

Process Mining bei Softwareprozessen: Ein Überblick

Ralf Kneuper, Dr. Ralf Kneuper Beratung, ralf@kneuper.de

1 Einführung

Process Mining umfasst eine Reihe von Methoden und Techniken, um Daten aus Ereignisdaten, den sogenannten Event Logs, zu analysieren und Informationen über die zu Grunde liegenden Prozesse und deren Bezug zu Prozessmodellen abzuleiten. Es ist verwandt mit Data Mining, betrachtet aber speziell die Analyse von Prozessdaten. Diese Beschränkung des Anwendungsgebietes ermöglicht es, spezifische Methoden und Algorithmen einzusetzen sowie spezifische, prozessbezogene Fragestellungen zu untersuchen wie beispielsweise den Vergleich des durchgeführten Prozesses mit dem definierten Prozess (im Rahmen der Qualitätssicherung) oder die Analyse von Engpässen.

Dieser Beitrag gibt einen kurzen Überblick über die verschiedenen Möglichkeiten, Process Mining im Zusammenhang mit Softwareprozessen einzusetzen.

2 Grundbegriffe des Process Mining

Nutzungsmöglichkeiten: Wie im Standardwerk [van11] zum Process Mining beschrieben, gibt es drei wesentliche Nutzungsmöglichkeiten dieses Ansatzes:

- **Entdeckung (Discovery):** Aus den verfügbaren Ereignisdaten wird der Prozess “entdeckt” oder identifiziert. Dies ist in erster Linie relevant, wenn es (noch) keine Beschreibung oder Definition des Prozesses gibt bzw. diese unvollständig ist. In diesem Fall wird beispielsweise der Kontrollfluss des tatsächlich durchgeführten Prozesses oder die zu Grunde liegende Organisationsstruktur abgeleitet.
- **Konformität (Conformance):** Der tatsächlich durchgeführte Prozess, wie er aus den Ereignisdaten erkennbar ist, wird mit dem definierten Prozess verglichen, um Abweichungen zu identifizieren. Auch in diesem Fall kann der Vergleich sich auf den Kontrollfluss des Prozesses (“Werden die richtigen Schritte in der richtigen Reihenfolge durchgeführt?”) oder auf die zu Grunde liegende Organisationsstruktur (“Werden die Schritte von der richtigen Rolle oder Organisationseinheit durchgeführt?”) beziehen. Hierbei handelt es sich also in erster Linie um ein Werkzeug zur Qualitätssicherung von Prozessen.
- **Verbesserung (Enhancement):** Aus den verfügbaren Ereignisdaten werden Prozessverbesserungspotentiale identifiziert. Diese können sich beispielsweise auf Engpässe im Prozess beziehen oder Geschäftsregeln für getroffene Entscheidungen ableiten und prüfen.

Eine eng mit Process Mining verwandte Disziplin mit großen Überschneidungen zum Process Mining ist (Business) Process Intelligence (BPI), eine Disziplin, die sich mit der Anwendung von Methoden und Techniken von Business Intelligence auf (Geschäfts-) Prozesse beschäftigt. BPI versucht, für Entscheidungen relevante Informationen aus Prozessdaten in eine Data Warehouse abzuleiten (siehe beispielsweise [CAM⁺09]). Beide Disziplinen haben große Überschneidungen und unterscheiden sich vor allem dadurch, dass BPI auch Metriken und ähnliche Informationen umfasst, andererseits bei der Analyse aber nicht unbedingt Bezug auf Prozessmodelle nimmt.

Auch wenn es mittlerweile weltweit verteilt eine Vielzahl von Gruppen gibt, die an Process Mining arbeiten und forschen, ist der Kern dieser Aktivitäten die Process Mining Group von Wil M.P. van der Aalst an der Universität Eindhoven, Niederlande. Van der Aalst ist auch der Autor des oben genannten Standardwerkes [van11] zum Thema Process Mining.

Werkzeuge: Da durch Process Mining meist große Datenmengen auszuwerten sind, ist diese Auswertung nur mit entsprechenden Werkzeugen realistisch möglich. Das mit Abstand am weitesten verbreitete Werkzeug für diese Aufgabe ist das von der Process Mining Group der Universität Eindhoven entwickelte Open-Source-Werkzeug ProM [Pro]. Zu ProM wurden auch von vielen anderen Forschungsgruppen Plugins entwickelt, so dass ein sehr hoher Anteil aller existierenden Methoden und Techniken zum Process Mining von ProM unterstützt werden. Leider ist die Bedienung von ProM allerdings recht gewöhnungsbedürftig, und die Dokumentation, insbesondere der enthaltenen Plugins teilweise sehr knapp.

