Institute of Architecture of Application Systems

Identifying Relevant Resources and Relevant Capabilities of Informal Processes

C. Timurhan Sungur[1], Uwe Breitenbücher[1], Oliver Kopp[2], Frank Leymann[1], Andreas Weiß[1]

[1]Institute of Architecture of Application Systems, University of Stuttgart, Germany
[2]Institute for Parallel and Distributed Systems, University of Stuttgart, Germany
firstname.lastname@informatik.uni-stuttgart.de

BibTeX:

```
@inproceedings{Sungur2017,
  author    = {C. Timurhan Sungur and Uwe Breitenb{\"u}cher and Oliver Kopp and
Frank Leymann and Andreas Wei{\ss}},
  title = {{Identifying Relevant Resources and Relevant Capabilities of
Informal Processes}},
  booktitle = {Proceedings of the 19th International Conference on Enterprise
Information Systems (ICEIS 2017)},
  year      = {2017},
  month = {April},
  pages = {295--307},
  publisher = {SciTePress}
}
```

These publication and contributions were presented at ICEIS 2017
ICEIS 2017 Web site: http://www.iceis.org/

© 2017 SciTePress. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the SciTePress.

Universität Stuttgart
Germany

Identifying Relevant Resources and Relevant Capabilities of Informal Processes

C. Timurhan Sungur[1], Uwe Breitenbücher[1], Oliver Kopp[2], Frank Leymann[1], and Andreas Weiß[1]
[1]*Institute of Architecture of Application Systems, University of Stuttgart, Germany*
[2]*Institute for Parallel and Distributed Systems, University of Stuttgart, Germany*
firstname.lastname@informatik.uni-stuttgart.de

Keywords: Informal Processes, Unstructured Processes, Resource Discovery, Capability Discovery, Relevant Resources, Relevant Capabilities

Abstract: Achieving goals of organizations requires executing certain business processes. Moreover, the effectiveness and the efficiency of organizations are affected by how their business processes are enacted. Thus, increasing the performance of business processes is in the interest of every organization. Interestingly, resources and their capabilities impacting past enactments of business processes positively or negatively can similarly have a positive or a negative impact in their future executions. Therefore, in our former work, we demonstrated a systematic method for identifying such resources and capabilities of business processes using interactions between resources of business processes without detailing the concepts required for this identification. In this work, we fill this gap by presenting a conceptual framework including concepts required for identifying resources possibly impacting business processes and capabilities of these resources based on their interactions. Furthermore, we present means of quantifying the significance of resources and their capabilities for business processes. To evaluate the identified resources and capabilities with their significance, we compare the results of the case study on the Apache jclouds project from our former work with the data collected through a survey. The results show that our system can estimate the actual values with 18% of a mean absolute percentage error. Last but not least, we describe how the presented conceptual framework is implemented and used in organizations.

1 INTRODUCTION

During their lifetime, organizations typically work to reach better states than their current ones, e.g., a more profitable state, a more knowledgeable state, and a more competitive state. To reach these desired states, organizations set and achieve their organizational *goals* [Etzioni, 1964]. Achieving these goals requires organizations to conduct a set of value adding activities in a certain order forming business processes of that organization. Documenting business processes supports humans when executing these processes, automating parts of executions, and improving these, as many of these business processes are executed repeatedly. In case business processes contain repetitive activities among different process enactments, business experts can document these repeating activities in business process models using activity-oriented approaches [Leymann and Roller, 2000, Weske, 2010]. These approaches enable modeling, executing, and improving business processes based on their activities. As activities and the order of them, i.e., the structure of processes, do not change in different executions, business process models can be used to prescribe their execution. In contrast to such *structured processes*, there are business processes which do not follow a clearly described sequence of activities, i.e., the activities and their execution order are not or only less structured in advance and evolve during execution. Thus, each execution is different from others. Different business process modeling and execution approaches have been proposed to deal with semi-structured and unstructured processes [Dustdar, 2004, Moody et al., 2006, Herrmann and Kurz, 2011, Sungur et al., 2014]. Semi-structured and unstructured processes typically involve activities for creating specific, individual knowledge on runtime. Due to strongly varying ways to achieve such goals, modeling *unstructured processes* based on activities typically does not increase the performance of these processes. Based on these properties, different works named these processes differently, such as *ad-hoc processes* [Dustdar, 2004], *unstructured processes* [Di Ciccio et al., 2015], *declarative flows* [van der Aalst et al., 2009], and *informal pro-*

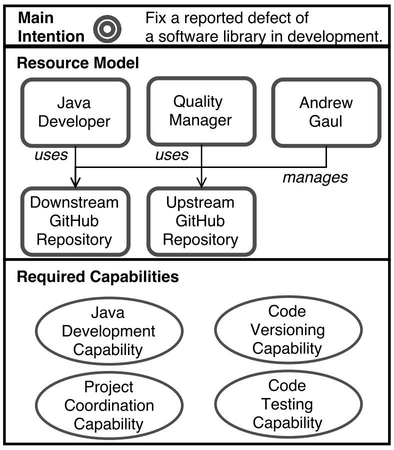

Figure 1: A simplified informal process model for a bug fixing process.

cesses [Sungur et al., 2014]. In the following, we refer to them as "informal processes".

Organizations desire to increase their performance, such as turnover, by improving the performance of their business processes. Increasing performance of activity-oriented processes has been on the focus of the research community for a long time [van der Aalst, 2016]. However, there is a lack of detailed systematic approaches for improving a business process containing negligible amounts of repeated activities shared among its executions. Improving the performance of unstructured processes involves allocating *relevant* capabilities provided by *relevant* resources. *Documented resources* are resources, whose participation in informal processes is already known and documented. For instance, an informal process for developing a open-source software is initiated with a Git repository. In this case, the Git repository is a documented resource due to being prescribed upon the initialization. *Relevant resources* are resources that interact with documented resources and other relevant resources in the course of an informal process. For instance, each pull request reviewer of a documented Git repository participating in an informal process is relevant resource due to interactions with a documented resource. Likewise, a *relevant capability* is a capability of a relevant or a documented resource of an informal process. Obviously, including certain relevant resources and capabilities in informal processes may increase their performance, such as including a person making frequent contributions in an open-source software development project. Thus, it is in the interest of organizations to efficiently find appropriate relevant resources and capabilities for enactments of such processes to optimize their executions. Unfortunately, this cannot be done a priori at design time since informal processes strongly vary from enactment to enactment [Di Ciccio et al., 2015]. Interestingly, previous enactments of such processes may give hints on appropriate relevant capabilities and resources that can be used.

