PROCESSPAGERANK – A NETWORK-BASED APPROACH TO PROCESS PRIORITIZATION DECISIONS

Complete Research

Lehnert, Martin, FIM Research Center, University of Augsburg, Universitaetsstrasse 12, 86159 Augsburg, Germany, martin.lehnert@fim-rc.de

Röglinger, Maximilian, FIM Research Center, University of Bayreuth, Friedrich-von-Schiller-Str. 2a, 95444 Bayreuth, Germany, maximilian.roeglinger@fim-rc.de

Seyfried, Johannes, FIM Research Center, University of Augsburg, Universitaetsstrasse 12, 86159 Augsburg, Germany, johannes.seyfried@fim-rc.de

Siegert, Maximilian, FIM Research Center, University of Augsburg, Universitaetsstrasse 12, 86159 Augsburg, Germany, maximilian.siegert@fim-rc.de

Abstract

Deciding which business processes to improve first is a challenge most corporate decision-makers face. The literature offers many approaches, techniques, and tools that support such process prioritization decisions. Despite the broad knowledge about measuring the performance of individual processes and determining related need for improvement, the interconnectedness of processes has not been considered in process prioritization decisions yet. So far, the interconnectedness of business processes is captured for descriptive purposes only, for example in business process architectures. This drawback systematically biases process prioritization decisions. As a first step to address this gap, we propose the Process-PageRank (PPR), an algorithm based on the Google PageRank that ranks processes according to their network-adjusted need for improvement. The PPR is grounded in the literature related to process improvement, process performance measurement, and network analysis. For demonstration purposes, we created a software prototype and applied the PPR to five process network archetypes to illustrate how the interconnectedness of business processes affects process prioritization decisions.

Keywords: Business Process Decision-Making, Business Process Architecture, Decision Support, PageRank, Business Process Improvement, Business Process Prioritization.

1 Introduction

Process orientation is a recognized paradigm of organizational design and a source of corporate performance (Dumas et al., 2013; Kohlbacher and Reijers, 2013). Business process management (BPM) in general and process decision-making in particular receive continued attention from practitioners and researchers (Buhl et al., 2011; vom Brocke et al., 2011). Fundamental to BPM is process improvement, a task that also requires prioritizing which processes to improve (Bandara et al., 2015; van der Aalst, 2013). Process prioritization requires to focus on processes that are of strategic importance or that show significant need for improvement (Bandara et al., 2015; Burlton 2015; Ohlsson et al., 2014). Most approaches to process prioritization neglect that processes are interconnected, a drawback that biases prioritization decisions and must be addressed in further research.

So far, the interconnectedness of processes is only captured for descriptive purposes, for example in process model repositories and business process architectures (BPA) (Dijkman et al., 2014; Malinova et al., 2014). While process model repositories organize large collections of process models to facilitate process modelling, composition, and execution, BPAs identify and visualize relations among processes

(La Rosa et al., 2011; Malinova et al., 2013). As for process prioritization, Bandara et al. (2015) state that available methods are "either of very high level and hence not of much assistance […], or, on the contrary, are so detailed that it can take a significant effort to simply identify the critical processes". However, improving a process according to one or several performance dimensions such as time, quality, or cost largely influences the performance of connected processes – and thus the overall performance of a company's processes (Leyer et al. 2015). Neglecting interconnections among processes also entails operational risks such as a change-related downtime of interconnected processes or disruptions and delays due to a change in process demand (Setzer et al., 2010). What is missing are approaches that provide concrete decision support on process prioritization integrating the need for improvement of single processes with their interconnectedness. Therefore, our research question is as follows: *How can process prioritization decisions be made in line with how processes are interconnected?*

As a first step to answer the research question, we interpret BPAs as networks with processes as interconnected nodes, combining network analysis and BPM research. In this analytical paper, we propose the ProcessPageRank (PPR), an adaptation of the Google PageRank that ranks processes according to their network-adjusted need for improvement and helps prioritize processes for improvement purposes.

The paper is organized as follows: In section 2, we sketch the foundations of BPM and network analysis, and derive high-level requirements. In section 3, we show how to transform BPAs into process networks, concretize the high-level requirements in terms of rationality postulates, and propose the PPR algorithm. In section 4, we apply the PPR to five process network archetypes and compare the results in a cross-case analysis. In section 5, we sum up key results and point to limitations as well as to future research.

2 Theoretical Background and Requirements

2.1 Business Process Management

Business Process Management (BPM) combines knowledge from information technology and management sciences, and applies this to corporate processes (van der Aalst, 2013). Processes split into core, support, and management processes (Harmon, 2010). Core processes are collections of events, activities, and decision points that involve actors and objects, collectively leading to valuable outcomes (Dumas et al., 2013). Support processes ensure that core processes continue to function, whereas management processes plan, organize, communicate, monitor, and control the activities within an organization (Harmon, 2010). In this paper, we focus on core and support processes and refer to both as processes.

Within the BPM lifecycle, process improvement is a fundamental activity (Zellner, 2011). The BPM literature offers numerous approaches to process improvement (Sidorova and Isik, 2010; Zellner, 2011). Many of these approaches focus on quantifying the performance and the need for improvement of single processes in terms of performance measures (Bolsinger, 2014; Dumas et al., 2013; Levina and Hillmann, 2012). Though relying on performance measures from different domains such as investment theory or social network analysis, these approaches share the individual process as unit of analysis. Few process improvement approaches take on a multi-process perspective. Lehnert et al. (2014), for example, propose a decision model to determine which projects an organization should implement in which sequence to balance the improvement of individual processes with the development of BPM capabilities. Ohlsson et al. (2014) propose a method for prioritizing process improvement initiatives. Thawesaengskulthai and Tannock (2008) compare popular quality management and continuous improvement initiatives to support the selection of process improvement projects. All these approaches do not cater for interconnections among processes.

Process performance and the effect of improvement projects are measured in terms of performance indicators (Leyer et al. 2015). Among others, performance indicators refer to the dimensions of the Devil's Quadrangle, i.e., time, cost, quality, or flexibility (Reijers and Liman Mansar, 2005). Some approaches also resolve the partly conflicting nature of these performance dimensions by means of integrated performance measures (Bolsinger, 2014). This leads to our first high-level requirement:

(R.1) *Performance of individual processes*: When prioritizing processes, the individual performance of the processes in focus must be measured in terms of one or more performance indicators and considered in the resulting ranking.

