, BPAs, and network analysis – particularly the Google PageRank. As for evaluation, the authors validated the *PPR*'s design specification against empirically validated and theory-backed design propositions. They also instantiated the *PPR*'s design specification as a software prototype and applied the prototype to a real-world BPA.

Keywords Business process management · Network analysis · PageRank · Business process architecture · Process interconnectedness · Process network · Process prioritization

1 Introduction

Process orientation is an acknowledged paradigm of organizational design and source of corporate performance (Dumas et al. 2013; Gaitanides 1983; Kohlbacher and Reijers 2013). Business process management (BPM) thus receives continued interest from industry and academia, supporting organizations in achieving operational excellence and capitalizing on improvement opportunities (Frese 1995; Mertens 1996; Rosemann and vom Brocke 2015; van der Aalst 2013; vom Brocke et al. 2011). Process improvement has been a top priority of process decision-makers for over a decade (Harmon and Wolf 2014). Despite the efforts put into process improvement, about 60% of related projects are reported to fail (Chakravorty 2010; Ohlsson et al. 2014). One key reason of this high failure rate is ineffective process prioritization (Olding and Rosser 2007).

The BPM literature offers several approaches that support process prioritization. Extant approaches are split into

Accepted after two revisions by Prof. Dr. Loos.

Electronic supplementary material The online version of this article (doi:10.1007/s12599-017-0490-4) contains supplementary material, which is available to authorized users.

M. Lehnert · J. Seyfried
Research Center Finance and Information Management,
University of Augsburg, Universitätsstraße 12, 86159 Augsburg,
Germany
e-mail: martin.lehnert@fim-rc.de

J. Seyfried
e-mail: johannes.seyfried@fim-rc.de

M. Röglinger (✉)
Research Center Finance and Information Management,
University of Bayreuth, Wittelsbacherring 10, 95444 Bayreuth,
Germany
e-mail: maximilian.roeglinger@fim-rc.de

two groups, i.e., performance-based and non-performance-based approaches. Performance-based approaches quantify the actual and target performance of processes, derive the related need for improvement, and rank processes based on their need for improvement (Bandara et al. 2015; Dumas et al. 2013; Leyer et al. 2015). Thereby, processes' need for improvement is quantified via performance indicators (e.g., time, cost, flexibility, or quality), whose realizations are eventually merged into integrated performance indicators (e.g., net present value or stakeholder service gap perception) (Bolsinger 2015; Hanafizadeh and Moayer 2008; Reijers and Mansar 2005; Shrestha et al. 2015). Non-performance-based approaches use decision criteria such as urgency, strategic importance, process dysfunctionality, difficulty of improvement, or perceived degree of change (Davenport 1993; Hammer and Champy 1993; Hanafizadeh and Osouli 2011). The link between both groups is that the process-specific need for improvement operationalizes process dysfunctionality.

Existing process prioritization approaches are subject to criticism. They have been characterized either as too high-level to be useful or as so detailed that the mere identification of critical processes requires significant effort (Bandara et al. 2015). Moreover, all approaches share the individual process as unit of analysis. They neglect whether and how processes are interconnected. Process interconnectedness has so far only been considered for descriptive purposes, e.g., in process model repositories and business process architectures (BPAs) (Dijkman et al. 2016; La Rosa et al. 2011; Malinova et al. 2014). It is vital, however, to account for process interconnectedness for prescriptive purposes, such as process prioritization (Manderscheid et al. 2015). This is for several reasons: First, improving a process affects the performance of other processes if they rely on the outcome of that process (Leyer et al. 2015). It may thus be reasonable to prioritize processes with a low stand-alone need for improvement if their outcome is used by many other processes. If process interconnectedness is ignored, prioritization decisions are biased and corporate funds may be allocated inefficiently. Second, neglecting process interconnectedness may entail risks such as downtimes or delayed executions in case of excess demand (Setzer et al. 2010). Beyond BPM-specific reasons, the need for considering interconnectedness as well as for identifying central nodes in networks has been recognized and addressed in many disciplines (e.g., project portfolio management, network analysis, enterprises architecture management) (Landherr et al. 2010; Probst et al. 2013; Winter and Fischer 2007). However, there is a lack of process prioritization approaches that not only consider the need for improvement of individual processes, but also their interconnectedness. Thus, we analyze the following research question: *How can processes be prioritized based on their individual need for improvement and interconnectedness?*

To address this question, we adopted the design science research (DSR) paradigm (Gregor and Hevner 2013). Our artifact is the *ProcessPageRank* (*PPR*). Belonging to the group of performance-based approaches, the *PPR* assists organizations in prioritizing their processes, ranking them based on their network-adjusted need for improvement. The *PPR* shows characteristics of a model and method (Gregor and Hevner 2013; March and Smith 1995). On the one hand, it includes constructs and relations, capturing the problem of interconnectedness-aware process prioritization (e.g., process networks, dependence intensity). On the other hand, the *PPR* specifies how process prioritization activities should be performed in a goal-oriented manner. The *PPR* builds on descriptive knowledge from process performance management and BPAs to conceptualize process performance and interconnectedness. To provide decision support, the *PPR* draws from prescriptive knowledge on network analysis. The *PPR* interprets processes as connected nodes and extends the Google PageRank as a popular centrality measure to identify central nodes in process networks. The *PPR* substantially extends our research on process prioritization by further specifying the need for improvement of individual processes considering multiple performance dimensions, substantiating process interconnectedness via dependence intensities, and advancing the evaluation (Lehnert et al. 2015).

This study follows the DSR methodology as per Peffers et al. (2007): Sect. 2 provides relevant theoretical background. Section 3 outlines the research method and evaluation strategy. In Sect. 4, we present the *PPR*, including the transformation of BPAs into process networks, the specification of input variables, and the *PPR* algorithm. In Sect. 5, we report on the results of our evaluation activities, before highlighting limitations and opportunities for future research in Sect. 6.

2 Theoretical Background

2.1 Process Performance Management and Business Process Architectures

BPM is the art and science of overseeing how work is performed to ensure consistent outcomes and take advantage of improvement opportunities (Dumas et al. 2013). It combines knowledge from information technology (IT) and management sciences (van der Aalst 2013). From a lifecycle perspective, BPM involves activities such as the identification, definition, modeling, implementation and execution, monitoring, control, and improvement of processes (Recker and Mendling 2016). Dealing with all processes of an organization, BPM offers an infrastructure for effective and efficient work (Harmon 2014). Processes, as BPM's unit of analysis, split into core, support, and management processes (Armistead et al. 1999). Core processes are collections of events, activities, and decision points involving actors and objects leading to valuable outcomes (Dumas et al. 2013). Support processes ensure that core processes continue to function, while management processes plan, organize, monitor, and control corporate activities (Harmon 2014). We focus on core and support processes, referring to both as processes.

To assess process performance and estimate the effects of improvement projects, performance indicators are an essential tool (Leyer et al. 2015). In process performance management, the realizations of performance indicators are typically compared with target values and admissible value ranges (Leyer et al. 2015). Complying with the predominating conceptualization of process performance as a multidimensional construct, performance indicators are grouped according to performance dimensions (Linhart et al. 2015). A popular framework is the Devil's Quadrangle that comprises flexibility, time, cost, and quality as dimensions (Reijers and Mansar 2005). The Devil's Quadrangle is so-named as improving one dimension weakens at least one other, disclosing trade-offs among performance dimensions to be resolved. To prioritize processes, process performance dimensions must be integrated in a way that accounts for trade-offs (Bolsinger 2015; Mansar et al. 2009). Thereby, the related multi-criteria decision problem is reduced to a single-criterion problem, a necessary task in normative analytical modeling and multi-criteria decision analysis (Cohon 2004; Meredith et al. 1989). The result is an integrated performance indicator. Examples for integrated performance indicators are the value contribution of a process (Buhl et al. 2011), the return on process transformation (vom Brocke and Sonnenberg 2015), the aggregated cash flow deviation from a threshold (Manderscheid et al. 2015), the business value score (Bandara et al. 2015), and the processes' individual need for improvement index (Lehnert et al. 2015).

Processes and their relations are typically modeled as BPAs. BPAs are structured overviews of an organization's processes and relations, potentially accompanied by guidelines that determine how to organize these processes (Dijkman et al. 2016). The top-most BPA level is also known as process map (Malinova et al. 2014). The four most frequent relation types in a BPA are specialization, decomposition, use, and trigger (Dijkman et al. 2016). Specialization relations express that a process is a specialized version of another process, inheriting all characteristics of the super-process. A decomposition expresses that a process is decomposed into multiple sub-processes. Use relations indicate that a process requires the output of another process to continue or complete its execution. That is, the performance of the using process depends, at least in parts, on the performance of the used process (Malone and Crowston 1994). Finally, trigger relations express that a process triggers the execution of another process without having to wait for the output of that process. In contrast to use relations, the performance of the triggering and the triggered processes are independent.

2.2 Network Analysis

In network analysis, centrality measures help determine central nodes in networks. If processes are interpreted as connected nodes, centrality measures help identify central nodes in process networks. With the *PPR* building on an extended Google PageRank, this section introduces the foundations of the PageRank. We justify in Sect. 4 why the extended Google PageRank is the only centrality measure that fully meets the requirements of interconnectedness-aware process prioritization. Two key reasons, which can already be named here, are that the PageRank copes with directed networks and is not biased by local patterns of single nodes. These properties are vital for interconnectedness-aware process prioritization because use relations among processes are directed and process prioritization must consider all processes from a BPA. To better illustrate the PageRank's components, we start with the eigenvector centrality, which is an immediate conceptual predecessor of the PageRank.

