

Mining of Agile Business Processes

Simon Brander¹, Knut Hinkelmann¹, Andreas Martin¹, Barbara Thönsen^{1,2}

¹ University of Applied Sciences Northwestern Switzerland FHNW

Institute for Information Systems

Riggenbachstrasse 16, 4600 Olten, Switzerland

{simon.brander|knut.hinkelmann|andreas.martin|barbara.thoenssen}@fhnw.ch

² University of Camerino

Dipartimento di Matematica e Informatica

Via Madonna delle Carceri 9, 62032 Camerino (MC)

{barbara.thoenssen@unicam.it}

Abstract

Organizational agility is a key challenge in today's business world. The Knowledge-Intensive Service Support approach tackles agility by combining process modeling and business rules. In the paper at hand, we present five approaches of process mining that could further increase the agility of processes by improving an existing process model.

Introduction

Organizational agility can be regarded as a key challenge for successful enterprises. McCauly (2010) defines an agile organization as one that can sense opportunity or threat, prioritize its potential responses, and effect efficiently and effectively. External influencers (assessed as opportunities and threats) as well as internal influencers (assessed as either strengths or weaknesses) require agility and the reaction to them.

By comparing an agile organization with a sports team, McCauly (2010) makes clear that agility requires both ad-hoc reactions on what is happening in a game (by sensing, prioritizing, and acting appropriately) and reactions in the long run (by revising their playbooks).

For an organization to be agile, it continuously has to improve its business processes on the one hand. On the other hand, the business processes have to be flexible enough to allow reactions in specific situations. This means that business processes with a fixed structure are not appropriate for knowledge-intensive work.

In (Witschel et al. 2010) we presented our approach for agile process modeling and execution. In this paper the focus will be on supporting the continuous improvement of agile business processes by the application of process and data mining technologies.

We give a short repetition of modeling agile processes in the next section. The subsequent chapter presents five scenarios for mining process and task data.

Agile Process Modeling and Execution

Current business process management has to deal with the emerging demand of being agile. The conventional approaches are seen as overstretched in several situations where agility is needed “[...] due to the lack of flexibility and the amount of overhead required for predefined process models” (Witschel et al. 2010).

The Knowledge-Intensive Service Support (KISS) deals with the demanding requirement of being agile (Feldkamp, Hinkelmann 2007) by combining conventional business processes with (business) rules. KISS addresses the

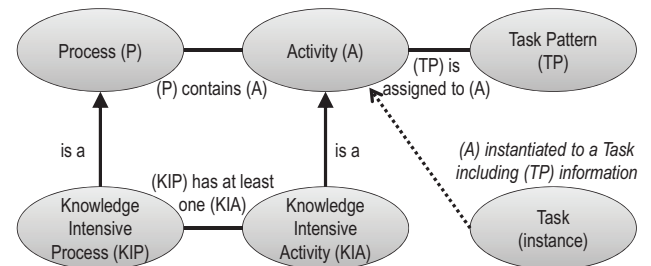


Figure 1: Concepts of KISSmir (based on (Feldkamp & Hinkelmann, 2007; Martin & Brun, 2010; Witschel et al., 2010))

common problems of dynamic or agile processes, such as exceptional situations, unforeseeable events, unpredictable situations, high variability, or highly complex tasks (Feldkamp, Hinkelmann 2007).

The “collaborative approach to maturing of process-related knowledge” introduced by Witschel et al. (2010) combines the knowledge intensive processes (KIP) of KISS with task patterns. Figure 1 illustrates the following definitions. “A knowledge intensive process (KIP) can be regarded as a collection of activities building the ‘skeleton’ of a business process, some activities of which can be knowledge intensive (called ‘KIA’)” (Witschel et al. 2010). Modeling knowledge-intensive processes corresponds to ad-hoc processes of BPMN where the process flow is not modeled in detail but activities can be executed in any order and any frequency (White & Miers 2008). “Whereas ordinary activities are always executed (i.e., in every process [instance]), KIAs are optionally executed depending on information specific for the certain process instance” (Witschel et al. 2010). This information can for example be application data, process data, functional data, or further information about needed resources. KISS supports the execution of knowledge-intensive processes by planning components and business rules (Feldkamp, Hinkelmann 2007).