Nicht ganz so mächtig, dafür aber gerade für Einsteiger sehr viel einfacher zu bedienen, ist das kommerzielle Werkzeug Disco [Flu] von Fluxicon, einem Unternehmen, das von ehemaligen Mitarbeitern der Process Mining Group der Universität Eindhoven gegründet wurde.

Ereignisprotokolle: Grundlage des Process Minings sind Ereignisprotokolle, die Informationen zu den Ereignissen enthalten, die bei der Ausführung eines Prozesses aufgetreten sind. Gemäß [van11, § 4.2] basiert ein Ereignisprotokoll auf folgenden Annahmen:

- Ein Prozess (bzw. seine Ausführung) besteht aus Fällen, beispielsweise kann der Prozess der Softwareentwicklung in einzelne Anforderungen gegliedert werden. Ein Fall besteht dann aus der Bearbeitung einer solchen Anforderung.

- Ein Fall besteht aus verschiedenen Ereignissen, die jeweils genau einem Fall zuzuordnen sind. Bei der Softwareentwicklung ist ein Ereignis dann beispielsweise die Analyse, Genehmigung oder Implementierung einer Anforderung.
- Ereignisse, die zu einem Fall gehören, sind geordnet. Aus dieser Ordnung lässt sich dann der im Prozess bestehende Kontrollfluss ableiten.
- Ereignisse können Attribute haben wie die dem Ereignis zugeordnete Aktivität, Zeitpunkt, Kosten oder beteiligte Ressource(n).

Daraus ergibt sich, dass ein für Process Mining zu verwendendes Ereignisprotokoll mindestens die relevanten Fälle, deren Reihenfolge sowie die zugeordneten Aktivitäten protokollieren muss, für weiterführende Aktivitäten auch die genannten weiteren Attribute.

Um derartige Ereignisprotokolle mit verschiedenen Werkzeugen generieren und bearbeiten zu können, wurde als Standardformat zuerst Mining eXtensible Markup Language (MXML) [Pro09] definiert, das mittlerweile durch das Format eXtensible Event Stream (XES) abgelöst wurde, siehe [van11, § 4.3] und [XES].

Das Werkzeug ProM hat einige Importfunktionen, um aus anderen Formaten derartige XES-Protokolle generieren zu können. Ein einfaches, XLS2XES genanntes Konvertierungswerkzeug, um aus Excel-Tabellen XES-Ereignisprotokolle zu erstellen, wurde vom Autor entwickelt und ist verfügbar unter <http://kneuper.de/ProcessMining/>.

3 Process Mining und Softwareentwicklungsprozesse

Obwohl die ersten Veröffentlichungen zu Process Mining (wenn auch noch nicht unter dieser Bezeichnung) sich auf Softwareprozesse bezogen, siehe [CW95, CW98], wurde Process Mining seither in erster Linie auf Geschäftsprozesse angewendet. Die Anwendung der Ideen und Konzepte von Process Mining auf das IT-Servicemanagement ist relativ einfach möglich, da es sich hier um relativ strukturierte, wiederholbare und häufig wiederholte Prozesse handelt. Aus diesem Grund bezog sich auch der *Business Process Intelligence Challenge* in den Jahren 2013 und 2014 auf IT-Servicemanagementprozesse, siehe [BPI13, BPI14]. Im IT-Servicemanagement kann Process Mining beispielsweise genutzt werden, um die Einhaltung von Service Level Agreements (SLA) zu überprüfen bzw. die Ursachen bei Nicht-Einhaltung zu analysieren. Hier gibt es gewisse Überschneidungen mit klassischen Kennzahlensystemen für das IT-Servicemanagement, aber durch den stärkeren Bezug auf den gesamten Prozess an Stelle einzelner Prozessschritte geht das Process Mining an dieser Stelle darüber hinaus.

Softwareentwicklungsprozesse sind allerdings meist weniger strukturiert und werden seltener durch-

geführt, so dass die Anwendung von Process Mining hier entsprechend schwieriger ist.

