Determining the Link and Rate Popularity of Enterprise Process Information*

Bernd Michelberger[1], Bela Mutschler[1], Markus Hipp[2], and Manfred Reichert[3]

[1] University of Applied Sciences Ravensburg-Weingarten, Germany
{bernd.michelberger,bela.mutschler}@hs-weingarten.de
[2] Group Research & Advanced Engineering, Daimler AG, Germany
markus.hipp@daimler.com
[3] Institute of Databases and Information Systems, University of Ulm, Germany
manfred.reichert@uni-ulm.de

Abstract. Today's knowledge workers are confronted with a high load of heterogeneous information making it difficult for them to identify the information relevant for performing their tasks. Particularly challenging is thereby the alignment of process-related information (*process information* for short), such as e-mails, office files, forms, checklists, guidelines, and best practices, with business processes. In previous work, we introduced the concept of *process-oriented information logistics* (POIL) to bridge this gap. POIL allows for the process-oriented and context-aware delivery of relevant process information to knowledge workers. So far, we have introduced concepts to integrate business processes with process information. A remaining challenge is to identify the process information relevant for a given process context. This paper tackles this challenge and extends our POIL approach with techniques and algorithms for identifying relevant process information. More specifically, we introduce two algorithms for determining the relevance of process information based on their link and rate popularity. We use a scenario from the automotive domain to demonstrate and validate the applicability of our approach.

Key words: process-oriented information logistics, process information relevance, link popularity algorithm, rate popularity algorithm

1 Introduction

Today's knowledge workers are confronted with a continuously increasing amount of heterogeneous information in their day-to-day operations [1]. Examples include e-mails, office files, process descriptions, forms, checklists, guidelines, working instructions, and best practices. This information may be accessed, for example, through shared drives, databases, portals, or enterprise information systems. Particularly, knowledge workers are not only interested in quickly

* This paper was done in the niPRO research project. The project is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 17102X10. More information can be found at http://www.nipro-project.org.

accessing this information, but also require comprehensive and aggregated information when performing a certain task [2, 3]. Identifying information required in this context, however, is much more time-consuming and complex than just managing information [4]. Problems frequently encountered include, for example, incomplete, incorrect, unpunctual, or outdated information [5].

A particular challenge is to align process-related information (*process information* for short) with business processes. In practice, process information is not only stored in large, distributed, and heterogeneous data sources, but usually managed separately from business processes. Shared drives, databases, portals, and enterprise information systems are used to manage process information. In turn, business processes are managed using process management technology. Hence, in practice, process information and business processes are often manually linked, i.e., process information is hard-wired to business processes, e.g., in Intranet portals linking specific process information with process tasks. However, this approach often fails due to high maintenance efforts and missing support for the specific requirements of individual process participants.

To tackle this challenge, in previous work, we introduced the concept of *process-oriented information logistics* (POIL) as new paradigm for delivering the right process information, in the right format and quality, at the right place, at the right point in time, and to the right people [6, 7]. Specifically, POIL shall enable a process-oriented and context-aware (i.e., personalized) delivery of relevant process information to knowledge workers. Goal is to no longer manually hard-wire process information to business processes, but to identify and deliver relevant process information to knowledge workers automatically.

This paper extends our POIL approach and introduces techniques for determining the relevance of process information based on two algorithms. The first one determines the *link popularity* of process information based on their relationships. The second one determines the *rate popularity* of process information based on user ratings. Section 2 sketches POIL. Section 3 provides formal definitions required for describing the algorithms. Section 4 introduces the algorithms in detail. Section 5 presents a scenario and a survey verifying the applicability of our algorithms. Section 6 discusses related work and Section 7 concludes the paper with a summary and outlook.

2 Process-oriented Information Logistics

Traditional *information logistics* (IL) approaches deal with the question of how to deliver information to knowledge workers as effectively and efficiently as possible [8]. For this purpose, basic principles from the fields of material logistics and lean management are applied. Examples include just-in-time delivery [9] and satisfaction of customer needs [10]. Particularly, IL aims at delivering that information to knowledge workers fitting their demands best. Thus, *information awareness* (e.g., awareness of information quality and flows) and, to a smaller extent, *context awareness* (e.g., awareness of the user context for the delivery of personalized information) adopt a key role in IL [11] (cf. Fig. 1).

Although IL is independent from the use of information and communication technology (ICT), the latter has been intensively used as IL enabler for several years. Consider ICT solutions in areas like business intelligence, management information systems, and enterprise content management. However, these solutions also suffer from shortcomings like limited applicability (e.g., only applicable within enterprises and not between them) [12], missing operational functionality (e.g., only the management level is addressed) [13], and lack of *process awareness* (e.g., delivering information without considering the current process context).

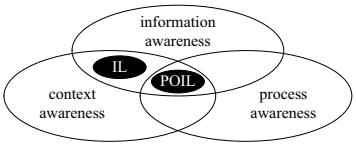


Fig. 1. Problem dimensions of IL and POIL.

Missing process awareness in contemporary IL solutions has guided our development of *process-oriented information logistics* (POIL) [6]. POIL aligns process information with business processes, both at the process schema and process instance level [14]. It enables a process-oriented and context-aware delivery of process information to knowledge workers. Thereby, POIL not only combines *information* and *context awareness*, but takes *process awareness* into account as well (cf. Fig. 1), i.e., awareness of process schemas and corresponding instances. Note that POIL focuses on knowledge-intensive business processes involving large amounts of process information, expertise, user interaction, creativity, and decision-making [15] such as the engineering of cars or the medical treatment of patients in hospitals.