The processes of an organization and their relations are typically modelled as BPAs. A BPA is an organized overview of an organization's processes and their relations, potentially accompanied by guidelines that determine how to organize these processes (Dijkman et al., 2014). The topmost level of a BPA is also referred to as process map or landscape (Malinova and Mendling, 2013). There are four kinds of relations occurring in a BPA, i.e., specialisation, decomposition, use, and trigger (Dijkman et al., 2014). The specialisation expresses that one process is a specialised version of another process. The decomposition expresses that a process is decomposed into multiple sub-processes. Use relations model situations where a process needs the output of another process to continue or complete its execution (synchronous communication). That is, the performance of the using process partly depends on the performance of the used process – not vice versa (Malone and Crowston, 1994). Trigger relations express that one process triggers the execution of another process without having to wait for the other process' output (asynchronous communication). The performance of the triggering and triggered process are independent. This leads to our second high-level requirement:

(R.2) *Relations among multiple processes*: When prioritizing processes, the relations among the processes in focus such as those captured in a BPA must be considered in the resulting ranking.

2.2 Network Analysis

Approaches to identifying important nodes in networks have been applied in fields like IT landscape management, biology, or power grids (Özgür et al., 2008; Simon and Fischbach, 2013; Wang et al., 2010). With the rise of online social networks (OSN), researchers from social network analysis found centrality measures to be very useful. Due to extensive research during the last years, the knowledge base regarding centrality measures can be considered quite mature (Probst et al., 2013).

In the OSN context, there are three especially popular approaches to measure the centrality of a distinct node, i.e., degree centrality (measures the amount of direct neighbours), closeness centrality (measures the shortest path to each node in the network), and betweenness centrality (measures the amount of shortest paths between every two nodes in the network that contain the node in focus) (Freeman, 1977). The drawback of these measures is that local patterns can have a disproportionally high influence on the centrality of a single node (Hanneman and Riddle, 2005). Another centrality measure that accounts for this problem and explicitly acknowledges that connections to influential nodes add more importance to a node than connections to less influential nodes, is the eigenvector centrality (Newman, 2003). The eigenvector centrality extends the concepts of degree and closeness centrality to a node's interconnectedness in the entire network (Hanneman and Riddle, 2005). A popular algorithm, based on the eigenvector of a network's adjacency matrix, is the Google PageRank.

Even though developed for determining the relative importance of a web page compared to all other web pages based on its link structure (Brin and Page, 1998), the PageRank has proven suitable for many other applications like key user identification, word sense disambiguation, or journal ranking (Chen and Chen, 2011; Heidemann et al., 2010; Mihalcea et al., 2004). The original PageRank algorithm as published by Brin and Page (1998) is shown in Formula (1).

$$PR(i) = c \cdot \sum_{j \in I_i} \frac{PR(j)}{|O_j|} \tag{1}$$

The PageRank rises with the number of links that point to node i. The higher the value of $PR(i)$ compared to the PageRank of all other nodes, the more central node i is in the network. The variable c is a constant used for normalization such that the sum of the ranks of all web pages is constant. The set I_i represents the links pointing to node i, and $|O_j|$ represents the number of outgoing links from node j.

The PageRank of node i can be interpreted as follows: For each incoming link, node i receives a share of the PageRank from the respective source node j. The share of its PageRank that node j gives to node i depends on how many links leave node j in total. As the PageRank has a recursive form, Brin and Page introduced the concept of the random surfer to solve the underlying eigenvector problem, where each node receives an initial PageRank of $1/n$ (Brin and Page, 1998). The idea is that a surfer travels through the network using the link structure. Each time the random surfer reaches a node, he randomly chooses one of the outgoing links with an equal probability and follows that link to the next node. Those nodes that the random surfer reaches more often are more central in the network. One drawback of the random surfer model is the problem of isolated networks, i.e., the random surfer cannot reach all nodes if the network consists of isolated sub-networks. Moreover, the random surfer can get stuck in nodes that only have incoming links. To address both drawbacks, the random surfer, at certain times, chooses not to follow the link structure, but to teleport to a random node in the network (Langville and Meyer, 2011). As for teleportation, the probability of reaching a node is equal, i.e., $1/n$, for all nodes independent from their interconnectedness. The question that remains is when the random surfer chooses to follow the link structure as opposed to teleporting. As a solution, the event of following the link structure gets assigned the probability d, whereas the probability of the teleportation is $(1-d)$. So, the teleportation factor $(1-d) \cdot 1/n$ represents the weight of each node without considering the link structure and no node can have a PageRank lower than this value. The probability d indicates which fraction of the PageRank stems from the link structure. When d converges to 1, PageRanks become very volatile to changes in the network structure. High values of d also increase the risk of rank sinks, i.e., nodes without outgoing links concentrate the weight whereas other nodes are ranked disproportionally low. By application on web pages, a d value of 0.85 has been identified as reasonable for addressing the trade-off of either not considering the interconnectedness enough or ending with a very volatile result (Langville and Meyer, 2011). These adjustments lead to the PageRank shown in Formula (2).

$$PR(i) = (1-d) \cdot \frac{1}{n} + d \cdot \sum_{j \in I_i} \frac{PR(j)}{|O_j|} \tag{2}$$

As mentioned, node i receives weight from node j if node j points to node i. The transferred weight depends on how many nodes leave node j, assigning an equal weight to each link. One can easily imagine that weighting all outgoing links equally is not always appropriate. In the case of web pages, for instance, the probability of a surfer following a distinct link depends on the anchor text of the link or on how prominent the link is placed. For that reason, an early adjustments to the PageRank was to give links individual weights (Langville and Meyer, 2011). The weight of the link that points from node j to node i is w_{ji}. Moreover, the probability of reaching an arbitrary node in the event of teleportation was previously described to be the same for each node in the network. However, in one of their early publications, Brin and Page (1998) already mention the possibility of customizing this probability. The only restriction is that each weight is from the interval [0; 1] and that the weights sum up to 1, since they are supposed to be probabilities. Therefore, each node can get assigned an individual weight k_i proportional to the weights of all nodes in the network (Langville and Meyer, 2011). The consideration of individual weights for nodes and links leads to Formula (3), which also serves as foundation of our PPR algorithm.

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

3 The ProcessPageRank

3.1 Translating Business Process Architectures into Process Networks

Building on the Google PageRank, the PPR algorithm requires a network with nodes and edges as input. Such a network can be derived from a BPA. Below, we address all components of a BPA, translate them

into elements of a process network, specify their notation, and indicate which additional information is needed to apply the *PPR* algorithm. Figure 1 shows a collection of interconnected processes as they are depicted in a BPA following the ArchiMate notation and how they are represented as a process network (Dijkman et al., 2014).