The eigenvector centrality extends the simple degree centrality, which only accounts for a node's direct neighbors, by taking the connectedness of neighboring nodes into account (Hanneman and Riddle 2005; Newman 2003). A node ranks higher if it has well-connected neighbors (Newman 2003). If x_i is node i's eigenvector centrality, it is higher if the centrality x_j of all nodes j that are direct neighbors of node j is higher. We define A as the adjacency matrix, where a_{ij} is 1, if node i is a direct neighbor of j, and 0 otherwise. Further, we define λ as the largest eigenvalue of the adjacency matrix. Based on this, the eigenvector centrality as proposed by Bonacich (1987) is computed as shown in Eq. (1)

$$x_i = \frac{1}{\lambda} \cdot \sum_j \left(a_{ij} \cdot x_j\right) \tag{1}$$

The eigenvector centrality serves as foundation for Brin and Page‘s (1998) PageRank. It works well for undirected networks, but has weaknesses when applied to directed networks, including the eigenvector centrality of nodes being 0 in certain constellations. Adding a constant term to a node's centrality irrespective of its connectedness prevents its centrality from becoming 0 and spreading that value through the network. To balance the constant and the network term, the factor $1/\lambda$ is replaced by the dampening factor d, weighting the network structure and constant terms with d and $(1-d)$, respectively. Another drawback of the eigenvector centrality is that if a node i has an ingoing edge from a node j, the weight that node i receives is the same irrespective of how many outgoing edges j has. Nevertheless, there are many applications where node i's centrality increases less strongly if node j has more outgoing edges (Brin and Page 1998). Adjusting the effect of one node on other nodes based on the number of outgoing edges can be accomplished by dividing x_j by the number of j's outgoing edges $|O_j|$. We refer to the set of outgoing edges of a node i as O_i, and to the set of ingoing edges as I_i. These adjustments lead to the PageRank as presented in Eq. (2) (Brin and Page 1998).

$$\begin{aligned} PR(i) &= (1-d)\frac{1}{n} + d \cdot \sum_{j}\left(a_{ij} \cdot \frac{PR(j)}{|O_j|}\right) \\ &= (1-d)\frac{1}{n} + d \cdot \sum_{j \in I_i} \frac{PR(j)}{|O_j|} \end{aligned} \tag{2}$$

The PageRank, as shown in Eq. (2), can be interpreted as follows: for each ingoing edge, node i receives a share of the PageRank of the respective source node j, which, in turn, depends on how many outgoing edges node j has. The dampening factor d balances the weight between the constant and network terms. With these adjustments, one can prove mathematically that the upper boundary of the interval containing d always equals 1 in case of an undirected network and, even though the mathematical proof does not hold in case of directed networks, in practice it will roughly be of order 1 (Newman 2003). Therefore, d should generally be chosen from the interval [0; 1]. However, if d converges to 1, PageRank values become highly susceptible to changes in the network structure. High d values increase the risk of rank sinks, i.e., nodes without outgoing edges have higher weight, while other nodes rank disproportionally low. When applying the PageRank to web pages, a d value of 0.85 is deemed reasonable to address this trade-off (Langville and Meyer 2011).

As mentioned, node i receives weight from node j if node j points to node i. This weight is determined based on node j's number of outgoing edges, assigning equal weight to each edge. However, weighting all outgoing edges equally is not always appropriate. In the case of websites, the importance of a distinct edge also depends on the anchor text of the link or on how prominently the link is located. Thus, an early adjustment to the PageRank was to allow individually weighted edges (Langville and Meyer 2011). The weight of an edge that points from node i to node j is denoted as w_{ij}. In the initial PageRank, the constant term is initialized with $1/n$. Each node (or webpage respectively) has the same initial weight. However, some nodes are more important than others, irrespective of their connectedness. Thus, Brin and Page (1998) expanded the concept of the constant term by allowing individual constant terms for each node. The only restriction is that each weight is from [0;1] and that the weights sum up to 1. This expansion is implemented by introducing an individual node weight k_i, which is proportional to the weights of all nodes in the network (Langville and Meyer 2011). The consideration of individual weights for nodes and edges leads to Eq. (3).

$$PR(i) = (1-d) \cdot \frac{k_i}{\sum_{t=1}^{n} k_t} + d \cdot \sum_{j \in I_i} \frac{PR(j) \cdot w_{ji}}{\sum_{k \in O_j} w_{jk}} \tag{3}$$

We rely on the extended PageRank, as shown in Eq. (3) as justificatory knowledge to derive the *PPR* algorithm in Sect. 4.3, enabling process prioritization that integrates the processes' individual need for improvement and interconnectedness.

3 Research Method and Evaluation Strategy

To design the *PPR*, we adopted the DSR paradigm, following the DSR methodology as per Peffers et al. (2007). The DSR methodology includes six phases, i.e., problem identification, definition of design objectives, design and development, demonstration, evaluation, and communication. Complying with the *design-evaluate-construct-evaluate* pattern advocated by Sonnenberg and vom Brocke (2012), we did not traverse these phases strictly sequentially, but switched between the design and development as well as the demonstration and evaluation phases.

As for problem identification, we justified the need for considering the interconnectedness of processes in process prioritization decisions as a valid DSR problem in Sect. 1. We also defined two design objectives drawing from extant knowledge related to process performance and BPA (Sect. 2.1). Both objectives provided guidance in the design and development phase as we operationalized them in terms of design propositions in line with prescriptive knowledge on network analysis (Sect. 2.2). The design objectives and related design propositions also helped validate the *PPR*'s design specification in the demonstration and evaluation phase. The design objectives are specified as follows:

(DO.1) Performance of individual processes When prioritizing processes for improvement purposes, the individual performance of these processes must be measured via performance indicators and considered in the resulting ranking.

(DO.2) Relations among multiple processes When prioritizing processes for improvement purposes, the relations among these processes must be considered in the resulting ranking.

In the design and development phase, we conceived the *PPR*'s design specification, building on normative analytical modeling and multi-criteria decision analysis (Cohon 2004; Meredith et al. 1989). We illustrate how to transform BPAs into process networks as well as which performance and interconnectedness data must be added to apply the *PPR* (Sect. 4.1). We then show how to determine relevant input parameters, i.e., the process need for improvement index and dependence intensity (Sect. 4.2). We finally derive the *PPR* algorithm as an extension of the Google PageRank in line with theory-backed and empirically validated design propositions (Sect. 4.3).

Our overall evaluation objective is to show that the *PPR* makes an appropriate contribution to the extant knowledge on process prioritization. To structure our evaluation, we adopted the evaluation framework by Sonnenberg and vom Brocke (2012). This framework comprises four activities (EVAL1–EVAL4) to cover the ex-ante/ex-post and artificial/naturalistic evaluation dimension (Venable et al. 2012). EVAL1 ensures the problem's meaningfulness from an academic and practical viewpoint. With EVAL1 strongly resembling the first phases of Peffers et al.'s (2007) DSR methodology, we do not provide further details here. EVAL2 aims to validate design specifications prior to their instantiation in terms of their alignment with the research problem, their real-world fidelity, and understandability. Thereby, EVAL2 distinguishes between an artificial and a naturalistic perspective. From an artificial perspective, we discussed the *PPR*'s design specification against design propositions. To do so, we first derived design propositions and validated them with industrial and academic BPM experts (Sect. 5.1). The actual discussion is presented together with the demonstration example (Sect. 5.3), because the *PPR* is a complex recursive algorithm. From a naturalistic perspective on EVAL2, we report on an in-depth interview with an expert from a global data-driven online retailer (Sect. 5.2). Regarding EVAL3, which takes an ex-post perspective and strives for validated instantiations, we implemented the *PPR* as a software prototype. In a previous study, we already applied a prior version of the prototype in a scenario analysis (Lehnert et al. 2015). In this study, we use the prototype to show the *PPR* in action based on a real-world BPA together with an efficiency and robustness analysis (Sect. 5.3). Taking an ex-post perspective, EVAL4 strives for validating the applicability and usefulness of artifact instantiations. Although our demonstration in EVAL 3 builds on a real BPA, it is not a full-fledged real-world case study. The reason is that the *PPR* is very data-intensive, a feature that currently causes considerable data collection effort in many organizations. In line with the uptake of process-aware information systems and the availability of process logs, however, we are confident that many organizations will be able to gather high-quality data with reasonable effort in the near future. We get back to this limitation in the conclusion.

4 The ***ProcessPageRank***

4.1 Transformation of Business Process Architectures into Process Networks

The *PPR* prioritizes processes while accounting for their individual need for improvement and interconnectedness. To do so, the *PPR* ranks the processes from a given BPA in line with their network-adjusted process improvement index (*NPNI*). As a prerequisite for the *PPR*'s application, we first transform all components of the given BPA into a process network and enrich the network with additional information (e.g., how often a process uses other processes). Figure 1 on the left shows connected processes as captured in a BPA using the ArchiMate notation (Dijkman et al. 2016). On the right, Fig. 1 illustrates the corresponding process network, which is used as input of the *PPR*.

To transform a BPA into a process network, we first define each process included in the BPA as a node in the process network. From a stand-alone perspective, we assume that each process has a process need for improvement index (*PNI*) that will be adjusted by the *PPR* in line with its interconnectedness. Thus, each process i features a PNI_i, which takes values from [0;1], where 0 and 1 indicate no or substantial need for improvement, respectively. The *PNI* operationalizes the concept of process dysfunctionality used in earlier process prioritization approaches. To quantify the *PNI*, we combine proven concepts of process performance management (i.e., the operationalization of process performance via multiple performance dimensions as well as the comparison of actual and target values) and multi-criteria decision analysis (i.e., the weighted aggregation of multiple decision criteria), which have not been combined so far. We provide more information about the *PNI* in Sect. 4.2.1. As a second step, we transfer the relations included in the BPA to the process network as follows:

> *Decomposition* A composed process is either modeled as a single process or all its component processes are modeled, depending on the intended level of granularity. In Fig. 1, processes 2–6 are modeled as components of process 1. The network only contains the component processes.

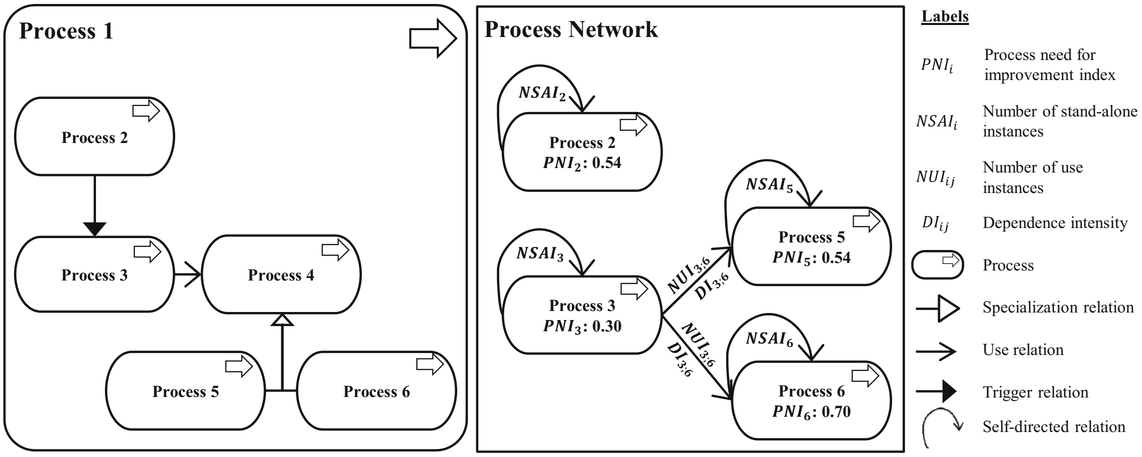


Fig. 1 Example of a BPA (*left*) and the corresponding process network (*right*)

Specialization Based on the idea that all relations of a super-process hold for its sub-processes, we only include sub-processes in the process network (Dijkman et al. 2016). In case a sub-process has additional relations with other processes, these relations must be transferred to the process network as well and treated as trigger or use relations, respectively. In Fig. 1, processes 5 and 6 specialize process 4. Hence, process 4 is not included in the process network. Processes 5 and 6 inherit the use relation between processes 3 and 4.