The core component of any POIL is a *semantic information network* (SIN), which comprises unified *information objects* (e.g., e-mails, guidelines, best practices), *process objects* (e.g., tasks, pools, lanes, data objects, events, task instances), and the *relationships* (e.g., "is similar to", "has same author as") between them. In particular, a SIN allows identifying objects linked to each other in the one or other way, e.g., information objects addressing the same topic or needed when performing a particular process task. In order to create a SIN, business processes and process information are transformed into unified process and information objects (cf. Fig. 2a-b). In the second step, these objects are semantically analyzed to detect their relationships (cf. Fig. 2c) [7, 16].

More precisely, the SIN is created in six consecutive phases (see [6] for details). Our main idea is to split up business processes into their constituent *process objects* and to integrate the latter with *information objects* in the SIN. For creating and maintaining a SIN, we apply algorithms provided by a semantic middleware we use to implement the SIN (see [6] for details). These algorithms, however, do not allow identifying relevant, i.e., currently needed, information objects within a SIN. What we additionally need are further algorithms. This is

indispensable in order to reach the aforementioned goals of POIL, i.e., to provide knowledge workers with the right process information.

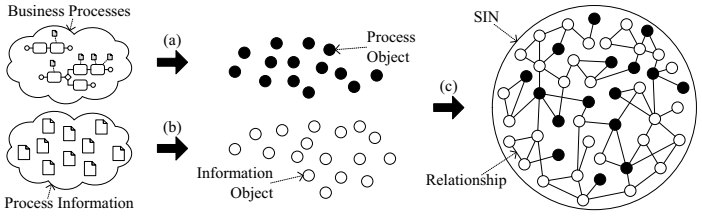


Fig. 2. Schematic and simplified creation of a SIN.

Generally, the SIN's relationships may exist between information objects (e.g., a guideline similar to another one), between process objects (e.g., an event triggering a subprocess), and between information and process objects (e.g., an instruction required for executing a task) (cf. Fig. 3a-c). Further, a relationship can be either *explicit* (i.e., hard-wired) or *implicit* (i.e., not hard-wired). Explicit relationships are, for example, modeled data flows in a process schema. Implicit relationships, in turn, are automatically identified by a variety of algorithms and link, for example, objects addressing the same topic or objects used in the same working context [6]. Moreover, relationships are labeled (e.g., "is a template") and weighted. A weight is expressed in terms of a number ranging from 0 to 1 (with 1 indicating the strongest possible relationship) [16]. This allows determining why objects are interlinked and how strong their relationship is.

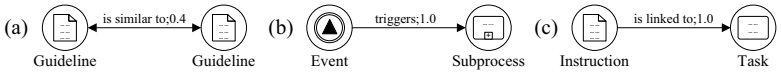


Fig. 3. Relationships between objects.

3 Preliminaries

Generally, a SIN is a labeled and weighted *directed graph*. Each directed *edge* $e = (u, v)$ represents a relationship and is associated with an ordered pair of *vertices* (u, v), which represents information and process objects; u is the source and v is the destination of e. Based on this, a SIN is formally defined as follows:

Definition 1 (SIN). *A labeled and weighted digraph is called semantic information network* $SIN = (V, E, L, W, f_l, f_w)$*, iff:*

- V *is a set of vertices representing information and process objects*

- *E is a set of edges representing relationships between objects*
- *L is a set of labels indicating relationship reasons*
- *W is a set of weights representing the relevance of relationships*
- *f_l is a labeling function with $f_l : E \to L$ assigning to each edge $e \in E(SIN)$ a label $f_l(e) \in L$*
- *f_w is a weighting function with $f_w : E \to W$ assigning to each edge $e \in E(SIN)$ a weight $f_w(e) \in W = [0, 1]$*

The membership of V and E in SIN is denoted as $V(SIN)$ and $E(SIN)$. A SIN constitutes a *finite graph*, i.e., V and E are finite sets [17]. A SIN may contain *slings* (i.e., $\exists\ e = (v, v)$, cf. Fig. 4a), *parallelism* (i.e., $\exists\ e = (u, v) \wedge f = (u, v)$, cf. Fig. 4b), and *anti-parallelism* (i.e., $\exists\ e = (u, v) \wedge f = (v, u)$, cf. Fig. 4c).

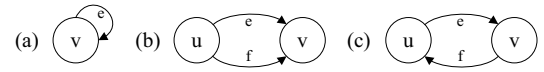


Fig. 4. Slings, parallelism and anti-parallelism of SINs.

In general, each vertex v may have several incoming and outgoing edges. The number of incoming edges of a vertex constitutes its *incoming degree*, whereas the number of outgoing edges is denoted as *outgoing degree*. The *total degree* of a vertex corresponds to the sum of its incoming and outgoing degrees. Vertices having no incoming edges are denoted as *unreferenced*. In turn, vertices without outgoing edges are called *non-referencing*. Finally, vertices being unreferenced as well as non-referencing are *isolated* [18].

Definition 2 (Degree). *The number of incoming and outgoing edges of a vertex $v \in V(SIN)$ is denoted as degree of v, where:*

- *$deg^-(v)$ is the incoming degree of a vertex $v \in V(SIN)$ which is denoted as $deg^-(v) = |E^-(v)| = |\{e = (x, y) \in E \mid y = v\}|$*
- *$deg^+(v)$ is the outgoing degree of a vertex $v \in V(SIN)$ which is denoted as $deg^+(v) = |E^+(v)| = |\{e = (x, y) \in E \mid x = v\}|$*
- *$deg(v)$ is the total degree of a vertex $v \in V(SIN)$ which is denoted as $deg(v) = deg^-(v) + deg^+(v) = |E(v)| = |E^-(v)| + |E^+(v)|$*

Vertices directly relating to a neighbored vertex are called *internal neighborhood*, whereas vertices referenced by another vertex are called *external neighborhood*. Then, the *total neighborhood* corresponds to the union of both internal and external neighborhood.

