Supporting Process Mining with Recovered Residual Data

Ludwig Englbrecht, Stefan Schönig, Günther Pernul

Supporting Process Mining with Recovered Residual Data

Ludwig Englbrecht[0000−0002−8546−3017], Stefan Schönig[0000−0002−7666−4482], and Günther Pernul

University of Regensburg, Regensburg, Germany
{ludwig.englbrecht,stefan.schoenig,guenther.pernul}@uni-regensburg.de

Abstract. Understanding how workflows are actually carried out within an organization can provide a crucial contribution to business process improvement. This paper presents a concept for reconstructing a business process by using file residuals on a hard-drive and without the need for existing event logs. Thereby, methods from the area of process mining are enriched with approaches from digital forensics investigations in a *Digital Trace Miner*. First, a framework that extracts traces originating from business process execution based on residual data is developed in order to link them to the processes. The traces from the extraction are used in a life-cycle to keep related data up-to-date. This approach has been implemented and evaluated by a prototype. The evaluation shows that this approach enables useful insights regarding the tasks performed on a suspect computer by associating recovered files by using file-carving mechanisms.

Keywords: Process Mining · Business Process Discovery · Digital Forensics · Digital Trace Mining

1 Introduction

Process mining is a powerful method to gain valuable insights about how tasks are actually executed within organizations [1]. These approaches use event logs from different sources and provide a human-understandable view of the activities employees (or machines) perform. Process mining techniques rely on appropriate and correct event logs that contain a timestamp, task information and a logical connection refereed as *case id*. In addition, event logs need to be present in an adequate amount and quality to rebuild a process.

During process execution, various files are created and deleted that yield big potentials as an alternative data source for process mining. Files that are written and deleted on the hard drive can be restored with tools from Digital Forensics (DF). DF tools for recovering files are based on the fact that files are not immediately deleted from the disc by the operating system. They exist as long as the actual data area on the hard drive is overwritten. There are two approaches to fulfill this task: *file-recovery* and *file carving*. Compared to *file recovery* the *file carving* approach recovers files and fragments of files when the

directory structure is corrupt or missing. The *file-carving* method can therefore rebuild more files as a *file recovery* mechanism.

Process mining is an established method for discovering process sequences. Methods and knowledge in this area can also provide a sharpened view for a digital investigation. A major existing problem in the research area of process mining is that event logs are often not available or of insufficient quality to successfully apply mining algorithms [1, 22, 9].

In this work, we tackle this research challenge by leveraging digital forensics approaches as a basis for reconstructing the sequence of activities as a supporting method for process discovery. Our main contribution is to present the potential of file residuals on a computer to reconstruct and rediscover executed processes. To the best of our knowledge, no work to examine these potentials has been devised. Therefore, we developed concepts and tools called *Digital Trace Miner* that investigate systems on the basis of process-based file transactions. We use a modular system that makes the method of generating fingerprints interchangeable. The approach has been implemented and evaluated extensively.

The paper is structured as follows: In the following, theoretical work that forms the basis of the framework will be discussed in more detail. In chapter three the basic principle of the approach for using residual data for process reconstruction is discussed. In chapter four, the implementation of the concept is shown. Therein is the core element of creating training data for process-related file residuals is described. In chapter five the concept and the prototypical implementation are evaluated. In chapter six the limitations are discussed. The last chapter provides a conclusion and presents future research.

2 Background and Related Work

Moving up the abstraction ladder to gain a better understanding of digital evidence is needed to produce a high level and human-understandable overview of the activities [12]. In this section the background and related work according to the combination of digital traces and business processes are presented.

2.1 Digital Forensics and Forensic Sciences

In general, the forensic sciences deal with the application of methods for conducting investigations in legal cases [17]. This means that forensic scientists have to adapt questions of a legal case into scientific questions and answer them by using appropriate and scientifically-validated methods [14]. For this, specific requirements like a well-defined and well-founded knowledge base, a scientific method, and an experimental base are needed [23, 8].

DF can therefore be understood as a forensic science which deals with the application of specific methods from computer science to answer questions for the legal system [10]. This requires DF to provide methods to preserve and process digital evidence with the highest possible objectivity within a DF investigation.

2.2 Trace in the Context of Business Processes

Business processes can be used to link information systems within the corporate organization. Consequently, processes serve as intermediaries between an application system and tasks within the company. Even if modeled business processes are present, this cannot be sufficient for a DF investigation. It is therefore crucial to record the processes as they are actually executed.

Process Mining (PM) is related to Business Process Intelligence (BPI) and can be used to support modeling and to optimize business processes of an organization [7]. Therefore, already existing log files are used to generate a process model for analysis [2]. To achieve this, the recorded activities from an event log file are used for the mining process which makes use of various algorithms to rebuild the workflow.

