Using a Deep Understanding of Network Activities for Workflow Mining

Mona Lange[1(✉)], Felix Kuhr[2], and Ralf Möller[1]

[1] Universität zu Lübeck, Lübeck, Germany
{lange,moeller}@ifis.uni-luebeck.de
[2] Hamburg University of Technology, Hamburg, Germany
felix.kuhr@tuhh.de

Abstract. Workflow mining is the task of automatically detecting workflows from a set of event logs. We argue that network traffic can serve as a set of event logs and, thereby, as input for workflow mining. Networks produce large amounts of network traffic and we are able to extract sequences of workflow events by applying data mining techniques. We come to this conclusion due to the following observation: Network traffic consists of network packets, which are exchanged between network devices in order to share information to fulfill a common task. This common task corresponds to a workflow event and, when observed over time, we are able to record sequences of workflow events and model workflows as Hidden Markov models (HMM). Sequences of workflow events are caused by network dependencies, which force distributed network devices to interact. To automatically derive workflows based on network traffic, we propose a methodology based on network service dependency mining.

Keywords: Workflow mining · Hidden Markov model · Network dependency analysis

1 Introduction

Workflow and business process models have recently gained a lot of traction in the cyber security community. This is due to the fact that they can be used as a foundation for operational impact assessment within a security information and event management system, or as a foundation for process-aware information systems. Workflows are often not documented and there have been reoccurring issues [16] with manually designed workflows. It is a very time consuming process to design hand-made workflow models and, thereby, expensive. Hand-made workflows are idealized descriptions of the process at hand and often describe more what should be done, than the actual process. Additionally, with a hand-made workflow it difficult to detect when concept drifts have occurred and the workflow model needs to be updated. Hence, the data mining community has paid a lot of attention to the automated acquisition of workflow models [5,13,14,17,18]. Based on event logs, workflow mining methods automatically deduce sequences of workflow events.

G. Friedrich et al. (Eds.): KI 2016, LNAI 9904, pp. 177–184, 2016.
DOI: 10.1007/978-3-319-46073-4_17

Currently, workflow mining is dependent on event logs and research in this domain can be divided into three topics: discovery, conformance and enhancement of workflows [15]. As we focus on workflow discovery in the context of this work, we list workflow discovery techniques in the following. A lot of workflow mining methods [5,13,14,17,18] rely on Petri nets, due to their similarities to workflow models established in business science. Other workflow mining methods [2,11] rely on HMMs, which are statistical models. Unlike Petri nets, HMMs are able to model properties such as the transition probability between workflow events. However, Petri nets can be efficiently mapped to HMMs [9,10]. Event logs are the basis for all previously listed techniques and are supposed to contain workflow data. Obtaining workflow information is not as easy and often experiments are conducted based on synthetic data sets [11]. A common limitation that all previously listed techniques have is that they rely on event logs containing workflow data, a data source that is hard to come by in every day enterprise networks.

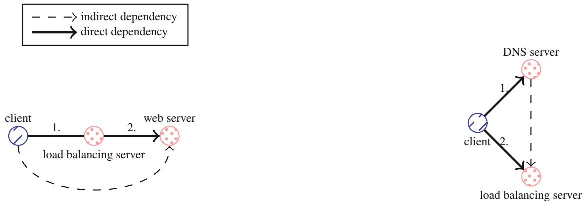
Network traffic contains traces of communicating network services, which interact to fulfill a higher mission. This communication leads to indirect dependencies, which are clues for a data-communication networks workflow. To achieve the goal of mining workflows based on network traffic, we rely on network service dependency discovery to identify workflow events. This is why we rely on an automatic network service dependency methodology called Mission Oriented Network Analysis (MONA) [7]. MONA was compared to three state of the art network service dependency discovery methodologies: NSDMiner [8], Sherlock [1] and Orion [3]. MONA was compared via F-measures to all these state of the art methodologies and was shown to have a better performance.

