# Refining process models through the analysis of informal work practice

Simon Brander[‡], Knut Hinkelmann[‡], Bo Hu[†], Andreas Martin[‡],
Uwe V. Riss[†], Barbara Thönssen[‡], Hans Friedrich Witschel[‡]

[†]SAP AG, Walldorf, Germany
{Bo01.Hu, Uwe.Riss}@sap.com
[‡]University of Applied Sciences Northwestern Switzerland (FHNW),
Riggenbachstr. 16, 4600 Olten, Switzerland
{Barbara.Thoenssen, Simon.Brander, Andreas.Martin, Knut.Hinkelmann,
HansFriedrich.Witschel**}@fhnw.ch
(Author names in alphabetic order)

**Abstract** The work presented in this paper explores the potential of leveraging the traces of informal work and collaboration in order to improve business processes over time. As process executions often differ from the original design due to individual preferences, skills or competencies and exceptions, we propose methods to analyse personal preferences of work, such as email communication and personal task execution in a task management application. Outcome of these methods is the detection of internal substructures (subtasks or branches) of activities on the one hand and the recommendation of resources to be used in activities on the other hand, leading to the improvement of business process models. Our first results show that even though human intervention is still required to operationalise these insights it is indeed possible to derive interesting and new insights about business processes from traces of informal work and infer suggestions for process model changes.

## 1 Introduction

Modelling business processes is a time-consuming, costly and error-prone task. Even with the greatest effort, it is often impossible to foresee all the situations that may occur during process execution. It is not even desirable to include all the possibilities due to a very practical reason: a complete model normally has high complexity and thus can be prohibitively expensive to manage and visualise. One compromise is to define only high-level structures (e.g. critical branches of business processes) either manually or semi-automatically with intervention from human experts. In order to reduce the workload of and dependency on process experts, approaches based on process mining have been put forward. They analyse the event logs of enterprise information systems so as to automatically suggest the model structures or to check the conformance of a model against the actual sequence of events reflected in the log.

** Contact author: Hans Friedrich Witschel

Log-based mining relies on the quality of event documentation that renders it less useful in situations where processes are handled through informal methods. In every-day work practice, it is not unusual for people to keep their own records (such as personal notes and draft documents) and information about the process activities out of the enterprise information system and hence invisible in event logs. Meanwhile, in modern organisations, it is also common for employees to conduct process-related communication via email—in addition to or as a replacement of communication through a workflow system. Emails, even though closely work related, demonstrate strong informal and personal characteristics. Although artefacts such as personal records and emails are not deemed the same as formally modelled knowledge, they carry important information about day-to-day business processes, a good understanding of which provide valuable input to the formal process models.

The work presented in this paper explores the potential of leveraging the traces of informal work and collaboration in order to improve process models over time. We look at two types of traces, namely emails and personal task instances gathered in a task management application. We propose methods that analyse the internal structure of emails and personal tasks in order to derive information about the business process, e.g. for adding activities or branches or recommending resources. It is important to understand that the suggested methods do not treat emails or tasks as atomic units – as is often done with events in system logs – but analyse their internal structure.

## 2 Related Work

In this section, we review the two research threads from which our approach is drawn. We first familiarise readers with the basic notions of *agile business processes* that will then be used throughout the paper when discussing the details and impacts of our algorithms. We then elaborate on the difference of our approach from the apparently similar process mining approaches that also rely on traces of work practice.

### 2.1 Agile Process Modelling and Execution

Agility is one of the major challenges that modern enterprises confront nowadays and a distinct character of successful companies. Organisational agility can be defined as an organisation's ability to "[...] sense opportunity or threat, prioritise its potential responses, and act efficiently and effectively." [14]

In terms of business processes, organisational agility implies a more permeable separation between build-time modelling and run-time execution, allowing a fluid transition between them [13]. The Knowledge-Intensive Service Support (KISS) [10] facilitates the shift form rigid process modelling approaches to flexible and agile processes. KISS gives the possibility to tackle exceptional situations, unforeseeable events, unpredictable situations, high variability, and highly complex tasks without having to change the process model very often.

This is done by combining conventional business process modelling with (business) rules—the process model can be seen as basic action flow and guidance where the business rules define the constraints. In order to provide the flexibility KISS introduced a special modelling construct called Variable Activity [1]. [18] has further developed the Variable Activity concept into *Knowledge Intensive Activity* (KIA). A knowledge-intensive activity is an activity that requires expertise or skills to be executed. The execution and results of KIAs depend on the actual context wherein the process is reinforced. Context data include application data, process data, functional data, or further information about needed resources [5]. A knowledge intensive process (KIP) is a special form of process where (some of) the activities are optionally executed—depending on information specific for the certain process instance—and in any order and in any frequency. Thus a KIP is comparable to ad-hoc processes known from BPMN [1]. The KISS approach is combined with task patterns into a collaborative process knowledge management and maturing system, called KISSmir [13,18].