Use Use relations are directly transferred to the process network. Each use relation is modeled as an edge from a using to a used process. As processes may use other processes several times per instance and period, each use relation has a weight representing the number of instances a process uses another process. We refer to this weight as the number of use instances NUI_{ij} between the processes i and j. Use relations capture dependencies among processes whose intensity may vary from process to process (Malone and Crowston 1994). Each use relation is therefore assigned a second weight, i.e., the dependence intensity DI_{ij} between the processes i and j. The *DI* indicates how strongly the performance of the using process depends on the used process. We formally introduce the *DI* in Sect. 4.2.

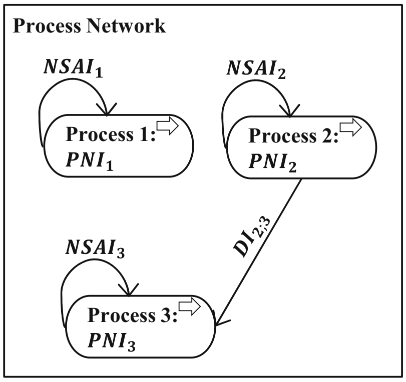
Trigger In line with the asynchronous communication property of trigger relations, the performance of triggering processes is independent from that of triggered processes. Triggering processes have "no interest" in triggered processes being improved. Thus, trigger relations need not be directly transferred to the process network. However, they influence the number of instances that a process is executed without using other processes. We model this number of stand-alone instances *NSAI* as weights of self-directed edges in the process network. In the *PPR* logic, self-directed edges and their weights prevent a process' *PNI* from being cascaded throughout the process network for those instances that do not use other processes. As processes may use other processes several times during the same instance within a distinct period, the *NSAI* does not necessarily equal the difference between the number of all instances and the number of all use instances.

4.2 Input Parameters of the *ProcessPageRank*

Processes are valuated via performance indicators, which are typically structured along the dimensions of the Devil's Quadrangle (i.e., time, cost, quality, and flexibility). The *PPR* considers the cost, time, and quality dimensions, as flexibility can be covered via other dimensions such as time (Ray and Jewkes 2004). As these performance dimensions must be treated differently in process networks, we first model the dimension-specific *PNI* and *DI* individually, and aggregate them in a second step building on ideas from multi-criteria decision analysis (Cohon 2004). Figure 2 shows an exemplary calculation of the *PNI* and the *DI* that illustrates the equations below. Please find an overview of all variables in Appendix A (available online via http://springerlink.com).

4.2.1 Process Need for Improvement Index

The dimension-specific process need for improvement index PNI_i^p reflects the urgency of process i to be improved regarding performance dimension $p \in \{Cost, Time, Quality\}$. To quantify the *PNI*, we compare the target state TS_i^p of a performance dimension with its actual state AS_i^p. This is sensible because, in process performance management, the realizations of performance indicators are typically compared with desired target values (Leyer et al. 2015). In the *PPR*, target and actual states are quantified via a single performance



Processes		Process 1	Process 2	Process 3
Cost	AS	45	80	90
	TS	25	60	75
Time	AS	50	75	55
	TS	40	70	40
Quality	AS	70	90	80
	TS	65	80	70
AE_i		1,500	1,000	2,000
$\boldsymbol{PNI_i}$		…	**0.444**	…

Relations	$\boldsymbol{DI_{2;3}}$
$BDI_{2;3}^{\text{Time}}$	10
$MDI_{2;3}^{\text{Quality}}$	0.5
$\boldsymbol{DI_{2;3}}$	**1.694**

$\rho^{\mathrm{C}} = 0.25; \rho^{\mathrm{T}} = 0.50; \rho^{\mathrm{Q}} = 0.25$

Exemplary calculation of $\boldsymbol{PNI_2}$

$$PNI_2^{\mathrm{C}} = \frac{(80-60)\cdot 1{,}000}{\max(30{,}000;\, 20{,}000;\, 30{,}000)} = 0.667$$

$$PNI_2^{\mathrm{T}} = \frac{(75-70)\cdot 1{,}000}{\max(15{,}000;\, 5{,}000;\, 30{,}000)} = 0.167$$

$$PNI_2^{\mathrm{Q}} = \frac{(90-80)\cdot 1{,}000}{\max(10{,}000;\, 20{,}000;\, 7{,}500)} = 0.500$$

$$PNI_2 = \frac{(0.667\cdot 30{,}000\cdot 0.25)+(0.167\cdot 30{,}000\cdot 0.50)+(0.500\cdot 20{,}000\cdot 0.25)}{(30{,}000\cdot 0.25)+(30{,}000\cdot 0.5)+(20{,}000\cdot 0.25)}$$

$$= 0.444$$

Exemplary calculation of $\boldsymbol{DI_{2;3}}$

$$DI_{2;3}^{\mathrm{C}} = \frac{90-75}{80-60} = 0.750$$

$$DI_{2;3}^{\mathrm{T}} = \frac{\min(10;55-40)}{75-70} = 2.000$$

$$DI_{2;3}^{\mathrm{Q}} = \frac{0.5\cdot(80-70)}{90-80} = 0.500$$

$$DI_{2;3} = \frac{(0.750\cdot 30{,}000\cdot 0.25)+(2.000\cdot 30{,}000\cdot 0.50)+(0.500\cdot 20{,}000\cdot 0.25)}{(30{,}000\cdot 0.25)+(30{,}000\cdot 0.5)+(20{,}000\cdot 0.25)}$$

$$= 1.694$$

Fig. 2 Exemplary calculation of the PNI and DI in a sample process network

indicator per dimension. In the cost dimension, we choose the *process costs per execution*, covering the costs of the process itself as well as the costs of used processes. As for time, we choose the *lead-time*, covering the total time for the completion of a process instance end-to-end. As for quality, we use the *error rate* because it has the same polarity as process costs and lead-time. We assume that each performance indicator covers the performance in the respective dimension and that the target state is never worse than the actual state. The *PPR* can also be extended to build on other indicators.

The PNI_i^p builds on the difference between the target and actual performance. The higher the difference, the higher the *PNI*. If processes A and B have the same difference between their actual and target states, but process A is executed more often, then process A should be improved first. Thus, the *PNI* of process A must be higher than that of process B. We thus multiply the difference between the actual and target states with the amount of executions AE_i. This makes the dimension-specific *PNI* comparable across all processes included in the process network. For the same reason, the dimension-specific *PNI* is normalized to the interval [0;1] against the highest dimension-specific *PNI* across all processes. As a result, we define the *PNI* for each performance dimension according to Eq. (4) If a process performs such badly that it cannot be used by other processes and does not deliver any useful output, it may be reasonable to improve this process first. To achieve this, the actual state can be set to an extremely high value, an intervention ensuring that the process is ranked first. Such a manual intervention, however, should be an exception as it bypasses the *PPR*'s prioritization logic.

$$PNI_i^p = \frac{(AS_i^p - TS_i^p)\cdot AE_i}{\max\limits_j\left[\left(AS_j^p - TS_j^p\right)\cdot AE_j\right]} \tag{4}$$

4.2.2 Dependence Intensity

The dependence intensity *DI* of a use relation indicates how strongly the performance of a using process depends on the performance of a used process. Figuratively, if a using process performs badly only due to the performance of a used process, the *PNI* of the using process depends highly on the used process' *PNI*. This phenomenon is captured in terms of a high *DI* between the using and used processes. Thus, the *DI* depends on the *PNI* of both the using and the used processes. The concrete modeling of the *DI* also depends on which performance dimension is analyzed.

4.2.2.1 Dependence Intensity in the Cost Dimension The dependence intensity *DI* can vary for different use relations. Consider a process B that has a significant difference between its actual and target performance (i.e., it performs poorly) but is executed infrequently. This leads to a moderately high PNI_{B}. Now consider a process C that has a small difference between its actual and target state (i.e., it performs far better than process B) but is executed frequently. This results in a moderately high PNI_{C}, equal to PNI_{B}. Finally, consider a process A that uses processes B and C equally often. Even though PNI_{B} and PNI_{C} are equal, from process A's perspective, improving process B is more desirable than improving process C, since the performance per instance of process B is worse and both processes are used equally often.

The *DI* captures this property as shown in Eq. (5). The worse the performance per instance of process *j*, the larger the impact of improving that process on a using process *i*. Thus, the larger the difference between the actual and the target performance of the used process *j* (i.e., the need for improvement), the larger the impact of improving process *j* on process *i*. Vice versa, the larger the difference between the actual and the target performance of the using process *i*, the smaller the impact of improving process *j* on the using process *i*. Consider process A performing poorly itself, it is more important to improve process A (from the perspective of process A) than to improve any used process. In contrast to the other performance dimensions, this effect always cascades through the process network in the cost dimension and it is independent of the specific design of the involved processes.

$$DI_{ij}^{\mathrm{Cost}} = \frac{AS_j^{\mathrm{Cost}} - TS_j^{\mathrm{Cost}}}{AS_i^{\mathrm{Cost}} - TS_i^{\mathrm{Cost}}} \tag{5}$$

4.2.2.2 Dependence Intensity in the Time Dimension The dependence intensity *DI* of the time dimension is an adjusted version of the cost-specific *DI*. Consider two processes A and B where A uses B. In general, an improvement in process B's lead-time will improve process A's lead-time as well. Now consider process A running two parallel streams I and II and process B being used in stream I. If both streams run equally fast, improving process B's lead-time only improves the lead-time of stream I, but not that of process A. This is as stream I then has to wait for stream II to finish. Process A's lead-time is thus not affected by improving process B. The same holds true if stream I is already faster than stream II before improving process B. Consider the lead-time for stream I being 10 min higher than for stream II. Improving process B's lead-time by 15 min results in stream I being 5 min faster than stream II. Process A as a whole, however, is only 10 min faster than before improving process B. Thus, the effect of improving process B's lead-time only partly influences process A.