In our previous work [Sungur et al., 2016], we have demonstrated the extended *Informal Process Execution (InProXec)* method enabling the identification of relevant resources and relevant capabilities on a case study. Identified relevant resources and relevant capabilities are presented to business experts as recommendations at *modeling time* to enforce their decision making processes with the information gathered from enactments of informal processes. Consequently, these recommendations do *not* target at providing a runtime support for human actors. In this work, we complement this demonstration with the conceptual foundations. Therefore, we present the following *contributions*: (i) a conceptual framework for identifying relevant resources and capabilities to support business experts during modeling (Sect. 4), (ii) steps required for *enabling* the generation of recommendations for informal processes using the conceptual framework presented (Sect. 4.1), and (iii) steps required for generating recommendations for informal processes using the conceptual framework (Sect. 4.2). In the following sections, we firstly present a motivating scenario (Sect. 2). After presenting the scenario, we recap the InProXec method (Sect. 3). Then, we present the contributions of this paper in Sect. 4. Finally, we present related work in Sect. 5 and conclude in Sect. 6.

2 MOTIVATING SCENARIO

The motivating scenario is based on the Apache jclouds project (https://jclouds.apache.org/) that exposes typical characteristics of how people work. The high-level goal of this project is developing Java libraries that unify access to functionalities of various cloud service providers. To achieve this goal, project members create an open-source project, in which as many contributors as possible are desired. As there are different cloud service providers in the market, the project is divided into sub-projects to integrate these different providers. Thus, different developers are experienced and responsible for different sub-projects. As the project is an open-source project, external contributors, i.e., people who are not part of the core project team, also participate in the development. For instance, to integrate OpenStack (https://www.openstack.org/), members of the OpenStack project create a sub-project implementing various APIs in Java, e.g., APIs for storage services. Such external experts contribute valuable knowledge about the project parts, e.g., the computational service API of OpenStack. Assigning the right resource to the right

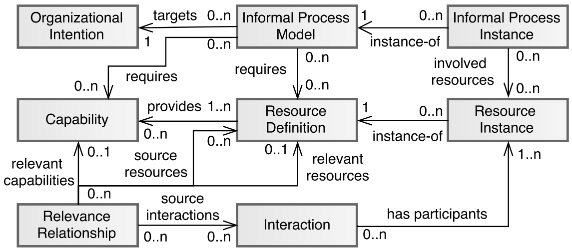

Figure 2: A view on real world entities of interest.

job will likely increase the efficiency of the process such as assigning a developer of the OpenStack storage API to a process of fixing a bug of the OpenStack storage API will reduce the time needed. Moreover, certain types of interactions between resources imply typical relevance relationships between them, e.g., a "commit" interaction implies a relevance relationship between the developer and the Git project containing the source of OpenStack storage API. Thus, this developer is a relevant resource for the respective process.

During the lifetime of the software development project, there are sub-processes that are executed repeatedly, such as reviewing code and fixing bugs. Each such sub-process represents an example for an informal process, as they involve activities for creating specific, individual knowledge at runtime. In case one capability is used during one instance of these repetitive executions, it is likely that it will be used in a later execution, too. Thus, such a capability can be considered as *relevant* for that specific process and the explicit inclusion of it can increase the efficiency of a process. The sub-process of reviewing code contributions of external committers requires a coordination capability, i.e., coordinating people during this revision process. In case this coordination capability is used by one of the documented resources during the enactment of sub-processes, this will imply that this capability can be relevant for the future executions, too. As a result, including such a relevant capability provided by a resource will most likely increase the efficiency of future executions of the same process. In this work, we present a conceptual framework for identifying relevant capabilities and resources by analyzing on interactions and capabilities of documented resources.

3 FUNDAMENTALS

In the following paragraphs, we first describe the Informal Process Essentials (IPE) approach [Sungur et al., 2014]. Hereafter, we describe the *four-phased Informal Process Execution (InProXec)* method [Sungur et al., 2015a, Sungur et al., 2016] enabling the usage of the IPE approach in organizations. In Fig. 2, we recap a formal model including real world entities and their relationships used in the InProXec method. All concepts illustrated in Fig. 2 are based on our previous work [Sungur et al., 2014, Sungur et al., 2015a, Sungur et al., 2015b, Sungur et al., 2016]. We introduced the concept of relevance relationships in our previous work [Sungur et al., 2016] but we did not detail these. In the current work, we present a conceptual framework generating relevance relationships.

Business processes have three dimensions: what, who, and which [Leymann and Roller, 2000]. The dimension "what" denotes activities of business processes conducted by actors represented by the dimension "who". To conduct activities, actors of business processes exploit other organizational resources, i.e., the dimension "which". In case of informal processes, activities are typically not predictable during modeling time due to the dominant change in activities at runtime. Moreover, modeling activities of informal processes is in many cases more costly than the performance increase provided by modeling. Thus, the IPE approach focuses on modeling and automatically allocating required resources of informal processes, i.e., the two dimensions "who" and "which", to support actors of processes and to reproduce desired outcomes of business processes. To model these two dimensions, business experts include resource definitions required in informal process models, as illustrated in Fig. 2. Each *resource definition* represents an organizational resource such as human resources, material resources, information resources, and IT resources. For instance, a resource definition can represent a human resource conducting required activities and a Git repository supporting these activities. As shown in Fig. 2, instead of directly referring to resources required in informal processes, business experts can model capabilities required in informal processes. Organizational *capabilities* represent abilities to perform a productive task using organizational resources. To guide resources during process enactments towards desired outcomes, business experts specify intentions of informal processes. Each *intention* describes desired outcomes of informal processes, e.g., a Java libraries that unify access to functionalities of various cloud service providers is the desired outcome of the motivating scenario. Furthermore, intentions enable tracking the *progress* of informal processes, such as two intentions completed out of three intentions implying a certain degree of the progress in an informal process. Resource definitions, capabilities, and intentions are stored in *informal process models*, as illustrated in Fig. 2. A simplified example informal process model of a process aiming at fixing a bug in the motivating scenario is shown in Fig. 1 with its resource definitions, capabilities, and the target intention. For further details about the approach, we refer the readers to [Sungur et al., 2014]. To enable the application of the IPE approach in organizations, we presented the InProXec approach [Sungur et al., 2015b]. In a previous work [Sungur et al., 2016], we demonstrated the application of an extended version of the InProXec approach to identify relevant resources and relevant capabilities of the motivating scenario. This previous work focuses on the *validation* of the extended method and does not describe the new concepts needed for the identification of relevant resource and relevant capabilities, which we address in the current work. Next, we give an overview of the InProXec method with four phases shown in Fig. 3.