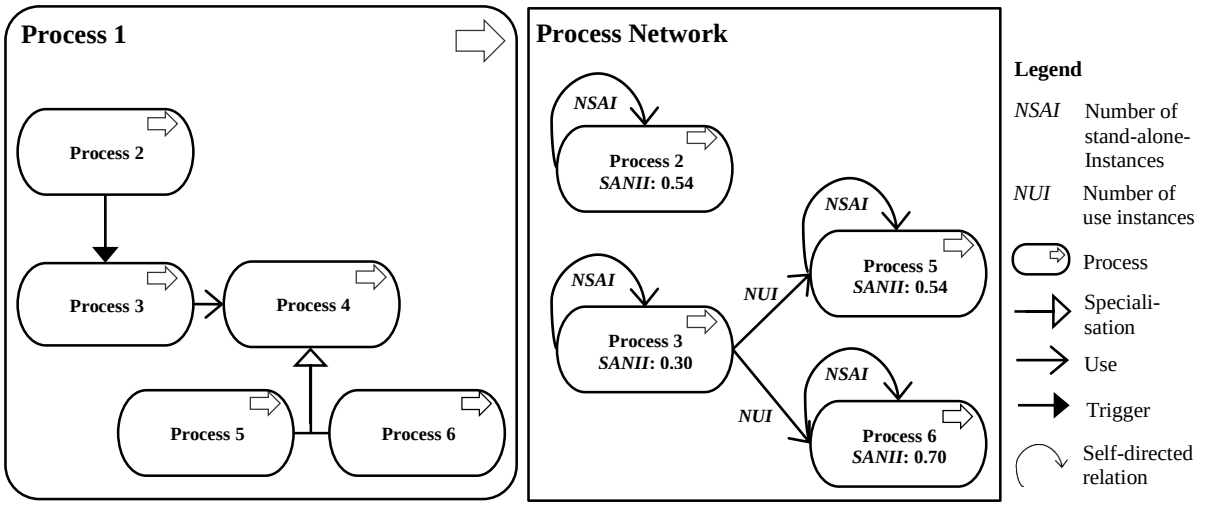


Figure 1. Example of a BPA (left) and a corresponding process network (right)

As a first step, we define each process of the BPA as a node in the process network. For a better understanding, we refer to each node in the process network as process. All processes from the BPA must be included in the process network. We assume that each process is measured in terms of its stand-alone need for improvement, e.g., according to the dimensions of the Devil's Quadrangle (Reijers and Liman Mansar, 2005). Each process has a stand-alone need for improvement index ($SANII$). The $SANII$ can take values from the interval [0; 1], where 0 indicates that the process does not need to be improved and 1 represents the highest possible stand-alone need for improvement. We refrain from further elaborating on how to build such an index and assume that the $SANII$ condenses information related to typical dimensions of process performance. Other important indicators are economic benefits and the contribution to a company's market position or innovation potential. One possible technique for building such an index is the Analytical Hierarchy Process, which has already been used for process redesign (Liman Mansar et al., 2009). Other methods from multi-criteria decision analysis include Techniques for Order Preference Similarity to Ideal Solution (Hwang and Yoon, 1981) or Multiple Attribute Utility Theory (Dyer, 2005). Furthermore, the $SANII$ must reflect the number of instances of a process in order to be able to differentiate between processes that perform equally well, but one is executed more often than the other. We assume that the $SANII$ of all processes can be compared.

As a second step, the relations among the processes as modelled in the BPA must be transferred to the process network. As for decomposition relations, either the decomposed process is modelled as a single process or all its component processes are modelled in the process network, depending on the intended level of granularity. In case of specialisation relations, we assume that all relations regarding the super-process hold true for each sub-process, which is why we only include sub-processes in the process network. Since the decomposition and specialisation relations from the BPA are more of a structural nature, we do not consider them explicitly in the process network (Figure 1). Use relations among the processes from the BPA are directly transferred to the process network. Each use relation is modelled in terms of a directed edge originating from the using process pointing to the used process. Because a process may use another process several times within a single instance, each use relation is assigned a weight that represents the number of instances a process is used by the other one. We refer to this weight as the number of use instances (NUI). Due to their asynchronous communication property, the trigger relations

from the BPA need not be directly transferred to the process network. Instead, all ingoing trigger relations of a distinct process are mapped to a self-directed relation of that process in the process network. The self-directed relation is assigned a weight that represents the number of all instances where the process is executed without using any other processes, i.e., where the process runs without its output being relevant for any other process instance, also including the number of related triggered instances. We refer to this weight as the number of stand-alone instances ($NSAI$). As one process may use different processes several times in the same instance, the weight of the self-directed edge does not necessarily equal the difference between the number of all instances and the weights of all outgoing use relations.

3.2 Rationality Postulates

To ensure that process prioritization decisions based on the PPR algorithm are rational, we define rationality postulates that the algorithm must not violate. Each rationality postulate is a concrete prioritization rule derived from the PageRank characteristics and the high-level requirements from above.

The first rationality postulate takes on the process perspective. In line with high-level requirement (R.1), process prioritization decisions must account for the individual performance of the processes under investigation. A process that ceteris paribus performs worse than another process must be ranked higher in a need for improvement ranking. Figuratively speaking, if two processes have the same interconnectedness, i.e., the same relations to the same processes with the same weights and their self-directed relations have the same weights, but one process performs worse, the process with the worse performance must be ranked higher. With the performance of a single process from the process network being reflected by the $SANII$, we postulate:

(P.I) For any two processes from the process network one of which, ceteris paribus, has a higher $SANII$, the network-adjusted need for improvement of this process must exceed the network-adjusted need for improvement of the process with the lower $SANII$.

In line with high-level requirement (R.2), the relations among the processes from the process network must be considered when prioritizing processes. If a process uses another process from the process network, the used process must be ranked higher because it is responsible for its own output and that of the using process. As a result, the using process also benefits from an improvement of the used process. In contrast, if a process depends on the output of another process, its improvement does not affect the used process. Therefore, it is rational that the using process loses an amount of its importance considering the number and intensity of use relations to other processes within the process network. We postulate:

(P.II) For any two processes from the process network one of which, ceteris paribus, …

a) …is used by an additional process or has a higher NUI for at least one of the ingoing use relations, the network-adjusted need for improvement of this process must exceed the network-adjusted need for improvement of the other process.

b) …uses an additional process or has a higher NUI for at least one of the outgoing use relations, the network-adjusted need for improvement of this process must be smaller than the network-adjusted need for improvement of the other process.

If the $SANII$ of two processes are equal, rationality postulate (P.II) assures that the PPR algorithm considers the interconnectedness of the processes from the process network. The more frequently a distinct process is used by other processes, the higher is its ranking because more processes depend on the output of this process. Postulate (P.II) also holds true for transitive use relations as the effects of improving a used process cascades to each directly and transitively using process. A simple example is an improvement project that decreases the cycle time of a process and the stand-alone need for improvement of this process. The reduced time of the improved process decreases the time a using process has to wait for the output of the improved process, which in turn most certainly positively affects the cycle time of any process that uses this intermediate process. Therefore, we postulate:

(P.III) For any two processes from the process network, which are both used by other (different) processes, the network-adjusted need for improvement of the process that is used by the process

with the higher network-adjusted need for improvement must ceteris paribus exceed the network-adjusted need for improvement of the other process.