Hence, even though a used process may seem to have high need for improvement due to a large difference between the actual and target lead-time, improving this process does not necessarily affect the using process to the same extent. Therefore, we define an upper boundary BDI_{ij}^{Time} for the *DI* associated with the time dimension as shown in Eq. (6). This boundary represents the maximum improvement of the used process *j* that can cascade to the using process *i*.

$$DI_{ij}^{\mathrm{Time}} = \frac{\min\left(\mathrm{BDI}_{\mathrm{ij}}^{\mathrm{Time}}; \mathrm{AS}_{\mathrm{j}}^{\mathrm{Time}} - \mathrm{TS}_{\mathrm{j}}^{\mathrm{Time}}\right)}{AS_i^{\mathrm{Time}} - TS_i^{\mathrm{Time}}} \tag{6}$$

4.2.2.3 Dependence Intensity in the Quality Dimension To calculate the dependence intensity *DI* associated with the quality dimension, it is necessary to consider the following property: if process A uses process B and process B creates defective output, the output of process A is likely to be faulty, too. Reducing process B's error rate, however, does not necessarily reduce process A's error rate to the same extent. For instance, if errors occur in process A and if we eliminate errors in process B, the errors in process A may still occur, and process A's error rate remains unchanged. In order to model this property, the quality-specific *DI* includes a moderator variable $MDI_{ij}^{\mathrm{Quality}}$ as shown in Eq. (7). The variable can be interpreted as the conditional probability of good quality in the using process *i* if the quality of the used process *j* is good after an improvement. Thus, it takes values from the interval [0;1]. The quality-specific *DI* has no fixed upper boundary.

$$DI_{ij}^{\mathrm{Quality}} = \frac{MDI_{ij}^{\mathrm{Quality}} \cdot \left(AS_j^{\mathrm{Quality}} - TS_j^{\mathrm{Quality}}\right)}{AS_i^{\mathrm{Quality}} - TS_i^{\mathrm{Quality}}} \tag{7}$$

4.2.3 Integration of the Dimension-Specific Input Parameters

We now integrate the dimension-specific process need for improvement indexes and dependence intensities into a single index to enable a prioritization across all performance dimensions and all processes included in the process network. Such an integration of multiple criteria into a single-criterion problem is a necessary step in multi-criteria decision analysis to provide decision support (Cohon 2004).

As an integrated indicator, the overall *PNI* must cater for trade-offs and the importance of the included performance dimensions. With all chosen performance indicators featuring the same polarity (i.e., low values are desirable), the overall *PNI* needs not resolve trade-offs. The dimension-specific *PNI* can be summed up, which is possible as they share the same measurement dimension (i.e., they are non-dimensional due to the normalization of the dimension-specific *PNI*). To capture that performance dimensions can be differently important, we use custom weights ρ^p that take values from the interval [0;1] and sum up to 1 (Keeney and Raiffa 1993). Like the dimension-specific *PNI*, the overall *PNI* must be normalized to be comparable across all processes. The overall *PNI* is shown in Eq. (8).

When aggregating the dimension-specific *PNI*, one must consider that they need not necessarily be included in the overall *PNI* as equally important, even if they are equal for two performance dimensions. The reason is that the dimension-specific *PNI* are relative measures, normalized using the highest dimension-specific value across all processes from the process network. Consider a process A that performs well regarding all performance dimensions. Further, consider the highest difference between the actual and the target cost value within the process network to be very high, while the highest difference in time is rather low. This makes process A's cost-specific need for improvement index rather low and the time-specific index rather high. Aggregating both indices with equal weight into process A's overall *PNI* would lead to an average value for process A, although it performs well in both performance dimensions. To prevent such a bias, we also consider the highest dimension-specific *PNI* values across all processes when aggregating the dimension-specific *PNI*. The higher the maximum *PNI* in a distinct dimension, the worse the performance of the processes in that dimension. Thus, the higher the *PNI* in one performance dimension, the higher its importance for the overall *PNI*.

$$PNI_i = \frac{\sum_p \left(PNI_i^p \cdot \max_j \left[\left(AS_j^p - TS_j^p \right) \cdot AE_j \right] \cdot \rho^p \right)}{\sum_p \left(\max_j \left[\left(AS_j^p - TS_j^p \right) \cdot AE_j \right] \cdot \rho^p \right)} \tag{8}$$

The same rationale holds for the aggregation of the dimension-specific dependence intensities. Their aggregation is analogous to that of the *PNI* as shown in Eq. (9).

$$DI_{ij} = \frac{\sum_p \left(DI_{ij}^p \cdot \max_j \left[\left(AS_j^p - TS_j^p \right) \cdot AE_j \right] \cdot \rho^p \right)}{\sum_p \left(\max_j \left[\left(AS_j^p - TS_j^p \right) \cdot AE_j \right] \cdot \rho^p \right)} \tag{9}$$

4.3 The *ProcessPageRank* Algorithm

In order to prioritize processes in line with their network-adjusted need for improvement index, the *PPR* further develops the extended PageRank from Eq. (3) by integrating the domain-specific input parameters introduced above. We chose the extended Google PageRank as foundation as it is the only centrality measure that integrates all components of process networks and that meets the requirements of interconnectedness-aware process prioritization. Neither the degree nor the eigenvector centrality cope with node and edge weights. Further, they primarily apply to undirected networks. As process networks are directed networks containing node and edge weights, only the Katz centrality and the PageRank apply to process prioritization. In the Katz centrality, however, the weight transferred from one node to another via an outgoing edge does not depend on other outgoing edges of that node. If we applied such a reasoning to process networks, processes would always assign the same weight to a used process irrespective of how many other processes it uses. However, if a using process transfers weight to a used process, it is very relevant to consider the characteristics of other use relations of the using process. In addition, the Katz centrality does not allow for adjusting the balance between a process' individual importance and its interconnectedness, another important feature of interconnectedness-aware process prioritization.

The extended PageRank encompasses two summands, weighted by the dampening factor. The first summand assigns each node a stand-alone weight. The second summand adjusts the stand-alone weight in line with the node's interconnectedness. The dampening factor indicates how strongly the interconnectedness adjusts the stand-alone weight. Following this structure, we first integrate the process need for improvement index *PNI* into the extended PageRank and, then, the number of use instances *NUI*, the number of stand-alone instances *NSAI*, and the dependence intensity *DI*. The integration of our input parameters is guided by the design objectives, we derived from the BPM literature. We operationalized the design objectives in terms of design propositions from a network analysis perspective and validated them with a group of BPM experts (Sect. 5.1).

4.3.1 *Integration of the Process Need for Improvement Index*

According to design objective (DO.1), process prioritization must consider the involved processes' individual performance. The *PPR* accounts for individual process performance via the *PNI*. To integrate the requirements of (DO.1) into the *PPR*, we formulated the following design proposition:

(P.1) For any two processes *i* and *j* from the process network: If, ceteris paribus, process *i* has a higher process need for improvement index than process *j*, then the network-adjusted need for improvement index of process *i* must exceed that of process *j*.

Figuratively, if two processes have the same interconnectedness (i.e., same relations with the same processes, same weights, and same self-directed relations) and the only difference is that one process performs worse, then the process with the worse performance must be ranked higher.

Equation (1) shows how the *PNI* is integrated into the *PPR*. On the one hand, the *PNI* is of course integrated into the first summand of the *PPR*, which reflects the stand-alone weight of each process. On the other, the *PNI* needs to be integrated into the second summand as it also influences to which extent the processes' weights are adjusted in line with their interconnectedness. We provide more information about this property in the next section.

4.3.2 Integration of the Process Network Structure

In line with design objective (DO.2), process prioritization should account for the relations among the processes from the process network. If a process uses another process, improving the used process gains importance as this positively affects the performance of both the used and the using process. The more intensely the using process uses the other process, the higher the effect of process improvement. As the intensity of use relations is represented by the dependence intensity *DI* and the number of use instances *NUI*, process prioritization must account for both parameters. This leads to the following design proposition for ingoing use relations:

(P.2) For any two processes i and j from the process network: If, ceteris paribus, process i is used by an additional process or has a higher number of use instances or a higher dependence intensity for at least one ingoing relation than process j, then the network-adjusted need for improvement index of process i must exceed that of process j.

A similar logic holds for outgoing relations. The more intensely a process uses other processes, the more important it is for this process to improve the used processes, the idea being that improving the using process has no effect on the used process, while, in general, improving the used process has a positive effect on the using process. Therefore, the more a process relies on other processes, the more important it is to improve the used processes, and the less important it is to improve the using process relative to the used processes. This leads to the following design proposition for outgoing use relations:

(P.3) For any two processes i and j from the process network: If, ceteris paribus, process i uses an additional process or has a higher number of use instances or a higher dependence intensity for at least one outgoing relation than process j, then the network-adjusted need for improvement index of process j must exceed that of process i.

The design propositions (P.2) and (P.3) focus on direct use relations. Accordingly, the more intensely a process is used by other processes in terms of *DI* or *NSAI*, the higher it should be ranked. Consequently, the more a process uses other processes, the lower it should be ranked, relative to used processes. Design objective (DO.2) does not only hold for direct use relations, but also for transitive relations. Consider a relation where process A uses process B, which in turn uses process C. As process A uses process B, process B should be ranked higher than process A. The same holds for the use relation between process B and C. Improving process C has a positive effect on process B, which transitively affects process A. Hence, the ranking of process C should be higher based not only on its relation with process B, but also based on the relation between processes A and B. This leads to the following final design proposition:

(P.4) For any two processes i and j from the process network that are both used by other (different) processes: If, ceteris paribus, process i is used by the process with the higher network-adjusted need for improvement index than process j, then the network-adjusted need for improvement index of process i must exceed that of process j.

The extended PageRank from Eq. (3) accounts for the network structure in its second summand. This summand includes an individual edge weight w_{ij} that enables incorporating a unique relative importance for each edge in the network. Below, we operationalize the edge weights such that the *PPR* implements the design propositions (P.2) to (P.4).

As stated in (P.2), a process should receive higher weights, the more often it is used by other processes. In the process network, we defined *NUI* and *NSAI* as weights of use relations and self-directed relations, respectively. Initializing the weight w_{ij} with the *NUI* and *NSAI* ensures two properties: First, if a process uses two other processes, one more frequently than the other, it transfers more weight to the process it uses more often, since the weight of the use relation is higher (P.3). Second, the process does not transfer weight in case it does not use other processes. As the weight of the self-directed relation represents the *NSAI* and the relation points to the process from which it originated, no weight is transferred.