Integrate Resources of Informal Processes (P_1). The first phase of the InProXec method has three objectives: (i) making resources required visible in modeling environments of informal processes, (ii) enabling the automated allocation of these resources during informal process model initialization, and (iii) enabling the generation of recommendations for informal processes. We have detailed means of achieving the objectives (i) and (ii) in our previous work [Sungur et al., 2015b]. In the current work, we present the means of achieving the objective (iii) in Sect. 4.1. The phase P_1 is a setup phase involving software development activities and must be initially executed before other phases. The Informal Process Essentials (IPE) approach focuses on modeling and automatically allocating actors and supporting resources of actors. Therefore, modeling environments of informal processes need to present business experts available actors and supporting resources of actors, i.e., *resources of interest*. Presenting resources from different resource domains requires understanding domain-specific resource definitions and transforming these into resource definitions of modeling environments of informal processes. To make organizational resources available in modeling environments of informal processes, technical experts develop software components called domain managers (the first objective of P_1).

Definition 1 (Domain Manager (DM))**.** *Domain managers are software components transforming domain-specific resource representations into resource definitions of modeling environments of informal processes.*

For instance, a domain manager of IT resources will create resource definitions for a new Git repository and for existing ones. After implementing domain managers, technical experts develop execution environment integrators to enable the automated allocation of resources for intentions of informal processes (the second objective of P_1).

Definition 2 (Execution Environment Integrator (EEI))**.** *Execution environment integrators are software components capable of executing life-cycle operations such as allocating and releasing resources during the enactment of an informal process.*

During resource allocations, EEIs convert resource definitions into *resource instances* containing instance descriptors. Each resource instance represents a unique allocated resource. We refer to resource definitions and instances in the following as resources if it causes no confusion. Resource definitions are a part of informal process models and are initialized upon initialization of the models. During initialization of an informal process model, a software component called an Informal Process Essentials runtime allocates all resources using available EEIs.

Definition 3 (Informal Process Essentials (IPE) Runtime)**.** *Informal Process Essentials runtimes are software components managing life-cycles of informal process instances using available execution environment integrators for each modeled resources definition.*

IPE runtimes correspond to business process execution engines of structured processes. We assume that before applying the InProXec method an IPE runtime already exists. Initializing an informal process model successfully results in an *informal process instance*. Informal process instances contain resource instances. During enacting informal processes, new intentions can emerge. These new intentions can be addressed with an updated set of allocated resource instances. Thus, the resources represented in informal process models and instances can vary. A valid analysis on resources of informal processes should not only consider resources addressed in informal process models but, rather, all resources in informal process instances of the process model. We refer to informal process models and instances in the following as informal processes as long as no confusion is possible. To address the third objective of P_1, i.e., enabling the generation of recommendations for informal process, technical experts develop software components capable of analyzing interactions of resources participating in informal processes and deriving recommendations from these interactions. *Informal process recommendations* include (i) relevant resources and capabilities of informal process models and (ii) new informal process models containing these identified resources and capabilities. For this analysis, technical experts develop software components collecting interactions of resources participating in informal processes. Moreover, they create software components capable of interpreting these interactions and generating informal process recommendations. Business experts exploit these recommendations to model more effective and efficient informal process models during the phase P_2, as presented in Fig. 3.

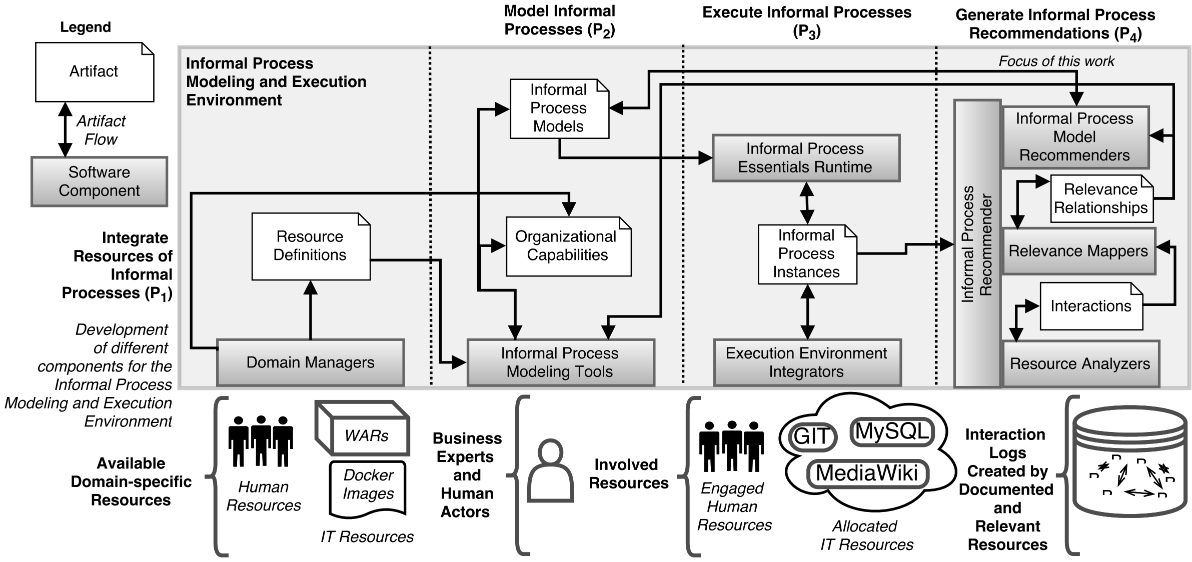

Figure 3: Phases of the InProXec method and its enabling system.

Model Informal Processes (P_2). The main objective of this phase is creating informal process models achieving organizational intentions. For this purpose, business experts model organizational capabilities provided by organizational resources, so that they can add these into informal process models. For instance, they specify a Java development capability provided by a set of human resources. Hereafter, they specify intentions of informal processes such as fixing a bug in a software library. To specify the means of achieving these intentions, business experts define informal process models using the resources integrated in P_1. After creating models, business experts initialize corresponding process models as described in P_3.