3.3 Adjustments to the Google PageRank

The high-level requirements introduced above regarding process prioritization decisions set the scope of the PPR algorithm. Therefore, the algorithm must integrate the stand-alone need for improvement of the processes under consideration with their interconnectedness from the process network. The Google PageRank seemed to be applicable to this problem as it integrates node weights and edge weights into a single index. Before it can be applied to process networks, the Google PageRank must be adjusted.

In section 2.1, we introduced the weighted PageRank algorithm, which can deal with individual weights of the edges between any two nodes in the network. Another extension of the PageRank enables using individual node weights (Brin and Page, 1998). The process network introduced in section 3.1 contains individual parameters for the processes as nodes as well as for the use and self-directed relations as directed edges. To take all parameters of the process network into account, we base the PPR algorithm on the most sophisticated version of the PageRank.

As described in the rationality postulates, a process should receive the more weight, the more it is used by other processes. Thus, the NUI and the $NSAI$ must be included in the algorithm. The first parameter we adjust is the edge weight in the PageRank formula w_{ji} to include the NUI as well as the $NSAI$. As previously described, the weight w_{ji} is used to control the relative importance of edges in the network. In line with the random surfer concept, it determines the relative probability for using a distinct outgoing edge of a distinct node in the event that the random surfer uses the network structure. Consequently, if an edge has a higher weight w_{ji}, more weight is transferred via that edge than via an edge with a lower w_{ji} coming from the same node. In our process network, the weight of a relation can represent the amount of use instances if the relation points from one process to another process. Otherwise, in case of a self-directed relation, the weights represent the amount of instances where the process does not use any other process. Using the weight of the use and self-directed relation as w_{ji} in the PageRank formula ensures two things: First, if a process uses two other processes, but one of them more often than the other, it transfers more weight to the process it uses more often since the weight of the use relation is higher. Second, the process does not transfer weight at times when it is executed without using other processes. Since the weight of the self-directed relation represents the number of instances where a process is executed without using another process and the relation points to the process from which it originated, no weight is transferred to another process. Figuratively speaking, if the random surfer chose the self-directed relation while traveling through the process network, he would end up at the same process where he started. Therefore, he does not take any weight to another process in case of choosing the self-directed relation.

Up to this point, a process transfers weight to other processes only according to the use relations. This circumstance implies that processes, which are used by the same process equally often, receive the same weight. As described in our rationality postulates above, the positive effect of improving a distinct used process on a distinct using process also depends on how high the stand-alone need for improvement of the used process was before. This is based on the following idea: Consider process i uses another process j. The higher the $SANII$ of process j, the higher the effect on process i and, therefore, the higher the network-adjusted need for improvement of process j. For example, if process i uses process j and the cycle time is the only indicator condensed in the $SANII$, the network-adjusted need for improvement of process j rises with a rising cycle time of process j, because i has to wait for j to finish. Hence, the higher the $SANII$ of the used process j, the more important it is for process i that process j is improved first. To be improved first, process j needs to rise in the ranking. Since this is in the interest of process i, it should consequently transfer the more weight to process j, the higher the $SANII$ of process j. Therefore, for the calculation of w_{ji}, the $SANII$ of process j must be included.

To integrate both effects just described into the weight w_{ji}, we multiply the NUI and the $NSAI$ with the $SANII$ of the node a relation points to. We refer to the $SANII$ of a process i as $SANII_i$ and to the NUI of a relation from process j to process i as NUI_{ji}. For better legibility, we refer to the $NSAI$ of a process i as NUI_{ji} with $i = j$. These adjustments result in Formula (4).

$$PPR(i) = \frac{1}{n} \cdot (1 - d) + d \cdot \sum_{j \in I_i} PPR(j) \cdot \frac{NUI_{ji} \cdot SANII_i}{\sum_{k \in O_j} NUI_{jk} \cdot SANII_k} \quad (4)$$

The second adjustment addresses the teleportation factor. As previously stated, this factor assigns each node an initial teleportation probability according to the random surfer model. It is equal for all nodes in the original model, but the extended model allows individual node weights. If one used the original form of the PageRank formula, where each node gets assigned the same node weight, isolated nodes without any ingoing or outgoing edges from or to other nodes end up being ranked equally. Moreover, it significantly influences the amount of weight that can be transferred away from the node (remember the recursiveness of the PageRank algorithm). To overcome this issue, we use the relative $SANII$ of a process as individual node weight. To do so, we scale the $SANII$ of a distinct process by the sum of the $SANII$ of all processes in the network to meet the requirements of the PageRank algorithm. This way, isolated processes get ranked according to their $SANII$ values and processes with a high $SANII$ value can transfer more weight to other processes. Integrating the relative $SANII$ as individual node weight into Formula (4) results in the final PPR algorithm, which is shown in Formula (5).

$$PPR(i) = \frac{SANII_i}{\sum_{j=1}^{n} SANII_j} \cdot (1 - d) + d \cdot \sum_{j \in I_i} PPR(j) \cdot \frac{NUI_{ji} \cdot SANII_i}{\sum_{k \in O_j} NUI_{jk} \cdot SANII_k} \quad (5)$$

Note that in addition to the adjustments to the formula, one also has to choose an appropriate value for the parameter d from the interval $[0; 1]$. As previously stated, d is set to 0.85 when ranking web pages (Langville and Meyer, 2004). The interpretation in the random surfer model is that the surfer uses a link from the current web page to get to the next web page with a probability of 0.85 as opposed to the case in which he teleports to a random web page within the network with a probability of 0.15. In case of the PPR, the parameter d balances the effects of a process' $SANII$ and the network structure on the ranking. Thus, d must be chosen carefully. If d is set to 0, the process network structure is not taken into account at all and the processes are ordered according to their $SANII$ values. If d is chosen very high, the network structure is considered to a great extent compared to the $SANII$. This would imply that the interconnectedness of a process has a much larger influence on its performance as the stand-alone criteria. To better understand the effect of a concrete d value, we analyse this parameter in detail in section 4.

4 Demonstration

For the demonstration, we implemented a software prototype[1] that can handle arbitrary process networks. We then applied the prototype to five archetypical cases and interpreted the PPR results for each case. Finally, we conducted a cross-case analysis to highlight differences among the single cases and to discuss the PPR algorithm against the high-level requirements and rationality postulates from above.