2.3 Related Work

In recent years the idea of extending the application of process mining by using alternative data sources has been founded. Bala [4] stated the application of using logs from project-generated artifacts for determining the actually performed steps of a software development project. He suggests using data from Version Control Systems (VCS) which are used to manage revisions to mine the actual project steps. Bala motivates for the interdisciplinary combination and demonstrates that process mining techniques can not be adopted directly.

In a subsequent work [5] this idea is further defined and presents an approach for extracting process knowledge from the historical data in the area of software development. Their work presents the baseline for extending the applicability of process mining techniques to software processes.

In the area of DF, Soltani and Seno [26] provide an overview of different works, which on the one hand collect evidence and on the other hand analyze it. Under the heading of preservation of evidence, the authors categorized various publications into four areas: *hard- and memory imaging tools, extracting data structures from hard/memory images, integrity issues and scalability issues* [26].

The authors refer to ten papers that have done research in this field, but none of these papers used process related file residuals. This is a research gap [26]. In this work we focus on the event reconstruction using signatures of applications, especially of process related file transactions.

In order to bring together past actions and assign them to processes, it is necessary to generate past action instances automatically. James and Gladyshev [15] take this up in their work. The authors developed a signature-based method for automatic analysis. They use the threshold of updates and bring low-level artifacts and high-level action in correlation. The goal of the paper is a fast and detailed reconstruction of action for DF. The problem that arises in their work is that they have built their use case for a small scope. They only use single computers, run a browser and rely on the computer's file system meta-data [15]. This is not appropriate for DF in companies. There are many systems connected in a common network and together they execute many processes .

One of the ten works that Soltani and Seno [26] referred to is the paper of Kälber et al. [16]. They present a new approach for automatic event reconstruction with the help of file system metadata. They developed a system that automatically takes fingerprints based on changes in the metadata of a system. This fingerprint generator serves as a basic concept for the goal of this work on automated event reconstruction based on process related file residuals.

Another basis for the approach of this work is the dissertation thesis of Meier [21]. It describes the importance and illuminates the basics of DF in enterprises and goes into more detail about different use cases. Among them, business processes in the sense of corporate forensics are also specified. This point is important for the development of our approach to use process related file transactions because Meier defines processes in terms of DF and the relationship between the level of processes and the level of low-level activities, such as state changes. Those are needed to be able to link actions to processes. However, Meier uses the concept of Differential Forensic Analysis to determine process related traces between a fixed time frame. We propose a more granular approach that observes and record file transformations.

In all the works mentioned above none of them is dedicated to the approach of DF of process related transaction files. The approach in this work enables to discover workflows based on file residuals systematically. The overall benefit is two-fold since recovered business-related files can be logically ordered and this abstraction provides useful insights for a further DF investigation.

3 Using File Residuals for Process Reconstruction

The idea of this paper is to combine the theoretical principles of past action instances of James and Gladyshev [15] with the inspiration of practical implementation of fingerprints based on file system metadata of Kälber et al. [16] and integrating it into the organizational environment as suggested by Meier [21]. The goal is to establish an approach for analyzing fingerprints from process related file transactions by its content and not by meta-data. This method is used as a basis for the training of a system intended for usage in a life-cycle.

To use process related file transactions to generate a fingerprint from an application an analyst has to connect the lowest level of information, like a state change of a system, with the highest activity of a process.

To establish the connection between the lowest level and the highest level, it is necessary to consider actors at an intermediate level. Actors at the intermediate level are the executors of the processes and, thus, also the executors of the action that triggers a state change. Humans or software applications at this level are responsible for state changes. By clicking on an order form people can change the status in the ERP system or an application can send an order command when the warehouse is empty. These three levels are shown in more detail in the following Fig. 1.

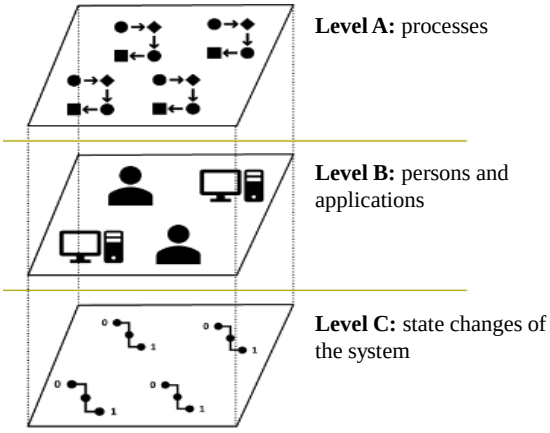

Fig. 1. Overview of the relation between different levels based on Meier [21]).