Definition 3 (Neighborhood). *Referencing and referenced vertices of a vertex $v \in V(SIN)$ are denoted as neighborhood of v, where:*

- *$\Gamma^-(v)$ is the internal neighborhood of a vertex $v \in V(SIN)$ which is denoted as $\Gamma^-(v) = V^-(v) = \{u \in V^-(v)\}$*

- $\Gamma^+(v)$ *is the external neighborhood of a vertex* $v \in V(SIN)$ *which is denoted as* $\Gamma^+(v) = V^+(v) = \{u \in V^+(v)\}$
- $\Gamma(v)$ *is the total neighborhood of a vertex* $v \in V(SIN)$ *which is denoted as* $\Gamma(v) = \Gamma^-(v) \cup \Gamma^+(v) = V(v) = \{u \in V(v)\}$

As set out in Definition 1, the function f_w assigns a weight to each edge e. This weight indicates the relevance of an edge and therewith the strength of the relationship between two vertices. However, in a SIN, there may be multiple edges between vertices with different weights. In order to determine the overall strength between two vertices, we calculate the average weight of all edges between them. The *average weight*$_\emptyset$ of a set of edges F can be calculated as follows:

$$avg_\emptyset(F) = \sum_{f \in F} \frac{f_w(f)}{|F|} \tag{1}$$

In practice, however, certain edges have to be weighted higher. As an example consider a "is similar to" relationship, which is usually more important than a "has same file extension as" relationship. Therefore, we additionally use a significance function f_s with $f_s : E \to \mathbb{N}_1$ assigning to each edge $e \in E$ a significance value $f_s(e) \in \mathbb{N}_1$. The higher a significance value is, the more important is an edge. The *average weight*$_\Delta$ of a set of edges F can be calculated as follows:

$$avg_\Delta(F) = \sum_{f \in F} \frac{f_s(f) * f_w(f)}{\sum_{g \in F} f_s(g)} \tag{2}$$

4 Determining the Relevance of Process Information

In two case studies as well as an online survey [19, 20], we already showed that knowledge workers spend considerable efforts to handle process information. One challenging task in this context is to identify relevant process information. In POIL, the SIN constitutes the basis for this task. However, additional techniques are needed to determine relevant process information, i.e., currently needed information objects in a SIN dependent on the process context (cf. Fig. 5).

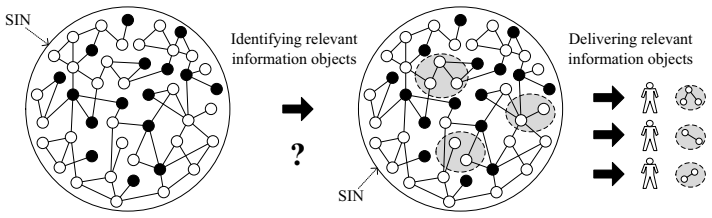


Fig. 5. Delivering relevant information objects.

In the following, we introduce two algorithms for identifying relevant information objects in a SIN. The first one determines the *link popularity* of information

objects based on the SIN's relationship structure. The second one determines the *rate popularity* of information objects based on user ratings. Note that the algorithms can be used independently, but can be combined as well.

4.1 Determining Link Popularity

In enterprises, process information is usually not explicitly linked to other process information or business processes. Therefore, it is not possible to take advantage of a rich relationship structure within an enterprise environment. Instead, process information is implicitly linked to other process information and business processes, e.g., dealing with the same topic or used in the same process context. A SIN makes such implicit relationships explicit by means of its edges. The SIN's relationship structure enables us to apply algorithms to identify strongly linked and therefore popular objects. The problem, however, is that existing link popularity algorithms are not sufficient in our context (as shown in the following). Thus, we extend them and introduce the *SIN LP algorithm*, which allows us determining the link popularity of information objects in a SIN (cf. Fig. 6).

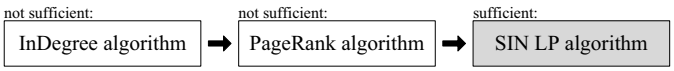


Fig. 6. Link popularity algorithms.

Basic to any link popularity algorithm is an *InDegree algorithm* [21] measuring the *link popularity* $LP(v)$ of an information object v by taking its number of incoming edges into account (cf. Formula (3)). The higher the number of incoming edges is, the greater the popularity of an information object becomes:

$$LP(v) = deg^{-}(v) \tag{3}$$

In a SIN, the InDegree is not really helpful since certain relationships might be more valuable than others. This issue, in turn, is picked up by the *PageRank algorithm* [22]: Relationships originating from information objects of high quality are considered being more valuable than relationships originating from information objects of low quality (cf. Formula (4)). Thus, the link popularity $LP(v)$ of an information object v is calculated as follows (with d corresponding to a damping factor ranging from 0 to 1):

$$LP(v) = (1-d) + d \sum_{w \in \Gamma^{-}(v)} \frac{LP(w)}{deg^{+}(w)} \tag{4}$$

However, like the InDegree, the conventional PageRank (originally designed for the web) is not applicable to a SIN since it only considers single relationships. In a SIN, there are multiple, weighted, and labeled relationships. Hence, we must extend the PageRank. First, we have to support multiple relationships:

$$LP(v) = (1-d) + d \sum_{w \in \Gamma^-(v)} |\{e = (w,v) \in E\}| * \frac{LP(w)}{deg^+(w)} \quad (5)$$

To also support weighted relationships, we further extend Formula (5) and include an average weighting function $avg_{\emptyset}$ (cf. Section 3):

$$LP(v) = (1-d) + d \sum_{w \in \Gamma^-(v)} avg_{\emptyset}(\{e = (w,v) \in E\}) * |\{e = (w,v) \in E\}| * \frac{LP(w)}{deg^+(w)} \quad (6)$$