### 2.2 Process Mining

Process mining discovers formal process models from usage data, e.g. event logs containing the recorded actions of real process executions (*cf.* [2,8,17]). Enterprise information systems, such as enterprise resource planning systems, are well suited to process mining. These systems use a structured approach, which means that they have a "predefined way of dealing with things" [16]. Unstructured systems, such as groupware systems, on the other hand, provide more liberties to the user and the exchange of information. Such freedom makes the use of event logs from unstructured systems more challenging. Let's take emails as an example. While an email header contains structured elements (e.g. sender, recipient, date/time, subject), its main content is stored in the unstructured body and is not readily available for process analysis. There are approaches and tools that focus on the process discovery from the event data and logs produced by unstructured systems. For instance, Dustdar et al. propose mining techniques for ad-hoc processes that are supported by a process-aware collaboration system named Caramba [9]. Van der Aalst and Nikolov describe a process mining approach and a tool that uses email logs as input and creates a process model out of it [15]. They rely heavily on the assumption that the emails are already appropriately tagged (e.g. task name and status in the subject) which is not necessarily true. Tagging emails is time consuming and error-prone. Di Cicco et al. go one step further and use information extraction to identify task names from the unstructured email body [7].

As opposed to the approaches mentioned in this section, we do not treat traces of people's work (such as events in event logs, tasks or emails) as atomic units, which essentially need to be put into the right order. Instead, we are interested in investigating the potential of discovering internal structure of such events by analysing the *contents* of traces (emails, tasks) and relating them to

[1] http://www.omg.org/spec/BPMN/2.0/PDF

the overall business process. We claim that process models may not be fine-grained enough in many cases and that therefore the treatment of events as atomic units will fail to reveal many possible refinements of such models. We also claim that traces of personal work (as opposed to events in information systems) may contain information of another quality, although this information may be hidden between noise, hence making human intervention necessary.

## 3 Discovering information work practice

As indicated by the number and diversity of ISO9001 certifications in Europe [11], Business Process Management becomes increasingly important not only for large enterprises but also for small- and medium-sized ones. Thus far, business processes are primarily modelled by process experts and delivered as one integrated package to the end users. Recently, an evident trend in businesses is the demand on adaptivity, against both the volatile markets and rapidly changing customer requirements. The conventional business process modeling approach, therefore, is under increasing challenge. On the one hand, it might take tremendous effort to reach an agreement on a standard business process model. On the other hand, in order to ensure generality, exceptions and specialities will be at sacrifice. A business process modeled based on expert knowledge is not likely to reflect actual work practices accurately. The situation will not be alleviated by process automation. For instance, a Workflow-Management-System (WfMS) often leaves out details of the modeled activities so as to provide certain flexibility to the end users. Such "gaps" are then filled by personal task execution preference and particulars. In practice, such information normally hides in the trace of informal communications (e.g. emails) and personal task "diaries" (e.g. personal notes and logs), and in many cases get lost into the vast amount of available information.

In the rest of this paper, we will assume the existence of a grossly crafted process model which is treated as the "seed" for further refinement. Due to its knowledge intensive nature, business processes can normally be refined in two general directions: further development of process knowledge (in terms of activity order, granularity, etc.) and further development of process-related knowledge (in terms of supporting information of business processes). In this section, we present two scenarios in which the mining of informal work can contribute to the improvement of business process models.

### 3.1 Resource Recommendation

When performing a task, a person often consults resources, the selection of which is based on her personal skills, experiences or preferences. A good understanding of such information would allow us to refine the corresponding process models or help improving execution of process instances. Hence we wish to discover the set of local resources (documents as well as task collaborators) that people use to accomplish their tasks

When working on an assigned task with the help of task management tools such as KISSmir [18], a person can add resources to that task. In order to enable automatic resource provision, accomplished 'historical' task instances are analysed. Figure 1 shows how this might work: assuming that the current task of checking a certain certification already has an association with a resource $R1$, we can recommend further resources such as $R4$ by analysing historical data, e.g. the co-occurrence of $R1$ with other resources in past task executions.

In addition, we propose to consider the set of emails associated to a certain activity across all process instances. This is done by extracting from emails the attachments (document resources) and embedded links (web pages), as well as senders and receivers (people). We then simply count the frequency with which resources occur in these emails and recommend the most frequent ones whenever a user works on the given activity again. An analysis of co-occurrence of email recipients can additionally be leveraged for recipient recommendation during email composition (cf. [6]).