As shown in the figure, there are three levels (A, B and C) that need to be connected. In order to identify the lowest level (C), the state changes the system, the theoretical principles of James and Gladyshev [15] are applied. These are of less practical nature so the work of Kälber at al. [16] is used as an inspiration for practical implementation. The second level (B) represents the people and physical things that trigger the actions for the changes. The top level (A) is represented by the processes with the respective process steps.

The second part of the framework is the life-cycle which is to be applied in companies to repeat the concept. This is done by extracting new information and linking old and new process information. The life-cycle consists of four phases that are repeated continuously. When the training on the respective system is completed and the link between low-level changes and business processes provides correct data, these data are fed into the life cycle. This is done in step one of the extractions. The other three stages serve to prepare, combine and abstract new and old information. These steps are repeated in a self-defined time period to increase accuracy.

Process mining is based on extracting knowledge from event logs which have been recorded by an information system. Those event logs are generated at level B and leave state changes at level C according to the previously described abstraction model in Fig. 1. The application of the life-cycle enables to link data from level C to a process description at level A.

There are various methods to create a process model from an event log. (e.g. by applying the alpha miner algorithm or a heuristic method). Both approaches intend to create a workflow net [3].

With the *HeuristicsMiner* algorithm, only the control flow perspective of a business process is used for reconstruction. Consequently, only the sequences of events of a certain case are considered. Furthermore, the *HeuristicsMiner* is a mining algorithm that can deal with noise. This can be used to extract the main

behavior from an event log. The attributes *case id*, *timestamp* and *activity* are required for the mining procedure. [27]

In order to generate a process model from an event log, it is important that causal dependencies are found within the log. This means if one activity is followed by another, there is a high probability that they are in a dependent relationship. The attribute timestamp is the exclusive characteristic of the order of events. With the aforementioned thesis that traces were left during process execution, we now have the basis to be able to reconstruct a process model on the basis of these traces.

3.1 Framework and Concept

The specific usage of this construct provides a guided structure to analyze specific areas on related information systems.

Figure 2 illustrates the architecture of the suggested Digital Trace Miner (DTM) and its application. A sub-process is a defined segment of a business process at a lower level. It consists of several tasks that are combined to form a logical group. The input can be understood as a combination of process knowledge and related traces on various systems (as well as applications). These traces are illustrated as small symbols within the swimlane of one specific system. For example, if sub-process D is executed, traces are generated by application 2. In every system, various applications are used and leave traces during the execution of business related tasks.

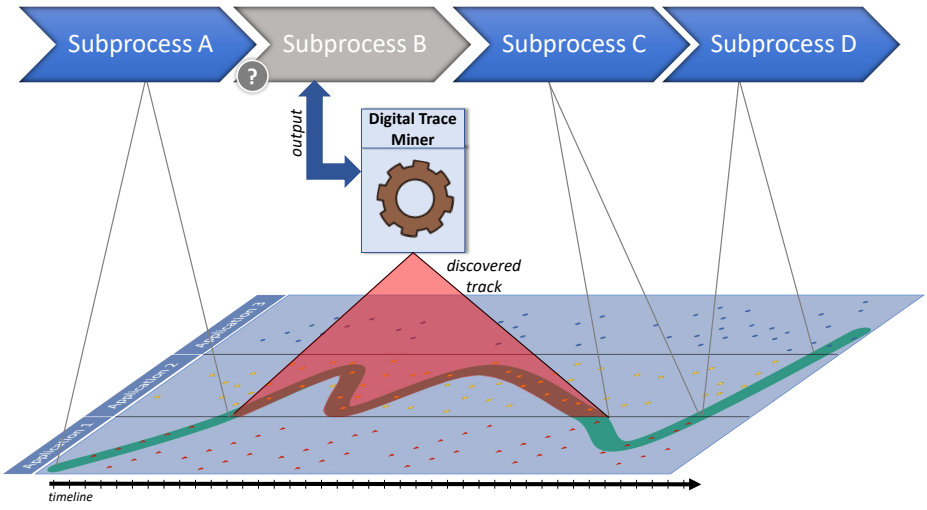

Fig. 2. Illustration of the concept of the Digital Trace Miner

During the execution of a business process various data is written to the hard disk and erased afterward. This data (or fragments of it) can be recovered

by using a file carver. However, the metadata of the deleted data is lost. If a sufficient amount of recovered data can be acquired, it is possible to reconstruct a business process. It is necessary to correctly determine the logical connection and order between the recovered data. A sequence of files can thus create a file structure that is very similar to event logs.

