Predicting User Interaction in Enterprise Social Systems Using Process Mining

Completed Research

Christoph Drodt
University of Koblenz-Landau
drodt@uni-koblenz.de

Mike Reuther
University of Koblenz-Landau
mreuther@uni-koblenz.de

Abstract

In this paper, we explore the potential of probabilistic process mining for Enterprise Collaboration Systems (ECS) to predict the behavior of systems users. Towards this objective, we discuss applicability, limitations and challenges of probabilistic process mining in the context of ECS. We argue that probabilistic process mining can be a valuable method for researchers as well as practitioners (managers of collaboration platforms). We create and examine two process models using a probabilistic finite automata algorithm on event data of an enterprise collaboration system to show the feasibility of probabilistic process mining in ECS. Our research illustrates the most probable sequence of user activities in the selected system and demonstrates a way to predict the communities that a user will likely be active in.

Keywords


Introduction

The amount of structured and unstructured data that is stored in Information Systems (IS) worldwide has been growing substantially in recent years. Most companies are facing the challenge of a historically developed, complex landscape of different IS including Process Aware Information Systems (PAIS) such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM) Systems as well as Enterprise Collaboration Systems (ECS) that support joint work. These IS store vast amounts of structured and unstructured data (Che et al. 2013) that contain valuable information about business activities. The discipline of Data Mining seeks to provide methods and algorithms for the analysis of data stored in IS (Chen et al. 2012). Methods from DM have been successfully applied to different types of IS for example to calculate churn rates in Customer Relationship Management (Au et al. 2003) or for the development of Business Intelligence (BI) Solutions (Elbashir et al. 2008). Over time, the methods and algorithms used to analyze data have become more sophisticated and shifted from being mostly of a descriptive nature to a predictive nature (Gandomi and Haider 2015). In recent years, a new method of analysis named Process Mining (PM) has established itself as a tool in many different domains like finance (Ngai et al. 2011) and medicine (Rebuge and Ferreira 2012). PM allows the discovery, conformance and enhancement of process models based on data derived from log files created by IS (van der Aalst 2011) and is capable of structuring huge amounts of complex and unstructured data as it can be observed in IS. The application of PM is not limited to academic research. According to a Gartner report (Kerremans 2018) the market value of PM solution will triple or quadruple in the next two years. Previous research has shown that PM can be successfully applied to Process Aware Information Systems (PAIS). In this paper, we turn our attention to Enterprise Collaboration Systems (ECS), which are different to PAIS because processes are not prescribed in the software but ECS are open to interpretive flexibility, which means that the user is free to choose the order of activities from a given set of functionalities. It is thus to be expected that different users will solve the same task in different ways, resulting into different patterns of use. Companies are still struggling with the introduction of ECS which is reflected in a relatively slow adoption rate (Hausmann et al. 2014). We argue that understanding the sequence of activities (i.e. the actual use) of an ECS would help the person responsible for the introduction of a collaboration solution to instruct and train employees in a better-
informed way. We believe that the application of PM with the aim of deriving meaningful process models will be a contribution to academic research as well as a valuable aid for collaboration professionals. In this paper, we thus explore the potential of probabilistic process mining for ECS in order to gain a deeper understanding of the way users act in ECS.

Related work

In this chapter, we provide the necessary background information of the software type under investigation: Enterprise Social Software (ESS). This is followed by an introduction to process mining and the particular algorithm that we chose for our analysis. This chapter also contains a summary of related work.