4.1 Single-Case Analysis

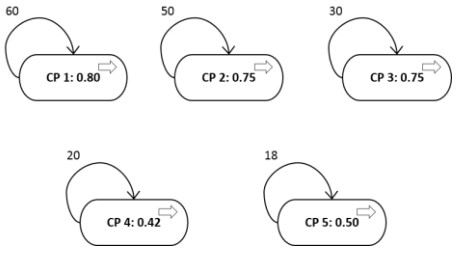
In the single case analysis, we apply the PPR to five cases each of which covers a distinct process network archetype. When choosing these cases, we had to consider four parameters, i.e., the stand-alone need for improvement as well as the ingoing, outgoing, and self-directed use relations of each process. The cases below cover changes in all these parameters. We deliberately constructed the cases presented here as small as possible to make the results more comprehensible. However, the prototype can also

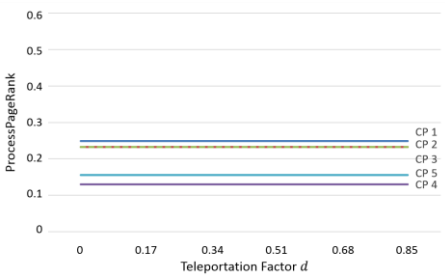
[1] The prototype is available from the authors upon request.

handle very large process networks. We simulated cases with up to 100,000 processes and up to 100,000 use relations per process.

Each case starts by briefly describing an exemplary situation where the case may occur in the real world. For illustrative purposes, we distinguish core processes (CP) and support processes (SP). We investigate how the PPR results change when the weighting between the $SANII$ and the process network is changed. We therefore analyse the PPR results subject to different d values from the interval [0.0000; 0.8500]. The PPR results are then interpreted for 0.3750 as an exemplary d value. This value appeared appropriate, as it assigns more weight to the stand-alone need for improvement, while still considering interconnectedness. Identifying a generally valid d is not the objective of this paper (see section 4.2 for a detailed discussion). For each case, we provide a table that shows the process network, a diagram of the PPR results as a function of d as well as the PPR results and the robustness interval for d = 0.3750. The robustness interval is the asymmetric interval around a chosen d value in which ranking not change.

4.1.1 Isolated Core Processes





Ranking for d=0.3750:

1.	CP 1	0.2485
2.	CP 2	0.2329
2.	CP 3	0.2329
4.	CP 5	0.1553
5.	CP 4	0.1304

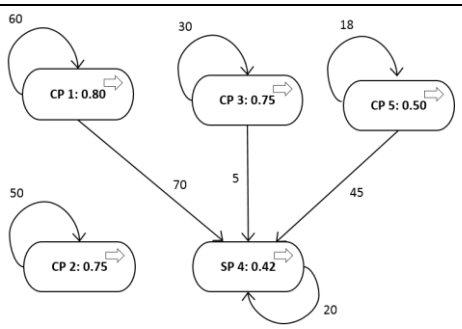
Robust interval for d:

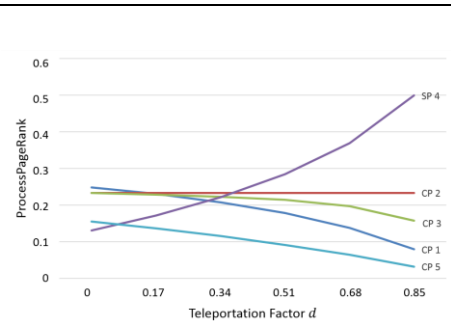
$d \in [0.0000; 0.8500]$

Table 1. Isolated Core Processes (Case 1)

In this case, we consider isolated core processes without use relations. An example would be a facility manager who performs tasks like mowing the lawn, clearing snow, or repairing something in the house. In the related process network, all processes only have a self-directed edge. As there are no use relations, the PPR results are independent of d. Consequently, the ranking only depends on the $SANII$ of the processes. The ranking therefore is perfectly robust in a trivial sense. As this case leads to the same PPR results as any other case where existing relations are ignored, it can serve as a benchmark for all following cases. We therefore use the same $SANII$ values in all cases.

4.1.2 Isolated Core Processes use one Support Process





Ranking for d=0.3750:

1.	SP 4	0.2329
2.	CP 2	0.2328
3.	CP 3	0.2216
4.	CP 1	0.2023
5.	CP 5	0.1104

Robust interval for d:

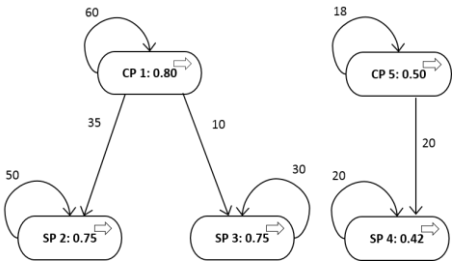
$d \in [0.3457; 0.3755]$

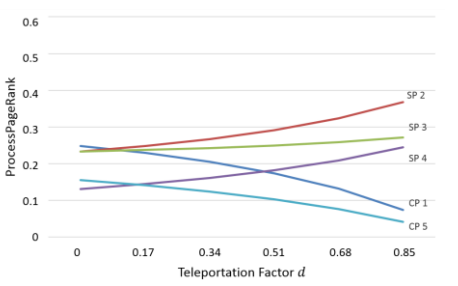
Table 2. Isolated Core Processes use one Support Process (Case 2)

In this case, we consider one support process used by many core processes. This setting can occur in a bank where core processes like opening an account or granting a loan use a support process that checks the client's credit history. The PPR value of the support process rises steeply with an increasing d as it receives weight from almost all core processes, while the PPR results of the using core processes drop. The PPR value of CP 2 is independent from d as it is not related to any other process. Comparing the

core processes CP 2 and CP 3 shows that, even though both processes have the same $SANII$, the rank of CP 3 drops below the rank of CP 2 with a rising d as it uses the support process and therefore transfers weight to it. Moreover, comparing the core processes CP 3 and CP 1 reveals that, even though CP 1 has a higher $SANII$ and both use the support process, their ranks develop differently and even switch at $d = 0.1881$. The reason is that the proportion of the instances where CP 1 uses the support process as opposed to being executed stand-alone is far greater than the corresponding proportion of CP 3. Therefore, CP 1 gives a higher proportion of its weight to the support processes than CP 3. The support process is already ranked first for a d value of 0.3750 because it receives weight of three other processes. For a d value of 0.3750, the robustness interval is [0.3457; 0.3755]. This interval is rather small and suggests that a decision-maker should take great care when choosing his teleportation factor (see section 4.2 for details).