So far, a process transfers weight to other processes according to use relations only. This implies that processes that are used equally often by the same process, ceteris paribus, receive equal weights. As described above, the positive effect of improving a distinct used process on a distinct using process also depends on the used process' *PNI*. Consider a process A that uses process B. The higher process B's *PNI*, the higher the effect on process A and, thus, the higher process B's network-adjusted need for improvement index $NPNI_{\mathrm{B}}$. For example, if process A uses process B and the lead-time is the only relevant indicator: $NPNI_{\mathrm{B}}$ rises with a rising lead-time of process B, because process A must wait for B. Hence, the higher process B's *PNI*, the more important it is for process A to improve process B first. Thus, process B must rise in the prioritization ranking. As this is in the interest of process A, it should transfer more weight to process B, the higher process B's *PNI*. Therefore, PNI_{B} must be included when calculating the weight w_{AB}. We therefore update the initialization of w_{ij} and include the used processes' *PNI* by multiplying them with the respective number of use instances *NUI*, or the number of stand-alone instances *NSAI* in the case of self-directed relations. For better legibility, we refer to the *NSAI* of a process i as NUI_{ij} with $i = j$. Taking into account all these adjustments results in Eq. (10).

$$NPNI(i) = (1-d) \cdot \frac{PNI_i}{\sum_{j=1}^{n} PNI_j} + d \cdot \sum_{k \in I_i} NPNI(k) \cdot \frac{NUI_{ki} \cdot PNI_i}{\sum_{l \in O_k} NUI_{kl} \cdot PNI_l} \qquad (10)$$

In Eq. (10), weight transfers within the process network depend on the *NUI* of the relation between two processes and on the *PNI* of the used process. However, weight transfers should also depend on the using processes' *PNI*. Consider two processes where process A uses process B. If processes are ranked according to Eq. (10), we get distinct values for these processes' *NPNI*. If we increase process A's amount of executions AE_{A} while keeping the number of use instances NUI_{AB} constant, process A's need for improvement index PNI_{A} rises. If process A's *PNI* rises, the weight transferred to process B also rises as the weight transferred to a used process is relative to the using process' *PNI*. If more weight is transferred to the used process B, its $NPNI_{\mathrm{B}}$ also rises even though the improvement of process B did not get more important as neither the NUI_{AB} nor any other variables for process B changed. To cater for this effect, we also include the dependence intensity *DI* in the weights. The resulting formula for w_{ij} is $(DI_{ki} \cdot NUI_{ki} \cdot PNI_i)$. However, if DI_{ij} is less than 1, only a fraction of the original weight is transferred from the using to the used process. The remaining weight stays with the using process. To consider this for each outgoing use relation of a process, we need to add the remaining weight, which is defined as $[(1 - DI_{ki}) \cdot NUI_{ki} \cdot PNI_i]$, to the self-directed relation. Applying this to Eq. (10) requires splitting the second summand into two sub-summands, which represent the weight transfers through use relations and through the self-directed relations, respectively. Integrating these changes leads to the final *PPR* algorithm that determines a network-adjusted need for improvement index *NPNI* for each process in the process network. Again, for better legibility, we refer to the *NSAI* of a process i as NUI_{ij} with $i = j$. Setting $DI_{ij} = 0$ for $i = j$ allows further simplifications. Together, this leads to Eq. (11). The complete *PPR* formula without the simplifications can be found in Appendix B.

$$NPNI(i) = (1-d) \cdot \frac{PNI_i}{\sum_{j=1}^{n} PNI_j} + d \cdot \left[\begin{array}{l} \sum\limits_{k \in I_i \setminus i} NPNI(k) \cdot \dfrac{DI_{ki} \cdot NUI_{ki} \cdot PNI_i}{\sum_{l \in O_k} NUI_{kl} \cdot PNI_l} \\ + NPNI(i) \cdot \sum\limits_{m \in O_i} \dfrac{(1 - DI_{im}) \cdot NUI_{im} \cdot PNI_m}{\sum_{q \in O_i} NUI_{iq} \cdot PNI_q} \end{array} \right] \qquad (11)$$

5 Evaluation

5.1 Validation of the Design Propositions

Before discussing whether the *PPR* meets the design propositions, we validated the propositions. This validation is a preparatory activity for the artificial perspective on EVAL2. On the one hand, the propositions align with descriptive knowledge on process performance management and BPAs and with the prescriptive knowledge on network analysis. One the other, we validated the design propositions via an online questionnaire with a group of ten BPM experts from industry and academia. Table 1 summarizes the experts' characteristics, where the bold numbers indicate how many experts meet a characteristic. For example, 2 experts were from academia, 6 from industry (4 from the IT domain, 2 from machine engineering, 1 from online retail, and 2 are unknown). Table 1 showcases that the experts had great experience in BPM, i.e., about eleven years on average.

After a brief introduction of the *PPR*'s idea, the questionnaire included four cases, each of which aimed to validate a distinct design proposition. The cases were very similar to enable the experts isolating the effects to be validated. Each case contained a process network with four processes (i.e., A to D) as well as use relations to capture the idea of the related design proposition. The cases also provided information about the process network (i.e., *PNI*, *NSAI*, *NUI*). Each case proposed a ranking and a rationale. The rationale was aligned with the related design proposition, unknown to the experts. For each case, we asked the experts whether they agree with the ranking and rationale. The complete questionnaire can be found in Appendix C. Table 2 overviews the cases, results, and expert comments.

The four cases were set up as follows:

> In case 1, all processes had the same *PNI* and each process had a self-directed relation with the same *NSAI*. There were no use relations among the processes as the case intended to validate design proposition (P.1), which requires the prioritization of processes with a higher *PNI*.

Table 1 Summary of characterizing data about experts

Industry	Academia	2	IT	4	Machine engineering	1	Online retail	1	Unknown	2
Number of employees	1–100	1	101–1000	4	1001–10,000	1	10,000+	3	Unknown	1
Years of experience in BPM	3–5	3	6–10	2	10–15	4	15+	1	Unknown	0

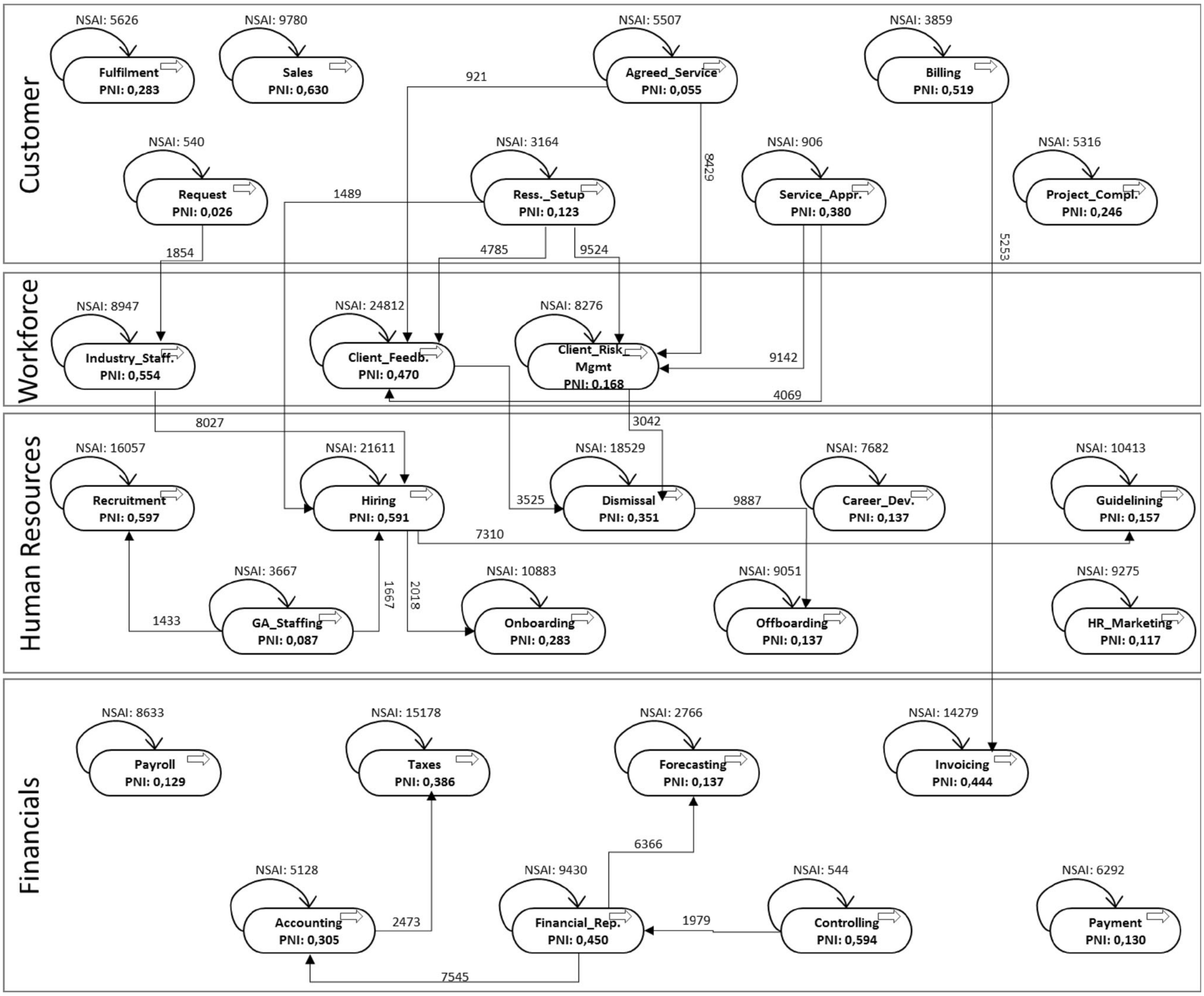


Fig. 3 Process network of the European nearshoring provider

Case 2 introduced use relations from process A to C and from process B to D, with a higher weight given to the latter use relation. This change aimed to validate design proposition (P.2), which requires the prioritization of one process over another if it is, ceteris paribus, used by an additional process, or if an existing use relation has a higher *NUI* or *DI* than another process.

Case 3 introduced another use relation from process B to C to validate (P.3). This design proposition ensures that a process is prioritized over another process if it, ceteris paribus, uses less processes or if the existing use relations have a lower *NUI* or *DI* than another process. While case 2 focused on a higher *NUI* on an existing relation, this case focuses on an additional relation.