In order to ensure that the mapping from residual data to tasks and even entire processes is possible, a three-step procedure is developed and applied in this paper. This procedure uses as input a set of recovered process relevant data. In the first step a rough presorting is performed.

Step 1: Selection of business relevant data. This is done by using an entropy-based similarity measurement according to Englbrecht and Pernul [11]. This approach is inspired by Mc Creight and Weber [20] and has also potential on this paper. It allows a rough but sufficient filtering of the residual data regarding potential process phases. Our concept is based on recovered files from file-carving tools. This technique enables the recovery of files without considering the metadata of the file-system. Since this approach also recovers a lot of irrelevant files for the business-process reconstruction a filtering mechanism needs to be applied.

To fulfill this aspect the concept of Englbrecht and Pernul [11] is used to recover only business-related files by comparing the recovered file residuals with a previously created repository. The similarity of two files is determined by the cosine similarity of the local Entropy structure of two files. This approach is shown in Fig. 3 and is built on techniques from the area of DF as well as on the entropy-based similarity measurement [20].

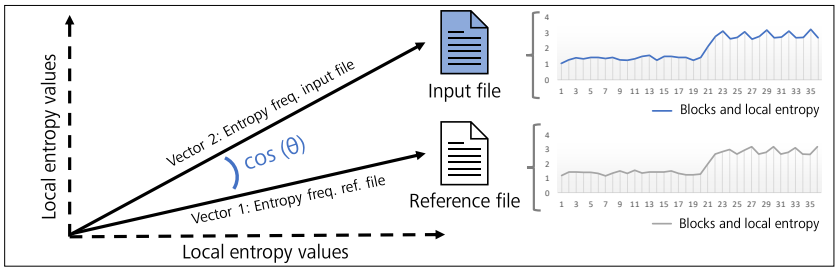

Fig. 3. Determining the similarity of two files based on [11]

Step 2: Reconstruct the order of file-versions. In the second step, the relationship of the files beyond the phases is determined. This is done using a similarity preserving hash function provided by Breitinger and Baier [6]. This mechanism can be used to determine data chunks from one file within another file. This allows determining the content-based relationship of two files and makes it possible to order the previously presorted data based on the content. This concept is shown in Fig. 4. Within this figure *a)* is a specific version of a file and *b)* is the transformation between two versions.

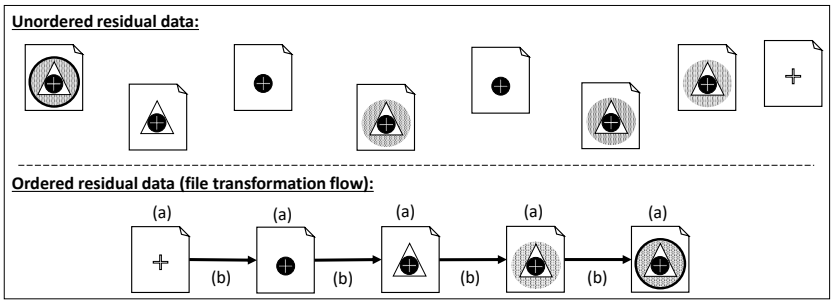

Fig. 4. Illustration of the relationship of data-chunks within multiple file versions

Step 3: Retrieve timestamps from the residual data. The third step of our concept is a fine adjustment of the result set. This is particularly important if a similar business process (same customer, same articles and the same process flow) was performed. Since the result of step 2 is just the order of the files the time dimension needs to be added. By analyzing the content of the files a timestamp can be extracted. This can be realized by using adequate regular expressions or NLP techniques. Nevertheless, this step provides the basis for the application of machine learning techniques to sharpen the differentiation of process instances.

4 Implementation

To demonstrate the applicability of the concept we implemented the core elements as dedicated web-services and realized the composition of those as a script. To fulfill the necessary tasks a profound recording of all relevant file-changes and the actual content of each file is necessary. For this we decided to use a modified version of an existing Continuous Data Protection (CDP) [25, 28] mechanism.

CDP is a data protection technology and is defined by the Storage Networking Industry Association (SNIA) as a methodology for continuously capturing and storing data changes, enabling data to be recovered from the past at any point in time. [25, 28] The main reason for using CDP instead of traditional backup technologies is the improvement in recovery point objective (RPO) and recovery time objective (RTO) metrics [19]. The RPO defines the time between two successful backups and thus the maximum loss of data during a successful recovery. If there is fully synchronized protection for a system, RPO is 0. CDP fulfills the metric RPO = 0, because it allows theoretically unlimited recovery points. [25] To ensure sufficient granularity of recovery, CDP systems can be implemented either at application, file or block level [18].

