Process mining applications in software engineering

Brian Keith[1] and Vianca Vega[1]

[1] Computing and Systems Engineering Department, Universidad Católica del Norte, Chile
{brian.keith, vvega}@ucn.cl

Abstract. Process mining is a field that uses elements from data mining and business process modeling to do tasks such as process discovery, conformance checking, and process improvement. This paper presents a study about the application of process mining techniques in the software development process. It shows a series of case studies that illustrate possible applications in the process and the product. Also, the main current challenges in applying process mining in software engineering are described. The objective of this paper is to show the importance and practical usefulness of applying process mining approaches in software engineering. The main result of this study is the fact that using process mining facilitates software process evaluation and auditing. The development of a methodology for applying process mining in software engineering is proposed as future work, considering the main challenges described previously.

Keywords: process mining, software development, process assessment, process improvement.

1 Introduction

Process mining is a discipline that lies between machine learning and data mining, and, process modelling and analysis. Its main objective is to discover processes, do conformance checking, and process improvement. Process mining seeks automatizing these three tasks by applying data mining techniques specially designed for dealing with process data [1-2].

Analyzing a process automatically and determining where deviations are produced may be used for taking corrective actions on how processes are done. This corresponds to the process improvement phase, where corresponding improvements are proposed, as a function of the information obtained [1].

Cook & Wolf in 1998 [3] intended to build software process models from data obtained from event records. Since then, process mining has greatly evolved and has become a field in itself, currently having applications beyond software engineering. Several studies applied in software engineering have been conducted, some of which are shown in the references [4-16].

Process mining applied in software engineering aims to solve monitoring and control problems and improving development processes, one of the critical problems

in software development [17]. In general, an improvement in the quality of the development process is expected to result in a better product quality. This reinforces the usefulness of applying process mining techniques in this field since it would eventually allow developing higher quality software.

Another relevant aspect to consider is that software processes are not usually explicitly modelled. Manuals supporting development work contain guides and abstract procedures. Therefore, there is a big gap between the "actual" process and the "official" process [17-18]. This gap may be measured with process mining techniques, allowing correction and improving the process if necessary.

According to the above and under the perspective that software processes may be considered as software in themselves [19], the application of process mining techniques could be seen as equivalent to the application of automatized testing methods in software development.

Based on the review of process mining studies in software engineering, two main application dimensions are observed: in the process (process evaluation and improvement [3, 4, 6, 9, 11, 12, 13, 14, 16] and process support [5, 7, 15]) and in the product [8, 10].

The purpose of this study is to examine process mining applications in software engineering. Some studies review and analyze the use of data mining techniques in software engineering as a whole (e.g., the systematic review in [20]); however, none of these literature reviews focuses on the application of process mining techniques in existing software processes.

This paper is organized as follows: first, the basic concepts of process mining are explained. Then, a review of the main process mining applications in software engineering is presented. Next, some specific applications are described in detail as case studies, analyzing their main results. Afterwards, some challenges process mining faces in this field are explained. Finally, the main results are summarized and the conclusions obtained in this study are shown.

2 Process Mining and Its Application in Software Engineering

The basic task of process mining is process discovery. This task begins with a dataset consisting of a record of all events occurring during the process development. With a sufficient amount of these data, it is possible to infer the process model, with adjustable levels of detail [1].

Process discovery automatization enables organizations to eliminate the problem of process modelling conducted by experts because, in many cases, assessments are influenced by the modeler's own bias [26]. In addition, existing algorithms allow obtaining greater levels of detail or more abstract models omitting certain cases, according to the desire of the final user of the model [1]. Certainly, the quality of the resulting process models is a function of the quality and quantity of available data [22]. Given the nature of available data, the disadvantage is that the algorithm only includes examples that have occurred, but it does not have examples of processes that cannot occur. The lack of negative examples is one of the weaknesses of these approaches [1].

Conformance checking begins with a defined formal process model and an instance of the actual process, and generates an analysis of the actual process deviation as compared to the ideal process [23]. This is very useful for auditing and has been successfully applied previously in different organizations [24]. Conformance checking allows determining how faithfully the process model is followed with respect to its formal definition. There are several methods to do this task. All of them allow defining the level of deviation of an actual process with respect to the formal process and can determine in what step of the process deviation occurs. This is applied in process control and auditing of organizations [1, 24].