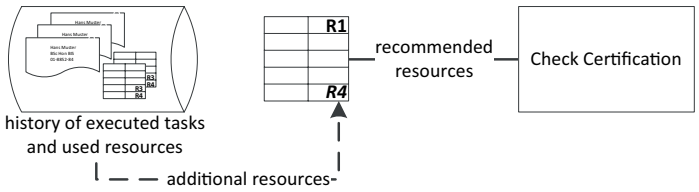


**Figure 1.** Resource recommendation based on historical data

### 3.2 Task Refinement

Email/task diary data contains valuable information about potential sub-structures of the business process that may not be reflected in the current process model. Such structures can be either implicit subtasks or context-based branches.

**Subtasks** Sometimes a process model is not fine-grained enough, i.e. its process activity might implicitly encapsulate various sub-activities that need to be performed (with or without a certain order) in order to accomplish a corresponding task. It might become necessary to differentiate and explicate such sub-structures when one has to collaborate with and/or delegate certain sub-tasks to others, or lift up the sub-structures to reflect them in the workflow, for better ICT support.

**Branches** Often, the need to perform a sub-task of a given task is not fixed and depends on the context wherein the task is carried out. For instance, the task of requesting an intern contract for a student might go along different routes depending on the student's nationality: contracts for foreign students might be more difficult and complicated to acquire than those for local students.

This is not always foreseeable until a concrete instance task calls for attention. These conditional branches should be reflected in the process model by decision points that check context attributes in order to select the most appropriate branches.

In both cases, we can expect to find corresponding evidence in email/task data, provided that the "seed" process model is sufficiently used, accumulating a large amount of data covering all possible alternatives.

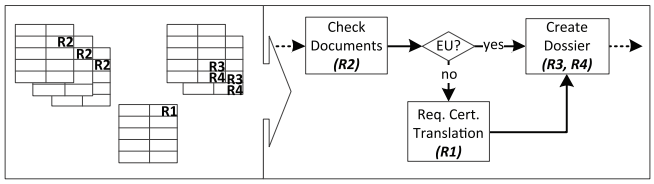


**Figure 2.** Task refinement based on clustering by resource usage

When using a task management system, users have the possibility of making subtasks explicit (creation of subtask entities, delegation), which can be exploited. However, often that is done in an implicit way, e.g. delegating work via email. Data mining methods can be employed to discover those sub-structures. Our assumption is that tasks/emails that belong to different branches and/or sub-activities will have different characteristics whereas tasks/emails that belong to the same subtask/branch will be substantially similar. By clustering all items that are associated with an activity (again across all process instances), we can expect that the resulting clusters will demonstrate branches and/or subtasks of the process. Figure 2 shows how tasks have been clustered based on attached resources, yielding three task clusters. We can now imagine that, as a result of inspecting the clusters, the two bigger ones form the basis of refining the task "Check documents" into one branch that uses resource $R2$ (which might be a document used for EU students) and another using resources $R3$ and $R4$ (for non-European students). Of course, automatic clustering is rarely fully reliable. Human analysis is necessary to quality-check the outcomes.

**Refinement based on task delegation** When executing KIAs, a participant can delegate parts of the task to colleagues. If this happens in a number of cases, it might be an indicator that the process model is not detailed enough and the knowledge-intensive task could be divided into several subtasks. We assume that a participant executing the task will document the delegation in the task description. We can analyse records of previous task executions to identify in which situations a task delegation happened. The criteria can be included in the process model as a decision point. By identifying subtasks, a KIA becomes better structured and evolves towards a knowledge-intensive process.

**Refinement based on task resources** Resource-based task refinement aims at refining the existing process model by using the stored information about a given task. With clustering, patterns of used resources are discovered. It is assumed that one cluster, i.e. one pattern of used resources equals a particular (sub-)task or branch. The existence of multiple patterns of resources associated with a single task, therefore, indicates the possibility of the process model refinement. The identified process model adaptations are then implemented to reflect such resource patterns / clusters. In order to determine whether a model refinement is appropriate, human intervention is necessary. Variables that define an identified pattern are presented to a human expert. The expert can also take a closer look at the individual resources (e.g. the contents of a text document) in a resource cluster to understand the rationale. This ensures that dependencies are discovered even if the system were not initially able to find them.

**Refinement based on emails** This kind of refinement resembles the previous one - all emails belonging to an activity (and across process instances) are clustered and the cluster representations are inspected by human experts to identify if they correspond to subtasks and/or branches within the activity. We consider two kinds of feature vectors to represent emails for clustering. The first kind is based on communication partners, i.e. senders and recipients of emails; here, we assume that different subtasks and/or branches of an activity require the interaction with different kinds of people.

On the other hand, even when communicating with the same people within a task, different aspects (i.e. subtasks) may be addressed, which may be detected by analysing the email's unstructured parts. Therefore, the second kind of feature vectors is built from the text of the email's subject and body. After automatic analysis is finished, the human expert is provided with a meaningful description of the clusters. For instance, an expert will need to know the number of emails in a cluster, the categories (i.e. roles) of communication partners together with their frequency, the most prominent keywords, derived via term extraction from the emails' subjects and bodies and a selection of example email subjects. This should allow her to see what the emails within a cluster have in common.