4.1.3 Isolated Core Processes use Isolated Support Processes





Ranking for d=0.3750:

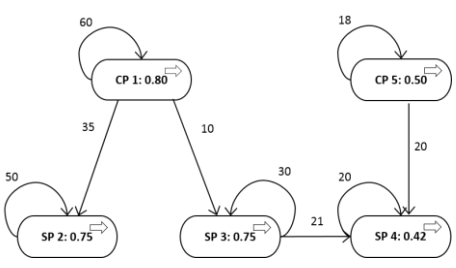
1.	SP 2	0.2713
2.	SP 3	0.2439
3.	CP 1	0.1991
4.	SP 4	0.1653
5.	CP 5	0.1204

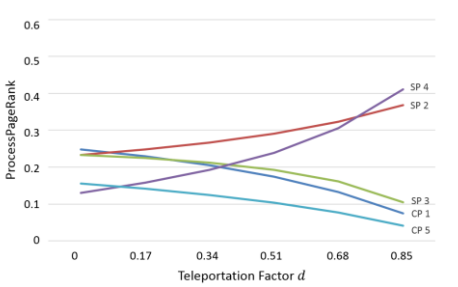
Robust interval for d:
$d \in [0.1527; 0.4849]$

Table 3. Isolated Core Processes use Isolated Support Processes (Case 3)

In this case, the process network consists of isolated sub-networks where each core process uses one or more support processes. This process network may occur in a post-merger situation where the processes of the merged companies have not been integrated yet. Both companies virtually run stand-alone, which is why their core and support processes are not connected. What is interesting in this case is the development of the PPR results of the support processes SP 2 and SP 3. Even though both processes have the same $SANII$ and an ingoing use relation from the core process CP 1, the PPR value of SP 2 rises faster than that of SP 3. The reason is that SP 2 is used much more often by CP 1 than SP 3. CP 1 thus transfers more weight to SP 2 than to SP 3. This case also illustrates the ability of the PPR algorithm to rank processes even if they are located in isolated sub-networks (enabled by the teleportation actions of the random surfer). For d = 0.3750, support process SP 2 is ranked first as it has a fairly high $SANII$ and receives additional weight from CP 1. Moreover, support process SP 4 is ranked higher than core process CP 5 because the weight transferred from CP 5 to SP 4 overcompensates for the lower $SANII$ of SP 4. The ranking is robust in the interval [0.1527; 0.4849], which can be considered to be very high.

4.1.4 Isolated Core Processes use unidirectionally interacting Support Processes





Ranking for d=0.3750:

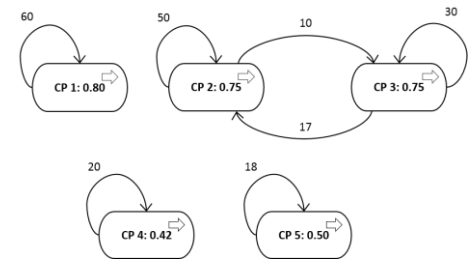
1.	SP 2	0.2713
2.	SP 3	0.2086
3.	CP 4	0.2006
4.	SP 1	0.1991
5.	CP 5	0.1240

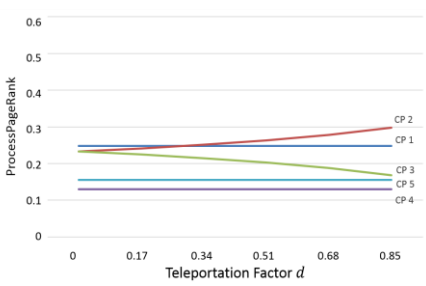
Robust interval for d:
$d \in [0.3717; 0.3979]$

Table 4. *Isolated Core Processes use unidirectionally interacting Support Processes (Case 4)*

This case is very close to the previous one. The only difference is that two of the support processes that were previously located in isolated sub-networks are now unidirectionally connected via an use relation. We still consider a company in a post-merger situation. This time, the company has already integrated one support process from one subsidiary into the BPA of the other subsidiary (i.e., a shared accounting support process). In this case, the PPR value of support process SP 4 rises very fast as it is used by two processes, i.e., SP 3 and CP 5, of which one has a pretty high $SANII$. Moreover, SP 3 is in turn used by core process CP 1. Even though SP 3 has an ingoing use relation, its PPR value drops. The reason is that SP 3 has both an ingoing and an outgoing use relation. Since the weight of the outgoing use relation is higher than that of the ingoing use relation, SP 3 transfers more weight to SP 4 than it receives from CP 1. For d = 0.3750, the support process SP 2 is ranked first even though SP 4 is used by two other processes of which one is also used by another process. However, the fact that SP 4 is used more often than SP 2 cannot overcompensate for the fact that the $SANII$ of SP 2 is almost twice as high as that of SP 4. The PPR results are volatile for small d values and robust in the interval [0.3717; 0.3979] for a chosen d of 0.3750. The reason is that the ranking of SP 4, the process with the lowest $SANII$, rises while the PPR results of the processes CP 1, SP 3, and CP 5 decrease. When d increases, SP 4 switches ranks with the other processes. After this calibration, the only change in the PPR results comes from SP 4. Since the PPR value of SP 2 grows with a raising d, it takes very high d values for the rank of SP 4 to excel that of SP 2.

4.1.5 Bidirectionally interacting Core Processes





Ranking for d=0.3750:

1.	CP 2	0.2536
2.	CP 1	0.2485
3.	CP 3	0.2122
4.	CP 5	0.1553
5.	CP 4	0.1304

Robust interval for d:

$d \in [0.2944; 0.8500]$

Table 5. Bidirectionally interacting Core Processes (Case 5)

In this case, core processes CP 2 and CP 3 use one another. All other processes operate stand-alone. The process network may represent a cross-selling situation, which could again turn up in a bank. Imagine a customer wants to open an account. The employee may suggest also opening a share deposit account. In such cases, process i, generally speaking, cannot use process j, while process j uses process i in the same instance. This implies for each process i, with an outgoing and an ingoing use relation to and from process j, that the number of instances where process i is executed without being used by process j must at least equal the amount of instances in which process i uses process j. The PPR value of CP 2 rises, while that of CP 3 decreases, even though both processes seem to have similar use relations. This circumstance is rooted in the different weights of the relations. Since the relative amount of instances in which CP 2 uses CP 3 as opposed to being executed without using another process is much smaller than that of CP 3, CP 2 transfers much less weight than CP 3. A changing d has no influence on the other processes since they have no relations with one another. For a d value of 0.3750, CP 2 is already ranked first. CP 1 is ranked second as it has a very high $SANII$ and does not give away any weight. All other processes stay on the same rank because the only other process that reacts to a rising d is CP 3. As there is only one change in the ranking, it is fairly robust. The ranking does not change beyond a d of 0.2944.

4.2 Cross-Case Analysis

In the single-case analysis, we discussed five process network archetypes to show how the PPR algorithm works in different situations. We now consider the effects of all cases and discuss them against the rationality postulates derived from the high-level requirements above.

Rationality postulate (P.I) requires processes to be ranked according to their $SANII$ if they only differ in their $SANII$. This particularly applies to isolated processes such as in the first case. There, all processes only have a self-directed relation and, independent of d, are ranked according to their descending $SANII$. Another example is the fifth case where the processes CP 1, CP 4, and CP 5 have the same PPR results independent of d. However, their rankings change because the PPR values of the other processes change. This behaviour can also be found in the second case for process CP 2. This shows that, even in case of isolated processes, the ranking must consider the interconnectedness of all processes.