Enterprise Social Software

ESS is a new kind of enterprise software that was stimulated by the emergence of social software (social media). The increasing use of social media (SM) and their "social software features" has changed the way people communicate and exchange information, and has raised employee expectations to use similar software features in their workplaces (Wehner et al. 2017). Major software vendors responded to this need with ESS, a specialized genre of collaboration software that provides social software capabilities such as IBM Connections, Yammer or Atlassian Confluence (Schubert and Schwade 2017). ESS provide basic functionality for the ECS of a company. Unlike the publicly available SM, their access is limited to the company employees (Figure 1). Typical "social features" include subscribing to / following information or people, commenting or flagging posts (tagging), or short expressions (microblogs) such as referrals or likes. These systems usually include extensive "awareness features" that help users identify new and potentially relevant content. The definition, terms and limits, as well as classification criteria have been addressed in previous research (Back and Irmler 2012; Kaplan and Haenlein 2010; Koch 2008; Levy 2009; McAfee 2006; Williams and Schubert 2011). ESS provides the functionality to build enterprise social networks (Wehner et al. 2016) and represents a new kind of information infrastructure (Hanseth et al. 1996; Monteiro et al. 2013). The features of ESS and SM are quite similar, but they differ strongly in their use. For example, in the enterprise context there could be guidelines or rules due to compliance and confidentiality. An interaction of employees could be part of a business process and therefore critical. Although the general research about online communities (Butler 2001; Faraj and Johnson 2011), Reddit (Olson and Neal 2015) and newsgroups (Joyce and Kraut 2006) is vast, the use case differs strong enough that the knowledge gained about public online communities cannot be transferred to ESS. As of now, the actual use of socially-enhanced ECS is not well understood. For example, a meta-analysis of the literature by Meske and Stieglitz (2014) showed that the measurement of the value added by ECS is problematic but highly relevant. In recent years, a new field of research, the so-called Social Collaboration Analytics (SCA), has emerged which allows a deeper analysis of ECS (Schubert and Schwade 2017). SCA is based on the methodologies and tools used in the research field Web Analytics (WA) and focuses on creating metrics in order to gain a better understanding of such systems. In their paper from 2017, Schubert and Schwade (2017) speculate that PM can be a contribution to SCA. In this paper, we will demonstrate a first successful application of Social Process Mining (SPM).

Figure 1: Terminology of ESS (Schubert and Schwade 2017)
Process Mining

The fundamental idea of process mining is to extract knowledge from event logs of real-world processes. To achieve this, process mining is split into three basic types: discovery, conformance checking and enhancement (van der Aalst et al. 2012). Discovery is used to generate process models from process logs and therefore to discover the underlying process with its activities. In order to test conformance, the discovered process is compared with the initially designed process to encounter differences. This helps to investigate whether the modeled process matches the actual real-world process. The third fundamental type is enhancement where the actual process is improved and extended by extracting information from an event log (van der Aalst et al. 2012). Since the emergence of Process Mining as a research field, many algorithms to support the goals of process mining have been developed. Exemplary pioneers are the $\alpha / \alpha + / \alpha++$-algorithm, the fuzzyminer and the heuristic miner (van der Aalst et al. 2003, 2004). In their application, these algorithms encountered typical problems including noise, incompleteness of process logs, complex routing constructs and short loops. A noisy process log holds alternative sequences of event flows which appear rarely. These alternatives can be considered as outliers but have an impact on the outcome of an algorithm. As the name suggests, an incomplete log includes traces (process instances) that were not completed. Today, new approaches are trying to address these problems to provide more accurate results. However, new techniques, algorithms and disciplines have been developed in the meantime in the field of process mining (Tiwari et al. 2008; Weerdt et al. 2012). A relatively young discipline is prediction, which can for example help to forecast machine maintenance cycles, production costs or a process outcome. Thus, different dimension types for prediction were addressed in recent studies. An overview of these techniques and dimensions was presented by Di Francescomarino et al. (2018) by creating the predictive process monitoring framework. They separated the field of predictive algorithms into the following categories: time, categorical outcome, sequence of outcome, risk, inter-case metrics and cost (Di Francescomarino et al. 2018).