## 4 Experiments

We employed well established data mining and data clustering algorithms when implementing the informal work based process model refinement approach. The system is still under development. A preliminary evaluation, however, proved the practical value of our approach.

### 4.1 Implementation

In this section, we focus more on the emails which present a real challenge due to their unstructured nature. Task diaries, on the other hand, can be processed in a more straightforward way.

**Mapping to activity** Collecting traces of informal work related to the activities of a business process requires mapping both task instances and emails to these activities. Mapping task instances to activities is straightforward with the help of the KISSmir system [18]—the task instances are assigned by the workflow engine and thus by definition belong to a certain activity of the underlying business process.

The real challenge for assigning emails to activities is due to a lack of direct association. As suggested in [15], mapping can be achieved in two steps a) mapping emails to a process instance and b) mapping emails to activities or tasks within that process instance. Van der Aalst and Nikolov [15] assume the existence of annotations of emails which are added either manually by end users or by a process-aware information system. They acknowledge that email annotation regarding the activity will not always be available and propose that this classification can alternatively be obtained automatically by matching the name of the task against the email subjects, where a partial match is acceptable. This approach presumes that the subject of each email that belongs to a certain activity of a business process will always contain the name of that activity. This is, however, only guaranteed if the email is written manually with significant attention or generated automatically by a task management system's email functionality. It may become less useful in other cases. We propose to utilise a classical machine learning approach that can (semi-)automatically assign emails to activities. This is done as follows:

1. *Acquiring training data* The users are asked to annotate a certain number (say 100) of emails manually as training data. For the first step, i.e. the identification of the process instance to which an email belongs, we follow the approach in [15] and let the user choose among a number of criteria, e.g. a contact linked to an email, to select all emails belonging to a process instance.
2. *Feature extraction* From each email in the training data and the set of remaining emails, we extract the email addresses of the sender and the recipients, the time-stamp, attachments and the text of the subject and body. We use these fields as features - where the text parts (subject and body) have to be broken into words that will constitute the features.
3. *Classification* We use standard classifiers from the machine learning literature to learn a model from the training data and apply it to unclassified emails.

Since the text-based features will result in a high-dimensional feature space and will thus outweigh the other features when combined into one feature vector, we propose to proceed with fusion methods that have been explored in the area of multi-media content, where low-dimensional image features have to be combined with high-dimensional textual features, *cf.* [3]. These methods either define and combine separate kernels (i.e. similarity functions) for the different features or build different classifiers for the different feature sets and combine these with a meta-classifier. In this work, we focus on the actual extraction

of process-relevant information from tasks and emails and therefore have not implemented/evaluated this automated mapping. A closer analysis will be the subject of future work.

**Communication partner categories** We classify communication partners in email exchange according to their job function or role in the business process. Such kind of information can be found in the employee directory. Representing this information formally, for example in an enterprise ontology allows inferring the required information of job function, role or project etc. automatically [12]. In our approach, end users can define categories of communication partners by specifying the category names followed by the keywords indicating what method should be used to filter the list of email contacts. Currently, three methods are implemented:

**LIST** enumerating a list of email addresses that together make up the category, *Process Owner = LIST(hans-friedrich.witschel@sap.com)*.
**CONTAINS** specifying a string (e.g. the domain) that all member email addresses must contain, *SAP = CONTAINS(@sap.com)*
**NOMATCH** reserved for all email addresses that are not qualified in the previous categories, *External = NOMATCH*

In order to classify a given email address $E$ using such a category definition, the email processing engine will start to match $E$ against categories defined via LIST, then - if no match was found - against the CONTAINS definitions and finally assign it to the NOMATCH category.

**Extracted features** After the assignment of task instances and emails to process and activity instances and categorisation of contacts in email communications, further information will be generated automatically, resulting in the following list of features associated to each email:

- Process instance ("Student Hiring of ⟨*Charly Brown*⟩")
- Activity instance ("Prepare Interview with ⟨*Charly Brown*⟩" in our example process)
- Type (simple email vs. meeting request)
- Subject
- Body
- Sender (name, email address and communication partner category)
- Recipients (same as sender)
- Time-stamp indicating when the email was sent
- The list of attachments of the email.

### 4.2 Experimental setup

In order to evaluate the methods outlined above, we performed some experiments with email data corresponding to a student recruitment process that researchers

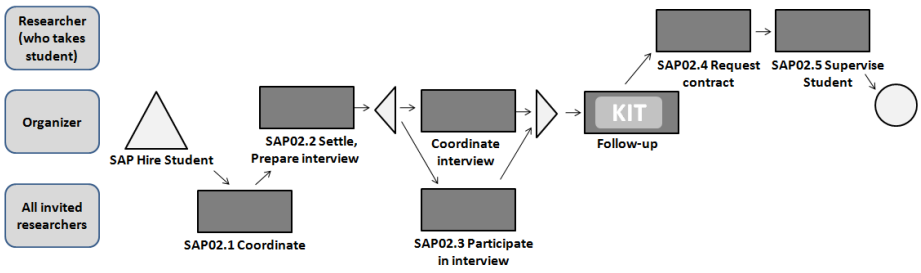


**Figure 3.** Student hiring process at SAP Research

at SAP Research perform when they want to hire intern or bachelor/master thesis students to work in research labs.