As stated in rationality postulate (P.II), the interconnectedness of a process is decisive for its ranking. Regarding ingoing use relations (P.II.a) the positive effect on the network-adjusted need for improvement becomes particularly apparent by comparing process SP 4 in the second and third case. In the second case, SP 4 has a higher NUI for the ingoing use relation it shares with SP 4 from the third case, and it has more ingoing use relations. As a result, the network-adjusted need for improvement rises much faster in the second case than in the third case. This effect can also be seen for processes SP 2 and SP 3 in the third case. Even though SP 2 and SP 3 have the same $SANII$ and have one ingoing use relation coming from the same process, the PPR value of SP 2 rises faster. This behaviour is justified by the higher NUI for the ingoing use relation of SP 2. As stated in rationality postulate (P.II.b), outgoing use relations negatively affect the network-adjusted need for improvement. Regarding the second case, one can see that the PPR value of process CP 3 stays constant while that of process CP 2 drops, even though both processes have the same $SANII$. The reason is that CP 3 has an outgoing use relation while CP 2 is isolated. The negative effect of outgoing relations is even stronger for processes CP 1 and CP 3. Even though CP 1 has the higher $SANII$, its network-adjusted need for improvement is lower for d values greater than 0.5740 since it transfers more weight to SP 4. In the fourth case, the support process SP 3 brings together the effects of rationality postulate (P.II.a) and (P.II.b), having both an ingoing and an outgoing use relation. As the weight given to SP 4 through the outgoing use relation overcompensates for the weight received through the ingoing use relation from CP 1, the PPR value of SP 3 drops with an increasing d. Regarding rationality postulate (P.II), decision-makers must be aware that not only the relations among processes must be carefully modelled, but also the weights of these relations as they can heavily influence the PPR results and thus the process prioritization decisions.

Rationality postulate (P.III) states that if a process uses another process, the transferred weight does not only depend on the stand-alone need for improvement of the using process but on the network-adjusted need for improvement. This effect is particularly evident in the fourth case where the PPR value of process SP 4 rises much faster than in the third case. The reason is that SP 4 has an additional ingoing use relation from SP 3. Whereas, in the fourth case, the PPR value of process SP 3 drops, it rises in the third case due to the ingoing use relation from CP 1. This shows that the fast rise of SP 4's PPR value in the fourth case also depends on the use relation from CP 1 to SP 3. The importance of the network-adjusted need for improvement of a process for the PPR results of related processes shows that not only direct, but also transitive relations are important. Another example for this behaviour are processes CP 2 and CP 3 in the fifth case. Even though both processes have the same $SANII$, the PPR value of CP 2 rises, while that of CP 3 drops. As a result, decision-makers must not prioritize processes based only on parts of a BPA, as such decisions are usually biased.

As seen in the single-case analysis, the interconnectedness of processes heavily affects process prioritization decisions. In the preceding cross-case analysis, we discussed that these effects may largely differ depending on the characteristics of the interconnectedness without violating the rationality postulates. As an additional factor, we evaluate the parameter d whose choice is particularly important in two situations. First, if a process that features both a low $SANII$ and either very many ingoing use relations or

at least one ingoing use relation with a high NUI (such as process SP 4 in the second and forth case), the PPR value of that process rises very steeply for a rising d and therefore causes many changes in the ranking. Second, if there is a process that features a high $SANII$ and either very many outgoing use relations or at least one outgoing use relation with a high NUI (such as process CP 1 in the second, third, and fourth case), the PPR value of this process drops very steeply for rising values for d and therefore causes many changes in the ranking. In sum, if the process network contains at least one such process the previously defined robustness interval for d will most likely be rather small, implying that the ranking might change significantly for small changes in d. Therefore, when the results show a small robustness interval for the chosen d value, decision-makers are advised to invest in identifying a more robust d value that still balances the stand-alone need for improvement and the effect of the process network in an appropriate manner. The diagrams included in the tables above assist in identifying such d values. Note that, as already mentioned above, identifying a generally valid d is not the objective of this paper. However, applying the PPR algorithm to a process network helps identify major problems rather easily. The results also show which processes should be improved to leverage the effect on other processes. These processes can then undergo an in-depth analysis using methods with a single-process perspective.

5 Conclusion

In this paper, we investigated the question how processes can be prioritized considering both their individual need for improvement and interconnectedness. Building on the seminal work of Brin and Page (1998), we proposed the PPR algorithm that ranks processes according to their network-adjusted need for improvement. The PPR algorithm requires a process network and some individual performance indicators as inputs. The process network can be derived from a business process architecture (BPA) while dealing with common relation types, i.e., trigger, use, specialisation, and decomposition. The performance indicators include the stand-alone need for improvement, the number of instances where the process is executed without using any other process, and the number of instances where the process uses other processes. On this foundation, we derived rationality postulates for process prioritization decisions and adapted the original PageRank algorithm accordingly. For demonstration purposes, we implemented a software prototype and applied the PPR algorithm to five process network archetypes. We showed that process prioritization decisions require the processes' stand-alone need for improvement, their interconnectedness, and the intensity of the relations among one another to be considered.

The PPR algorithm is beset with limitations that should be addressed in future research. First, we assumed that the stand-alone need for improvement index is a single performance indicator, neglecting that process performance is a multi-dimensional construct. Future research should analyse how to build a stand-alone need for improvement index that reflects multiple dimensions of process performance. The index should also account for economic benefits to be more helpful for practitioners (Buhl et al. 2011). Second, the PPR algorithm, as developed so far, only focuses on the need for improvement and blinds out the effects of improvement projects. Such projects, however, may change the ranking. Besides the effects on single processes, it would be interesting to analyse how strongly improvement projects impact other processes and cascade through the process network. Third, the PPR algorithm would benefit from considering an economic perspective to process improvement. In real-world settings, improvement projects typically are differently expensive and have different effects on the processes' need for improvement. Hence, we encourage future research to investigate how an economic perspective can be integrated. Fourth, in line with the analytical nature of this paper, we illustrated the properties of the PPR algorithm by means of five process network archetypes and a cross-case analysis. Nevertheless, it would further benefit from real-world case studies.

Acknowledgements

This research was (in part) carried out in the context of the Project Group Business and Information Systems Engineering of the Fraunhofer Institute for Applied Information Technology FIT.

References

Bandara, W., A. Guillemain and P. Coogans (2015). ”Prioritizing Process Improvement: an Example from the Australian Financial Services Sector.” In: *Handbook on Business Process Management 2*. 2nd Edition. Heidelberg: Springer, 289–310.

Bolsinger, M. (2014). "Bringing value-based business process management to the operational process level." *Information Systems and e-Business Management* (ahead-of-print).