Einige Ansätze zur Nutzung von Process Mining im Bereich der Softwareentwicklung wurden allerdings gemacht, die über die Anwendung auf Softwareentwicklungsprozesse hinausgehen:

- Anwendung auf die *Softwareentwicklungsprozesse* selbst: dies ist die bereits angesprochene und offensichtlichste und direkteste Form. Sie wird in Abschnitt 4 unten ausführlicher behandelt.
- Nutzung von Process Mining als Werkzeug zur *Analyse von Anforderungen*: Dieser Ansatz wird in Abschnitt 5 ausführlicher behandelt.
- Analyse der *Ausführung* eines Softwaresystems, zum Beispiel zur Analyse und Steigerung der Performanz: Voraussetzung dafür ist, dass das System entsprechend instrumentiert wurde, um seine Ausführung in einem Event Log zu protokollieren. Ein sehr ähnlicher Ansatz, allerdings nicht unter dem Namen Process Mining, ist beispielsweise in [LMP08] beschrieben. Ansonsten sind dem Autor dieses Beitrags keine Veröffentlichungen zu diesem Ansatz bekannt, auch wenn er wiederholt in verschiedenen Diskussionsforen diskutiert wurde.

Ebenfalls verwandt mit Process Mining, auch wenn es sich nicht direkt auf Prozesse bezieht, ist das Mining von Software Repositories (MSR). Dieses Thema wird u.a. in einer Reihe von Teilkonferenzen der ICSE-Konferenzen (siehe <http://msrconf.org>) behandelt, mit einem breiten Spektrum von Fragen. Dieses umfasst Fragen zu den Entwicklern (z.B. "Welche Arten von Fragen stellen Entwickler in relevanten Foren? Wie kooperieren Entwickler in Projekten?"), zu Produkten und deren Qualität (z.B. "Welche Arten von Fehlern und Warnungen treten häufig auf?") und schließlich auch zu den hier behandelten Prozessen.

4 Mining von Softwareentwicklungsprozessen

Die ersten Veröffentlichungen zu Process Mining bei Softwareentwicklungsprozessen waren [CW95, CW98], die drei verschiedene Methoden zur Entdeckung von Prozessen auf Basis von "algorithmischer Inferenz von Grammatiken", von Markov-Modellen und von neuronalen Netzwerken beschrieben.

Während die Algorithmen und Werkzeuge zum Process Mining bei Softwareentwicklungsprozessen im Wesentlichen die gleichen sind wie für allgemeine Geschäftsprozesse, sind die Datenquellen sowie die Vorbereitung der Daten deutlich anders. Bei Softwareentwicklungsprozessen sind die Daten meist in einem Repository enthalten, beispielsweise in Konfigurationsmanagement-Werkzeugen wie Git oder Subversion, oder Werkzeugen für Anforderungsmanagement und Bug Tracking, beispielsweise BugZilla

oder Jira. Sehr relevant, aber meist wenig strukturiert und daher schwierig zu analysieren sind Mail-Datenbanken oder Diskussionsforen innerhalb von Projekten. Dieses Thema wird beispielsweise in der bereits angesprochenen MSR-Konferenzreihe behandelt.

Wenn die Daten dann als Ereignisprotokoll aufbereitet sind, können sie ähnlich wie bei allgemeinen Geschäftsprozessen mit den üblichen Methoden des Process Mining analysiert werden.

Die Analyse von Repositories im Software Engineering, um daraus für das Process Mining geeignete Daten bzw. Ereignisprotokolle zu gewinnen, wurde insbesondere von Poncin in seiner Masterarbeit [Pon10] und den weiteren Publikationen [PSv11b, PSv11a] betrachtet. Als Teil dieser Arbeit entwickelte Poncin das Rahmenwerk FRASR, mit dem die Vorbereitung von Ereignisdaten aus verbreiteten Repositories wie Subversion und Git unterstützt wird.

Auf Basis dieser Vorbereitung von Ereignisdaten untersuchte Poncin dann beispielhaft die Zuordnung von Projektrollen zu Einzelpersonen, also die Frage, inwieweit man aus den Ereignisdaten ableiten kann, welche Person welche Prozessrolle innehat (beispielsweise bei einem Open Source-Projekt). Als zweites Beispiel untersuchte er, inwieweit die tatsächliche Bearbeitung von Fehlermeldungen einem definierten Lebenszyklus entspricht.

Eine ähnliche Anwendung von Process Mining beschreibt [GTK14]. Hier wird Process Mining als Werkzeug zur Prüfung von Softwareentwicklungsprozessen beschrieben mit den folgenden drei Komponenten:

- *Process Variant Miner*: Diese Komponente untersucht, welche Prozessvarianten in der Entwicklung tatsächlich umgesetzt wurden.
- *Conformance Analyser*: Diese Komponente vergleicht die Umsetzung des Prozesses gegen seine Spezifikation zur Überprüfung der Konformität.
- *Statistical Analyser*: Aufbauend auf den Ergebnissen der ersten beiden Komponenten wertet diese Komponente die Ergebnisse statistisch aus, um Aussagen über die Stabilität des Prozesses und seiner Varianten zu liefern.