Instances of activities are called tasks as described in Witschel et al. (2010) and it does not matter if the activity is knowledge-intensive or not. “A task is a definition of a particular item of work that specifies the requirements and the goal of this work” (Witschel et al. 2010). A task pattern is like a template of a task that contains all relevant information and experience for the task execution. Examples are information about resources that can be used, sub-tasks that should be performed or problem / solution information, which enable users to share their experience (Riss, Rickayzen, Maus, van der Aalst 2005). According to Witschel et al. (2010) it can be seen as a “[...] mediator between process modeling and individual task execution”.

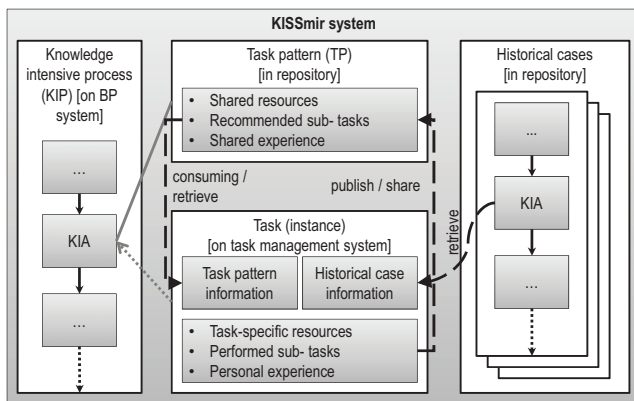


Figure 2: The KISSmir system (based on (Martin & Brun, 2010; Witschel et al., 2010))

To point that out, we have a one-to-one relationship between an activity of an agile business process and a task pattern in the introduced approach.

The KISSmir system (Martin, Brun 2010), developed during the EU-funded project MATURE, is a process handling and information sharing tool and implements the principles of knowledge-intensive processes and tasks / task patterns (Witschel et al. 2010). “It focuses on the modeling of knowledge intensive business processes and aims at the following goal: The individual knowledge and experience used when carrying out a process should be shared with other employees in an organization” (Martin, Brun 2010). This can be seen in Figure 2. The heart in the KISSmir system is the execution of a (knowledge-intensive) process. A task pattern that contains task-specific information (e.g. general task information, shared resources, recommended subtasks, shared experience) is assigned to each activity. If the activity should be performed, it is instantiated to a task using the task pattern. The task contains task-specific resources provided by the process or the user, subtasks or personal experience etc. The user or assignee of the task has the possibility to add additional information to the task. If he does so, he is asked if he wants to share this information by publishing or adding it to the task pattern. KISSmir also gives the user the possibility to access or retrieve historical cases that are similar to the one at hand. The mining approaches of agile business processes (shown in the next section) rely on the described KISSmir system.

Mining of Agile Process Data

KISS and KISSmir support one aspect of agility - the flexible reaction on specific situations. In this section we deal with the second aspect of agility - the continuous adaption of business processes in a dynamic environment. This is achieved by applying process mining in order to suggest potential process improvements.

Van der Aalst and Weijters (2004) use the term process mining “for the method of distilling a structured process description from a set of real executions”. According to Van der Aalst et al (2007), one can further distinguish three perspectives in process mining. The *process perspective* concerns “the ordering of activities”, which is represented by the control flow. The *organizational perspective* focuses on the involved people and their relations. The *case perspective* covers case properties, such as the execution path or instance-specific values.