Execute Informal Processes (P_3). This phase has the objective of initializing informal process models and achieving organizational intentions specified in P_2. First, business experts select appropriate informal process models based on the present organizational context. Upon an initialization request, the IPE runtime uses EEIs to allocate resources documented in informal process models. For instance, they send an invitation for participation to human resources defined in the respective informal processes. In case they agree on the participation, they are considered to be allocated. Another example of a resource allocation is creating a new Git repository on GitHub for an informal process. Allocating all resources specified in an informal process instance converts the state of the instance into *achieving*, i.e., achieving intentions of the respective informal process. During the state achieving, actors of informal processes work towards the intentions of informal processes using other supporting resources collaboratively. As shown in Fig. 2, collaborations result in *interactions* containing information about actual executions of informal processes. After achieving intentions of informal processes, actors or business experts stop informal process instances. Hereafter, the employed IPE runtime deallocates resource instances using available EEIs.

Generate Informal Process Recommendations (P_4). The objective of this phase is generating informal process recommendations from interactions occurring among resources of informal processes. These recommendations include resources and capabilities possibly relevant for informal process models and new informal process models using these relevant resources and capabilities. The starting point of generating recommendations are the resources instances *documented* in informal process instances, as shown in Fig. 3. Thus, we start by defining "documented resources".

Definition 4 (Documented Resource)**.** *A documented resource of an informal process model is a resource instance allocated on purpose for an instance of the process model during the process initialization or the process enactment.*

In other words, a documented resource is not limited to the resources represented in informal process models but, rather, they include all allocated resources during the course of different process enactments. Documented resources provide a basis for identifying relevant resources of informal processes. Furthermore, the definition of relevant resources is built on top of documented resources:

Definition 5 (Relevant Resource)**.** *A relevant resource of an informal process is a resource instance interact-*

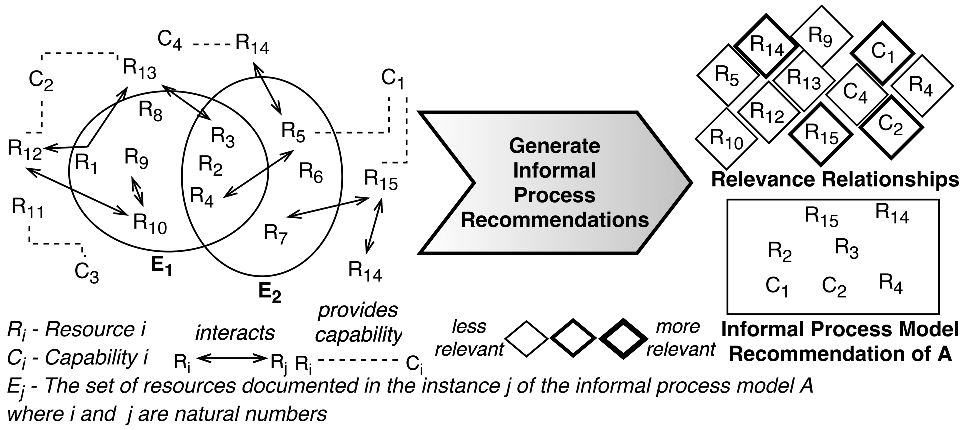

Figure 4: An illustration of generating informal process recommendations.

ing directly or indirectly with a documented resource. Indirectly means that there is an interaction path from a documented resource including other resources until the relevant resource is interacted with.

In Fig. 4, we present two informal process instances E_1 and E_2. According to the definition of documented resources, all resources in two circles are documented resources, i.e., $R_1 - R_{10}$. Moreover, all resources communicating with these are relevant resources, such as R_{15}. By communicating with the relevant resource R_{15}, the resource R_{14} becomes relevant. To define relevant capabilities, we exploit relevant resources, as follows:

Definition 6 (Relevant Capability)**.** *A relevant capability of an informal process is an existing or a derived capability of a relevant or a documented resource.*

In Definition 6 existing capabilities refer to modeled capabilities of resources during P_2. For instance, in Fig. 4, C_1 is a relevant capability as it is a capability of both R_{15} and R_5. The derived capabilities are capabilities created using different properties of resource interactions, such as type and frequency of an interaction. An example of such a capability is a coordination capability derived using reviews of a pull request, which are interpreted as coordination of pull requests of a software project. To fulfill the objective of generating informal process recommendations, relevant resources and relevant capabilities are needed to be identified. Therefore, during P_4, software components built in P_1 collect interactions of relevant resources and, hereafter, interpret these interactions to assign different degrees of relevance for relevant resources and capabilities. Using these assigned degrees, software components built in P_1 create informal process model recommendations. Informal process model recommendations contain relevant resources and capabilities found previously and are based on the original informal process model. Next, we present a conceptual framework (Sect. 4) for enabling the generation recommendations for informal processes (Sect. 4.1) and generating the recommendations (Sect. 4.2).

4 CONCEPTUAL FRAMEWORK FOR GENERATING INFORMAL PROCESS RECOMMENDATIONS

In this section, we present key concepts used to generate informal process recommendations. Interacting resources build a unweighted bi-directional resource-interaction graph where resources are nodes of the graph and interactions are the edges connecting these nodes, as shown in Fig. 4. The shortest path between two resources in a graph is the path with the least number of edges connecting the two resources. In a graph, the *distance* between two resources is the number of edges in the shortest path connecting these. Based on this distance definition, we define the distance of a relevant resource as follows:

Definition 7 (Distance of a Relevant Resource)**.** *The distance of a relevant resource to a documented resource is the number of edges in the shortest path connecting these resources.*

In Fig. 4, the distance of R_{14} is 2 due to the interactions between R_7/R_{15} and R_{15}/R_{14}. According to Definition 7, the documented resource used to identify a relevant resource is the starting point for calculating the distance of a relevant resource. As shown in Fig. 3, we initially exploit information available in *interactions* to identify relevance relationships containing relevant resources and relevant capabilities. Therefore, resource interactions over different mediums need to be investigated, e.g., physical interactions, emails, Git commits, and Wiki edits. To collect interactions of a documented resources, we employ resource analyzers:

Definition 8 (Resource Analyzer)**.** *Resource analyzers are software components mapping resource instances to a set of interactions of a certain type occurred in a certain time span.*

Technical experts create resource analyzers to enable the generation of informal process recommendations, as detailed in Sect. 4.1. Resource analyzers are used to collect interactions of documented and relevant resources recursively, i.e., they collect first resources in distance 1 and, then, 2 until the given distance. Moreover, they collect interactions in certain *time spans* to limit the collected interactions to the interactions happened during executions of informal processes. For instance, a resource analyzer of a GitHub repository can collect interactions of the last three months from this repository to identify developers working on it during execution of a bug fixing process spanning the previous three months. Hereafter, each of the developers found can be further investigated with their corresponding resource analyzers to collecting further interactions resolving in additional relevant capabilities and relevant resources, such as additional GitHub repositories. Each interaction collected by resource analyzers contains a relevant resource and relevant capabilities of an informal process model. To associate these relevant resources and capabilities with informal process models, *relevance relationships* are used.