In addition to the traditional applications of CDP, it can also be used to generate data for usage in DF. In this context this technology has not been considered in the literature so far, but it can be of elementary importance. In Fig. 5 we illustrate the possible use of CDP in the context of DF for the creation of application fingerprints based on the actual file content without the need for meta-data.

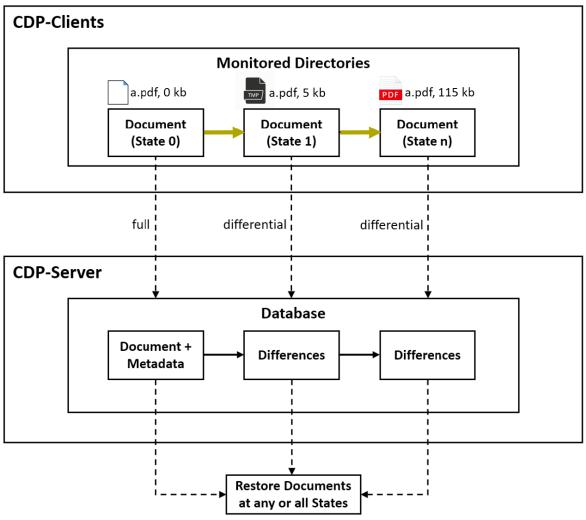

Fig. 5. Usage of CDP in the context of Digital Forensics

As shown in Fig. 5, the file *a.pdf* is monitored by a CDP client. The file can have n states, in which it may have different content and size. The respective states are transmitted from the CDP client to the CDP server. If the file is changed, the changes are saved and the metadata is updated accordingly.

Since block-based CDP systems are required to achieve RPO = 0, an architecture that uses block-based CDP is modified as a technique for recording business related file transformations.

The CDP client monitors all file system operations performed on the client's drive. When a file system operation modifies data on a client drive, the changes are detected at the block level and transmitted to the CDP server. If the file is modified at a later time, the process is repeated and only the modified file blocks and again the associated metadata are transmitted to the CDP server.

The CDP server stores the block data centralized for all clients in the architecture. The metadata, which includes a timestamp of the change, is stored separately for each client. Thus, all changes on the file systems of all clients are stored centrally on the CDP server. In turn, all changes that have been made in the course of business process execution are also recorded.

This CDP approach has been used as a baseline for the acquisition of the training-data for the Digital Trace Miner. During the execution of a business process at an employee PC all file transformations are recorded by using CDP. To adopt this approach the tool Sauvegarde[1] has been extended. In order to indicate the extension our new version is called SauvegardeEX and is available on GitHub[2].

[1] https://github.com/dupgit/sauvegarde
[2] https://github.com/LudwigEnglbrecht/sauvegardeEX

Since the original software is intended to capture all data and restore them at the client system we enriched the server component. We modified the restore component and installed it on the server side to restore all files at the server. Also a modular extension has been added to perform analysis on the restored data at the server. This modification was necessary to first restore all business related data and to extract a list of all local entropy values per restored file. This is necessary for to measurement of the similarity and to classify new files according to a process phase as stated in sub-section 3.1.

The final modified version of Sauvegarde is illustrated in Fig. 6. It describes the concrete system architecture including the technologies used to implement it. The system architecture for the implementation consists of a set of clients, a

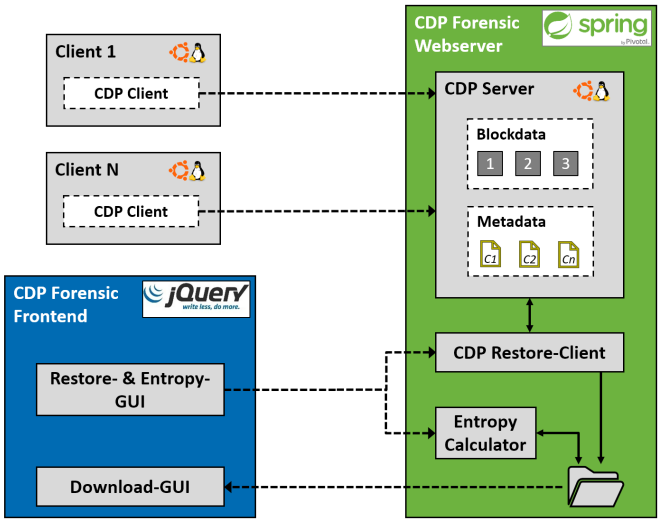

Fig. 6. Architecture of *SauvegardeEx*

web service and a frontend for user interaction.