We illustrate the mining approaches by using the admission process for a Master Program at the University of Applied Sciences Northwestern Switzerland as an example. The process is illustrated in Figure 3. Each rectangle represents an activity (e.g. “Check

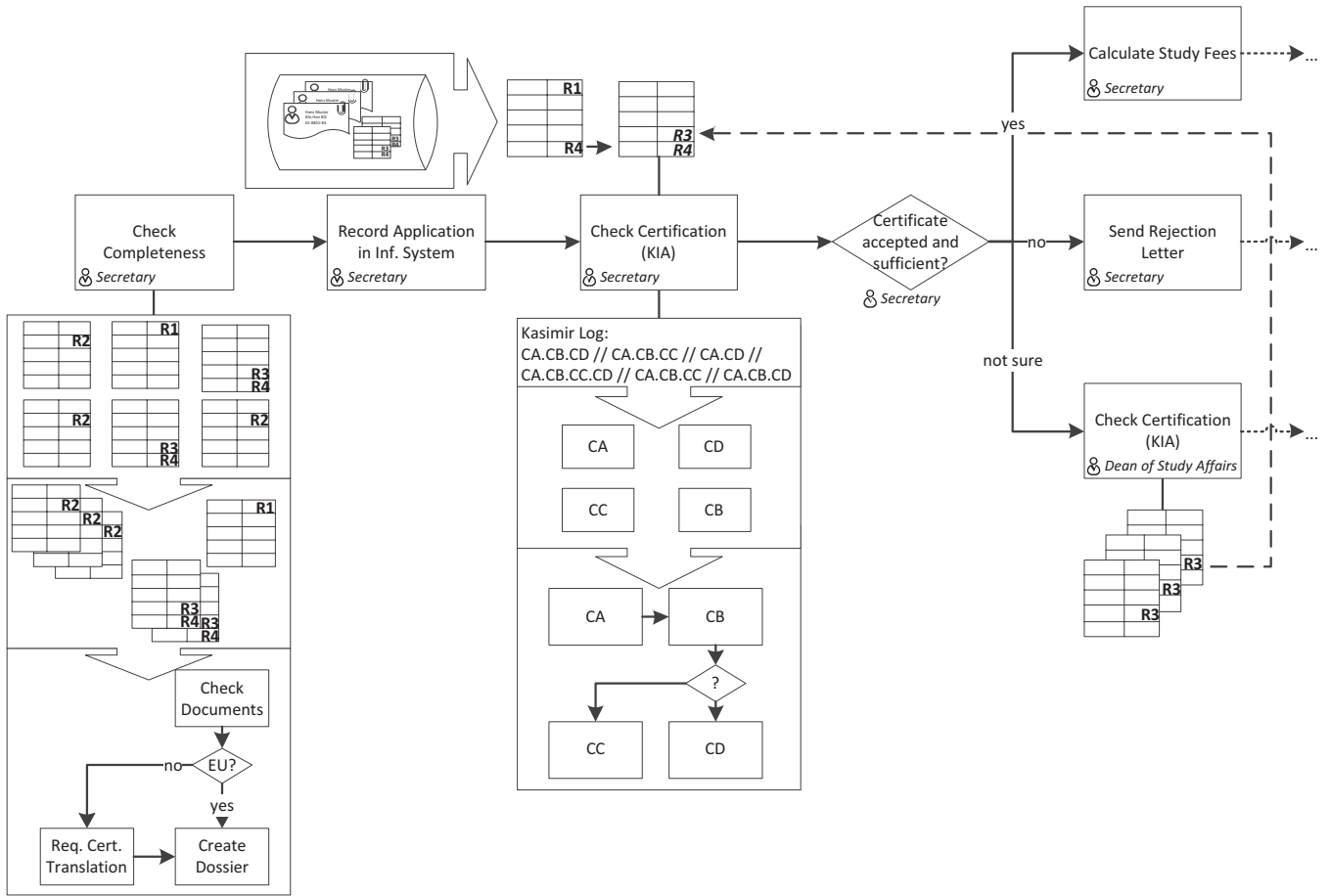


Figure 3: Section of the matriculation process at the School of Business (FHNW)

Completeness”) and the responsible role (e.g. “Secretary”). The tables connected to an activity represent a task. The R_i in the tables represent resources suggested by the KISSmir system and also resources actually used by the participant during execution. Examples of such resources are documents, websites, or people.