Auf Grund der meist geringeren Strukturierung und damit verbundenen größeren Varianz der Softwareentwicklungsprozesse, sowie der geringeren Anzahl von Wiederholungen der gleichen Prozesse, ist die Anwendung von Process Mining hier nicht im gleichen Umfang wie beispielsweise im IT-Servicemanagement bei Prozessen wie Incident Management oder Change Management möglich. Auch in der Softwareentwicklung gibt es aber Prozesse, die häufig wiederholt werden, beispielsweise die Fehlerbehebung oder, bei iterativem Vorgehen, der Ablauf bei der Bearbeitung einer einzelnen Anforderung oder Story mit Analyse der Aufgabenstellung, Testfalldefinition, Design und Implementierung, Testdurchführung (meist in mehreren

Stufen), und Commit der Ergebnisse. In diesen Fällen ist damit auch Process Mining anwendbar, um beispielsweise Aussagen über die Konformität zum vereinbarten Vorgehen oder über Prozessschritte, bei denen häufig Verzögerungen auftreten, zu machen.

Sowohl [Pon10] also auch [GTK14] enthalten zusätzlich einen guten Überblick über relevante Literatur zum Process Mining von Softwareentwicklungsprozessen, so dass sie eine hilfreiche Basis für eine tiefergehende Einarbeitung in dieses Thema bilden.

5 Process Mining zur Anforderungsanalyse

In diesem Fall wird Process Mining nicht auf Softwareentwicklungsprozesse angewendet, sondern ist ein Schritt in einem solchen Prozess, der dabei hilft, Anforderungen zu identifizieren und zu analysieren. Dieser Ansatz wurde beispielsweise in [MFJv08] sowie [RLv14, RMLv14] behandelt.

Im Wesentlichen gibt es dabei zwei Anwendungsszenarien für Process Mining:

- Wenn ein neues System entwickelt werden soll, dann werden durch Process Mining die bisher genutzten Geschäftsprozesse analysiert, um zu verstehen, was die Benutzer tun und welche Schritte das neue System unterstützen könnte und sollte, beispielsweise um häufige Fehler zu vermeiden oder lange dauernde Prozesse zu beschleunigen.
- Wenn entsprechende Software bereits existiert und eingesetzt wird, dann kann Process Mining von Nutzungsprotokollen dabei helfen, die Nutzung dieser Software besser zu verstehen und dadurch die Software zu verbessern. Wie interagieren die Benutzer mit dem System, welche Schritte führen sie dabei aus, und welche typischen Probleme treten dadurch auf, sei es durch Fehlbedienung oder weil erwartete Funktionen nicht vorhanden sind?

Bei einer iterativen Vorgehensweise kann das zweite Szenario in gewissem Rahmen auch bei der Entwicklung neuer Software genutzt werden, da man hier schon sehr früh existierende Software hat, die testweise oder sogar produktiv genutzt werden kann. Auch hier wird die Nutzung dieser neuen Softwareversionen, beispielsweise am Ende eines Scrum-Sprints, protokolliert und kann dann für die Planung der folgenden Sprints durch Methoden des Process Mining ausgewertet werden. Diese Vorgehensweise ist beispielsweise in [RMLv14] ausführlicher beschrieben.

6 Ausblick

Während Process Mining für die Prozesse des IT-Servicemanagements wie beispielsweise Incident Management oder Change Management problemlos anwendbar ist und in Unternehmen gelegentlich auch

angewendet wird, gilt dies für Softwareentwicklungsprozesse in geringerem Umfang. Auch hier ist Process Mining allerdings zumindest auf manche Prozesse der Softwareentwicklung anwendbar, je nach Grad der Strukturierung des Prozesses mehr oder weniger leicht. Daneben kann Process Mining auch als ein Schritt innerhalb der Softwareentwicklung eingesetzt werden, um die Anforderungen an ein Softwaresystem zu analysieren.

Offene Fragen für die weitere Forschung in diesem Feld sind daher beispielsweise:

- Für welche Teilprozesse der Softwareentwicklung ist Process Mining geeignet, für welche eher nicht?
- Welche nützlichen Informationen können realistisch durch Process Mining abgeleitet werden?
- Ist Process Mining beispielsweise ein gutes Werkzeug für die Qualitätssicherung von Softwareprozessen?
- Welche Unterschiede gibt es beim Process Mining für unterschiedliche Typen von Prozessmodellen, z.B. zwischen Plan-getriebener und agiler Entwicklung?
- In welchen Fällen ist Process Mining ein geeignetes Werkzeug zur Analyse von Anforderungen, und wie kann es in diesen Fällen am besten in die Softwareentwicklung integriert werden?