Note that Formula (6) only deals with equally weighted relationships. To finally support differently weighted relationships, we have to extend it by the average weighting function avg_{Δ} (cf. Section 3):

$$LP(v) = (1-d) + d \sum_{w \in \Gamma^-(v)} avg_{\Delta}(\{e = (w,v) \in E\}) * |\{e = (w,v) \in E\}| * \frac{LP(w)}{deg^+(w)} \quad (7)$$

Based on Formula (7) it becomes possible to determine the link popularity of information objects in a SIN. Note that this corresponds to the solution of a system of equations. In our approach we use an approximate, iterative calculation of the link popularity, i.e., we assign an initial $LP(v) = init$ to each information object v. The link popularity $LP(v)$ is then iteratively determined for each information object v as follows (let i be the number of iterations)[1]:

Input: $SIN = (V, E, L, W, f_l, f_w)$; d; i; $init$;
Result: $LP(v)$ for each $v \in V(SIN)$;
foreach $v \in V(SIN)$ **do** $LP(v) = init$;
foreach $e \in E(SIN)$ **do** $f_s(e)$;
for $j = 1$ **to** i **do**
  foreach $v \in V(SIN)$ **do**
    $pop = 0$;
    foreach $w \in \Gamma^-(v)$ **do**
      $pop \stackrel{+}{=} avg_{\Delta}(\{e = (w,v) \in E\}) *$
        $|\{e = (w,v) \in E\}| * LP(w) \ / \ deg^+(w)$;
    end
    $LP(v) = (1-d) + d * pop$;
  end
  $j = j + 1$;
end

Algorithm 1: SIN Link Popularity Algorithm.

In summary, algorithm 1 allows determining the link popularity of information objects based on the SIN's relationship structure in an iterative way.

[1] Our implementation can be found at http://sourceforge.net/projects/linkinganalyzer/

4.2 Determining Rate Popularity

This section introduces another algorithm that allows determining the rate popularity of process information based on user ratings. In enterprises, existing IL solutions often allow users to rate the quality of process information, e.g., by means of "like buttons" or "five stars ratings". The set of ratings R can then be used to determine the *rate popularity* $RP(v)$ of an information object v. However, ranking information objects based on user ratings is a non-trivial task. Like before, we first show that existing algorithms are not sufficient in POIL and then introduce our *SIN RP algorithm*, which allows us determining the rate popularity of information objects in a SIN (cf. Fig. 7).

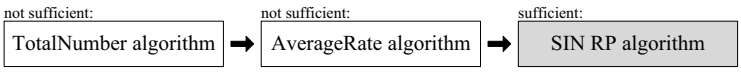


Fig. 7. Rate popularity algorithms.

An approach to determine rate popularity $R(v)$ of an information object v is to rank information objects by their *total number* of ratings $|R(v)|$:

$$RP(v) = |R(v)| \tag{8}$$

Another approach is to determine the rate popularity $RP(v)$ based on the *average user rating* using $avg(R(v))$ of an information object v:

$$RP(v) = \sum_{r \in R(v)} \frac{r}{|R(v)|} \tag{9}$$

However, applying Formulas (8) or (9) is not appropriate in a SIN. Both formulas tend to prefer older information objects available for a longer time (i.e., there was more time for users to rate for these information objects). This shortcoming is rather problematic in enterprise environments with continuously emerging information objects. Using Formula (9) results in another problem: Assume that in a "five stars rating" there is an information object with an overall weight of 4.8, which is based on hundreds of individual ratings. Additionally assume that another information object is rated by one knowledge worker with 5.0. The latter information object is then directly ranked on the first position. To avoid this, all ratings must be taken into account.

Thus, we calculate the rate popularity consistent with *Bayesian interpretation* [23]. Formula (10) allows calculating the average rating $avg(R)$ of all information objects. Formula (11) then calculates the rate popularity $RP(v)$ of a single information object v taking both the set of ratings R and the information objects' age into account. Thus, we avoid that information objects with few, but favorable ratings are ranked on the first positions:

$$avg(R) = \sum_{v \in V} \frac{|R(v)| * avg(R(v))}{|R|} \tag{10}$$

$$RP(v) = \frac{\frac{\left(\frac{|R|}{|\{v \in V | R(v) > 0\}|} * avg(R)\right) + \left(|R(v)| * avg(R(v))\right)}{\frac{|R|}{|\{v \in V | R(v) > 0\}|} + |R(v)|}}{age(v)} \quad (11)$$

Algorithm 2 shows how the rate popularity value for each information object v is calculated taking the set of available user ratings R into account[2]:

Input: $SIN = (V, E, L, W, f_l, f_w)$; R;
Result: $RP(v)$ for each $v \in V(SIN)$ where $|R(v)| > 0$;
foreach $v \in V(SIN)$ **do**
  if $|R(v)| > 0$ **then**
    $avg(R) \stackrel{+}{=} |R(v)| * avg(R(v)) \ / \ |R|$;
  end
end
foreach $v \in V(SIN)$ **do**
  if $|R(v)| > 0$ **then**
    $pop = ((|R| \ / \ |\{v \in V \mid R(v) > 0\}| *$
      $avg(R)) + (|R(v)| * avg(R(v))))$;
    $pop = pop \ / \ (|R| \ / \ |\{v \in V \mid R(v) > 0\}| +$
      $|R(v)|)$;
    $RP(v) = pop \ / \ age(v)$;
  end
end

Algorithm 2: SIN Rate Popularity Algorithm.

In summary, algorithm 2 allows determining the rate popularity of information objects based on user ratings in an easy way.

5 Validation

In order to prove that our algorithms support knowledge workers when performing knowledge-intensive tasks, we use a real-world scenario from the automotive domain (cf. Section 5.1). Specifically, we implement our algorithms (cf. Section 5.2) and then compare their outcome with results of a survey among automotive engineers who were asked to manually determine the relevance of process information related to the considered scenario (cf. Section 5.3). Doing so, we aim to show that our algorithmic results can indeed replace the costly and time-intensive human determination of relevant process information.