Figure 3 shows the most relevant parts of this process (as the seed process) from a researcher's perspective.

The data we used for the experiments was extracted from the Inbox and calendar of one of the authors by means of a script that outputs all emails, appointments and meeting requests that contain a user-specified string (e.g. part of the student's name) up to a certain date - namely one week after the date when the student started working at SAP Research or the date when his/her application was rejected. It represents 9 cases of student hiring undertaken by the owner of the Inbox (i.e. 9 process instances) and consists of approximately 350 emails after some manual cleaning.

The emails were processed as discussed in the previous sections. They are manually associated to the activities of the process in Figure 3. Then, the process owner (i.e. the owner of the email Inbox) defined 8 categories of communication partners, namely process owner, teammates, team assistants (responsible for collecting contract requests), HR (the human resources department of SAP), manager, applicant, SAP (all SAP employees) and external parties which represent the primary categories of communication partners in this process. Overall, there are about 60 emails representing the activity of screening and selecting applicants, 20 emails about coordinating with colleagues, 30 emails preparing the interview, 10 emails representing the actual interview, 40 emails that follow up on an interview, 90 emails about contract requests and about 100 emails representing supervision of the student.

### 4.3 Results

In our experiment, we concentrated on the task refinement part since our data set was not large enough to enable resource recommendation. For task refinement, the emails were divided into 7 subsets, each corresponding to one of the seven activities "screen and select applicants", "coordinate", "settle and prepare interview", "conduct interview", "interview follow-up", "request contract" and "supervise student" of the student hiring process.

| Nr | Size | Keywords (Weight) | Senders ($f$) | Recipients ($f$) | Example subjects ($f$) |
|---|---|---|---|---|---|
| 1 | 13 | look (13.19), well (12.96), algebra (10.26), thing (10.26), winner (10.26), get (10.26), linear (10.26), list (10.03), seem (7.07), diploma thesis (5.0), linear algebra (4.0), extern referenzcode (4.0), favorite subject (2.0), thesis student (2.0), reference code (2.0), mix impression (2.0) | teammates (13) | process owner (13), teammates (13) | AW: Students (4), RE: Students, suggestion (2), AW: Our list of students (2), WG: Student Applications (2), RE: Students (2), ... |
| 2 | 9 | experience (13.86), look (13.19), think (11.23), relevant (9.32), skill (8.25), decide (7.9), topic (7.9), send (7.49), suggestion (7.03), work (6.59), student application (5.0), programming skill (2.0) | process owner (9) | teammates (16) | Students (2), Students, suggestion (2), ... |
| 3 | 7 | diploma thesis (24.0), diploma (20.43), internship (17.33), application (14.29), karlsruhe (13.86), extern (12.43), thesis (12.32), referenzcode (11.79), please (11.68), extern referenzcode (10.0), mature (8.8), deadline (8.52), application process (7.0), whole application (7.0), thesis student (7.0), reference code (7.0) | process owner (7) | TAs (7), teammates (7) | RE: Student Applications (5), RE: Student applications CW 27 (1), RE: Student applications CW 47 (1) |
| 4 | 7 | diploma thesis (12.0), diploma (8.67), guy (7.69), sure (6.52), talk (6.52), anybody (6.52), wait (6.52), tomorrow (6.52), extern referenzcode (4.0), application process (2.0), whole application (2.0), mature team (2.0) | TAs (5), teammates (2) | process owner (7), SAP (2) | $NAME$ and $NAME$ (2), AW: Student Applications (2), ... |
| 5 | 7 | diploma thesis (27.0), diploma (20.43), extern (14.92), referenzcode (14.14), internship (12.48), extern referenzcode (12.0), thesis (11.47), please (10.51), application (10.21), karlsruhe (9.9), deadline (7.21), application process (6.0), whole application (6.0), thesis student (6.0) | TAs (7) | SAP (21), process owner (1) | Student Applications (6), ... |
| 6 | 4 | experience (30.5), grade (26.07), little (22.82), program (20.19), claim (19.55), mark (15.8), background (14.26), maybe (14.26), work (12.45), program skill (3.0), background somewhat (2.0), grade check (2.0), basic knowledge (2.0) | process owner (4) | empty (4) | [BLOCK] check out students (2), [BLOCK] check student applications (1), [BLOCK] catch up (1) |
| 7 | 3 | thesise (9.89), internship (8.32), external (7.9), hourly (5.71), please (5.26), thesis (5.1), application (5.1), above (4.95), diploma (4.33), deadline (3.93), application process (3.0), whole application (3.0), thesis student (3.0), reference code (3.0), new application (3.0) | TAs (3) | SAP (9), TAs (3) | Student applications CW 27 (1), Student applications (1), Student applications CW 47 (1) |