In the following we show five process mining approaches. These approaches are innovative because they utilize the process agility as well as the used resources attached to tasks and task patterns.

In a nutshell, the Subtask Discovery, Determine Subtask Execution Order, and Resource-Based Task Refinement approaches cover the process perspective, while the case perspective is part of the Resource Recommendation and Retrospective Resource Recommendation approaches. The organizational perspective is not directly addressed.

Resource Recommendation

The first process mining approach deals with the case perspective. When someone works on an assigned task, she/he can add resources to the task instance. When executing a task, different participants might use different

resources depending on the concrete case. One reason for this is that people executing the task can have different experiences and skills. Another cause is the fact that each case is (at least to some extent) unique. Therefore, there is only a small chance that one set of resources covers all cases.

The KISSmir system stores all executions of a given task. Based on the case data (the independent variables) the process miner attempts to discover potentially useful resources (the dependent variable). Once such sets of case

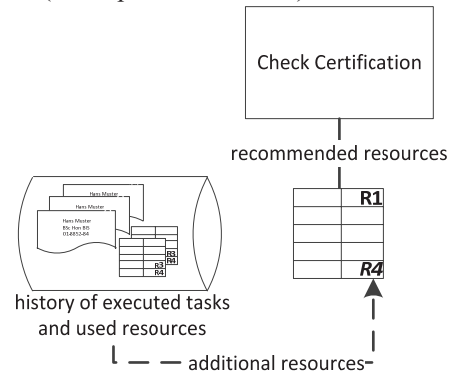


Figure 4: Additional resources based on historical case analysis

data and resources are found, the system makes use of them by suggesting potentially useful resources in future executions of the task.

Assume that in the “Check Certification” activity of or sample process, a participant found a website listing all the accepted university programs of some foreign countries. He adds this website to the task description. If in future a case of an applicant from one of these countries is to be dealt with, the system can automatically suggest the website thus raising awareness by the users that did not know of this resource.

Retrospective Resource Recommendation

When decisions about applications have to be made, we often find the following three types of cases: There are obvious cases that can immediately be rejected or accepted by a clerk or assistant. However, there is a third group of cases that require detailed investigation by an expert. This investigation is expensive and thus it should be avoided.

The mining approach for learning decision criteria analyses the results of the detailed investigation trying to identify the decision criteria or at least finding those resources the expert used to support his decision making.

If in future similar cases the clerk is made aware of this resource, a detailed investigation by the expert could be avoided.

Retrospective resource recommendation is quite similar to the resource recommendation as described in the previous subsection. Again, the use of resources is analyzed to suggest potential resources in future cases. However, the recommendations are not attached to the task in which the resources was used but to a previous task.

In the example process, a secretary might not be sure whether a certificate can be accepted or not. The dean of the appropriate study program then has to decide whether the degree satisfies the program requirements or not. The dean might use resources that the secretary is not aware of to come to a decision. This could, for example, be a document on the network drive that contains a list of universities in a country and the various earlier decisions. If the secretary can decide future cases by himself using this list, the expensive involvement of the expert – in the given case the dean – can be reduced.