5.1 Real-World Scenario

Our scenario (cf. Fig. 8) deals with the review of product requirements documented in functional specifications at a large automotive manufacturer [19].

[2] Our implementation can be found at http://sourceforge.net/projects/ratinganalyzer/

Goal is to improve as well as to approve such specifications. The underlying review process is knowledge-intensive, i.e., it comprises large amounts of process information (e.g., review protocols, checklists, review templates, guidelines), user interaction (e.g., "perform review meeting", "send review comments"), and decision-making (e.g., should the document be approved or not?). Three roles are involved: (1) The *author* provides the specification to be reviewed. (2) The *review moderator* organizes the review meetings. (3) The *reviewer* finally analyzes the provided specification and documents errors, ambiguities, and uncertainties.

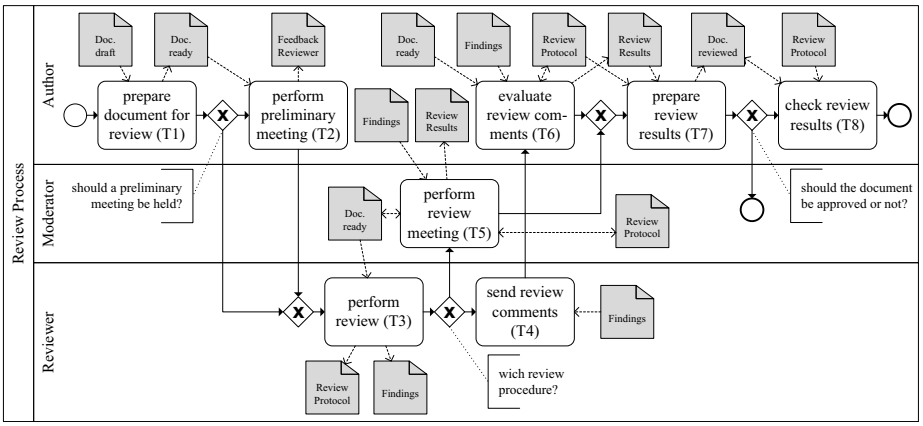


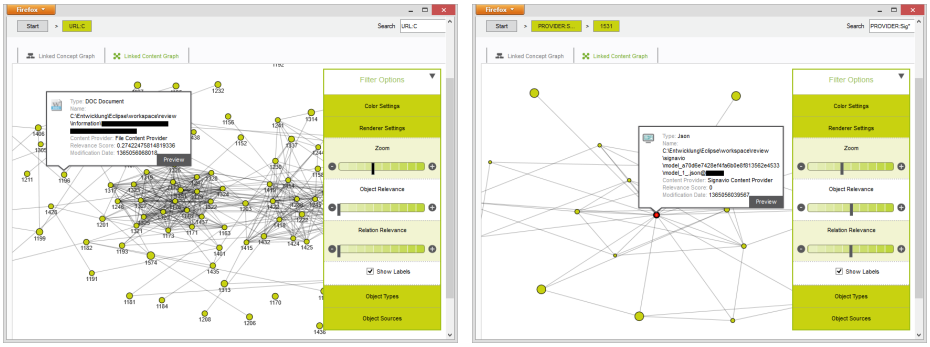
Fig. 8. Process schema of our automotive scenario (BPMN model).

The review process starts with the preparation of the document to be reviewed (task T1). This step is performed by the document's author. Based on this initial preparation, the author decides whether or not a preliminary review meeting becomes necessary (task T2). Afterwards, the document is reviewed (task T3). Based on the review's outcome, the reviewer decides whether an additional review meeting is needed (task T4) or whether it is sufficient to directly send findings and comments to the author (task T5). The latter then evaluates review results (task T6) and updates the document accordingly (task T7). If the document's overall review status is rejected, it will not be approved. In turn, if its overall review status is accepted, the author can finally approve the document (task T8). For each of these process steps, a variety of process information is needed; e.g., guidelines, templates, meeting protocols, or working instructions.

5.2 Implementation

Based on the scenario discussed we first implemented the corresponding SIN - altogether comprising one process schema modeled with Signavio Process Editor, three process instances created and managed with the Activiti BPM Platform, and about 300 documents (i.e., process information) such as reviews, review

protocols, templates, guidelines etc. For creating the SIN we use the semantic middleware iQser GIN Server as well as several Java open-source plugins we developed[3]. The implemented SIN includes 348 objects (45 process objects, 303 information objects) and 65.991 relationships (77 process object relationships, 65.319 information object relationships, and 595 cross-object-relationships) [6]. While Fig. 9a shows the entire SIN of our scenario, Fig. 9b only depicts objects (i.e., information and process objects) directly related to task T3. Note that due to privacy reasons, the document names are blacked out.



(a) Entire SIN of our scenario. (b) Task T3 and related objects.

Fig. 9. The implemented SIN.

We then additionally implemented our algorithms in a proof-of-concept prototype called iGraph[4], a web-based Java/Scala application. iGraph uses the web application framework Play, the Twitter Bootstrap framework, the JavaScript libraries Data-Driven Documents (D3) and jQuery, HyperText Markup Language (HTML) 5 templates, and Cascading Style Sheets (CSS) 3.