**Table 1.** Clusters based on communication partner categories for activity "Screen and select applicants".

As a next step, each email was represented as a feature vector. As outlined in Section 4.1, we chose two kinds of feature vectors, one based on communication partner categories, the other based on the text from subject and body of the emails. This means that emails attachments and embedded links were not considered in this set of experiments.

| Nr | Size | Keywords (Weight) | Senders ($f$) | Recipients ($f$) | Example subjects ($f$) |
|---|---|---|---|---|---|
| 1 | 45 | permit (34.81), german (23.28), salary (22.09), regard (20.78), obtain (20.53), student (18.5), internship (18.38), restrict (17.68), hour (17.68), number (17.31), contract request (14.0), start date (10.0), phone number (5.0), student contract (5.0), work duration (3.0), full time (2.0) | process owner (19), external (6), HR (5), teammates (5), TAs (4), applicant (4),... | proces owner (25), teammates (13), TAs (5), ... | [BLOCK] catch up (3), Re: Your master thesis (3), RE: Praktikumsplan $NAME$ (3),... |
| 2 | 9 | description (173.29), next (112.16), task (91.71), call (88.46), phone (85.39), phone call (60.0), e mail (54.0), project (48.9), process (46.96), e (46.63), period (42.44), need (41.07), start date (18.0), contract period (12.0), current performance (12.0), performance record (12.0), subject description (11.0), contract status (10.0), mature project (9.0) | process owner (5), applicant (4) | applicant (5), process owner (4) | RE: contract status (3), ... |
| 3 | 8 | name (141.54), internship (46.78), family (46.65), use (46.65), offer (45.54), first (44.48), last (36.68), father (31.1), successful (28.09), submission (25.28), last name (24.0), first name (23.0), family name (21.0), father name (14.0), paper work (8.0), request contract (8.0), start date (8.0), mature project (8.0) | process owner (6), applicant (2) | teammates (9), applicant (5), process owner (2) | RE: Your internship (5), Re: Your internship (2), ... |
| 4 | 6 | integration (106.64), thesis (106.49), supervisor (70.18), schema (66.65), need (60.32), case (56.08), social (53.32), month (51.76), position (50.79), e mail (36.0), thesis contract (19.0), service integration (18.0), business schema (18.0), thesis position (18.0), month thesis (12.0), shipment address (12.0) | process owner (4), external (2) | teammates (4), TAs (2), process owner (2), external (2) | Re: PS: Internship (2), FW: PS: Internship (2), RE: PS: Internship (2) |
| 5 | 6 | e mail (24.0), studiengang (22.35), betreuung (15.76), pdf (14.9), inhaltliche (14.9), hier (13.28), professor (13.28) | process owner (4), HR (1), applicant (1) | TAs (2), process owner (2), HR (2),... | RE: Details ber Diplomarbeit (1), ... |
| 6 | 3 | integration (53.32), thesis (33.69), schema (33.32), social (26.66), position (25.4), internship (18.38), service (18.19), case (17.41), business (16.02), service integration (9.0), business schema (9.0), thesis position (9.0), thesis contract (8.0), month thesis (5.0), shipment address (5.0) | process owner (2), external (1) | teammates (3), external (2), process owner (1) | RE: PS: Internship (2), Re: PS: Internship (1) |

**Table 2.** Clusters based on email text for activity "Request contract".

For the first kind of feature vectors, if $c_1, ...c_n$ is the set of communication partner categories, then an email is formally represented by a vector $\boldsymbol{e} = (r_{c_1}, ..., r_{c_n}, s_{c_1}, ..., s_{c_n})$ where $r_{c_i}$ is the number of recipients of the email belonging to category $c_i$ and $s_{c_i}$ is the number of senders (0 or 1) of the email belonging to category $c_i$.

For the second kind of vectors, the text of the emails was tokenised into one-word units (terms), which were weighted using *term frequency-inverse document frequency* (tf.idf). Thus, if $t_1, ..., t_n$ is the set of distinct one-word units that occur in any of the emails of the entire corpus, then an email is represented by

a vector $\boldsymbol{e} = (w_1, ..., w_n)$ where $w_i$ denotes the tf.idf weight of term $t_i$ for the current email.

Using the Weka machine learning library[2], the vectors were clustered using the expectation maximisation algorithm and employing cross validation to determine the number of clusters automatically.