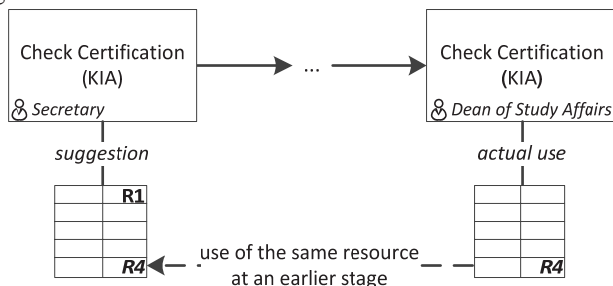


Figure 5: Resources used by experts recommended at earlier stage to normal performers

Resource-Based Task Refinement

The resource-based task refinement can be associated with the process perspective. As the previous suggestion, the resource-based task refinement also uses the fact that different resources are used to complete a task. The goal of this approach is to refine the existing process model. This is achieved by using the stored information about executions of a given task.

In the analysis phase, patterns of used resources are discovered. In the following refinement phase, it is assumed that one pattern of used resources equals a particular task. Different patterns of used resources for one task indicate a potential opportunity to refine the process model. This is done in the refinement phase: It consists of making sense of the found patterns. In order to determine whether a model refinement is appropriate, variables that could define whether an identified pattern applies to the case are presented to a human expert. The expert also takes a closer look at the resources (e.g. the contents of a text document) in a resource pattern to make more sense of the patterns and the respective triggers. This ensures that dependencies are discovered even if the system were not initially able to find them. The final step is the process refinement phase, in which the identified process model adaption is implemented. Resource patterns are reflected in adjusted or additional tasks, and gateways represent identified triggers.

If a candidate obtained her Bachelor’s degree from a university outside of the European Union, the secretary might, for example, use a specific website of the Swiss government to check whether she needs to provide additional documents, such as a certified English attestation that states the equivalence of the degree in the Bologna system. A consequence of analyzing the used resources might be that the task “Check Completeness” (of an application) can be concretized into a gateway element with a root for holders of European diplomas and a second root for all other holders with the additional task “Request Certified Translation”.

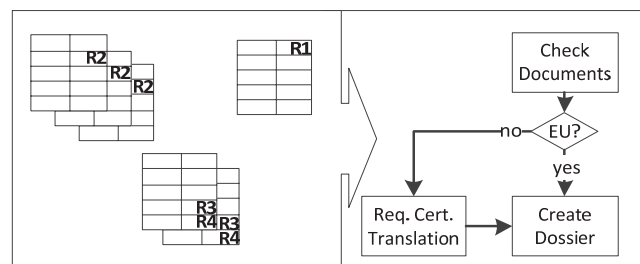


Figure 6: Task refinement based on use of resources

Subtask Discovery

When executing knowledge-intensive activities, it may happen that a participant delegates parts of the task to a colleague. If this happens in a number of cases, this 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 a task records the delegation in the task description. The subtask discovery approach analyses records of previous task executions to identify in which situations a task delegation happened. The criteria can be included as a decision point into the process model. By identifying subtasks, a knowledge-intensive activity gets more structured and evolves towards a knowledge-intensive process.

In the sample process this can happen if in order to validate the qualification a reference persons is contacted. This activity can be added as a subtask.

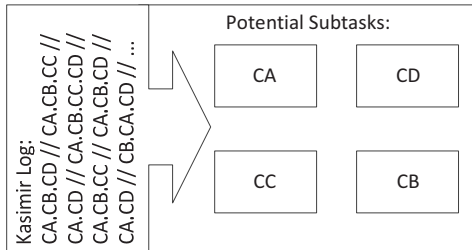


Figure 7: Subtask discovery based on previous subtasks

Determine Subtask Execution Order

As described above, the KISS part of KISSmir allows to model knowledge-intensive processes where the order of activities is not defined and each activity can be executed several times.

The Determine Subtask Execution Order analyses the actual execution order of subtasks in a knowledge-intensive process. If the same order of subtasks is found in many instances, the creation of a more detailed structure of the process model can be suggested. Case variables are used to explain and combine deviations in a similar process model artifact. If the record also contains details about the delegation of subtasks to other performers or roles, it can also be suggested for future implementations of the task in the new process model.