Case 4 validates design proposition (P.4), which considers transitive relations within process networks. To do so, we kept the use relations from case two between the processes A and C as well as between B and D, and we gave them equal weights. However, we changed PNI_{B} to

Table 2 Results of validating the design propositions

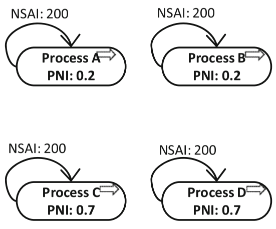
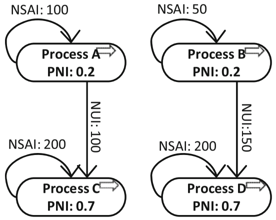
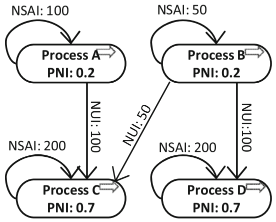
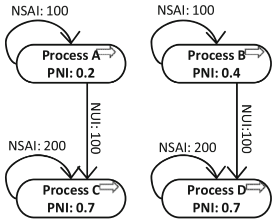
Case		Process	Rank	Comments	
CASE 1		Process A	2	E02	Process prioritization largely depends on whether the process is a business or support process.
		Process B	2	E05	Suggestion to integrate additional criteria needed for process prioritization.
		Process C	1	E08	True, if differentiation between business and support process is contained in the PNI.
		Process D	1		
		Agreement	**8 / 10**		
CASE 2		Process A	3	E02	Process prioritization largely depends on whether the process is a business or a support process.
		Process B	4	E04	To consider including differentiation between business and support processes.
		Process C	2	E05	Value from process improvement should be taken into account.
		Process D	1	E06	Process A should be prioritized over Process B since it is executed more often than Process B.
		Agreement	**7 / 10**	E08	Generally agree with prioritization, but Process A should be prioritized over Process B due to the higher NSAI.
CASE 3		Process A	3	E02	Process prioritization largely depends on whether the process is a business or a support process.
		Process B	4	E04	To consider including differentiation between business and support processes.
		Process C	1	E06	Process A should be prioritized over Process B since it is executed more often than Process B.
		Process D	2		
		Agreement	**8 / 10**		
CASE 4		Process A	4	E02	Process prioritization largely depends on whether the process is a business or a support process.
		Process B	3	E06	The argument generally seems reasonable. More information on how the *PNI* is constructed, and how the PNI of processes are related is needed to fully support the statement.
		Process C	2		
		Process D	1		
		Agreement	**8 / 10**		

Table 3 Results of applying the PPR to the provider's process network

Process	Area	*PNI*	*NPNI*	Rank *PNI*	Rank *NPNI*	Rank difference
Client feedback	WF	0.487	0.097	2	1	1
Hiring	HR	0.477	0.095	4	2	2
Taxes	F	0.435	0.094	6	3	3
Invoicing	F	0.534	0.092	1	4	−3
Payment	F	0.482	0.074	3	5	−2
HR governance	HR	0.228	0.060	13	6	7
Payroll	F	0.374	0.057	7	7	0
Client risk management	WF	0.229	0.044	12	8	4
Onboarding	HR	0.196	0.042	16	9	7
Forecasting	F	0.119	0.042	20	10	10
Resource setup	C	0.472	0.041	5	11	−6
Industry staffing	WF	0.226	0.033	14	12	2
Financial reporting	F	0.249	0.032	11	13	−2
Accounting	F	0.307	0.028	10	14	−4
Customer request	C	0.358	0.027	8	15	−7
Controlling	F	0.334	0.026	9	16	−7
Sales	F	0.146	0.022	17	17	0
Fulfillment	C	0.130	0.020	19	18	1
Billing	F	0.209	0.016	15	19	−4
Service approval	C	0.146	0.011	18	20	−2
Recruitment	HR	0.054	0.008	23	21	2
Service ADJUSTMENT	C	0.085	0.007	21	22	−1
HR marketing	HR	0.042	0.006	24	23	1
GA staffing	HR	0.080	0.006	22	24	−2
Offboarding	HR	0.026	0.006	28	25	3
Project completion	C	0.033	0.005	26	26	0
Career development	HR	0.029	0.004	27	27	0
Dismissal/resigning	HR	0.036	0.004	25	28	−3

HR human resources processes, *F* financial processes, *WF* workflow processes, *C* customer processes

a higher value, such that the network-adjusted index $NPNI_B$ also rose relative to process A.

Only one expert (E02) disagreed with all proposed rankings and rationales, arguing that process prioritization depends on whether a process is a business or a support process. Our response to this comment is twofold. First, if a business process uses a support process, this will affect the performance of the business process. If the support process is, in fact, the bottleneck of the business process, improving the support process should be prioritized. Second, if decision-makers intend to focus on improving business processes as compared to support processes, they can capture this preference when instantiating the *PNI*. The *PNI* is lower if a process' target state is lower because it depends on the difference between the target and actual performance. If decision-makers have a low aspiration regarding the performance of support processes, the target state should not be as high as if the decision-maker expected excellent performance. Thus, the *PNI* of support processes decreases with low performance aspirations, which in turn leads to a higher ranking of business processes in general.

Experts E08 and E04 argued that some way to include a differentiation between business and support processes may be helpful. Nevertheless, they agreed with the rankings and rationales. Expert E05 suggested that more than one variable should be used to characterize processes and disagreed with the first case. However, the *PNI* is a variable that characterizes a process' need for improvement according to multiple performance dimensions. As the questionnaire focused on validating the design propositions, we only briefly introduced the *PNI*'s constituents. Expert E05's suggestion to include the value of improvement projects can be captured via the *PNI*. The *PNI* depends, among others, on the target performance, which can be derived using benchmarking, project candidate evaluation, or expert estimations. If the target performance is set to the expected target performance after the implementation of an improvement project, the value of the improvement is considered in process prioritization. Two experts (E06, E08) commented that process A should be prioritized over process B in cases two and three (E06) due to a higher *NSAI*. However, this was due to an incorrect interpretation of the *NSAI* as the amount of instances of the process, instead of the number of instances the process was executed without using other processes. For the last case, expert E06 disagreed with the statement considering (P.4) due to a lack of information given on the construction of the *PNI*, but confirmed the reasoning. We resolved other misinterpretations in brief bilateral interactions with the experts.

In sum, nine out of ten experts approved our design propositions fully or to great extent. This result corroborates the experts' strong consensus. Two experts explicitly commented that they very much liked the idea of considering interconnectedness when prioritizing processes. Based on these design propositions, we discuss in Sect. 5.3 whether the *PPR*'s design specification aligns with the research problem and contributes to extant knowledge, as part of EVAL2.

5.2 Expert Interview at a Global Online Retailer

As a naturalistic validation of the *PPR*'s design specification, we conducted a 3-h semi-structured interview where we discussed the *PPR*'s design specification with an industry expert (IE) who also participated in the validation of the design propositions. This interview covers the naturalistic perspective on EVAL2. The interview was structured along predefined evaluation criteria, i.e., real-world fidelity, understandability, expected impact on the artifact environment, and applicability (Sonnenberg and vom Brocke 2012).

The IE is working at a data-driven global online retailer that sells a wide range of products and has over 100,000 employees. That company permanently strives for new business opportunities, entailing a constant need for process redesign. It also aims for operational excellence, an objective requiring effective process prioritization. The IE has over 15 years of BPM experience and change management, and is working as a senior process manager at one of the retailer's distribution centers. The IE's main responsibility is process improvement, which makes process prioritization an integral task of his daily business. The company's strong focus on data and the IE's experience make the IE a suitable discussion partner for challenging the *PPR*. The IE expressed great interest in the idea of including process interconnectedness into process prioritization and hoped getting the opportunity to integrate the *PPR* in his company. The IE agreed with the *PPR*'s design specification, deeming the *PPR* a valid solution to the problem including process interconnectedness into process prioritization. Below, we outline the IE's subjective assessment of the evaluation criteria mentioned above.

As for real-world fidelity, the IE agreed that the *PPR* covers most constellations that occur in his company as it integrates the processes' individual need for improvement, the processes' interconnectedness, the number of use instances, and a dimension-specific dependence intensity. The IE considered the *PPR* as flexible and applicable to numerous real-world settings as it includes various possibilities for customization, e.g., the ability to adapt the target state and to weigh the included performance dimensions depending on the application context. The IE also mentioned that in a human-intensive work environment such as that of his company, he would appreciate a way to include specific staff requirements within the *PNI*, such as hazard potential or ease of training. However, the IE agreed that such effects would not cascade through the process network, a circumstance that makes including this additional dimension in the *PPR* rather easy. The IE also confirmed that the *PPR* is understandable for experienced experts such as typically involved in process prioritization decisions.

Regarding the *PPR*'s impact on artifact environment and users, the IE expected that already a discussion of the *PPR*'s problem statement would change the way users think about process prioritization. In the IE's opinion, using the *PPR* would facilitate a mindset shift as users tend to treat business processes as isolated entities. Further, the IE indicated that the *PPR* is likely to harmonize and promote the traceability of process prioritization decisions via clear guidelines on how to incorporate the interconnectedness. In the past, the IE tried to include process interconnectedness on his own experience, but lacked capabilities to quantify relevant constructs. According to the IE, the *PPR* solves this issue and supports users by making the integration of such effects less dependent on subjective influences. Further even if decision-makers account for relations among processes when prioritizing processes in their area of responsibility, processes from other areas of responsibility as well as the dependencies considering those processes are not included. Therefore, the *PPR* enables companies to create an integrated process prioritization across all departments.

The IE confirmed that the *PPR* would be applicable in his company as the company is highly process-oriented and collects almost all parameters via BPM tools. This is why most of the *PPR*'s input parameters can be gathered in a relatively short time span. The IE considered changing employee mindset as the key challenge associated with the *PPR*'s application. In his opinion, employees of data-driven companies are more receptive to data-driven models such as the *PPR*. However, he also assessed that companies that are not as data-driven, will have more problems with collecting all input parameters. The more data-driven a company, the more easily to apply the *PPR*.

5.3 Demonstration Example at a European Nearshoring IT Provider

5.3.1 Case Company and Business Process Architecture

To show the *PPR* in action and to demonstrate the applicability of our software prototype, we present a demonstration example based on a real BPA. This BPA was provided by a BPM expert who is working at a European nearshoring IT provider and who also participated in the design propositions' validation. To meet the requirements of an artificial ex-post evaluation (EVAL3), we transformed the BPA into a process network, applied the *PPR*, and discussed the results. In addition, we used the results to illustrate that the *PPR* implements the design propositions, as this is hard to show exclusively based on the design specification. This analysis covers the artificial perspective of EVAL2.

The European nearshoring IT provider has over 1000 employees, operating its headquarters in Romania. The provider serves customers from industries like IT, automotive, or logistics – mainly based in Europe, but also in the United States. The provider supports customers in all steps of the software development lifecycle as well as in application management. Serving major international companies makes excellent processes one of the providers' primary goals. To enhance its BPM capabilities and get an overview of its processes, the provider recently developed a BPA. On the top-most level, the BPA included 48 processes and 30 use relations. The BPA covered business, support, and management processes structured along four process areas, i.e., customer, workforce, human resources, and financial processes. Relations among these processes exist within and across process areas. In this BPA, processes from the upper areas use processes from the lower areas. Figure 3 shows the process network that we derived from the provider's BPA.

As the BPA was under construction when we investigated the provider, detailed performance data was not available yet. This is why we had to generate data for the purposes of this demonstration example. However, the example comes very close to a real-world case study because of the included real-world processes and relations, but it is not a full-fledged one due to the lack of performance data. Please find more information about how we transformed the given BPA, how we generated suitable input data, and about which data we used in Appendix D.