Table 1 displays the clustering results for the activity "screen and select applicants", using feature vectors based on communication partner categories (where $f$ gives the frequency values). The keywords and phrases were extracted using the TE (term extraction) module of the ASV toolbox [4]. They carry weights as computed by this module; the other characteristics of the clusters (categories of senders and recipients, subjects) are represented together with their frequency of occurrence in the cluster.

For instance, the first line of Table 1 describes a cluster consisting of 13 emails. All of these emails were sent by one of the teammates of the process owner; and they were all received by the process owner and other teammates (for recipients, this table does not distinguish between *To* and *CC* fields of an email). Some subjects of the emails in this cluster are given in the last column of the table (e.g. the subject "AW: Students" occurred 4 times) and the most important keywords extracted from the emails of the cluster ("look", "well", "algebra", ...) are given in the third column.

Table 2 presents clusters for the activity "Request contract", derived with feature vectors based on the text from emails' subject and body.

### 4.4 Discussion

**Interpretation of example clusters** For the clusters in Table 1, a human can easily derive an interpretation of clusters in terms of sub-activities by looking at communication partners and keywords:

- Clusters 5 and 7 are very similar and show communication from the team assistants towards all colleagues (including the process owner). They announce new student applications and ask for feedback.
- Cluster 6 contains private appointments only, where - as the keywords suggest - the process owner has taken notes about the applicants' qualifications.
- Clusters 1 and 2 consist of communication between the process owner and his teammates; keywords such as "linear algebra" and "programming skill" signal that the goal of this communication is the discussion of the applicants' qualification and hence to reach an agreement of whom to invite for an interview.
- In Cluster 3, the process owner communicates with team assistants; as can be seen from the subjects of emails in the last column, these emails contain feedback about applicants.

[2] http://www.cs.waikato.ac.nz/ml/weka/

– Cluster 4 is less clear in its interpretation; it consists mainly of emails sent by the team assistants to the process owner. A drilldown helped to understand that most of these serve to clarify questions around further procedure, management approval and deadlines.

From this analysis, we can quickly suggest to divide the activity "screen and select applicants" into sub-activities "receive notification of new applications", "screen applications", "discuss applications with teammates", "give feedback to team assistants" and "if in doubt, clarify further procedure". Note that the labeling of clusters and the final suggestion of how to adapt the process model is still a manual effort.

As Table 2 shows for the activity "request contract", the analysis is not always that easy. An interpretation could be as follows. It is evident that Cluster 3 shows the communication between the process owner and the applicant, trying to obtain the necessary data ("start date", "last name") to fill in the contract request form. Cluster 2 is less well defined, but seems to be mainly about communicating the status of the contract request. Clusters 4, 5 and 6 seem to be a mixture of planning work together with the student and administrative issues (e.g. answering the applicant's questions about university supervisors). Cluster 1 is by far the largest and consists of a mixture of topics, an important one being the exact modalities of the contract ("salary", "hour", "work duration", etc).

This analysis could suggest a subdivision of the activity into "get student data for filling request form", "clarify contract modalities", "answer student questions" and "check and communicate contract status".

**Analysis of the approach** Although we cannot show all the cluster results here for space reasons, we would like to summarise the insights that we gained through qualitative analysis of the clustering data. In future work, it will also be interesting to apply quantitative measures (such as precision and recall) by comparing clustering results to previously defined gold standards. However, in this work the data set was too small to allow for meaningful quantitative results.

We found that in general, clustering with feature vectors based on communication partner categories resulted in more easily interpretable and meaningful clusters than when using text-based features. In addition, for some activities, such as "request contract" (see above), clusters are more imbalanced in size, more difficult to interpret and less revealing. This is the case regardless of whether we use communication partner categories or text-based feature vectors. However, for 5 of the 7 activities, the clusters are very clear and easy to interpret.

From the authors' experience, we could identify cases where the clustering fails to reveal interesting branches that could help to subdivide the activity, e.g. the differentiation between intern and thesis contracts and European and non-European students in the "request contract" activity. The reasons for such failures were twofold:

– Imbalance and sparseness problem: some interesting branches (e.g. non-European students) are not represented by enough process instances in the data in order to be detected.

– Noise problem: interesting differentiations such as between internship or thesis are buried under more obvious (and hence less interesting) ones.

Of course, some of these problems may be resolved by using a bigger data set; the qualitative results presented here can thus only show a general direction and highlight some interesting aspects that still need to be quantified – which is our goal in future research.

## 5 Conclusions

Conventionally, business processes are modeled manually by process experts. With the growing demand on flexibility and agility, process mining and collaborative approaches start to gain attention as low-cost and low-overhead alternatives. In this paper, we proposed a framework that takes advantage of informal work practice in refining manually crafted process models. This is based on our observation that i) gaps between predefined models and day-to-day work practice is inevitable and thus local adaptation is necessary; ii) in many cases local adaptation carries informal characteristics and is not transparent to an enterprise information system and thus most of such valuable knowledge gets lost within the vast amount of information available to an employee; and iii) local adaptation can be leveraged to reflect work practice at the model level.