Finally, process improvement is based on the results obtained from the previous tasks. It begins with the actual process model and a performance indicator, generating a new process model which should improve results as compared to the original model [1]. A simple example is the elimination of a bottleneck detected via process discovery. It is possible to use simulation techniques to analyze possible improvements proposed and compare them according to a well-defined metric [25].

Process mining has been successfully used in various application domains [27-31]. There is an IEEE process mining community including more than 60 organizations [4]. However, despite process mining success, a limitation is that techniques are seldom used in operational environments. It has been shown that it is possible to use process mining for supporting operational decisions [31].

The big amount of data generated by the software development process allows applying data analysis techniques and using its results to guide process optimization. There are several examples of data mining techniques for software process analysis and improvement [20].

In order to apply process mining techniques, data available must fit the metamodel described in Fig. 1. Each process consists of activities and is associated with a series of process instances. Each process instance consists of one or more events. These events must indicate the activity they correspond to and have a description, a time to be executed, and a person in charge. It is possible to enrich the event record with more data, as available [16].

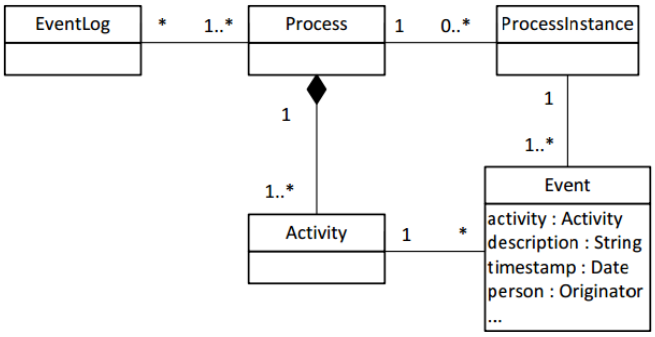

Fig. 1. Process mining metamodel [16]

Models may be worked out through different formal notations, from lower level models corresponding to transition diagrams to more advanced notations such as Petri networks, BPMN or UML [32].

Table 1 shows the dimensions of possible applications of process mining applications in software engineering.

Table 1. Process mining applications in software engineering

Dimension	*Application*	*Example*	*References*
Process	Process evaluation and improvement	Finding discrepancies between the formal and actual process.	[3], [4], [6], [9], [11], [12], [13], [14] & [16]
	Process support	Process discovery as a complement to requirement elicitation.	[5], [7] & [15]
Product	In the product	Analyzing users' behavior.	[8] & [10]

Fig. 2 illustrates some of the possible process mining applications in the context of software development. The life cycle presented is generic and follows the phases of the Rational Unified Process (RUP) [33].

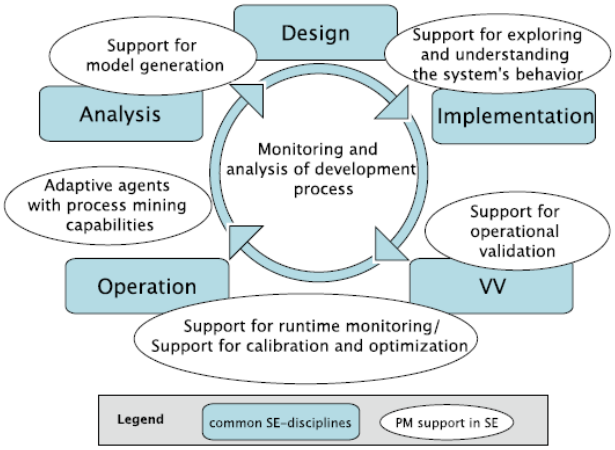

Fig. 2. Process mining applications in RUP [5]

In software engineering, there are several sources from which data could be obtained (e.g., version control repositories, bug tracking systems, and e-mail). Owing to the many available sources, a powerful step to prepare data is required before applying data mining procedures [16].

The broad applicability of process mining in software engineering is due to its generic techniques. These can be applied in any type of process, if correct data are available. Process mining can also be used for mining the use of software products developed. A case study analyzes two industrial applications, particularly a reservations system and a travel portal [8].