Definition 9 (Relevance Relationships)**.** *A relevance relationship specifies the degree of relevance of a resource or a capability with an informal process model.*

A relevance relationship is considered as an association with informal process models, because these relationships rely on interactions of documented and relevant resources of informal process models. The degree of the relevance of a resource or a capability depends on different factors such as (i) the type, the frequency, and contents of an interaction, (ii) the degree of relevance of a relevant resource providing a relevant capability, and (iii) the distance of a relevant resource. Thus, each relevance relationship contains the degree of a relevance derived using interactions, i.e., the *correlation coefficient* of the relevance relationship. A relevance relationship implies either (i) a positive relevance, (ii) an irrelevance, or (iii) a negative relevance, that is (i) a positive correlation coefficient, (ii) a zero correlation coefficient, and (iii) a negative correlation coefficient, respectively. A positive relevance means that a resource definition should be included in an informal process model. An irrelevance means that the considered resource has no impact on the informal process. In contrast, a negative relevance means, a resource definition should not be included in a process model. To generate recommendations for informal processes (Sect. 4.2), relevance relationships are identified. Thus, there is a need for a means of determining relevance relationships using existing entities such as interactions or other relevance relationships. Therefore, we employ *relevance mappers*:

Definition 10 (Relevance Mapper)**.** *Relevance mappers are software components enriching a set of relevance relationships using (i) interactions collected by resource analyzers and (ii) existing relevance relationships identified by other relevance mappers.*

For instance, a relevance mapper converts an interaction between a Git repository and a developer into a relevance relationship containing the relevant Git resource with a positive correlation coefficient. In case such a relevance relationship is already present, the relevance mapper updates the correlation coefficient of the corresponding relevance relationship. Technical experts build these relevance mappers to enable the generation of recommendations for informal processes, as shown in Fig. 3 and detailed in Sect. 4.1.

To calculate the correlation coefficients of relevance relationships, we created the following requirements: A relevance mapper can calculate the correlation coefficient of a relevance relationship using different information sources, such as interaction types, interaction contents, and identified relevance relationships. For instance, a commit interaction implies a higher relevance in a development project than an email interaction (the type of an interaction). Moreover, the number of lines committed is also important during the calculation of correlation factors (the contents of an interaction). Thus, such different information sources should have effect in the calculated correlation coefficients (Rq_1). Furthermore, different information sources can imply the relevance or the irrelevance of the same relevant resource or the same relevant capability. For instance, multiple commits made by the same developer will increase the relevance of the respective developer. Thus, the resulting correlation coefficient of relevant resource or a relevant capability should represent an aggregated value calculated based on different information sources (R_2). Typically, relevant resources with a larger distance are less relevant for a specific informal process. For instance, in an informal process a Git repository is included. Moreover, this Git repository is updated by a relevant human resource, who works on another repository, which is less relevant than the human resource used to identify the latter repository. Thus, relevant resources with a larger distance and their relevant capabilities should be less relevant (Rq_3).

Based on these requirements, we designed the function presented in Equation (1) within the range of

$\mathbb{R}$. The function maps either a relevant resource or a relevant capability (rc) based on the interactions (I) and relevance relationships (R) to a new correlation coefficient ($cc(rc, I \cup R)$). The function iterates through interactions and relevance relationships and relies on other functions depending on an interaction or a relevance relationship, that are, a relevance factor ($rFactor(rc, ir)$) and the distance ($d(rc, ir)$) of the relevant capability. Consequently, the value of the correlation coefficients depend on interactions and relevance relationships ($I \cup R$) of a relevant resource or relevant capability (rc) given.

$$cc(rc, I \cup R) = \sum_{ir \in I \cup R} \frac{rFactor(rc, ir)}{d(rc, ir)} \quad (1)$$

where $cc(rc, I \cup R) \in \mathbb{R}$, $rFactor(rc, ir) \in \mathbb{R}$, and $d(rc, ir) \in \mathbb{Z}^+$.

To align the variable impact of different types of interactions or relevance relationships used to calculate a new correlation coefficient, we use the function relevance factor ($rFactor(rc, ir)$). Interestingly, contents of interactions or relevance relationships can be considered during the calculation of values of $rFactor(rc, ir)$ for each interaction or relevance relationships, too. As a consequence of considering contents, an interaction or a relevance relationship can map to a relatively smaller or a larger value. For instance, a relevance mapper of a GitHub interactions can analyze collected interactions representing issues in the *time span* of an informal process instance. Furthermore, the relevance mapper can map to a higher or a smaller value based on the contents of issues, such as by doing a topic analysis and matching these with intentions of an informal process instance [Li and Yamanishi, 2003]. Consequently, different issues created out of the scope of an informal process instance are eliminated. Such interactions typically exist within the interactions containing unstructured text resulting in ambiguous interpretations. In contrast, the meaning of structured interactions is typically unambiguous and represents typically a certain productive or unproductive activity, such as a Git commit interaction. In relevance relationships, we represent (i) resources conducting productive and unproductive activities and (ii) capabilities of these resources with different correlation coefficients. Moreover, based on the sign of a $rFactor$, the next calculated value of a correlation coefficient is either higher or lower. To this end, a negative relevance factor will imply unproductive activities deduced using certain types of interactions or relevance relationships, such as spam emails. Thus, the equation satisfies Rq_1. Furthermore, the summation of calculated values for each different interaction or relevance relationship results in a correlation coefficient representing different these. Consequently, the equation fulfills Rq_2. More distanced relevant resources and their capabilities result in a smaller correlation coefficient and, thus, are less relevant using the variable ($d(rc, ir)$) in the equation. Consequently, Rq_3 is satisfied.