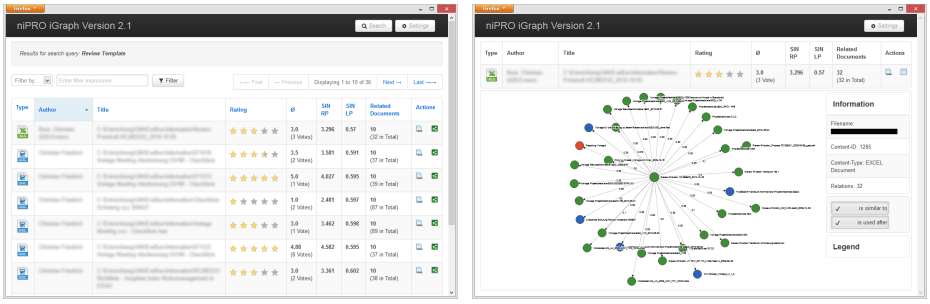
The iGraph user interface provides two views: a *table-based* and a *graph-based view*. The former lists information objects identified based on a document search query (cf. Fig. 10a). The latter illustrates the relationships of selected SIN objects (i.e., process or information objects); Fig. 10b, for example, depicts information objects linked to process task T3 of our scenario.

5.3 Empirical Validation

Using iGraph, we construct a survey (cf. Section 5.3). In this survey, automotive engineers evaluate previously calculated results of the link and rate popularity algorithms. Doing so, we aim to show that our algorithmic results can indeed replace the costly human determination of relevant process information. More

[3] These plugins are available at http://sourceforge.net/directory/?q=nipro

[4] A screencast presenting the iGraph prototype is available at http://nipro.hs-weingarten.de/screencast



(a) Table-based view. (b) Graph-based view.

Fig. 10. Different views of iGraph.

specifically, the goal is to prove the accuracy of our algorithms. Particularly, the survey was guided by two research questions: (RQ1) "How do results of the SIN LP algorithm match with user-generated evaluations on the relevance of process information?" and (RQ2) "How good is the ranking of process information based on our SIN RP algorithm compared to other ranking approaches?"

We performed the survey in late April 2013. The questionnaire comprised 18 questions. Overall, 20 automotive experts participated. Most of them work at electric/electronic engineering departments, but there were also participants from other departments. All participants were selected due to their expert knowledge regarding the considered review scenario.

RQ1 (Investigating Link Popularity). To investigate RQ1, we use iGraph to calculate two link popularity result lists (as input values we set $init = 0.45$, $i = 12$, $d = 0.5$, and double-weight "is similar to"-relationships): (a) the top eight documents according to the SIN LP algorithm for process task T1 and (b) the top eight documents according to the SIN LP algorithm for process task T3. Table 1 shows the documents the SIN LP algorithm returns for T1 and T3.

We then asked survey participants to evaluate - based on their practical experiences - the relevance of the documents returned by the SIN LP algorithm for the tasks T1 ("prepare document for review") and T3 ("perform review"). As can be seen in Table 1, the survey participants confirm the relevance for the majority of the 16 documents identified by our SIN LP algorithm. Note that we consider a document as being relevant if more than 50% of the survey participants confirm relevance. Results show that our algorithm is indeed well working, especially since the algorithm's overall accuracy can be further improved, for example, by combining it with other algorithms (e.g., the SIN RP algorithm).

RQ2 (Investigating Rate Popularity). To investigate RQ2, we first calculate a ranking of review templates applying the SIN RP algorithm (note that we use real ratings we obtained from the automotive manufacturer supporting the survey). Fig. 11 shows the calculated ranking of review templates. Additionally,

Table 1. SIN LP algorithm validation results.

Case	ID	Type	$LP(v)$	#Marked	Ratio	Is Relevant?
Task T1	1231	Review Template	0.443	12	60.0 %	■
	1210	Process Overview	0.442	20	100.0 %	■
	439	Review Template	0.441	4	20.0 %	□
	432	Specific Review	0.439	17	85.0 %	■
	811	Guideline	0.435	4	20.0 %	□
	439	Protocol	0.434	2	10.0 %	□
	578	Checklist	0.434	19	95.0 %	■
	777	Guideline	0.432	19	95.0 %	■
Task T3	1210	Process Overview	0.443	17	85.0 %	■
	879	Protocol	0.442	19	95.0 %	■
	431	Specific Review	0.441	10	50.0 %	□
	432	Specific Review	0.439	9	45.0 %	□
	741	Review Template	0.435	7	35.0 %	□
	439	Review Template	0.434	6	30.0 %	□
	578	Checklist	0.434	18	90.0 %	■
	729	Review Template	0.432	19	95.0 %	■

□ = no ■ = yes

in order to evaluate the SIN RP rating, we calculate three further rate-based rankings. More specifically, we calculate the additional rankings based on Formula (8) (a ranking based on the total number of ratings) and Formula (9) (a ranking based on the average rating). Finally, we also create a random ranking.

We then asked survey participants to evaluate - based on their practical experiences - both the plausibility and the usefulness of the four rankings. Fig. 12 shows that 16 out of 20 participants consider the ranking created with our SIN RP algorithm as the most plausible one. The ranking based on the total number of ratings is considered as the second most plausible one (three votes). The ranking based on the average rating only received one vote.

As aforementioned, we also asked the participants to evaluate the usefulness of the rankings based on a Likert Scale [24] ranging from "not at all useful" (1) to "very useful" (5)). Fig. 12 shows that 87.5% of the participants state that the ranking created with our SIN RP algorithm is "useful" or "very useful". Again, survey results show that our algorithm is indeed well working.

Conclusion. Our empirical validation confirms that most of the documents returned by our SIN LP algorithm are indeed relevant ones. Moreover, our empirical research also shows that the link popularity is a good indicator for identifying relevant process information, especially since results of the SIN LP algorithm can be further refined for specific process tasks by applying the SIN LP algorithm to only specific parts of a SIN (e.g., to a specific process task, corresponding task instances, or related information objects).

Type	Author	Title	ID	Rating	Ø	SIN RP	SIN LP	Related Documents	Actions
			389		5.0 (1 Vote)	3.502	0.401	17	
			1261		4.8 (22 Votes)	4.335	0.421	15	
			1051		4.2 (10 Votes)	3.770	0.411	10	
			422		3.7 (12 Votes)	3.541	0.418	34	
			567		3.2 (4 Votes)	3.316	0.421	21	
			1022		2.8 (15 Votes)	3.030	0.413	22	
			1023		1.6 (12 Votes)	2.421	0.396	32	
			846		1.4 (8 Votes)	2.512	0.382	8	

Fig. 11. Rating.