There are methodological proposals to apply process mining in software engineering, along with frameworks particularly designed to facilitate these tasks [9, 16, 35, 36]. Nevertheless, the development of a standard methodology is still an open problem in this area. This is due to the fact that software process intelligence is a very recent field still evolving [36].

3 Case Studies

This section shows the results of some studies in the literature. These studies were selected on the basis of their illustrative value. The case studies have been classified according to the process involved. Only the main results are shown here, in-depth detail of the implementations and modeling can be found in the referenced works.

Configuration management process. SCM data can be used to build explicit process models. This corresponds to an example of process discovery [9]. Using the audit data generated by the configuration management system, which correspond to process instances, the model can be derived from them using process mining algorithms. This model may be verified by a process engineer, who must later decide what changes must be made to optimize and manage it better [9].

By complementing process data with the person in charge of its activity, it is possible to build a model of the social network of the process and also analyze individuals who do each activity [1]. The application of these techniques reveals that, in some cases, people in charge of code development are the same individuals who are later in charge of its testing. This is certainly not desirable because, in ideal situations, they should be different people [9].

Testing process. It is possible to apply process mining techniques to obtain models from testing scripts used during the testing phase of software development. This allows generating new testing cases and facilitates understanding the different testing cases used [7]. Another study proposes the use of process mining techniques to analyze software systems online under real circumstances. The objective is to learn as much as possible from information obtained from these data in execution time and then use this knowledge to diagnose problems and recommend actions [10].

The first study [7] is an example of a process mining application as a supporting tool in the process because the purpose is to support the testing phase, but not to analyze the process. The second study [10] is an example of process support and also an application in the product because process mining tools are directly applied in the software to support the testing phase.

Change control process. This study fully describes the application of process mining techniques in data generated by the change control process in charge of a Change Control Committee in the context of a software development project [4].

One of the main results obtained revealed that in most process instances (about 70%) the analytical phase was omitted and change requests were directly solved without a previous analysis. This indicates that the omission of the analytical task is a structural problem in this organization [4]. This type of discrepancy must be detected automatically by applying process mining because it provides relevant knowledge about how processes operate in the development processes of an organization.

CMMI conformity assessment. This case corresponds to the application of conformity review techniques to evaluate a process according to CMMI [12, 13]. Studies have been previously done to assess conformity as compared to other

standards. For example, an Australian software development company was assessed by taking standard ISO/IEC 12207 as a reference in 2006 [38].

CMMI model describes the typical elements processes must include to be effective [37]. In general, CMMI assessment consists of two steps: collecting data and then interpreting them according to the CMMI model [39]. Process assessment can be qualitative or quantitative, and it may rely on expert judgment [40]. Process mining may help in the data collecting phase. The use of automated process mining tools can help create objective assessments by reducing human intervention [1, 13].

Another aspect to which process mining can contribute is CMMI assessment cost reduction. One of the limitations of small environments in the application of CMMI process assessment is its high cost. The same study proposes a tool based on process mining which, once configured, constantly assesses the conformity of development processes with CMMI. This tool, called Jidoka4CMMI, provides a framework that defines testing cases and verifies if a process fits CMMI recommendations [13].

Another study applied process mining techniques to evaluate the real process done by developers in an organization. Then, discrepancies found between the actual process and the model described were assessed. The results of this assessment became a plan to work and improve the process [14].

4 Challenges

Some of the general challenges faced by process mining in the field of software engineering are presented below [1, 2, 13, 17, 22, 40, 41, 44].

Process knowledge. In many companies, information systems are agnostic to business processes. Actions done by the systems are kept in some kind of record archive, but the systems do not identify the process or process instance they correspond to [1, 41]. The problem of applying process mining in records generated by process-agnostic systems is one of the main challenges faced by process mining when applying it in the entrepreneurial world. Some of the problems emerging during data acquisition can be solved by defining and using standards such as MXML [42] or OpenXES [43], both based on XML [41].

In the case of process-agnostic systems, their records may not have enough data to apply process mining algorithms. In some cases, the opposite may occur since the amount of details is excessive and the complexity of the resulting model makes it impractical. Every good model must keep a balance between precision and the representation of reality and computing simplicity [2, 41].