Since CDP is not designed for use in DF, a web service is implemented in addition to the CDP server to provide the functionalities required in the context of DF, such as adapted reconstruction logic and persistent data storage for reconstructed business processes. The web service contains a restore client that is used to reconstruct individual file states and communicates with the CDP server. In addition, the web service contains a persistent data storage, in which business processes are stored after the reconstruction is completed and made available for download and to perform analysis.

The frontend provides the graphical user interface and is used to control the business process reconstruction. All processes that are executed on the web server in this respect are initiated at the frontend. To do this, the instructions for execution are transferred to the web service interfaces provided for this purpose.

The reconstructed business process data can then be viewed at the frontend and obtained for further forensic analysis.

SauvegardeEx builds the fundamental base for the Digital Trace Miner since with this approach the necessary test data can be acquired. This means a computer of an employee can be observed during the execution of a business process. The modified version of Sauvegarde enables a possibility to restore any files at any stage with the necessary metadata.

Since our approach is intended to discover a business process based on recovered files (residual data) from file-carving where metadata is absent the following tasks need to be performed:

1. Perform file-carving on a hard-drive to recover files
2. Using the acquired data via SauvegardeEx for classification
3. Rebuild the order of the files
4. Extract information about the execution of the activities
5. Aggregate the previous information to event-logs

5 Evaluation

To evaluate our concept we created a scenario[3] that produces files from a common workflow of creating an offer for a customer. The process consists of four main steps and starts when a customer sends a request for an offer via an XML file. We aligned the process on a concept where businesses interchange data via standardized exchange formats. This provides the possibility to simulate a process without the need to align with specific ERP software. During this simulation for each process instance a four-fold transformation of an XML-file has been performed. This means after the *offer request* form a customer this XML file has been subsequently enriched with data. The process is shown in Fig. 7.

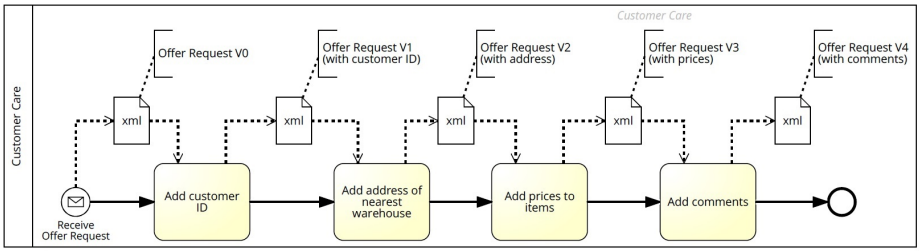

Fig. 7. The used business process for the evaluation[4]

As stated before our main focus of the concept is to discover processes based on residual data by building event-logs where no event logs are available. For

[3] Evaluation script and data: https://github.com/LudwigEnglbrecht/DTMEvaluation
[4] The process has been modeled with *Signavio*. https://www.signavio.com/

this reason our process is a sub-process of a larger business process where other sub-processes are well covered by event-logs.

Step 1: Learning the file-transformation structure. The process of the creation of an offer for a customer has been simulated by execution the whole process for 6.063 times. Every process instance is assigned to a different customer. During the simulation of the process execution our prototype *SauvegardeEx* observes the hard-drive where the different versions of the offer request are written and saves every version of the file at the server.

We started the client of *SauvegardeEx* at the PC where the previously mentioned process instances (and the file transformations) are executed. Every version of the altered file is transmitted to the *SauvegardeEx* Server for further analysis. The original version of Sauvegarde enables the recovery of every file version at the client. We extended this feature to export all versions of the files.

This step ends with a profound observation of all file changes and the recording of all file versions.

Step 2: Creating file-residuals of the process. As a subsequent step of our evaluation all file residuals of the process instances are deleted from the hard-drive via the operation system deletion command. This means no special erase software to overwrite deleted data is used.

The hard-drive on which the process instances have been executed and all files have been deleted is used now as an input for the *Digital Trace Miner*. We used the included file-carver of Autopsy 4.15.0[5] to reassemble all (possible) files on this hard-drive. As a result a lot of files have been recovered but the meta-data is still missing.

The result of this step are all recovered files of a hard-drive by using file-carving techniques without meta-data. Those files need to be further processed by classifying the files to specific process phases and ordered according to their process-instance to derive a *case id*.

Step 3: Classifying file-residuals to process phases. As stated in the concept of our approach the recovered files (without meta-data) needs to be classified according to their process phase. This is done by using a web-service of the Entropy-based file classification [11].

The first classification of step 3 provides the information to what phase of a business process a recovered file belongs. This is also the base for linking the recovered files according to their order and to obtain a workflow based on the file residuals.