Brin, S. and L. Page (1998). "The anatomy of a large-scale hypertextual Web search engine." *Computer Networks and ISDN Systems* 30 (1-7), 107-117.

Buhl, H. U., M. Röglinger, S. Stöckl and K. S. Braunwarth (2011). ”Value Orientation in Process Management.” *Business & Information Systems Engineering* 3 (3), 163-172.

Burlton, R. (2015). Delivering business strategy through process management. In: *Handbook on Business Process Management 2*. 2nd Edition. Heidelberg: Springer 45–78.

Chen, Y. and X. Chen (2011). "An evolutionary PageRank approach for journal ranking with expert judgements." *Journal of Information Science* 37 (3), 254-272.

Dijkman, R., I. Vanderfeesten and H. A. Reijers (2014). "Business process architectures: overview, comparison and framework." *Enterprise Information Systems* (ahead-of-print), 1-30.

Dumas, M., M. La Rosa, J. Mendling and H. A. Reijers (2013). *Fundamentals of Business Process Management.* Heidelberg: Springer.

Dyer, J. S. (2005). "Maut – Multiattribute Utility Theory." *Multiple Criteria Decision Analysis: State of the Art Surveys.* New York: Springer.

Freeman, L. C. (1977). "A set of measures of centrality based on betweenness." *Sociometry* 40 (1), 35-41.

Hanneman, R. A. and M. Riddle (2005). *Introduction to Social Network Methods.*URL: http://faculty.ucr.edu/~hanneman/nettext/ (visited on 11/26/2014).

Harmon, P. (2010). *Business Process Change.* 2nd Edition. Burlington: Morgan Kaufmann.

Leyer, M., D. Heckl and J. Moormann (2015). "Process performance measurement." In: *Handbook on Business Process Management 2*. 2nd Edition. Heidelberg: Springer 227–242.

Heidemann, J., M. Klier and F. Probst (2010). "Identifying Key Users in Online Social Networks: A PageRank Based Approach." In: *Proceedings of the 31th International Conference on Information Systems*. Saint Louis: USA.

Hwang, C. and K. Yoon (1981). *Multiple Attribute Decision Making.* Heidelberg: Springer.

Kohlbacher, M. and H. A. Reijers (2013). "The effects of process-oriented organizational design on firm performance." *Business Process Management Journal* 19 (2), 245-262.

La Rosa, M., H. A. Reijers, W. M. P. van der Aalst, et al. (2011). "APROMORE: An advanced process model repository." *Expert Systems with Applications* 38 (6), 7029-7040.

Langville, A. N. and C. D. Meyer (2011). *Google's PageRank and beyond: The science of search engine rankings.* Princeton University Press.

Langville, A. N. and C. D. Meyer (2004). "Deeper inside pagerank." *Internet Mathematics* 1 (3), 335-380.

Lehnert, M., A. Linhart and M. Röglinger (2014). "Chopping Down Trees vs. Sharpening the Axe–Balancing the Development of BPM Capabilities with Process Improvement." *Business Process Management*, Springer International Publishing, 151-167.

Levina, O. and R. Hillmann (2012). "Network-based business process analysis." In: *System Science (HICSS), 45th Hawaii International Conference on. IEEE*, 4356-4365.

Limam Mansar, S., H. A. Reijers and F. Ounnar (2009). "Development of a decision-making strategy to improve the efficiency of BPR." *Expert Systems with Applications* 36 (2), 3248-3262.

Malinova, M., H. Leopold and J. Mendling (2014). "A Meta-Model for Process Map Design." In: *CAISE Forum 2014*. Thessaloniki: Greece.

Malinova, M., H. Leopold and J. Mendling (2013). "An Empirical Investigation on the Design of Process Architectures." In: *Proceedings of the 11th International Conference on Wirtschaftsinformatik.* Leipzig: Germany.

Malinova, M. and J. Mendling (2013). "The Effect Of Process Map Design Quality On Process Management Success." In: *Proceedings of the 21st European Conference on Information Systems (ECIS).* Utrecht: Netherlands.

Malone, T. W. and K. Crowston (1994). "The interdisciplinary study of coordination." *ACM Computing Surveys (CSUR)* 26 (1), 87-119.

Mihalcea, R., P. Tarau and E. Figa (2004). "PageRank on semantic networks, with application to word sense disambiguation." In: *Proceedings of the Proceedings of the 20th international conference on Computational Linguistics*, 1126.

Newman, M. (2003). "Mixing patterns in networks." *Physical Review E* 67 (2), 26126.

Ohlsson, J., S. Han, P. Johannesson, F. Carpenhall and L. Rusu (2014). "Prioritizing Business Processes Improvement Initiatives: The Seco Tools Case." In: *Advanced Information Systems Engineering.* LNCS 8484, 256-270.

Özgür, A., T. Vu, G. Erkan and D. Radev (2008). "Identifying Gene-Disease Associations Using Centrality on a Literature Mined Gene-Interaction Network." *Bioinformatics* 24(13), 277-285.

Probst, F., L. Grosswiele and R. Pfleger (2013). "Who will lead and who will follow: Identifying Influential Users in Online Social Networks." *Business & Information Systems Engineering* 3 (3), 179-193.

Reijers, H. A. and S. Liman Mansar (2005). "Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics." *Omega* 33 (4), 283-306.

Setzer, T., Bhattacharya, K., and Ludwig, H. (2010). "Change scheduling based on business impact analysis of change-related risk". In: *Network and Service Management, IEEE Transactions on*, 7(1), 58-71.

Sidorova, A. and O. Isik (2010). "Business process research: a cross-disciplinary review." *Business Process Management Journal* 16 (4), 566-597.

Simon, D. and K. Fischbach (2013). "IT Landscape Management Using Network Analysis." *Enterprise Information Systems of the Future.* Heidelberg: Springer, 18-34.

Thawesaengskulthai, N. and J. D. T. Tannock (2008). "Pay-off selection criteria for quality and improvement initiatives". *International Journal of Quality & Reliability Management* 25 (4), 366-382.

van der Aalst, W. M. P. (2013). "Business Process Management: A Comprehensive Survey." *ISRN Software Engineering*, vol. 2013 (Article ID 507984).

vom Brocke, J., J. Becker, A. Maria Braccini et al. (2011). "Current and Future Issues in BPM Research: A European Perspective from the ERCIS Meeting 2010." *Communications of the Association for Information Systems* 28 (1), 25.

Wang, Z., A. Scaglione and R. J. Thomas (2010). "Electrical centrality measures for electric power grid vulnerability analysis." In: *49th IEEE Conference on Decision and Control*, 5792-5797.

Zellner, G. (2011). "A structured evaluation of business process improvement approaches." *Business Process Management Journal* 17 (2), 203-237.