Data quality. Process mining strongly depends on the quality of data collected and stored. Big amounts of data must be frequently filtered due to incomplete process instances. Nevertheless, useful datasets can be obtained through careful data preparation, although they were not data originally stored to apply process mining. In spite of this, process mining efficiency and effectiveness can be substantially benefited from well-structured and well-defined datasets and also collection guides that allow obtaining high-quality datasets [17, 22].

Another problem in datasets is noise. This may sometimes arise due to unexpected processes (activities which are not supposed to occur or occurring in incorrect positions) [41]. The integrity and quality of the data is an important factor in the quality of process assessment results [40], this becomes even more important in conjunction with the natural necessities of process mining techniques.

Tool integration. All software process improvement methodologies share a common pre-requirement: the process and the product must be constantly measured. To do this, development process data must be continuously collected and analyzed. Nevertheless, manual data collection and analysis require an important additional effort [13].

A key requirement for software process follow-up is that they should not interfere with the process under assessment. This is seldom the case for traditional approaches because they require the active participation of all process actors so that they can report their activities. An alternative approach is automatic data collection in work environments (e.g., IDEs and design tools). The use of these integrated tools may facilitate data obtainment to apply process mining techniques without interfering with software developers' work [13].

Result evaluation. Another problem is the assessment of the resulting process because it is necessary to define a standard and rigorous procedure to assess the quality of the output generated by the process mining algorithm [1]. It is possible to measure the quality of the model through the discrepancy between the records of original events and the model obtained. Another problem is to choose an adequate model representation because, ultimately, the model will not be useful if it is not possible to understand and analyze it properly [41].

Another challenge for the implementation of process mining techniques is that algorithms must be properly parametrized. The search of optimum parametrization can be a complex task due to the number of available parameters [1, 41].

Usability. According to the Process Mining Manifesto [2], one of the challenges that must be faced to improve process mining usability is its integration with other methodologies and analytical techniques. For example, simulation tools may complement process analysis by assessing alternative implementations [44].

In general, a redesign project requires more than one instrument. Although process mining provides tools for diagnosing and analyzing processes, they must be complemented with other methodologies (e.g. simulation) or process improvement tools to allow understanding and planning the redesign process properly [1, 2, 44].

5 Conclusions

This study addresses the main aspects necessary to understand the importance of process mining and its application in the field of software engineering. In addition, the different challenges and problems faced by process mining are identified. On the other hand, the study shows that process mining may be applied in the development process and also in the product developed. Several applications illustrating these

concepts are shown. Given the general character of process mining algorithms, it is possible to apply them in almost all entities following a sequence of logical steps if data are in a proper format.

Process mining is an area with a great potential of applications in the field of software engineering, particularly in the study of software development processes. Since process mining is a relatively new field as compared with the disciplines forming it, its growth potential is quite high, both on a theoretical and practical basis.

The capacity to discover actual processes, assess its conformity with respect to the official model, and find improvement opportunities, make process mining cause a high impact on an organizational basis. Case studies show that it is possible to use process mining to detect process discrepancies, obtaining many data that can be used for planning eventual corrections and improvements.

The challenges that must be faced in the field of process mining are also described. One of the main challenges is the availability of suitable data provided by information systems aware of business processes. This idea may be generalized to continuously assess process performance. This may allow managers to do a better job since they can be warned about possible process deviations and problems appearing in the real-time process in due time.

Although there are methodologies to apply process mining techniques in general, the development of a standard methodology to implement process mining techniques in organizations is still an open problem. Particularly, the need of a methodology is considered to explain the development of an activity to incorporate process knowledge to information systems naturally agnostic to it. Once this is done, tasks would somehow follow the traditional data mining approach, for which several methodologies are available.

Finally, the importance of process mining and automatic approaches of process assessment are shown both on a general basis and in software development processes.