Step 4: Rebuilding an order of the file-residuals. In step 4 the order of the file-residuals is rebuilt by using the approximate hashing function of Gupta and Breitinger [13]. This implementation has been used since the approach uses Cuckoo Filter to improve the approximate hashing process. Each recovered file of phase *n* is compared with all files of phase *n-1*. The detection of data chunks in a file that is classified as one of the previous phases allows the indication of a sequence.

[5] https://www.autopsy.com/download/

Step 5: Aggregation of the information to an event-log. In this step all information of the previous steps is aggregated to event-logs by a python script. This aggregation is the core of the proposed *Digital Trace Miner* since with this a high-level abstraction of the file residuals (digital trances) is possible.

During the execution of the evaluation, it was possible to find 5.100 process instances based on the file residuals. Process instances which could only be partially restored are not included in this result. One reason for the incomplete reconstruction is that not all data can be completely restored by file carving. This disadvantage is discussed in the following chapter.

6 Limitations and Discussion

The presented approach describes how the actual execution of a business process on the computer can be detected step by step on the basis of file residuals without the existence of event logs. The concept offers a useful supplement for the discovery of sub-processes where no useful event logs are available. However, the concept and the proposed Digital Trace Miner has some limitations.

The most challenging aspect of the concept is when little or no data is stored on the computer since a majority of the files are processed in a cloud environment. This means that even the recovery of deleted data from the hard disk provides no added value. In order to be able to analyze data stored outside the computer under investigation, other methods of DF must be included (e.g. network forensics or cloud forensics).

Another challenge for the approach is if deleted files can not be restored or have been overwritten over time. The presented approach assumes that unallocated space on the hard-drive has not been overwritten until the application of the DTM. To overcome this issue approaches from process mining according to the handling of missing event-log entries can be applied.

Further, if there is only an encrypted drive present, the aforementioned concept cannot be applied. To address this issue it is necessary to also extract the decryption key from memory with adequate forensic tools.

7 Conclusion and future work

The discovery and reconstruction of actually executed business processes on a specific system can bring valuable insights for process analysis and process mining. We present an approach to discover a business process based on file residuals that is built on approaches from DF. In this paper, first a framework that extracts traces originating from a business process execution based on residual data was developed in order to link them to processes. After that a modified CDP software for the observation and recording of business-process related file transformations has been presented. A structured classification and ordering process rebuilds the necessary information to create an event-log that can be processed further.

Since our approach has just been applied to a single process we will adjust the validation to sharpen the differentiation of different process variants. Due to the loss of information by the deletion of files the application of machine learning can bring a benefit to rebuild lost information and to improve the accuracy of the Digital Trace Miner.

The present approach in its current form is designed for a single object of investigation. However, since an application communicates with a remote peer such as a server during the execution of a task, it would be reasonable to implement a cross-system view. This will lead to more reliable results because the file residuals on one system can be combined with data from another system and the current limitation (by overwriting the deleted files) can be overcome.

In future work we will study and enhance the credibility of the linkage between process tasks and the produced file residuals by applying concepts of a whole-system data provenance approach [24]. This can support our approach by automating the creation of training data and meeting the requirements of DF.

References

1. van der Aalst, W.M.P., Adriansyah, A., et al.: Process mining manifesto. In: Business Process Management Workshops. vol. 99, pp. 169–194 (2011)
2. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.J.M.M., van Dongen, B.F., de Medeiros, A.K.A., Song, M., Verbeek, H.M.W.: Business process mining: An industrial application. Inf. Syst. **32**(5), 713–732 (2007)
3. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9), 1128–1142 (2004)
4. Bala, S.: Mining projects from structured and unstructured data. In: Gulden, J., Nurcan, S., Reinhartz-Berger, I., Guédria, W., Bera, P., Guerreiro, S., Fellmann, M., Weidlich, M. (eds.) CEUR Workshop Proceedings. vol. 1859, pp. 133–137. CEUR-WS.org (2017)
5. Bala, S., Mendling, J.: Monitoring the software development process with process mining. In: Shishkov, B. (ed.) Business Modeling and Software Design. pp. 432–442. Springer International Publishing, Cham (2018)
6. Breitinger, F., Baier, H.: Similarity preserving hashing: Eligible properties and a new algorithm mrsh-v2. In: Rogers, M.K., Seigfried-Spellar, K.C. (eds.) Digital Forensics and Cyber Crime - 4th International Conference, ICDF2C 2012, Lafayette, IN, USA, October 25-26, 2012. vol. 114, pp. 167–182. Springer (2012)
7. Castellanos, M., de Medeiros, A.K.A., Mendling, J., Weber, B., Weijters, A.J.M.M.: Business process intelligence. In: Cardoso, J.S., van der Aalst, W.M.P. (eds.) Handbook of Research on Business Process Modeling, pp. 456–480. IGI Global (2009)
8. Cohen, F.: Toward a Science of Digital Forensic Evidence Examination. In: Advances in Digital Forensics VI, IFIP Advances in Information and Communication Technology, vol. 337, pp. 17–35. Springer Berlin Heidelberg (2010)
9. Dakic, D., Stefanovic, D., Lolic, T., Narandzic, D., Simeunovic, N.: Event log extraction for the purpose of process mining: A systematic literature review. In: International Symposium in Management Innovation for Sustainable Management and Entrepreneurship. pp. 299–312 (2019)