Literatur

- [BPI13] BPIC: *Third International Business Process Intelligence Challenge (BPIC'13)*. <http://www.win.tue.nl/bpi/2013/challenge>, 2013.
- [BPI14] BPIC: *Third International Business Process Intelligence Challenge (BPIC'14)*. <http://www.win.tue.nl/bpi/2014/challenge>, 2014.
- [CAM⁺09] M. Castellanos, A.K. Alves De Medeiros, J. Mendling, B. Weber und A.J.M.M. Weijters: *Business process intelligence*. In: Cardoso J. und Wil M.P. van der Aalst (Herausgeber): *Handbook of research on business process modeling*, Information Science Reference, Seiten 456–480. Hersey, 2009.
- [CW95] Jonathan E. Cook und Alexander L. Wolf: *Automating Process Discovery through Event-Data Analysis*. In: *Proc. of the 17th Intern. Conf. on Software Engineering, Seattle, Washington, USA*, Seiten 73–82, 1995.
- [CW98] Jonathan E. Cook und Alexander L. Wolf: *Discovering Models of Software Processes from Event-Based Data*. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.
- [Flu] Fluxicon Process Laboratories: *Disco—Discover your processes*. <http://fluxicon.com/disco/>.
- [GTK14] Tuğba Gürgen, Ayca Tarhan und N. Alpay Karagz: *An Integrated Infrastructure Using Process Mining Techniques for Software Process Verification*. In: R. Perez-Castillo und M. Piattini (Herausgeber): *Uncovering Essential Software Artifacts through Business Process Archeology*, Kapitel 14, Seiten 364–382. IGI Global, 2014.
- [LMP08] Davide Lorenzoli, Leonardo Mariani und Mauro Pezzè: *Automatic Generation of Software Behavioral Models*. In: *Proc. of the 30th International Conference on Software Engineering (ICSE '08)*, Seiten 501–510. ACM, 2008.
- [MFJv08] Laura Maruster, Niels Faber, Ren J. Jorna und Rob J. F. van Haren: *A process mining approach to analyse user behaviour*. In: *WEBIST 2008, Proceedings of the Fourth International Conference on Web Information Systems and Technologies, Funchal, Madeira, Portugal*, Band 2, Seiten 208–214. INSTICC Press, 2008.
- [Pon10] Wouter Poncin: *Process Mining Software Repositories*. Diplomarbeit, Eindhoven University of Technology, August 2010. Verfügbar unter http://www.frasr.org/downloads/2010-08-20_ThesisWouterPoncin.pdf.
- [Pro] Process Mining Group, Eindhoven Technical University: *ProM Tools*. <http://www.promtools.org/doku.php>.
- [Pro09] Process Mining Group, Eindhoven University of Technology: *MXML (Mining eXtensible Markup Language)*. <http://www.processmining.org/logs/mxml>, 2009.
- [PSv11a] Wouter Poncin, Alexander Serebrenik und Mark van den Brand: *Mining student capstone projects with FRASR and ProM*. In: *SPLASH 2011 Educators' Symposium, Portland OR, USA*, 2011. Verfügbar unter http://www.frasr.org/downloads/2011-10_SPLASH-ETS.pdf.
- [PSv11b] Wouter Poncin, Alexander Serebrenik und Mark van den Brand: *Process mining*

software repositories. In: *15th European Conference on Software Maintenance and Reengineering, March 1-4, 2011, Oldenburg, Germany, 2011*. Verfügbar unter http://www.frasr.org/downloads/2011-03_CSMR.pdf.

- [RLv14] Vladimir A. Rubin, Irina A. Lomazova und Wil M.P. van der Aalst: *Agile Development with Software Process Mining*. In: *Proceedings of the 2014 International Conference on Software and System Process (ICSSP 2014)*, Seiten 70–74. ACM, 2014.
- [RMLv14] Vladimir A. Rubin, Alexey A. Mitsyuk, Irina A. Lomazova und Wil M. P. 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, ESEM '14*, Seiten 57:1–57:8. ACM, 2014.
- [van11] Wil M.P. van der Aalst: *Process Mining. Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.
- [XES] *XES Extensible Event Stream*. www.xes-standard.org.