References

1. W. Van Der Aalst, “Process mining: discovery, conformance and enhancement of business processes”. Springer Science & Business Media, 2011.
2. W. Van Der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. van den Brand, R. Brandtjen, J. Buijs et al., “Process mining manifesto,” in Business process management workshops. Springer, 2011, pp. 169–194.
3. J. E. Cook and A. L. Wolf, “Discovering models of software processes from event-based data”, ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 7, no. 3, pp. 215–249, 1998.
4. J. Šamalíková, J. J. Trienekens, R. J. Kusters, and A. T. Weijters, “Discovering changes of the change control board process during a software development project using process mining,” in Software Process Improvement. Springer, 2009, pp. 128–136.
5. L. Cabac and N. Denz, “Net components for the integration of process mining into agent-oriented software engineering,” in Transactions on Petri Nets and Other Models of Concurrency I. Springer, 2008, pp. 86–103.
6. A. Colombo, E. Damiani, and G. Gianini, “Discovering the software process by means of stochastic workflow analysis,” Journal of Systems Architecture, vol. 52, no. 11, pp. 684–692, 2006.

7. V. Shah, C. Khadke, and S. Rana, “Mining process models and architectural components from test cases,” in Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on. IEEE, 2015, pp. 1–6.
8. V. A. Rubin, A. A. Mitsyuk, I. A. Lomazova, and W. M. van der Aalst, “Process mining can be applied to software too!” in Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, 2014, p. 57.
9. V. Rubin, C. W. Günther, W. M. Van Der Aalst, E. Kindler, B. F. Van Dongen, and W. Schäfer, “Process mining framework for software processes,” in Software Process Dynamics and Agility. Springer, 2007, pp. 169–181.
10. W. v. d. Aalst, “Big software on the run: in vivo software analytics based on process mining (keynote),” in Proceedings of the 2015 International Conference on Software and System Process. ACM, 2015, pp. 1–5.
11. R. Santos, T. C. Oliveira et al., “Mining software development process variations,” in Proceedings of the 30th Annual ACM Symposium on Applied Computing. ACM, 2015, pp. 1657–1660.
12. J. Samalikova, R. J. Kusters, J. J. Trienekens, and A. Weijters, “Process mining support for capability maturity model integration-based software process assessment, in principle and in practice,” Journal of Software: Evolution and Process, vol. 26, no. 7, pp. 714–728, 2014.
13. S. Astromskis, A. Janes, A. Sillitti, and G. Succi, “Continuous CMMI assessment using non-invasive measurement and process mining,” International Journal of Software Engineering and Knowledge Engineering, vol. 24, no. 09, pp. 1255–1272, 2014.
14. M. Huo, H. Zhang, and R. Jeffery, “An exploratory study of process enactment as input to software process improvement,” in Proceedings of the 2006 international Workshop on Software Quality. ACM, 2006, pp. 39–44.
15. W. M. Van der Aalst, “Business alignment: using process mining as a tool for delta analysis and conformance testing,” Requirements Engineering, vol. 10, no. 3, pp. 198–211, 2005.
16. Poncin, Wouter; Serebrenik, Alexander; Van Den Brand, Mark. “Process mining software repositories,” in Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on. IEEE, 2011. pp. 5-14.
17. J. Samalikova, R. Kusters, J. Trienekens, T. Weijters, and P. Siemons, “Toward objective software process information: experiences from a case study,” Software Quality Journal, vol. 19, no. 1, pp. 101–120, 2011.
18. W. S. Humphrey, “A discipline for software engineering”. Addison-Wesley Longman Publishing Co., Inc., 1995.
19. L. Osterweil, “Software processes are software too,” in Proceedings of the 9th international conference on Software Engineering. IEEE Computer Society Press, 1987, pp. 2–13.
20. M. Halkidi, D. Spinellis, G. Tsatsaronis, and M. Vazirgiannis, “Data mining in software engineering,” Intelligent Data Analysis, vol. 15, no. 3, pp. 413–441, 2011.
21. W. M. van der Aalst, “Process mining in the large: A tutorial,” inBusinessIntelligence. Springer, 2014, pp. 33–76.
22. R. Bose, R. S. Mans, and W. M. van der Aalst, “Wanna improve process mining results?” in Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on. IEEE, 2013, pp. 127–134.
23. W. Van der Aalst, A. Adriansyah, and B. van Dongen, “Replaying history on process models for conformance checking and performance analysis,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 182–192, 2012.
24. W. M. van Aalst, K. M. van Hee, J. M. van Werf, and M. Verdonk, “Auditing 2.0: using process mining to support tomorrow’s auditor,” Computer, vol. 43, no. 3, pp. 90–93, 2010.