In our previous work [Sungur et al., 2016], we presented a case study validating the conceptual framework presented here using the Apache jclouds project described in the motivating scenario. Thereby, we used our implementation of the framework[1] for generating relevance relationships and new informal process models based on the relationships. During the generation of relevance relationships, we validated that the equation does not depend on the order of interactions given by making multiple runs on the same interaction sets with different orderings. Furthermore, recently, we evaluated the equation presented in Equation (1) empirically by comparing the correlation coefficients of the relevance relationships of resources generated in the case study with a set of correlation coefficients collected in a survey with 13 experienced GitHub users.

The experience of the participants on using GitHub in our survey varies between 6 months to 72 months with an average of 34 months. We provided each participant numbers and types of interactions of 21 resources identified in our case study. For instance, we presented a GitHub user with 10 commits and 2 pull requests. More specifically, we presented 1069 interactions in 8 different types collected during our case study distributed to 21 different resources. Furthermore, each expert evaluated each resource with the same interactions twice to be able to check the quality of given answers. Based on this list and on the experience of participants, we asked each participant to assign an integer value between 1 (not relevant) and 5 (relevant) for each resource listed. Consequently, in our comparison we had to map from the automatically generated values onto the scale of 1 to 5, as the actual value of the positive correlation coefficient is in range of $\mathbb{R}$. Different interpretations of different types of interactions result in a standard deviation between 0 and 0.19 for the values of different resources assigned by participants. Moreover, the statistical correlation coefficient between the automatically generated values and the average of the user assigned values is 0.93 meaning that the generated results and collected results have a strong linear positive correlation. Individually considered, the statistical correlation coefficient between the generated values and each data set provided by each participant vary between 0.79 and 0.98. Furthermore, the root mean square error and mean absolute percentage error values [Armstrong, 1978] are 0.62 and 18%, respectively.

[1] https://gitlab.com/timur87/informal-process-recommender

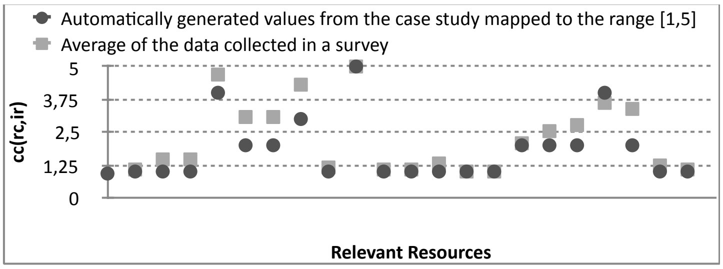

Figure 5: Comparison of correlation coefficients derived by experienced GitHub users and generated automatically.

Fig. 5 presents a comparison of the generated values and the average of the values provided by participants. Moreover, the results differ at certain points. This is firstly because of the assigned relevance factors ($rFactor(rc, ir)$) representing the increase rate of correlation coefficient per interaction did not meet the consensus of the participants, which can resolved by changing the assigned relevance factors. Secondly, the results provided by human resources are positive integers between 1 and 5 resulting in a lower precision. In contrast, the correlation coefficient generated based on the conceptual framework is with a higher precision.

Relevance relationships are already recommendations by themselves through presenting relevant capabilities and resources and their degree of relevance during P_2. Using relevance relationships, its possible to create *informal process model recommendations*, that is, new informal process models based on relevant resources and capabilities. For instance, in case a relevance relationship with a correlation coefficient 0.9 between a developer and an informal process model exists, an informal process model recommendation contains this relevant resource. As presented in Fig. 3, to automate generating informal process model recommendations from relevance relationships, we introduce the concept of *informal process model recommenders*, which are developed during P_1 of our method (Fig. 3).

Definition 11 (Informal Process Model Recommender)**.** *An informal process model recommender generates a new informal process model using relevance relationships.*

During P_2, generated recommendation models are presented to business experts. Finally, we introduce the concept of *informal process recommenders* orchestrating resource analyzers, relevance mappers, and informal process model recommenders to generate informal process recommendations.

Definition 12 (Informal Process Recommender)**.** *An informal process recommender generates informal process recommendations, i.e., relevance relationships and informal process model recommendations, using resource analyzers, relevance mappers, and informal process model recommenders.*

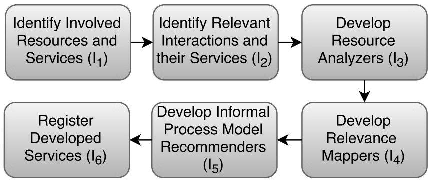

Figure 6: Detailed steps of enabling the generation of recommendations for informal processes.

In the following subsections, we describe necessary steps for enabling the generation of recommendations for informal processes (Sect. 4.1) and generating recommendations for informal processes (Sect. 4.2) using the conceptual framework in organizations.

4.1 Enabling the Generation of Recommendations for Informal Processes using the Conceptual Framework

This section presents the additional steps executed in Fig. 6, i.e., $I_2 - I_6$, after executing the existing step I_1 to achieve the objective of enabling the generation of recommendations for informal processes of the phase P_1 of the InProXec method. As organizations are living organisms, resources playing a role in informal processes change continuously. Thus, steps presented in this section can be executed repeatedly during the lifetime of an organization to adapt newly emerging intentions of informal processes. We validated the following steps in the context of a case study presented in our former work [Sungur et al., 2016].

Identify Involved Resources and Services (I_1): At first, business experts and technical experts identify possible actors of informal processes and their supporting resources in the context of organizational intentions using their experience on the past and present informal processes. They work together, as both organizational knowledge of business experts and IT knowledge of technical experts are required. Additionally, they can make interviews with human actors of informal processes to identify further resources involved in informal processes, if it is needed.

Identify Relevant Interactions and their Services (I_2): Using the resources determined in I_1, technical experts and business experts identify resource interactions providing information about the relevance of resources and their capabilities, i.e., *relevant interactions*, in the context of organizational intentions. Finding relevant interactions is followed by identifying services capable of delivering these interactions, e.g., GitHub and MediaWiki services.