In general, input data required to apply the *PPR* can be collected from various sources. As for the *PNI*, actual performance data of the involved processes can be gathered from process performance management systems or extant enterprise systems (e.g., enterprise resource planning, supply chain management, or workflow management systems). Analogous to other decision models, target performance values and weights of performance dimensions must be set by experts (e.g., BPM experts, process owners, corporate controllers, or senior managers). Experts can use internal or external benchmarks and/or apply methods from corporate planning and forecasting, consensus measurement, or multi-criteria decision analysis (e.g., Delphi studies, analysis of historical data, Analytical Hierarchy Process). The same holds for process-specific performance boundaries regarding time and quality. The amount of executions can be retrieved from enterprise systems or estimated based on expert assessments. The dependence intensity can be quantified as the conditional probability of good performance of using processes if used processes perform well. Dependencies among the processes can be derived based on a BPA or from process models. As for the dampening factor, only heuristics are available in the literature. An appropriate company-specific value can only be determined via a scenario analysis. Finally, we would like to highlight that process logs are a very valuable data source for the *PPR*. Given high-quality process logs, parameters including the actual performance, amount of executions, dependencies, and their intensity can be mined. In such settings, only target values, weights, and boundaries must be estimated.

With the process network containing many processes and relations, it becomes obvious that, in industry-scale settings, there generally is neither a trivial nor an intuitive answer to the question how to prioritize processes for improvement purposes. To prioritize processes in line with their individual need for improvement and interconnectedness, prescriptive knowledge as provided by the *PPR* is necessary. As a recursive algorithm whose complexity heavily grows with the number of processes and relations, the *PPR* cannot be feasibly applied without a software instantiation. We thus implemented a software prototype that efficiently handles arbitrary process networks and analyzes the robustness of prioritization results in line with the decision-makers' preferences. In fact, it took the *PPR* prototype less than a minute to process the network at hand on an ordinary workstation, including the robustness analysis.

5.3.2 Analysis of the Results

Table 3 shows the results of applying the *PPR* to the process network we derived based on the European nearshoring IT provider's BPA. Note that these results are case-specific. We do not claim that these results are generalizable due to the high number of input parameters. From the left to the right, Table 3 includes the involved processes and process areas (HR: human resources, WF: workforce, F: financials, C: customer). It also lists the processes' individual need for improvement index *PNI*, the network-adjusted need for improvement index *NPNI*, the related rankings, and rank differences. Please consider that the *PNI* and *NPNI* values cannot be directly compared as each *PNI* stems from the interval [0;1], whereas the *NPNI* values sum up to 1. Instead, the rankings and rank differences should be used to interpret the *PPR* results. Table 3 is sorted descending according to the *NPNI* and the resulting ranking.

A first view on the results shows that the process network contains processes with a moderately high individual need for improvement (e.g., Client Feedback, Hiring) and processes with a very low individual need for improvement index (e.g., Project Completion, Career Development). In line with the *PPR*'s constitutive idea, we see processes whose network-adjusted rank is higher or lower than their individual rank as well as processes whose network-adjusted rank equals the individual rank. For example, the Forecasting process is ranked higher than from a stand-alone perspective. The opposite holds true for the Customer Request and Controlling processes. This is because the *PPR* adjusts the processes' individual need for improvement according their interconnectedness, with interconnectedness being measured via the number of use and stand-alone instances as well as the dependence intensity. Overall, the stand-alone and the network-adjusted ranking are positively correlated, featuring a Spearman rank correlation coefficient of 0.88. Even if some processes show greater differences regarding their individual and network-adjusted ranks, the *PPR* does not confound, but carefully adjust the individual ranking. This is reasonable as we applied the *PPR* using a dampening factor of 0.5, meaning that the processes' individual need for improvement and interconnectedness affect the network-adjusted need for improvement in equal shares. Other values for the dampening factor would have yielded other network-adjusted rankings. A value of 0.5 is reasonable, as it is unrealistic in industry that the processes' interconnectedness receives substantially more weight than their individual need for improvement. This assessment was confirmed by our BPM experts and in particular by the expert working for the nearshoring provider.

An in-depth analysis reveals that customer processes – except for Customer Request and Resource Setup – tend to have lower individual ranks and drop in the network-adjusted ranking. The reason is that most customer processes have a rather low *PNI* and many outgoing relations. No customer process is used by other process. The ranks of workforce processes, however, are rising as they are intensively used by customer processes. Changes in the ranking of human resources processes are diverse. Some processes rise (e.g., HR Governance), some drop (e.g., GA Staffing), and others remain unchanged (e.g., Career Development) in the ranking. One reason is that human resources processes feature a different interconnectedness regarding use relations. In addition, human resource processes have a very low individual need for improvement, except for Hiring. Financial processes mostly drop in the ranking, but stay in the upper half of the network-adjusted ranking. The reason is that financial processes have a comparatively high individual need for improvement. The only exception is the Forecasting process that has a rather low individual need for improvement, is directly used by Financial Reporting as well as transitively by Controlling. By trend, processes (i.e., Hiring, Client Feedback, Client Risk Management) that are often used by other processes and/or have a high individual need for improvement, raise in the network-adjusted ranking. Processes (i.e., Resource Setup, Customer Request) that use many processes and are not used by other processes drop in the network-adjusted ranking. The three best-ranked processes (i.e., Client Feedback, Hiring, Taxes) are heavily used and have a high need for improvement. Other process parameters such as the dependence intensity and the amount of executions, which are only shown in the Appendix, corroborate these results.

The demonstration example confirms that the *PPR* implements the design propositions derived in Sect. 4.3. As we brought forward the key arguments above, we provide only a short justification here. Design proposition (P.1), which deals with the processes' individual need for improvement, becomes manifest in the processes Payment and Payroll. Payment has a higher *PNI* than Payroll. Both processes have no connections to other processes. Consequently, Payment has a higher *NPNI* than Payroll. Design propositions (P.2) and (P.3), which address direct ingoing and outgoing use relations, can be discussed based on the processes GA Staffing and Recruitment. Without considering network effects, GA Staffing is ranked better than Recruitment. As GA Staffing uses Recruitment, the *NPNI* of Recruitment exceeds that of GA Staffing, in line with design proposition (P.2). This case also holds true as for design proposition (P.3). As GA Staffing uses Recruitment, the *NPNI* of Recruitment exceeds that of GA Staffing. The processes Invoicing and Taxes help discuss design proposition (P.4), dealing with transitive relations. Both processes are used by a single but different process and do not use other processes. Although Invoicing has a higher individual need for improvement than Taxes, it is used by a process with a lower *NPNI* (i.e., Billing) than Taxes (i.e., Accounting). Together with the effects of the amount of executions and the number of use instances, Taxes is in the end ranked better in the network-adjusted ranking. When discussing the design propositions, consider that design propositions are idealized axioms building on a 'ceteris paribus' assumption. While the design propositions help guide the design of the *PPR*, their effects are not strictly separable in practice. Typically, design propositions take effect simultaneously if the *PPR* is applied to prioritize processes in real-world settings.

To assist decision-makers in assessing the quality of the *PPR* results and identifying those input parameters that strongly influence process prioritization decisions, we finally report on the robustness analysis offered by our software prototype. The prototype uses simulation where decision-makers can define the number of iterations, the value range to be analyzed, the category of input parameters to be investigated (e.g., number of use and stand-alone instances, amount of executions, custom weights, dampening factor, and the processes' actual and target performance). In each iteration, the prototype randomly draws values of the chosen parameter category from the predefined intervals. The prototype finally compares the simulation results with the original results using the average Spearman rank correlation coefficient. In our demonstration example, we chose 1.000 iterations and set the value range of the input parameters to [−30; +30%]. The average Spearman rank correlation coefficient was 0.980 when varying the number of use and stand-alone instances and amount of executions. Furthermore, it was 0.992 for the dampening factor and 0.994 for the custom weights. These results show that the *PPR* results are very robust regarding variations of these parameters. Hence, estimation inaccuracies hardly affect the *PPR* results. This is good as these input parameters tend to be hard-to-estimate. By contrast, varying the processes' actual and target performance influences the *PPR* results more strongly. A variation within the interval [−10; +10%] yields an average rank correlation coefficient of 0.468. This is reasonable as the actual and target performance are relevant for each process. It would be surprising if the *PPR* results did not change in case of different performance values. Further, process performance is easier to estimate compared to other parameters such that a higher variation is tolerable.

As part of EVAL3, this demonstration example illustrated that the *PPR* efficiently applies to larger process networks – in this case: based on a real BPA of a European nearshoring IT provider – and yields interpretable results. The results were robust regarding inaccuracies of hard-to-estimate input parameters (e.g., the number of use and stand-alone instances) as well as sensitive regarding input parameters related to process performance, which are comparatively easy to assess. The example also showed that the *PPR* implements the design propositions, an investigation that covers the artificial perspective of EVAL2.

6 Conclusion

6.1 Summary and Contribution

With process prioritization being a critical success factor of effective process improvement, this study investigated how business processes should be prioritized based on their own need for improvement and interconnectedness. Adopting the DSR paradigm, we developed the *ProcessPageRank* (*PPR*) that ranks processes from a given BPA in line with their network-adjusted need for improvement. The *PPR* draws from descriptive knowledge on process performance management and BPAs as well as from prescriptive knowledge related to network analysis, particularly the Google PageRank. The *PPR* interprets processes as connected nodes and extends the Google PageRank as a popular centrality measure to identify central nodes in process networks. The network-adjusted need for improvement integrates the processes' individual need for improvement, building on multiple process performance dimensions (i.e., cost, quality, time), with their interconnectedness in the process network, captured via use relations. In the *PPR*, use relations are annotated with the number of use instances (i.e., how often a process uses another process) and a dependence intensity (i.e., how strongly a process' performance depends on the processes it uses) in order to not only reflect whether, but also how intensely processes are interconnected.

Following the evaluation framework as per Sonnenberg and vom Brocke (2012), we validated the *PPR*'s design specification by conducting an in-depth expert interview at a global online retailer and discussing it against design propositions in the course of a demonstration example. We derived the design propositions from the descriptive knowledge on process performance management and BPA, operationalized them using prescriptive knowledge on network analysis, and validated them with BPM experts from academia and industry. Finally, we instantiated the *PPR*'s design specification as a software prototype and applied the prototype to a real BPA from a European nearshoring IT provider.

The *PPR* adds to the prescriptive knowledge on process prioritization as it is the first approach to account for process interconnectedness when prioritizing processes for improvement purposes. The *PPR* also is the first approach to apply the mature knowledge on centrality measures to process decision-making in general as well as to process prioritization in particular.

6.2 Limitations and Future Research

While validating the *PPR*'s design specification and applicability, we identified directions in which the *PPR* should be advanced. Below, we present these directions together with ideas for future research.