2 Network Service Dependency Discovery

In the following, we will only use the term workflow, however it should be noted that earlier publications use the terms workflow and business process models interchangeably [5]. Companies, organizations and enterprises have a workflow, which translates into network activities within their data communication network. Workflows often cause reoccuring network activity patterns. We understand these network activity patterns as workflows and network service dependency analysis has the purpose of detecting workflow events. and we rely on an automatic network service dependency methodology called Mission Oriented Network Analysis (MONA) [7]. In the following, we thus rely on the same network model introduced by MONA.

2.1 Indirect Dependencies

In the context of this work, we introduce network dependency analysis as a basis for workflow mining. For this purpose, indirect dependencies correspond to workflow events. Similarly to previous work, we distinguish remote-remote (RR)



(a) Remote-remote indirect dependency.

(b) Local-remote indirect dependency.

Fig. 1. Example for indirect dependencies.

dependencies and local-remote (LR) dependencies [3]. Examples for both dependency types are shown in Fig. 1. For a remote-remote (RR) dependency, first one remote host must be contacted before issuing a request to another remote host. Figure 1a shows an RR dependency $ISDEP_{RR}$ and Fig. 1b shows and LR dependency $ISDEP_{LR}$. The set of all RR and LR dependencies is defined as $ISDEP = \{ISDEP_{LR}, ISDEP_{RR}\}$. Following MONA [7], normalized cross correlation provides us with a heuristic for learning indirect dependencies $ISDEP$. An indirect dependency event $\iota_i = \{\delta(s_i^j, s_k^l), \delta(s_m^j, s_o^n)\}$ is based on direct dependency events δ. The set of all indirect dependencies $ISDEP$ translates into a set of indirect dependency events $\Omega = \{\iota_0, \ldots, \iota_n\}$. MONA creates probabilities $\varrho(\tau_{delay}) \in P_\varrho$ ranging between $\varrho(\tau_{delay}) = [0, \ldots, 1]$ and provides a set of observed workflow events $F \subseteq \Omega$.

$$p(\iota_{hg}(\delta_h(s_i^j, s_k^l), \delta_g(s_m^j, s_o^n)) \mid \delta_h(s_i^j, s_k^l) \wedge \delta_g(s_m^j, s_o^n)) = \varrho_{r,s}(\tau_{delay}) \tag{1}$$

Obviously, there is a level of uncertainty associated with detected indirect dependencies. By understanding indirect dependencies as indirect dependency events, we are able model the probability of uncertain event by relying on Kolmogorov axioms of probability theory [6]. An example for this probability space is illustrated in Fig. 1a and contains a set of workflow events $F \subseteq \Omega$, showing an RR dependency and consists of a client d_c sending an HTTP request to a load balancing server d_{lbs}. The load balancing server d_{lbs} then sends an HTTP request to a webserver d^{ws}. The RR dependency, shown in Fig. 1a, can be written as $\iota_{01}\left(\delta_0(s_*^c, s_{80}^{lbs}), \delta_1(s_*^{lbs}, s_{80}^{ws})\right)$. Figure 1b shows an LR dependency, where the client d_c sends a request to a DNS server d_{DNS}. Afterwards, the client d_c sends a load balancing server d_{lbs}. The LR dependency, shown in Fig. 1b, can be written as $\iota_{23}\left(\delta_2(s_*^c, s_{53}^{DNS}), \delta_3(s_*^c, s_{80}^{lbs})\right)$.

3 Workflow Mining

Normalized cross correlation provides an heuristic approach for estimating workflow events and the result is described by a probability space (Ω, F, P_ϱ) described in Sect. 2. Based on the probability space, we define the problem of mining workflows as detecting the most likely sequence of hidden states. This is a problem often associated with HMMs. Using HMMs, it is possible to identify the probability whether a specific workflow is in place or not. We are interested in the most likely sequence of workflows in a given communication data network. The most likely sequence of hidden states can be calculated using the dynamic programming Viterbi algorithm [4]. An HMM $\lambda = (a_{ij}, e_{\iota_{kl}}, \pi)$ representing a workflow is defined as follows:

- n states $\Omega = \{\iota_0, \ldots, \iota_{n-1}\}$,
- an alphabet $\Delta = \{\delta_0, \ldots, \delta_{m-1}\}$ of m symbols,
- a transition probability matrix $a_{ij} = \iota_i \times \iota_j$,
- emission probabilities $e_{\iota_{kl}}(\delta_l)$ representing the probability of a state ι_{kl} emitting symbol δ_l and
- initial state distribution vector $\pi = \pi_o$.