10. Dewald, A., Freiling, F.C.: From Computer Forensics to Forensic Computing: Investigators Investigate, Scientists Associate (2014)
11. Englbrecht, L., Pernul, G.: A privacy-aware digital forensics investigation in enterprises. In: Volkamer, M., Wressnegger, C. (eds.) ARES 2020: The 15th International Conference on Availability, Reliability and Security, Virtual Event, Ireland, August 25-28, 2020. pp. 58:1–58:10. ACM (2020)
12. Garfinkel, S.L.: Digital forensics research: The next 10 years. Digital Investigation **7**, 64–73 (2010)
13. Gupta, V., Breitinger, F.: How cuckoo filter can improve existing approximate matching techniques. In: James, J.I., Breitinger, F. (eds.) Digital Forensics and Cyber Crime - 7th International Conference, ICDF2C 2015, Seoul, South Korea, October 6-8, 2015. vol. 157, pp. 39–52. Springer (2015)
14. Inman, K., Rudin, N.: Principles and Practice of Criminalistics: The Profession of Forensic Science. Protocols in forensic science, CRC Press, Boca Raton, Fla. (2000)
15. James, J.I., Gladyshev, P.: Automated inference of past action instances in digital investigations. International Journal of Information Security **14**(3), 249–261 (2015)
16. Kälber, S., Dewald, A., Freiling, F.C.: Forensic application-fingerprinting based on file system metadata. In: Seventh International Conference on IT Security Incident Management and IT Forensics. pp. 98–112 (2013)
17. Kent, K., Chevalier, S., Grance, T., Dang, H.: Guide to Integrating Forensic Techniques into Incident Response: NIST SP 800-86 (2006)
18. Li, H., Xiao, F., Xiong, N.: Efficient metadata management in block-level CDP system for cyber security. IEEE Access **7**, 151569–151578 (2019)
19. Lu, M., Chiueh, T.: File versioning for block-level continuous data protection. In: 29th IEEE International Conference on Distributed Computing Systems (ICDCS 2009), 22-26 June 2009, Montreal, Québec, Canada. pp. 327–334. IEEE Computer Society (2009)
20. McCreight, S., Weber, D.: System and method for entropy-based near-match analysis (2010), US Patent App. 12/722,482
21. Meier, S.: Digitale Forensik in Unternehmen. Universität Regensburg (Jan 2017)
22. de Murillas, E.G.L., van der Aalst, W.M., Reijers, H.A.: Process mining on databases: Unearthing historical data from redo logs. In: BPM. pp. 367–385 (2016)
23. Palmer, G.: A road map for digital forensic research: Report from the first digital forensic research workshop (dfrws). In: First Digital Forensic Research Workshop, Utica, New York. pp. 27–30 (2001)
24. Pasquier, T., Han, X., Goldstein, M., Moyer, T., Eyers, D., Seltzer, M., Bacon, J.: Practical whole-system provenance capture. In: Proceedings of the 2017 Symposium on Cloud Computing. pp. 405–418. ACM (2017)
25. Sheng, Y., Wang, D., He, J., Ju, D.: TH-CDP: an efficient block level continuous data protection system. In: International Conference on Networking, Architecture, and Storage. pp. 395–404 (2009)
26. Soltani, S., Seno, S.A.H.: A survey on digital evidence collection and analysis. In: 2017 7th International Conference on Computer and Knowledge Engineering (ICCKE). pp. 247–253. IEEE (2017)
27. Weijters, A., van der Aalst, W.M.P., de Medeiros, A.A.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP **166**, 1–34 (2006)
28. Yu, X., Tan, Y., Sun, Z., Liu, J., Liang, C., Zhang, Q.: A fault-tolerant and energy-efficient continuous data protection system. J. Ambient Intell. Humaniz. Comput. **10**(8), 2945–2954 (2019)