The combination of the subtask discovery and the determination of subtask execution order corresponds to what is also known as (business) process discovery, which for example Cook and Wolf (1998) define as “methods for automatically deriving a formal model of a process from basic event data collected on the process”.

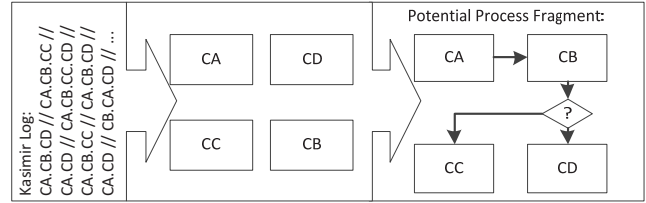


Figure 8: Process fragment discovery based on analysis of previous cases

Conclusion

The introduced approach aims for bridging the gap between process modeling (design time) and process execution (run time). Most process models base on in-depth business process analysis, representing possible requirements and constraints at a certain point of time as close as possible. After that, the model is transferred into an executable version, deployed and used. Even though a process model has been well defined, business changes quickly and the model might be already outdated when it is executed for the first time. If this is the case, users start creating workarounds to cope with the changes. In general, those 'deviations' are not captured, analyzed or mined for process improvements. Only when processes are to be improved and the management cycle starts again, models are adapted. With our approach, actions that users take during process execution are audited continuously and, therefore, suggestions for improvements can be derived at any time without additional effort and according to actual needs. Beside that, process optimization is achieved for knowledge-intensive processes and activities. Mining instances of these parts allow for constantly improving suggestions of resources, tasks or sequences of activities.

An extension of our mining approaches would be to identifying workflow resource patterns as introduced by Russell et al. (2004). This would require that we record the use of resources in more detail.

Acknowledgement

This work has been funded by the European Commission, Unit for Technology-Enhanced Learning (TEL) under the 7th Framework Programme, Contract no. 216356.

References

- Cook, J. E.; and Wolf, A. L. 1998. Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology*, 7(3), 215-249.
- Feldkamp, D.; and Hinkelmann, K. 2007. KISS-Knowledge-Intensive Service Support: An Approach for Agile Process

Management. *Advances in Rule Interchange*. Retrieved from <http://www.springerlink.com/index/47848h0041817171.pdf>

Martin, A.; and Brun, R. 2010. Agile Process Execution with KISSmir. *5th International Workshop on Semantic Business Process Management collocated with 7th Extended Semantic Web Conference*. Heraklion, Greece.

McCauly, D. 2010. Achieving Agility. In: Swensson, Keith, D. *Mastering the Unpredictable*. Tampa, Florida: Meghan-Kiffer Press.

Riss, U. V.; Rickayzen, A.; Maus, H.; and van der Aalst, W. M. P. 2005. Challenges for business process and task management. *Journal of Universal Knowledge Management*, 0(2), 77-100.

Russell, N.; ter Hofstede, A.H.M.; Edmond, D.; and van der Aalst, W.M.P. 2004. BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven, 2004.

Van der Aalst, W. M. P.; Reijers, H.; Weijters, A.; Van Dongen, B.; Alves De Medeiros, A.; Song, M.; and Verbeek, H.M.V. 2007. Business process mining: An industrial application, *Information Systems*, 32 (5), 713-732.

Van der Aalst, W. M. P.; and Weijters, A. J. M. M. 2004. Process mining: a research agenda, *Computers in Industry*, 53 (3), 231-244.

White, S.A.; and Miers, D. 2008. BPMN - Modeling and Reference Guide. Future Strategies Inc.

Witschel, H.-F.; Hu, B.; Riss, U. V.; Thönssen, B.; Brun, R.; Martin, A.; and Hinkelmann, K. 2010. A collaborative approach to maturing process-related knowledge. In *Proceedings of the 8th International Conference on Business Process Management*. New York, USA: Springer.