We refer to a sequence of observed symbols as $O = \delta_0, \delta_1, \delta_2, \ldots$ and a sequence of states as $Q = \iota_0, \iota_1, \iota_2, \ldots$. Based on tumbling windows $w_{t_i} \in W$

$$W = w_{t_0}, \ldots, w_{t_i}, \ldots, w_{t_{n-1}} \tag{2}$$

with a shift Δ_t, we derive indirect dependency events based on observed direct dependencies. Direct dependencies imply that network packets are exchanged between two network services. Normalized cross correlation is a heuristic approach, hence observed workflow events can be untrue and existing indirect dependencies might not be detected. However, repeatedly reoccurring workflow events are very likely to be actual workflow events.

The parameters of a_{ij} and $e_{\iota_{kl}}(\delta_l)$ the HMM $\lambda = (a_{ij}, e_{\iota_{kl}}, \pi)$ can be learned over multiple tumbling windows $w_{t_i} \in W$ and $t_i \in [t_0, \ldots, t_p]$ by:

$$a_{ij} = \frac{A_{ij}}{\sum_{q=\{0,\ldots,p\}} A_{iq}}, \tag{3}$$

where A_{ij} is the number of observed state transitions from state ι_i to ι_j over p tumbling windows and it is normalized over all of ι_i's outgoing state transitions. An emission probability $e_{\iota_{kl}}(\delta_l)$ is derived as

$$e_{\iota_{kl}}(\delta_l) = \frac{E_{\iota_{kl}}(\delta_l)}{\sum'_{\forall \iota_{xl}, \delta_l \subset \iota_{xl}} E_{\iota_{xl}}(\delta_l)}, \tag{4}$$

where $E_{\iota_{kl}}(\delta_l)$ is the number of times that state ι_{kl} is observed, when symbol δ_l is emitted. This is normalized over the number of occurances of all states $\iota_{xl} \subset \delta_l, \iota_{xl} \in F$, which also emit symbol δ_l. By observed network traffic within an monitored network traffic, this HMM allows us to automatically derive workflows based on network service dependency analysis. This introduced methodology is applied to a real-life network and the results of this experimental evaluation are shown in the next section.

4 Experimental Evaluation

The disaster recovery site of an energy distribution network, provided an Italian water and energy distribution company, was available for non-invasive experimentation. We integrate our framework into this test network to test the ability of our newly introduced workflow mining approach to rediscover workflows based on network traffic. The implementation of our introduced methodology works online and continuously analysis network traffic, which is mirrored by routers and switches in the test environment. Based on this set-up, we are able to analyze detected network service dependencies, which constitute workflow events and evaluate, whether our novel workflow mining approach is able to rediscover workflows. Figure 2 shows all network service dependencies detected by MONA. Based on Supervisory Control and Data Acquisition (SCADA) protocols, remote terminal units TTY-T116 to TTY-T164 in substations of medium voltage and high voltage, acquire data from electrical devices (e.g., programmable logic