The proposed approach focuses on two types of informal data of work practice, i.e. emails and personal task diaries. Different from apparently similar approaches, we reduce the overhead of manual annotation with the help of standard data mining / clustering algorithms. Hidden process patterns discovered from the informal work data are then used to fine-tune a "seed" process model, improving the model to reflect real-life work practice. We experimented the idea in a proof-of-concept implementation. The preliminary evaluation with emails from one of the co-authors has confirmed our intuition and the practical value of our approach. Further evaluation in a larger scale and with high diversity is forthcoming. Apart from optimising the learning algorithms, we aim to conceive other application scenarios that leverage the hidden "treasure" of informal work.

## Acknowledgements

This work is supported by the European Union IST fund through the EU FP7 MATURE Integrating Project (Grant No. 216356).

## References

1. A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Toward a technology for organizational memories. *IEEE Intelligent Systems*, 13(3):40–48, 1998.

2. R. Agrawal, D. Gunopulos, and F. Leymann. Mining process models from workflow logs. In Hans-Jörg Schek, Gustavo Alonso, Felix Saltor, and Isidro Ramos, editors, *Advances in Database Technology EDBT'98*, volume 1377 of *Lecture Notes in Computer Science*, chapter 31, pages 467–483. Springer, Berlin; Heidelberg, 1998.
3. S. Ayache, G. Quénot, and J. Gensel. Classifier fusion for SVM-based multimedia semantic indexing. In *Proceedings of ECIR'07*, ECIR'07, pages 494–504, Berlin; Heidelberg, 2007. Springer.
4. C. Biemann, U. Quasthoff, G. Heyer, and F. Holz. ASV Toolbox – A Modular Collection of Language Exploration Tools. In *6th Language Resources and Evaluation Conference (LREC)*, 2008.
5. S. Brander, K. Hinkelmann, A. Martin, and B. Thönssen. Mining of agile business processes. In *Proceedings of the AAAI Spring Symposium on AI for Business Agility*, 2011.
6. V. R. Carvalho and W. W. Cohen. Recommending Recipients in the Enron Email Corpus. Technical Report CMU-LTI-07-005, Carnegie Mellon University, Language Technologies Institute, 2007.
7. C. Di Ciccio, M. Macella, M. Scannapieco, D. Zardetto, and T. Cartacci. Groupware Mail Messages Analysis for Mining Collaborative Processes. Technical report, Sapienza, Università di Roma, 2011.
8. J.E. Cook and A.L. Wolf. Discovering models of software processes from event-based data. *ACM Trans. Softw. Eng. Methodol.*, 7(3):215–249, July 1998.
9. S. Dustdar, T. Hoffmann, and W.M.P. van der Aalst. Mining of ad-hoc business processes with TeamLog. *Data and Knowlegde Engineering*, 55(2):129–158, 2005.
10. D. Feldkamp, K. Hinkelmann, and B. Thönssen. KISS: Knowledge-Intensive Service Support: An Approach for Agile Process Management. pages 25–38. 2007.
11. International Organization for Standardization. ISO Survey 2009. Available at http://www.iso.org/iso/survey2009.pdf, accessed in March 2011.
12. K. Hinkelmann, E. Merelli, and B. Thönssen. The role of content and context in enterprise repositories. In *Proceedings of the 2nd International Workshop on Advanced Enterprise Architecture and Repositories-AER*, 2010.
13. A. Martin and R. Brun. Agile Process Execution with KISSmir. In *5th International Workshop on Semantic Business Process Management*, 2010.
14. D. McDauley. In the face of increasing global competition and rapid changes in technology, legislation, and knowledge, organizations need t o overcome inertia and become agile enough to respond quickly. Organizational agility might indeed be one. In Keith D. Swenson, editor, *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*, pages 257–275. Meghan-Kiffer Press, 2010.
15. A. Nikolov and W.M.P. van der Aalst. EMailAnalyzer: An E-Mail Mining Plug-in for the ProM Framework, 2007.
16. Wil M. P. van der Aalst. Exploring the CSCW spectrum using process mining. *Advanced Engineering Informatics*, 21(2):191–199, April 2007.
17. Wil M. P. van der Aalst and A. J. M. M. Weijters. Process mining: A research agenda. *Comput. Ind.*, 53(3):231–244, 2004.
18. H.F. Witschel, B. Hu, U.V. Riss, B. Thönssen, R. Brun, A. Martin, and K. Hinkelmann. A Collaborative Approach to Maturing Process-Related Knowledge. In Richard Hull, Jan Mendling, and Stefan Tai, editors, *Business Process Management*, pages 343–358, Berlin, Heidelberg, 2010.