Develop Resource Analyzers (I_3): To generate recommendations for informal processes (Sect. 4.2), initially, interactions containing documented and relevant resources should be collected. To collect interactions identified previously in I_2, technical experts create resource analyzers. Resource analyzers use the selected interaction services such as the GitHub service from I_2 to collect interactions containing documented and relevant resources. Moreover, each resource analyzer converts different addressing schemes used in different kinds of interactions into a system-specific format of the corresponding execution environment of informal processes. Each resource analyzer typically addresses a certain type of resources, such as human resources, and a certain interaction service, such as GitHub to assign a single responsibility to each analyzer.

Develop Relevance Mappers (I_4): After having a mechanism to collect interactions using resource analyzers, technical experts proceed with developing services for interpreting these interactions. Depending on the selected resources in I_1, selected interactions in I_2, and their services, technical experts develop relevance mappers to generate relevance relationships containing relevant resources and relevant capabilities.

Develop Informal Process Model Recommenders (I_5): Additionally, to automate creating informal process model recommendations, technical experts can develop informal process model recommenders, which incorporate relevance relationships to provide improved informal process models containing relevant resources and capabilities. During this step, technical experts define a specific threshold value to eliminate certain relevant resources and capabilities. For instance, involving a contributor who made just one commit, i.e., having a low correlation coefficient, can be more costly than the value he adds. This threshold is set for eliminating such cases.

Register Developed Services (I_6): After creating resource analyzers, relevance mappers, and informal process model recommenders, technical experts register these to an informal process recommender (Definition 12) to enable an automated discovery (I_6). Consequently, the infrastructure is ready to generate the recommendations in P_4.

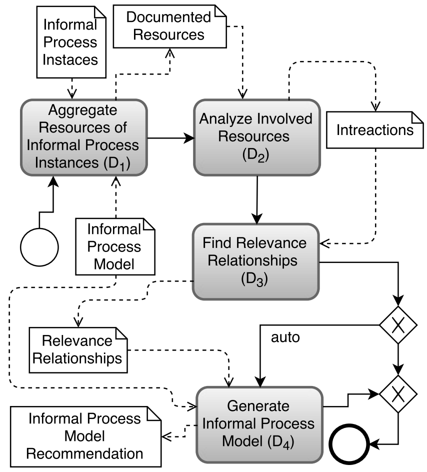

Figure 7: Generating recommendations for informal processes.

4.2 Generating Recommendations for Informal Processes using the Conceptual Framework

Generating recommendations for informal processes happens first after executing the steps described previously. The steps shown in Fig. 7 are executed by an informal process recommender automatically during P_4 of the InProXec method. As resources of the respective informal process may not be accessible after its finalization, this phase needs to be executed before finalizing informal process instances. We validated the presented steps in the context of a case study presented in our former work [Sungur et al., 2016].

Aggregate Resources of Informal Process Instances (D_1): As shown in Fig. 7, the generation process is initiated with an informal process model and its corresponding instances. In case multiple informal process instances are provided as input, a merger component aggregates all documented resources contained in different process instances. Duplicate resource instances are eliminated to avoid extra computational effort, that is collecting the same interactions more than once for the same resource instance.

Analyze Documented Resources (D_2): After the aggregation, the set of documented resources go through resource analyzers developed during I_3 to collect the relevant interactions identified in I_2, such as Git commits and pull requests.

Find Relevance Relationships (D_3): After collecting interactions in D_3, relevance mappers created in I_4 derive relevance relationships containing relevant resources and relevant capabilities. Derived relevance relationships are passed through relevance mappers iteratively. Thus, in case relevance mappers deduce a specific relevance relationship more than once, they update the correlation coefficient of the relationship.

Generate Informal Process Model (D_4): The output of D_3 is a set of relevance relationships, as presented in Fig. 7. Afterwards, the business expert can either decide to stop with identified relevance relationships or to continue with automatic generation of an informal process model recommendation. Informal process model recommenders exploit relevance relationships to generate a new informal process model, i.e., an informal process model recommendation, based on a provided informal process model.

In P_2, business experts can use either (i) generated relevant resources and relevant capabilities contained in relevance relationships or (ii) generated informal process model recommendations. The first option provides business experts more flexibility and the second one has the advantage of causing less effort as it reduces the steps to be taken by business experts in P_2. Although informal process model recommendations may reduce the effort of business experts, they need to be assessed by business experts before they can be used. Thus, in both cases business experts evaluate generated recommendations.

5 RELATED WORK

The approach presented has commonalities with Expert Finding Systems (EFS) [Schall, 2009, Balog et al., 2006], as EFSs typically address finding the right people for certain topics. In the context of our work, the concept of resource differs and is not restricted to human resources, i.e., experts. Thus, our approach does not only recommend experts but also other supporting relevant resources, e.g., a specific Wiki resource. Begel et al. [Begel et al., 2010] present a framework called Codebook creating a graph of interrelated resources using interactions of people, work items, files, and source code. Although Codebook is capable of finding different type of interrelated relevant resources, the framework does not address the degree of resource relevance and relevant capabilities. Dorn et al. [Dorn et al., 2011] proposed a skill-dependent recommendation model for finding experts with a better connectivity and matching skills. In this work, we also rely on interactions between resources to make recommendations. However, we do not specify how the correlation among resources are calculated, but rather we leave that as an extension point depending on the context of the problem and specific resource interactions. As future work, we will exploit existing research in these area, such the work of Dorn et al. [Dorn et al., 2011] and the work of Campbell et al. [Campbell et al., 2003], to improve relevance mappings.

The method presented makes no assumptions on the existence of reusable structured activities or a business process execution engine executing these activities defined in an automated fashion. In case structured activities and a business process engine are present in the environment, business process mining approaches can be exploited to improve business process executions [van der Aalst, 2016]. Different approaches provide recommendations for future resource allocations of business processes using event logs of process executions [Arias et al., 2016, van der Aalst and Song, 2004, Yang et al., 2012]. These approaches focus on interactions between allocated resources and business process execution engines. In contrast, our approach focuses on every meaningful resource interaction including interactions with business process execution engines. Our approach provides no replacement for such activity-oriented approaches, but it stands rather complementary. Dorn and Dustdar [2011] presented an activity recommendation system relying on message-exchanges for unstructured processes. Furthermore, Folino et al. [Folino et al., 2014, Folino et al., 2015b, Folino et al., 2015a] present means of identifying activities of informal processes using a clustering based discovery approach on event logs.