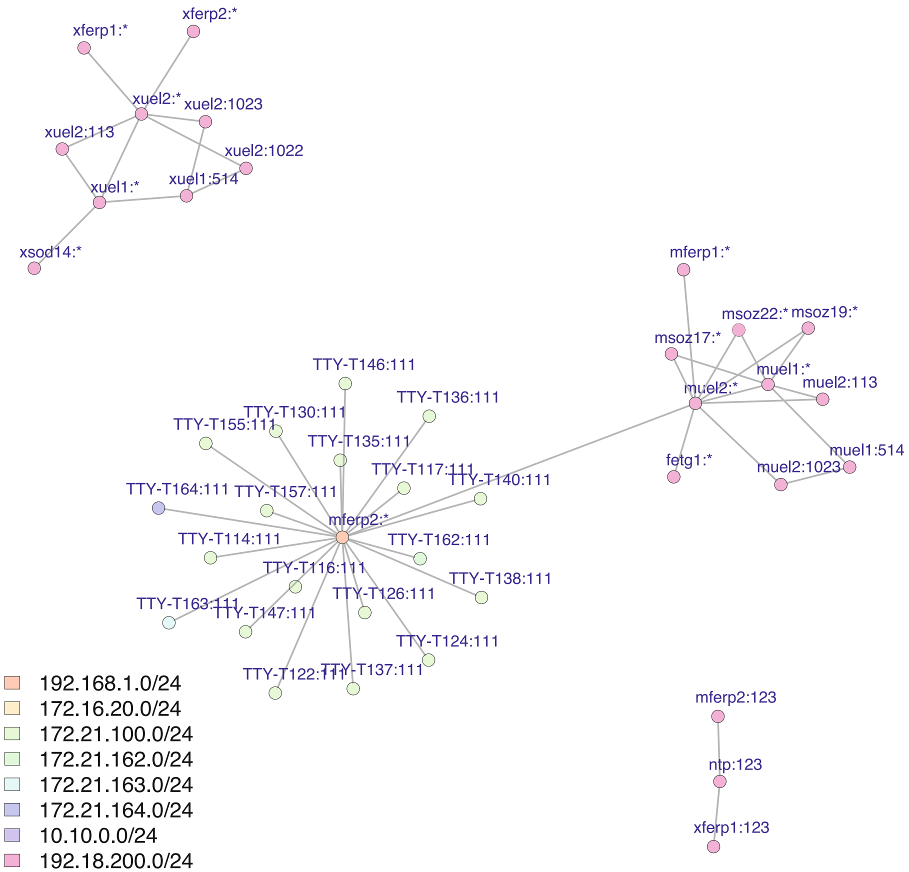


Fig. 2. Network service dependency analysis in an energy distribution network.

controllers, sensors, etc.), and send them via front end servers mferp1, mferp2 to the supervisory scada servers muel1 and muel2 of the power grids main office. These network service dependencies were classified as complete and correctly identified by network operators.

Figure 3 shows an excerpt of workflows derived based on network service dependencies. These workflows are plotted in BPMN 2.0 [12]. To point out workflows spanning multiple subnetworks, we model subnetworks as swimlanes. Our experimental analysis consists of comparing automatically derived workflows to workflows provided by network operators beforehand. Based on this analysis, we conclude whether our introduced online workflow mining approach is able to rediscover workflows. This experimental evaluation showed that our introduced workflow mining approach is able to rediscover workflows based on network traffic.

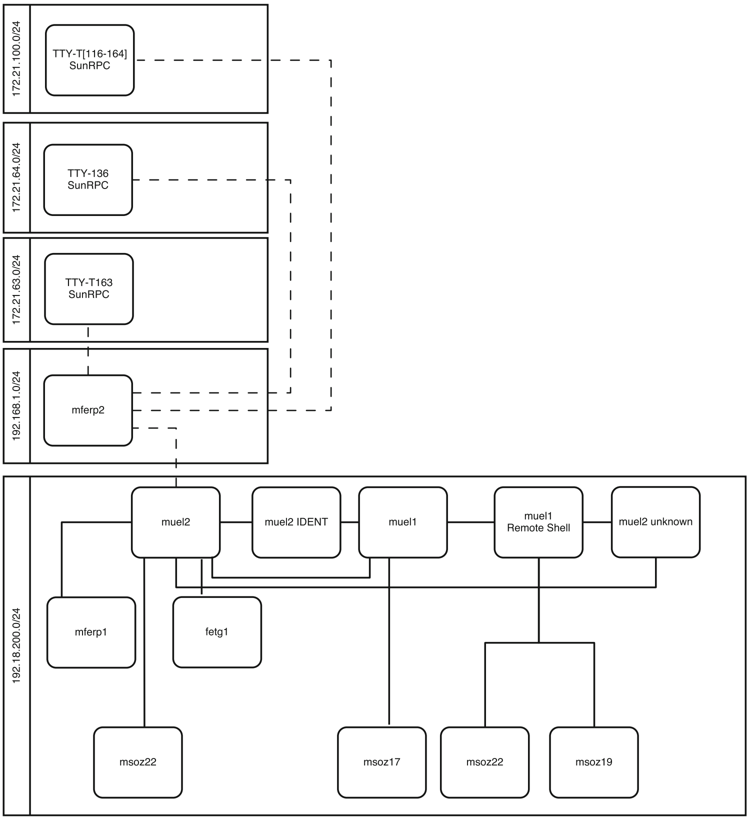


Fig. 3. Excerpt of workflows derived from network traffic in an energy distribution network.