Process Mining in ESS

PM techniques have been applied to different non PAIS and in the context of groupware. It has e.g. been used to analyze Microsoft Sharepoint log data (Naderipour 2011) or online learning platforms (Beheshtitha et al. 2015; Bogarin et al. 2014; Maita et al. 2017). Wang et al. (2014) manually coded a forum in order to mine a knowledge sharing process. Older studies used PM to analyze interaction patterns from e-mail logs (van der Aalst and Nikolov 2010) or mined event logs to identify social networks (van der Aalst et al. 2005). However, these studies show that unstructured processes – and collaboration activities tend to be ad hoc and thus unstructured – are hard to understand. The mined processes are complex and often lead to “spaghetti”-like models. This gets amplified in the context of collaboration by the non-prescriptiveness of the collaborative processes. Additionally, “real life” datasets are often incomplete and noisy. Some traditional algorithms like the heuristic miner and fuzzy miner are two robust techniques that solved the problem of noise and missing values in event logs but still struggle with extreme “spaghetti”-like models (Weijters and van der Aalst 2003). Another approach by Diamantini (2016) used graphs to explore the behavior patterns of actors in a hospital information system, which requires a transformation of the dataset into graphs. However, to the best of our knowledge, an exploratory study of real life event data of an ESS with the RegPFA algorithm has never been done before.

Research methodology

We used the CRISP-DM research methodology to structure our research as suggested by Schubert and Schwade (2017). In this research, we performed the first five of the six CRIPS-DM steps (business understanding, data understanding, data preparation, modeling, evaluation and deployment) (Pete et al. 2000). We were familiar with the target platform (UniConnect) because we had been actively using it for several years (business understanding). The data understanding was established with the help of database experts from the management team of UniConnect. The dataset was then prepared and the best fitting algorithm was selected. 14 models were extracted, evaluated and interpreted. Two of these models are described in this paper.
**Dataset**

The ESS used for this research is based on the commercial collaboration software IBM Connections, which (similar to a PAIS) generates an event log for all user activity on the platform. The platform UniConnect is a collaboration platform with numerous integrated applications such as wikis, blogs, forums, microblogs, task management, file libraries and social network capabilities and is organized in communities (workspaces). At the time of data extraction, more than 3,300 users from 35 universities are working with UniConnect. Our dataset is an excerpt of the \texttt{f\_trx\_events} table of the Metrics database from the year 2015. The dataset contains 12 columns, with 350,000 rows which can create problems with runtime complexity. Attributes of low relevance needed to be filtered in order to reduce the complexity. Therefore, only the fields \texttt{user\_id}, \texttt{community\_id}, \texttt{event\_ts} and \texttt{event\_name} were used, which represent the user and his interaction with the system. A large part of the dataset contained events showing the read and visit events of users, which can be used to show the user navigation through the platform. In this analysis, the authors focus on the active contribution of the users and less on the information consumption. Therefore, visit and read events were left out which helped to reduce the complexity of the dataset.

**The Process Mining Algorithm**

Following the **predictive process monitoring framework** (see section Process Mining), we aim at predicting the interaction of ESS users with the system and therefore focus on the **outcome** category. As we have an event log with rare contextual data, a large number of approaches can be discarded from the list of possibilities, where only one algorithm was appropriate. We needed a working implementation to achieve our goal. Inspecting the remaining techniques we decided to use a very recent approach, developed by Breuker et al. (2016). The \textit{RegPFA} is based on probabilistic finite automata (PFA) and returns a probabilistic process model in the form of a state-based transition system. The benefit of using PFA is a context-based probability estimation. Thus, predicting the next event does not rely on the current event, but on the whole current process instance. For the parameter estimation, which starts with a standard structure and estimates the parameter, this artifact uses an Expectation Maximization-based technique. As this easily leads to the common problem of overfitting, the PFA is extended by Bayesian regularization to avoid this (Breuker et al. 2016). The \textit{RegPFA} is split into two components: The Analyzer and the Predictor. Where the Predictor returns the next event most likely to happen, the Analyzer is responsible for creating the necessary probabilistic process model. To achieve this, an event log with the attributes \textit{case ID, activity ID} and a \textit{timestamp} is needed. A \textit{case} is one instance of the process, which consists of activities. The \textit{timestamp} states when an \textit{activity} has taken place. Additionally, the \textit{RegPFA} requires hyperparameters, most importantly the \textit{state} range and the model scorer. The last setting indicates the criterion on which the best fitted model is picked. One can choose between the Akaike information criterion (Akaike 1998), the cross entropy (Rosenfeld 2000) and the heuristic information criterion (HIC) (Breuker et al. 2016). Whenever the chosen criterion is decreased, the produced model has improved. The best fitted model is the one with the lowest score (Breuker et al. 2016). Finding the best values for the hyperparameters to achieve the best fitted model is part of our research process.