Vasconcelos et al. [Vasconcelos et al., 2001] present necessary elements and their relationships to model organizational goals, resources, processes, and their associations. However, their work does not aim at improving these processes but enables traceability between goals, processes, and resources. Adaptive Case Management [Herrmann and Kurz, 2011] and Case Handling [van der Aalst et al., 2005] offer the notion of case to avoid context tunneling and rigid activity structures. During the execution of a case, actors can collect relevant information in a case and reuse this information in the future. Using the case data, the *documented* information and actors can be captured. In contrast, the concept of resource in our approach is more generic enabling the identification of any kind of *relevant* resources.

6 CONCLUSION AND OUTLOOK

In this work, we presented a conceptual framework for identifying resources and capabilities that may be relevant for future executions of business processes. Identified resources and capabilities are associated with values representing their degree of relevance to business processes. To evaluate our approach, we conducted a survey and compared the collected data in the survey with the automatically generated results. Moreover, we presented steps for enabling the generation of recommendations for informal processes and necessary steps involved during the automated generation of recommended resources and capabilities.

In the future, we will investigate further empirical evaluation methods for the system presented, such as the application of recall and precision metrics from the field of information retrieval. Moreover, we will develop additional interaction interpreters based on existing approaches, such as expert finding systems. We will additionally present the implementation of new types of resources that are allocated during enactments of business processes in an ad-hoc fashion based on emerging requirements of human actors.

ACKNOWLEDGMENTS

This work has been partially supported by Graduate School of Excellence advanced Manufacturing Engineering (GSaME), DFG Cluster of Excellence in Simulation Technology (EXC 310/2), and SmartOrchestra (Research Grant 01MD16001F, BMWi).

REFERENCES

Arias, M. et al. (2016). A framework for recommending resource allocation based on process mining. In *BPM 2015*, number 256 in LNBIP. Springer.

Armstrong, J. S. (1978). *Long-range Forecasting: From Crystal Ball to Computer*. John Wiley & Sons Inc.

Balog, K., Azzopardi, L., and de Rijke, M. (2006). Formal models for expert finding in enterprise corpora. In *SIGIR '06*, pages 43–50. ACM.

Begel, A., Khoo, Y. P., and Zimmermann, T. (2010). Codebook: discovering and exploiting relationships in software repositories. In *ICSE '10*, volume 1, pages 125–134. ACM.

Campbell, C. S., Maglio, P. P., Cozzi, A., and Dom, B. (2003). Expertise identification using email communications. In *CIKM '03*, pages 528–531. ACM.

Di Ciccio, C., Marrella, A., and Russo, A. (2015). Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches. *JoDS*, 4(1):29–57.

Dorn, C. and Dustdar, S. (2011). Supporting dynamic, people-driven processes through self-learning of message flows. In *CAiSE 2011*, volume 6741 of *LNCS*, pages 657–671. Springer.

Dorn, C., Skopik, F., Schall, D., and Dustdar, S. (2011). Interaction mining and skill-dependent recommendations for multi-objective team composition. *DKE*, 70(10):866 – 891.

Dustdar, S. (2004). Caramba—a process-aware collaboration system supporting ad hoc and collaborative processes in virtual teams. *DPD*, 15(1):45–66.

Etzioni, A. (1964). *Modern Organizations*. Prentice Hall.

Folino, F., Guarascio, M., and Pontieri, L. (2014). Mining predictive process models out of low-level multidimensional logs. In *CAiSE 2014*, pages 533–547. Springer.

Folino, F., Guarascio, M., and Pontieri, L. (2015a). Mining multi-variant process models from low-level logs. In *BIS 2015*, volume 208 of *LNBIP*, pages 165–177. Springer.

Folino, F., Guarascio, M., and Pontieri, L. (2015b). On the discovery of explainable and accurate behavioral models for complex lowly-structured business processes. In *ICEIS 2015*, pages 206–217. SCITEPRESS.

Herrmann, C. and Kurz, M. (2011). Adaptive case management: Supporting knowledge intensive processes with it systems. In *S-BPM ONE 2011*, volume 213 of *CCIS*, pages 80–97. Springer.

Leymann, F. and Roller, D. (2000). *Production Workflow: Concepts and Techniques*. Prentice Hall PTR.

Li, H. and Yamanishi, K. (2003). Topic analysis using a finite mixture model. *IPM*, 39(4):521 – 541.

Moody, P., Gruen, D., Muller, M., Tang, J., and Moran, T. (2006). Business activity patterns: A new model for collaborative business applications. *IBMSJ*, 45(4):683–694.

Schall, D. (2009). *Human Interactions in Mixed Systems - Architecture, Protocols, and Algorithms*. PhD thesis, TU Wien.

Sungur, C., Dorn, C., Dustdar, S., and Leymann, F. (2015a). Transforming collaboration structures into deployable informal processes. In *ICWE 2015*, volume 9114 of *LNCS*, pages 231–250. Springer.

Sungur, C. T., Binz, T., Breitenbücher, U., and Leymann, F. (2014). Informal Process Essentials. In *EDOC 2014*, pages 200–209. IEEE.

Sungur, C. T., Breitenbücher, U., Leymann, F., and Wettinger, J. (2015b). Executing informal processes. In *iiWAS '15*, pages 54:1–54:10. ACM.

Sungur, C. T. et al. (2016). Identifying relevant resources and relevant capabilities of collaborations - a case study. In *EDOCW 2016*, pages 1–4. IEEE.

van der Aalst, W. (2016). *Process Mining: Data Science in Action*. Springer.

van der Aalst, W., Pesic, M., and Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *CSRD*, 23(2):99–113.

van der Aalst, W. and Song, M. (2004). Mining social networks: Uncovering interaction patterns in business processes. In *BPM 2004*, volume 3080 of *LNCS*, pages 244–260. Springer.

van der Aalst, W., Weske, M., and Grünbauer, D. (2005). Case handling: a new paradigm for business process support. *DKE*, 53(2):129 – 162.

Vasconcelos, A. et al. (2001). A framework for modeling strategy, business processes and information systems. In *EDOC 2001*, pages 69–80.

Weske, M. (2010). *Business Process Management: Concepts, Languages, Architectures*. Springer.

Yang, H., Wen, L., Liu, Y., and Wang, J. (2012). An approach to recommend resources for business processes. In *OTM 2012*, volume 7567 of *LNCS*, pages 662–665. Springer.