Regarding its design specification, the *PPR* quantifies the need for improvement of individual processes based on performance indicators to operationalize process dysfunctionality. Even though the *PPR* allows for the integration of indicators from virtually any performance dimension, we only specified it for the cost, time, and quality dimensions as well as for indicators with the same polarity. Thus, the *PPR* may be extended to include other performance dimensions, depending on the domain where it is applied. In addition, the *PPR* prioritizes processes according to their network-adjusted need for improvement. Depending on the project candidates available for process improvement, however, improving the process with the highest network-adjusted need for improvement is not necessarily optimal. If processes A and B are ranked first and second, but the project candidate for process B requires far lower investment than that for process A, it might be reasonable to improve process B first. The same holds if a much less risky project candidate is available for process B. This argument relates to the 'difficulty to improve' construct used in non-performance-based process prioritization approaches. Thus, the *PPR* may be extended regarding an economic valuation and a project management perspective. Regarding the validation of the design propositions based on which we developed the *PPR*, we concede that the expert group only included ten members, even if these experts were very experienced. Regarding the in-depth interview with the expert from the global online retailer, we admit that the expert's assessment may be positively biased towards data-driven BPM approaches due his great experience and the retailer's BPM capabilities.

Currently, the *PPR*'s applicability is limited due to its high data requirements. While some parameters can be retrieved from enterprise systems or derived with reasonable effort (e.g., actual performance and number of executions), other parameters must be assessed by domain experts (e.g., target performance, weights of performance dimensions, the dampening factor). This limitation, however, does not only apply to the *PPR*, but to all data-driven BPM approaches, e.g., process mining, process intelligence, or predictive performance monitoring. Due to the uptake of process-aware information systems, we are confident that high-quality process (log) data will be available in the near future to enhance the *PPR*'s applicability. In such settings, only the performance target and boundaries as well as dimension-specific weights must be estimated by experts. Although the presented demonstration example builds on a real-world BPA and was inspired by our industry experience, it is not a full-fledged real-world case study. Depending on available process data, future research should focus on conducting further interviews in different contexts to further validate the *PPR*'s real-world fidelity and case studies to validate the *PPR*'s applicability. Thereby, future research should set up a knowledge base to institutionalize data collection routines. To facilitate future case studies, we recommend advancing the software prototype in such a way that it can be used more conveniently and implements more sophisticated analysis functionality.

References

Armistead C, Pritchard JP, Machin S (1999) Strategic business process management for organisational effectiveness. Long Range Plan 32(1):96–106

Bandara W, Guillemain A, Coogans P (2015) Prioritizing process improvement: an example from the Australian financial services sector. In: vom Brocke J, Rosemann M (eds) Handbook on business process management 2, 2nd edn. Springer, Heidelberg, pp 289–307

Bolsinger M (2015) Bringing value-based business process management to the operational process level. Inf Syst eBus Manag 13(2):355–398

Bonacich P (1987) Power and centrality: a family of measures. Am J Sociol 92(5):1170–1182

Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine. Comput Netw 30(1–7):107–117

Buhl HU, Röglinger M, Stöckl S, Braunwarth KS (2011) Value orientation in process management. Bus Inf Syst Eng 3(3):163–172

Chakravorty SS (2010) Where process-improvement projects go wrong. Wall Street J 255(19)

Cohon JL (2004) Multiobjective programming and planning. Dover Publishings, New York

Davenport TH (1993) Need radical innovation and continuous improvement? Integrate process reengineering and TQM. Strateg Leadersh 21(3):6–12

Dijkman R, Vanderfeesten I, Reijers HA (2016) Business process architectures: overview, comparison and framework. Enterp Inf Syst 10(2):129–158

Dumas M, La Rosa M, Mendling J, Reijers HA (2013) Fundamentals of business process management. Springer, Heidelberg

Frese E (1995) Grundlagen der Organisation: Konzept – Prinzipien – Strukturen, 6th edn. Gabler, Wiesbaden

Gaitanides M (1983) Prozessorganisation: Entwicklung, Ansätze und Programme prozessorientierter Organisationsgestaltung. Vahlen, München

Gregor S, Hevner AR (2013) Positioning and presenting design science research for maximum impact. Manag Inf Syst Q 37(2):337–356

Hammer M, Champy J (1993) Reengineering the corporation – A manifesto for business revolution. Bealey, London

Hanafizadeh P, Moayer S (2008) A methodology to define strategic processes in organizations: an exploration study in managerial holding companies. Bus Process Manag J 14(2):219–227

Hanafizadeh P, Osouli E (2011) Process selection in re-engineering by measuring degree of change. Bus Process Manag J 17(2):284–310

Hanneman RA, Riddle M (2005) Introduction to social network methods. University of California, Riverside

Harmon P (2014) Business process change, 3rd edn. Morgan Kaufmann, Burlington

Harmon P, Wolf C (2014) The state of business process management 2014. http://www.bptrends.com/bpt/wp-content/uploads/BPTrends-State-of-BPM-Survey-Report.pdf. Accessed 3 Aug 2016

Keeney RL, Raiffa H (1993) Decisions with multiple objectives: preferences and value trade-offs. Cambridge University, Cambridge

Kohlbacher M, Reijers HA (2013) The effects of process-oriented organizational design on firm performance. Bus Process Manag J 19(2):245–262

La Rosa M, Reijers HA, van der Aalst WMP, Dijkman RM, Mendling J, Dumas M, Garcia-Banuelos L (2011) APROMORE: an advanced process model repository. Expert Syst Appl 38(6):7029–7040

Landherr A, Friedl B, Heidemann J (2010) A critical review of centrality measures in social networks. Bus Inf Syst Eng 2(6):371–385

Langville AN, Meyer CD (2011) Google's PageRank and beyond: the science of search engine rankings. Princeton University, Princeton

Lehnert M, Röglinger M, Seyfried J, Siegert M (2015). ProcessPageRank - A Network-based Approach to Process Prioritization Decisions. In: Proceedings of the 23rd European Conference on Information Systems (ECIS), May 2015

Leyer M, Heckl D, Moormann J (2015) Process performance measurement. In: vom Brocke J, Rosemann M (eds) Handbook on Business Process Management 2, 2nd edn. Springer, Heidelberg, pp 227–241

Linhart A, Manderscheid J, Röglinger M, Schlott H (2015) Process improvement roadmapping – How to max out your process. In: Proceedings of the 36th International Conference on Information Systems (ICIS). Fort Worth, December 2016

Malinova M, Leopold H, Mendling J (2014) A meta-model for process map design. CAiSE Forum. pp 16–20

Malone TW, Crowston K (1994) The interdisciplinary study of coordination. ACM Comput Surv 26(1):87–119

Manderscheid J, Reißner D, Röglinger M (2015) Inspection coming due! How to determine the service interval of your processes. In: Motahari-Nezhad HR, Recker J, Weidlich M (eds) Business process management – BPM2015, LNCS, vol 9253. Springer, Heidelberg, pp 19–34

Mansar SL, Reijers HA, Ounnar F (2009) Development of a decision-making strategy to improve the efficiency of BPR. Expert Syst Appl 36(2):3248–3262

March ST, Smith GF (1995) Design and natural science research on information technology. Decis Support Syst 15(4):251–266

Meredith JR, Raturi A, Amoako-Gyampah K, Kaplan B (1989) Alternative research paradigms in operations. J Oper Manag 8(4):297–326

Mertens P (1996) Process focus considered harmful? WIRTSCHAFTS-INFORMATIK 38(4):446–447

Newman M (2003) The structure and function of complex networks. SIAM Rev 45(2):167–256

Ohlsson J, Han S, Johannesson P, Carpenhall F, Rusu L (2014) Prioritizing business processes improvement initiatives the seco tools case. In: Jarke M, Mylopoulos J, Quix C, Rolland C, Manolopoulos Y, Mouratidis H, Horkoff J (eds) Advanced information systems engineering, LNCS, vol 8484. Springer, Heidelberg, pp 256–270

Olding E, Rosser B (2007) Getting started with BPM, Part 3: understanding critical success factors. GR Reports. Gartner, Stamford

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manag Inf Syst 24(3):45–77

Probst F, Grosswiele L, Pfleger R (2013) Who will lead and who will follow: identifying influential users in online social networks. Bus Inf Syst Eng 5(3):179–193

Ray S, Jewkes EM (2004) Customer lead time management when both demand and price are lead time sensitive. Europ J Oper Res 153(3):769–781

Recker J, Mendling J (2016) The state of the art of business process management research as published in the bpm conference. Bus Inf Syst Eng 58(1):55–72

Reijers HA, Mansar SL (2005) Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega 33(4):283–306

Rosemann M, vom Brocke J (2015) The six core elements of business process management. In: Brocke J, Rosemann M (eds) Handbook on business process management 1, 2nd edn. Springer, Heidelberg, pp 105–122

Setzer T, Bhattacharya K, Ludwig H (2010) Change scheduling based on business impact analysis of change-related risk. IEEE Transact Netw Serv Manag 7(1):58–71

Shrestha A, Cater-Steel A, Toleman M, Tan WG (2015) A method to select IT service management processes for improvement. J Inf Technol Theory Appl 15(3):31–56

Sonnenberg C, vom Brocke J (2012) Evaluations in the science of the artificial – reconsidering the build-evaluate pattern in design science research. In: Peffers K, Rothenberger M, Kuechler B (eds) Proceedings of the 7th International conference on design science research in information systems: advances in theory and practice (DESRIST 2012), Las Vegas, pp 381–397

van der Aalst WMP (2013) Business process management: a comprehensive survey. ISRN Softw Eng 2013:507984-1–507984-37. doi:10.1155/2013/507984

Venable J, Pries-Heje J, Baskerville R (2012) A comprehensive framework for evaluation in design science research. In: Peffers K, Rothenberger M, Kuechler B (eds) Proceedings of the 7th International conference on design science research in information systems: advances in theory and practice (DESRIST 2012), Las Vegas, pp 423 – 438

vom Brocke J, Sonnenberg C (2015) Value-orientation in business process management. In: vom Brocke J, Rosemann M (eds) Handbook on business process management 2, 2nd edn. Springer, Heidelberg, pp 101–132

vom Brocke J, Becker J, Braccini AM, Butleris R, Hofreiter B, Kapočius K, De Marco M, Schmidt G, Seidel S, Simons A, Skopal T, Stein A, Stieglitz S, Suomi R, Vossen G, Winter R, Wrycza S (2011) Current and future issues in BPM research: a European perspective from the ERCIS meeting 2010. Commun Assoc Inf Syst 28(1):393–414

Winter R, Fischer R (2007) Essential layers, artifacts, and dependencies of enterprise architecture. J Enterp Archit 3(2):7–18