**Analysis and results**

In the exploratory approach we used different permutations of the attributes to extract different models. This enabled different perspectives on the dataset, which then were interpreted. In the following, the process of selecting the attributes, generating a log file and mining the model is described. Later, the resulting models are presented and interpreted. Note that PM is not limited to the following use cases. It can help to analyze and interpret log files in various ways.

**Pre-Processing and the Mining-Process**

For the pre-processing part, we chose Disco (https://fluxicon.com/disco/). When importing the dataset into Disco, one can select the three necessary attribute \textit{case ID, activity ID} and a \textit{timestamp}. For each model we selected different attributes to achieve different models. Additionally, it was necessary to apply filters on the imported data to reduce confusing data for the interpretation (see section Dataset). In the next step...
we exported the data to the Extensible Event Stream (XES) format, which is the standard for process mining. For the PM step, we selected ProM 6.8 (http://promtools.org/doku.php?id=prom68) as the RegPFA is currently only available as a plugin for this tool. After loading the XES-file and starting the RegPFA, the hyperparameters need to be defined. It turns out that finding the right settings is a long-lasting process, as the mining process takes a lot of time (multiple days per model). One setting of the hyperparameters is the pruning ratio, which affects the resulting transition system. Lowering the ratio displays less probable activities and therefore clears up the transition system. In the beginning, we chose a high pruning ratio and therefore received only unreadable models (“spaghetti”-like models). Figure 2 illustrates this phenomenon. Our aim was to find a readable model that can be interpreted. Thus, finding the right pruning ration was another important part of the process. In the following we present the two models and their interpretation.

Figure 2: Example of a "spaghetti"-like model

Model 1: Predicting Users Activities

In the first model, the most common user activity sequences are modeled. As described earlier, the user_id is used as case ID, event_name as activity ID and event_ts as timestamp. A filter was applied for event_op_name to clear out empty and therefore for this model irrelevant rows. Additionally, all visit-activities were removed. Applying these filters led to 14% fewer cases (987 in total) and 80% fewer events (70,676 in total). The process mining step resulted in a model with 73 states, according to the HIC score, which we chose for both models. After applying different pruning ratios, the resulting transition system that displayed the best balance between too much and too few information was found at a pruning ratio of 0.001 (Figure 3).

Figure 3 represents the mined model. Even with the chosen pruning ratio, it is hard to interpret. Also changing the ratio just slightly in both directions resulted in either a model with two states or a "spaghetti" model. For this reason, we used this visualization and tried to interpret parts of it. One example is presented in Figure 3 where one part is magnified. With a probability of 87% a file library is created, followed by the creation of a file collection with the probability of 32%. Afterwards a community is created (28%), subsequently followed by joining a community (28%). Our interpretation of this path is, that when creating a new community, a file library and a file collection is created by the system automatically (beforehand). Also, the community join after creating a community seems like a logical step, and as it is not performed by the user manually leads to the same conclusion that this event is triggered automatically by the software. This supports the assumption, that this model not only shows user activities but also system activities that are initiated by certain user activities and might be mandatory to reach a desired state. Following the transition on the left, the path ends in the same state as the path described before. This transition is the removal of a community-invite before joining a community. This is another activity, which is not manually performed by a user and seems to be a system action, when a user follows an invitation to a community.

Concluding, we can observe process steps that are necessary for a single user action which are not actually executed by the user, but by the system.
Model 2: Predicting User Activities in Communities

For this model, we are targeting the communities with the highest number of user interactions. Accomplishing this, \textit{user\_id} is used as case ID, \textit{community\_id} as activity ID and \textit{event\_ts} as timestamp. Like in model 1, a filter was applied for the attribute \textit{community\_id} where the standard community (ID 0) was removed. As every user automatically joins and interacts in this community, no benefit would be gained from including this community into the analysis. After the rows were filtered out, only 95\% of the cases (total of 1,093) and 56\% of the events (total of 197,937) remained.