The results of the SIN RP algorithm are considered as useful by the survey participants. In fact, most participants state that the ranking of documents as suggested by the SIN RP algorithm is both plausible and useful. Additionally, our SIN RP algorithm avoids the problematic situation that process information with only a few good user ratings is directly ranked on the first position of a ranking. Finally note that the results of the SIN RP algorithm can be easily further improved, for example, by taking into account the expertise of knowledge workers, i.e., ratings of experienced knowledge workers might be weighted higher.

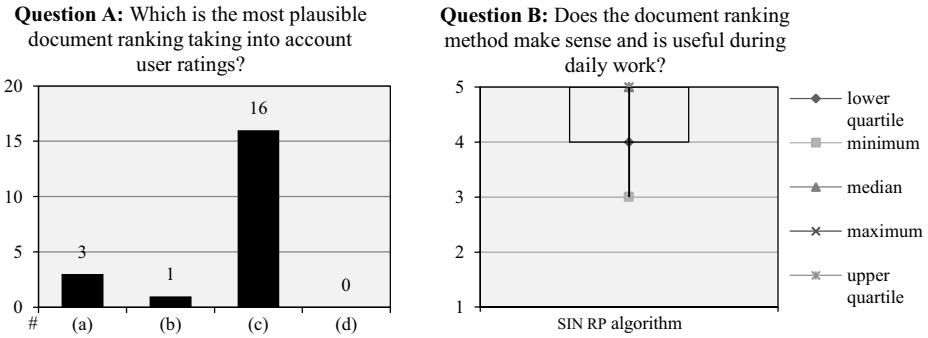


Fig. 12. SIN RP algorithm validation results.

In summary, the popularity values of our algorithms clearly help to determine the relevance of process information. However, as it is difficult to determine the

overall relevance of process information based on a single algorithm, we will combine our algorithms when further extending our POIL framework.

6 Related Work

As discussed in Section 2, various ICT solutions have been proposed to enable IL and hence to identify relevant information. As examples consider data warehousing (DWH), business intelligence (BI) solutions, decision support systems (DSS), and enterprise content management (ECM). However, these approaches suffer from several weaknesses. For example, DWH rather focuses on the creation of an integrated database [25]. Traditional BI, in turn, addresses data analytics and is usually isolated from business process execution [26]. Conventional DSS support complex business decision-making at the management level [27]. By contrast, ECM deals with the management of information across enterprises referring to related strategies, methods, and tools [28].

There exists a wide range of link popularity algorithms. Best known is the PageRank algorithm [22]. However, relationships being more valuable than others are picked up by other algorithms as well, e.g., the Hits algorithm [29] or the weighted PageRank algorithm [30]. An algorithm combining both PageRank and Hits is the Salsa algorithm [31]. Another evolution of the PageRank is the Topic-Sensitive PageRank algorithm [32], which additionally considers topics. However, all these algorithms have been originally developed for the web and cannot be directly, i.e., without modification, applied to POIL. Particularly, they do not allow dealing with the specific characteristics of a SIN.

Research done by others also influenced the development of our rating popularity algorithm. An approach to improve search results based on user ratings, for example, is presented in [33]. In [34], a study on rate popularity algorithms and their pros and cons is presented. Similar to our algorithm, a self-learning algorithm is presented in [35], which addresses both user ratings and content relevance. Notwithstanding, like the link popularity algorithms, existing rate popularity algorithms cannot be directly applied to a SIN.

7 Summary and Outlook

This paper presented two algorithms for determining the relevance of process information in POIL. The first one determines the popularity of process information based on the relationships of a SIN. The second one determines the popularity of process information based on user ratings. We applied our algorithms to a real-world scenario, i.e., validated them based on an implementation and a survey in the automotive domain.

In future, we will develop additional algorithms for determining the relevance of process information. In particular, we will focus on self-learning algorithms enabling us to take into account our POIL context framework [36].