25. W. Van Der Aalst, “Process mining: overview and opportunities,” ACM Transactions on Management Information Systems (TMIS), vol. 3, no. 2, p. 7, 2012.
26. F. Gottschalk, W. M. van der Aalst, and M. H. Jansen-Vullers, “Mining reference process models and their configurations,” in On the Move to Meaningful Internet Systems: OTM 2008 Workshops. Springer, 2008, pp. 263–272.
27. R. Mans, M. Schonenberg, M. Song, W. M. van der Aalst, and P. J. Bakker, “Application of process mining in healthcare–a case study in a dutch hospital,” in Biomedical Engineering Systems and Technologies. Springer, 2008, pp. 425–438.
28. A. Mitsyuk, A. Kalenkova, S. Shershakov, and W. van der Aalst, “Using process mining for the analysis of an e-trade system: A case study,”, p. 15, 2014
29. N. Trcka and M. Pechenizkiy, “From local patterns to global models: Towards domain driven educational process mining,” in Intelligent Systems Design and Applications, 2009. ISDA’09. Ninth International Conference on. IEEE, 2009, pp. 1114–1119.
30. C. Zinski. “Internet Banking: The Trade Name Trend”. Banking & Financial Services Policy. 2001.
31. W. M. van der Aalst, M. Pesic, and M. Song, “Beyond process mining: from the past to present and future”, in Advanced Information Systems Engineering. Springer, 2010, pp. 38–52.
32. A. A. Kalenkova, W. M. van der Aalst, I. A. Lomazova, and V. A. Rubin, “Process mining using BPMN: relating event logs and process models,” Software & Systems Modeling, pp. 1–30, 2015.
33. P. Kruchten, “The rational unified process: an introduction”. Addison-Wesley Professional, 2004.
34. M. Jans, M. Alles, and M. Vasarhelyi, “The case for process mining in auditing: Sources of value added and areas of application,” International Journal of Accounting Information Systems, vol. 14, no. 1, pp. 1–20, 2013.
35. J. Stolfa, S. Stolfa, M. A. Kosinar, and V. Snasel, “Introduction to integration of the process mining to the knowledge framework for software processes,” in Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015. Springer, 2016, pp. 21–31.
36. A. Sureka, A. Kumar, and S. Gupta, “Ahaan: Software process intelligence: Mining software process data for extracting actionable information,” in Proceedings of the 8th India Software Engineering Conference. ACM, 2015, pp. 198–199.
37. H. Glazer, J. Dalton, D. Anderson, M. D. Konrad, and S. Shrum, “CMMI or agile: why not embrace both!” 2008.
38. I. ISO, “IEC 12207 systems and software engineering-software life cycle processes,” International Organization for Standardization: Geneva, 2008.
39. S. U. Team, “Standard CMMI appraisal method for process improvement (SCAMPI) a, version 1.3: Method definition document,” 2011.
40. Bourque, Pierre, et al. “Guide to the software engineering body of knowledge (40 (R)): Version 3.0”. IEEE Computer Society Press, 2014, ch. 8, sec. 3.2, p. 154.
41. A. Burattin, “Obstacles to applying process mining in practice”, in Process Mining Techniques in Business Environments. Springer, 2015, pp. 59–63.
42. W. M. van der Aalst, B. F. van Dongen, C. W. Günther, R. Mans, A. A. De Medeiros, A. Rozinat, V. Rubin, M. Song, H. Verbeek, and A. Weijters, “Prom 4.0: comprehensive support for real process analysis”, in Petri Nets and Other Models of Concurrency–ICATPN 2007. Springer, 2007, pp. 484–494.
43. C. W. Gunther, “XES standard definition”. www.xes-standard.org, 2009, Cited on, p. 72.
44. S. Aguirre, C. Parra, and J. Alvarado, “Combination of process mining and simulation techniques for business process redesign: a methodological approach”, in Data-Driven Process Discovery and Analysis. Springer, 2012, pp. 24–43.