Inspecting the results of the mining process, the lowest HIC score and therefore best model was achieved at a model with 72 states. Again, after testing different values, the best visualization was achieved with a pruning ratio of 0.001. Figure 4 represents this model. Comparing Figure 4 with 2 reveals a huge difference in states and transition which is due to a different amount of values in the \textit{activity\_ID} attribute.

As Figure 4 states, this model can be split into three clusters. The upper one self-loops with the community 28136, which represents a case study group that is part of a tutorial. Looking deeper into the tutorial, a lot of joint work was necessary in the case study group, which worked in this community. Community 10523 in the left cluster is the German support community of UniConnect where users can ask for help or report problems. Both clusters have a relatively low probability of 7\% and 5\% that a user will be active in. The more interesting cluster is the right cluster including the two communities 25109 and 45847. Both communities belong to lectures and therefore activities like file up-/download, providing information and asking questions in forums or wikis are taking place. What makes this cluster so interesting is, that both lectures belong to the same study program, where one is obligatory and the other is not. Additionally, both lectures are given by the same person. We can interpret, that it is highly likely that when a student is registered to one teaching module, this person is also registered to the other one.

To summarize, it is possible to find communities, which a user will participate in when joined in a particular community. In this special context we can draw conclusions on the interest of students in lectures.
Conclusions

After examining current PM approaches, we selected the RegPFA process algorithm to analyze a real-world dataset. To generate different models, different permutations of the attributes were chosen. The models represent the user activities and the communities with the most frequent activities. By lowering the pruning ratio, activities with a low probability were removed to gain the most probable activities for each model on the one hand and readable and interpretable models on the other hand. We were able to interpret the models in the context of UniConnect. Model 1 revealed that the system performs complementary actions to complete certain user tasks. In model 2, clusters of communities in which the same users were active could be identified. Overall, we showed that it is possible to gain information from ESS log files using PM techniques. In particular, the RegPFA algorithm was able to handle the high amount of complex information and extract readable and interpretable models. Not only were we able to extract information, we were also able to interpret the resulting models based on observations and with the help of our existing domain knowledge. The PM algorithm can be applied to generate recommendations for users to join a community. This could help students to avoid enrolling in the wrong course and therefore to gain information more quickly. The RegPFA is able to provide this information without the knowledge of sensitive user data, e.g. the course of studies, the matriculation number and an e-mail address. In the enterprise context this could be used to also identify join recommendations for communities without the knowing the context of communities and user data.

The dataset we used was limited to the year 2015. Some cases of collaboration activity on the platform might have started in 2014 and continued into 2015. The same applies to continuous collaboration that started in 2015 and continued in 2016. As a consequence, many traces are incomplete and the interpretation is limited to the prediction of the specific timeframe of a year. Expanding this timeframe could lead to a more accurate prediction. Furthermore, the dataset was derived from a university setting, which does not portray a typical “enterprise use” of the commercial collaboration software IBM Connections. Mining a log file from real enterprise data might present additional/differing challenges. We would like to point out that the interpretation of the results requires a domain specialist who can provide a meaningful interpretation of the models.

At this stage we demonstrated the applicability of PM to ESS but were limited to generating predictive process models. Additionally, we used the platform-level perspective for our analysis which hindered the interpretability of our models due to their size and complexity. In future work we want to compare models from different communities (e.g. project community vs student community), user groups (e.g. contributors vs authors) and time periods. Furthermore, we want to apply other approaches like pattern miner, Graph
miner, association rules, organizational mining or collaborative filtering. We believe that further research in this field will enable the operators of ESS platforms to realize the benefits of using PM which is already well known in other fields like ERP. The possibility of creating fully automated and accurate models of user interaction that represent business processes will help in managing and controlling those business processes. The effectiveness of trainings can be measured and could help with the adoption of ESS. The interpretation of behavior patterns would benefit the academic domain of Computer Supported Cooperative Work (academic) as well as the actual use of ESS in organizations (practice).

REFERENCES


Predicting User Interaction in ESS Using PM