5 Conclusion

We introduced an approach to mine workflows online, based on network traffic via network service dependency discovery. To the best of our knowledge this is the first workflow mining approach, which is able do deduce a HMM based workflow model by analyzing network traffic. We integrate this online workflow mining approach into the data-communication network of an energy distribution network. In the context of our experimental evaluation, we came to the conclusion that network operators have a high level understanding of workflows in their monitored network. However, they lack a detailed understanding on what applications and network services are involved. This was generally due to this network relying heavily on third party software that are often also updated and maintained by the third party. Thus, we concluded that deriving manual workflow models is costly and requires specialist know how. Luckily, network traffic based workflow mining support network operators in understanding workflows in their monitored network on application layer level.

Acknowledgments. This work was partly supported by the Seventh Framework Programme (FP7) of the European Commission as part of the PANOPTESEC integrated research project (GA 610416).

References

1. Bahl, P., Chandra, R., Greenberg, A., Kandula, S., Maltz, D.A., Zhang, M.: Towards highly reliable enterprise network services via inference of multi-level dependencies. In: ACM SIGCOMM Computer Communication Review, vol. 37, pp. 13–24. ACM (2007)
2. Blum, T., Padoy, N., Feußner, H., Navab, N.: Workflow mining for visualization and analysis of surgeries. Int. J. Comput. Assist. Radiol. Surg. **3**(5), 379–386 (2008)
3. Chen, X., Zhang, M., Mao, Z.M., Bahl, P.: Automating network application dependency discovery: experiences, limitations, and new solutions. OSDI **8**, 117–130 (2008)
4. Forney, G.D.: The viterbi algorithm. Proc. IEEE **61**(3), 268–278 (1973)
5. Herbst, J.: A machine learning approach to workflow management. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 183–194. Springer, Heidelberg (2000)
6. Kolmogorov, A.N.: Foundations of the Theory of Probability. Chelsea publishing company, New York (1950)
7. Mona Lange, R.M.: Time Series data mining for network service dependency analysis. In: The 9th International Conference on Computational Intelligence in Security for Information Systems. Springer, Heidelberg (2016)
8. Natarajan, A., Ning, P., Liu, Y., Jajodia, S., Hutchinson, S.E.: NSDMiner: automated discovery of network service dependencies. IEEE (2012)
9. Priyadharshini, V., Malathi, A.: Analysis of process mining model for software reliability dataset using HMM. Indian J. Sci. Technol. 9(4), 1–5 (2016)
10. Rozinat, A., Veloso, M., van der Aalst, W.M.: Using hidden markov models to evaluate the quality of discovered process models. Extended Version. BPM Center Report BPM-08-10, BPMcenter. org (2008)

11. Silva, R., Zhang, J., Shanahan, J.G.: Probabilistic workflow mining. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 275–284. ACM (2005)
12. Silver, B., Richard, B.: BPMN Method and Style, vol. 2. Cody-Cassidy Press, Aptos (2009)
13. Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). doi:10.1007/978-3-642-28108-2_19
14. Aalst, W.M.P.: Business process management demystified: a tutorial on models, systems and standards for workflow management. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) ACPN 2003. LNCS, pp. 1–65. Springer, Heidelberg (2004). doi:10.1007/978-3-540-27755-2_1
15. Van Der Aalst, W.M.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011)
16. Van Der Aalst, W.M., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.: Workflow mining: a survey of issues and approaches. Data Knowl. Eng. **47**(2), 237–267 (2003)
17. Van Der Aalst, W.M., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9), 1128–1142 (2004)
18. Dongen, B.F., Aalst, W.M.P.: Multi-phase process mining: building instance graphs. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) ER 2004. LNCS, pp. 362–376. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30464-7_29