References

1. Öhgren, A., Sandkuhl, K.: Information Overload in Industrial Enterprises - Results of an Empirical Investigation. in: Proc. 2nd European Conf. on Information Management and Evaluation, pp. 343-350, London (2008)
2. Mundbrod, N., Kolb, J., Reichert, M.: Towards a System Support of Collaborative Knowledge Work. in: Proc. 1st Int'l Workshop on Adaptive Case Management (ACM'12), pp. 31-42, Tallinn (2012)
3. Bocij, P., Chaffey, D., Greasley, A., Hickie, S.: Business Information Systems: Technology, Development and Management for the E-Business. Prentice Hall (2006)
4. Rowley, J.: The wisdom hierarchy: representations of the DIKW hierarchy. in: J. of Information Science, 33(2), pp. 163-180 (2006)
5. Michelberger, B., Mutschler, B., Reichert, M.: Towards Process-oriented Information Logistics: Why Quality Dimensions of Process Information Matter. in: Proc. 4th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'11), LNI 190, pp. 107-120, Hamburg (2011)
6. Michelberger, B., Mutschler, B., Reichert, M.: Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes. in: Proc. 16th IEEE Int'l EDOC Conf. (EDOC'12), pp. 21-30, Beijing (2012)
7. Hipp, M., Michelberger, B., Mutschler, B., Reichert, M.: A Framework for the Intelligent Delivery and User-adequate Visualization of Process Information. in: Proc. 28th Symp. On Applied Computing (SAC'13), pp. 1383-1390, Coimbra (2013)
8. Heuwinkel, K., Deiters, W.: Information logistics, E-Healthcare and Trust. in: Proc. Int'l Conf. e-Society (IADIS'03), 2, pp. 791-794, Lisbon (2003)
9. Deiters, W., Löffeler, T., Pfennigschmidt, S.: The Information Logistics Approach Toward User Demand-driven Information Supply. in: Proc. Conf. on Cross-Media Service Delivery (CMSD'03), pp. 37-48, Santorini (2003)
10. Womack, J.P., Jones, D.T.: Lean Thinking: Banish Waste and Create Wealth in Your Corporation. Free Press (2003)
11. Michelberger, B., Andris, R., Girit, H., Mutschler, B.: A Literature Survey on Information Logistics. in: Proc. 16th Int'l Conf. on Business Information Systems (BIS 2013), LNBIP 157, pp. 138-150, Poznań (2013)
12. Dinter, B., Winter, R.: Information Logistics Strategy - Analysis of Current Practices and Proposal of a Framework. in: Proc. 42nd Hawaii Int'l Conf. on System Sciences (HICSS-42), pp. 1-10, Hawaii (2009)
13. Winter, R.: Enterprise-wide Information Logistics: Conceptual Foundations, Technology Enablers, and Management Challenges. in: Proc. 30th Int'l Conf. on Information Technology Interfaces (ITI'08), pp. 41-50, Dubrovnik (2008)
14. Rinderle, S., Reichert, M., Dadam, P.: On Dealing with Structural Conflicts between Process Type and Instance Changes. in: Proc. 2nd Int'l Conf. Business Process Management (BPM'04), pp. 274-289, Potsdam (2004)
15. Gronau, N., Müller, C., Korf, R.: KMDL - Capturing, Analysing and Improving Knowledge-Intensive Business Processes. in: J. of Universal Computer Science (JUCS), 11(4), pp. 452-472 (2005)
16. Wurzer, J., Mutschler, B.: Bringing Innovative Semantic Technology to Practice: The iQser Approach and its Use Cases. in: Proc. 4th Int'l Workshop on Applications of Semantic Technologies (AST'09), pp. 3026-3040, Lübeck (2009)
17. Diestel, R.: Graph Theory. Springer (2010)
18. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating Web Spam with TrustRank. in: Proc. 13th Int'l Conf. on Very Large Data Bases (VLDB'04), 30, pp. 576-587, Toronto (2004)

19. Michelberger, B., Mutschler, B., Reichert, M.: On Handling Process Information: Results from Case Studies and a Survey. in: Proc. 2nd Int'l Workshop on Empirical Research in Business Process Management (ER-BPM'11), LNBIP 99, pp. 333-344, Clermont-Ferrand (2011)
20. Hipp, M., Mutschler, B., Reichert, M.: On the Context-aware, Personalized Delivery of Process Information: Viewpoints, Problems, and Requirements. in: Proc. 6th Int'l Conf. on Availability, Reliability and Security (ARES'11), pp. 390-397, Vienna (2011)
21. Borodin, A., Roberts, G.O., Rosenthal, J.S., Tsaparas, P.: Link Analysis Ranking: Algorithms, Theory, and Experiments. in: J. of ACM Transactions on Internet Technology (TOIT), 5(1), pp. 231-297 (2005)
22. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford University (1999)
23. MacKay, D.J.C.: Information Theory, Inference and Learning Algorithms. Cambridge University Press (2003)
24. Likert, R.: A Technique for the Measurement of Attitudes. in: Archives of Psychology, 140, pp. 1-55 (1932)
25. Lechtenbörger, J.: Data warehouse schema design. Infix Akademische Verlagsgesellschaft Aka GmbH, PhD Thesis, University of Münster (2001)
26. Bucher, T., Dinter, B.: Process Orientation of Information Logistics - An Empirical Analysis to Assess Benefits, Design Factors, and Realization Approaches. in: Proc. 41st Hawaii Int'l Conf. on System Sciences (HICSS-41), pp. 392-402, Hawaii (2008)
27. Janakiraman, V.S., Sarukesi, K.: Decision Support Systems. Prentice-Hall (2004)
28. Cameron, S.A.: Enterprise Content Management: A Business and Technical Guide. British Informatics Society (2011)
29. Kleinberg, J.M., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.S.: The Web as a Graph: Measurements, Models, and Methods. in: Proc. 5th Annual Int'l Conf. on Computing and Combinatorics, pp. 1-17, Tokyo (1999)
30. Xing, W., Ghorbani, A.: Weighted PageRank Algorithm. in: Proc. of the 2nd Annual Conf. on Communication Networks and Services Research, pp. 305-314, Fredericton (2004)
31. Lempel, R., Moran, S.: The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. in: Proc. 9th Int'l World Wide Web Conf. on Computer Networks, pp. 387-401, Amsterdam (2000)
32. Haveliwala, T.H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. in: J. of IEEE Transactions on Knowledge and Data Engineering, 15(4), pp. 784-796 (2003)
33. Vassilvitskii, S., Brill, E.: Using Web-Graph Distance for Relevance Feedback in Web Search. in: Proc. 29th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 147-153, Seattle (2006)
34. Lowd, D., Godde, O., McLaughlin, M., Nong, S., Wang, Y., Herlocker, J.L.: Challenges and Solutions for Synthesis of Knowledge Regarding Collaborative Filtering Algorithms. Technical Report, Oregon State University (2004)
35. Bian, J., Liu, Y., Agichtein, E., Zha, H.: A few Bad Votes too Many?: Towards Robust Ranking in Social Media. in: Proc. 4th Int'l Workshop on Adversarial Information Retrieval on the Web, pp. 53-60, Bejing (2008)
36. Michelberger, B., Mutschler, B., Reichert, M.: A Context Framework for Process-oriented Information Logistics. in: Proc. 15th Int'l Conf. on Business Information Systems (BIS'12), LNBIP 117, pp. 260-271, Vilnius (2012)