

Ansätze zum Ähnlichkeitsabgleich von deklarativen Geschäftsprozessmodellen

Michaela Baumann¹, Michael Heinrich Baumann², Lars Ackermann¹, Stefan Schönig¹,
Stefan Jablonski¹

¹ Universität Bayreuth, Institut für Informatik, vorname.nachname@uni-bayreuth.de

² Universität Bayreuth, Institut für Mathematik, michael.baumann@uni-bayreuth.de

Abriss

Prozesse in Organisationen lassen sich grundsätzlich in zwei Klassen unterteilen: Strikte Routineprozesse und flexible Prozesse mit einem sich dynamisch entwickelnden, von Fall zu Fall unterschiedlichen Kontrollfluss. Aufgrund der unterschiedlichen Natur strikter und flexibler Prozesse entstanden auch zwei verschiedene Klassen von Prozessmodellierungssprachen: Imperative und deklarative. Während imperative Modelle die exakte Schritt für Schritt-Reihenfolge der Aufgaben in einem Prozess beschreiben, formulieren deklarative Modelle lediglich Regeln, welchen die Prozessausführung folgen muss. Aufgrund immer größer werdender Prozessmodellsammlungen ist es notwendig, unabhängig vom Modellierungsansatz, ähnliche Modelle zu erkennen und zusammenzuführen, was mit dem Begriff Ähnlichkeitsabgleich bezeichnet wird. Für imperative Prozessmodelle existiert bereits eine Vielzahl von Methoden zur Lösung dieses Problems. Da es für deklarative Prozessmodelle jedoch keine allgemein gültigen Lösungen gibt, werden in der vorliegenden Arbeit grundlegende Möglichkeiten des Ähnlichkeitsabgleichs diskutiert. Abschließend wird zu offenen Problemen und den nächsten Schritten Stellung genommen.

1 Einleitung

Das *Geschäftsprozessmanagement (GPM)* stellt eine etablierte Methode zur Modellierung, Ausführung und Analyse von Prozessen dar (Dumas et al. 2013). Prozesse in Organisationen lassen sich grundsätzlich in zwei Klassen unterteilen (Jablonski 1994): Strikte Routineprozesse mit eindeutig vorgeschriebenem und vorab spezifizierbarem Kontrollfluss und flexible, entscheidungsintensive Prozesse mit einem sich dynamisch entwickelnden und von Fall zu Fall unterschiedlichen Kontrollfluss. Strikte Prozesse stellen beispielsweise automatisierbare Verarbeitungsprozesse von Dokumenten dar. In der Kundenbetreuung kann man eher flexible Prozesse beobachten, welche von Kunde zu Kunde stark abweichen und sich an den jeweiligen Kundenbedürfnissen und der aktuellen Situation orientieren. Aufgrund der unterschiedlichen Natur strikter und flexibler Prozesse entstanden auch zwei Paradigmen von Prozessmodellierungssprachen: Imperative und deklarative. Erstere fokussieren das exakte

Beschreiben der Reihenfolge der Arbeitsschritte in einem Prozess. Deklarative Modelle hingegen formulieren lediglich Regeln für die Prozessausführung, wobei grundsätzlich jede Ausführung, die keine Regel verletzt, valide ist. Derartige Regeln können die Reihenfolge der Prozessschritte, aber auch die Abhängigkeiten derselben von Daten, Personen und Werkzeugen repräsentieren. Während sich Routineprozesse häufig besser mittels imperativer Modellierungssprachen repräsentieren lassen, eignen sich für flexible Prozesse meist eher deklarative Notationen (Pichler et al. 2012, van der Aalst et al. 2009). Die *Business Process Model and Notation (BPMN)* wird zur Gruppe der imperativen, die *Case Management Model and Notation (CMMN)* hingegen zu den deklarativen Modellierungssprachen gezählt.

Aufgrund der wegen Mehrfachmodellierungen immer größer werdenden Sammlungen von Prozessmodellen, wird es zunehmend schwieriger, diese zu verwalten. Mehrfachmodellierung kann dabei unter anderem durch verschiedene Versionen desselben Prozessmodells im Rahmen des *Business Process Reengineerings (BPR)* (Tka und Ghannouchi 2012), zielgruppenspezifische Modellanpassungen im Zuge der Internationalisierung (Dijkman et al. 2011) oder Unternehmensfusion verursacht werden. Zur Verwaltung großer Prozesssammlungen ist es notwendig, ähnliche Modelle zu erkennen und zusammenzuführen, was mit dem Begriff Ähnlichkeitsabgleich bezeichnet wird (Dijkman et al. 2011). Ein solcher Abgleich dient daneben auch der Wiederverwendbarkeit von Teilprozessen sowie dem Vergleich mit Referenzprozessen (Konformitätsprüfung) und Vorgaben (Einhaltungs-/Complianceprüfung). Die größte Herausforderung für einen solchen Abgleich ist die Heterogenität in Konzeption (Granularität der Aktivitäten) und Realisation (Repräsentation des Verhaltens, Formulierung der Beschriftungen und verwendete Terminologie) der abzugleichenden Modelle. Das angesprochene Granularitätenproblem kann durch sogenannte M:N-Abbildungen behandelt werden, wobei versucht wird, Teile eines Prozessmodells auf Teile eines anderen Prozessmodells abzubilden.

Für imperative Prozessmodelle existiert bereits eine Vielzahl von Methoden, die mehr oder weniger zur Lösung des Abgleichproblems beitragen. Da es für deklarative Prozessmodelle bislang jedoch kaum Lösungen gibt, bzw. die vorhandenen nur sehr fallspezifisch angewendet werden können, aufgrund der Komplexität dieser Modelle hier ein automatisierbarer Abgleich jedoch besonders wichtig ist, ist es die wesentliche Aufgabe der vorliegenden Arbeit, weitere Möglichkeiten des Ähnlichkeitsabgleichs zu diskutieren. Dazu wird nachfolgend auf grundlegende Konzepte deklarativer Prozessmodelle, den aktuellen Stand der Wissenschaft sowie einige Ansätze für den Ähnlichkeitsabgleich eingegangen. Abschließend wird zu offenen Problemen und den nächsten Schritten Stellung genommen.

2 Deklarative Prozessmodelle

Fahland et al. (2009) sehen den wesentlichen Unterschied zwischen imperativen und deklarativen Prozessmodellen darin, wie ermittelt werden kann, ob ein gegebenes Verhalten konform bezüglich des Prozessmodells ist oder nicht. Im Falle der imperativen Modelle muss ein Verhalten mittels eines Pfades durch das Modell rekonstruiert werden können. Im Gegensatz dazu wird bei deklarativen Prozessmodellen geprüft, dass zu keinem Zeitpunkt eine der enthaltenen Regeln verletzt worden ist. Ausführungspfade lassen sich auch für deklarative Modelle generieren, wobei bedacht werden muss, dass es eine sehr große Anzahl an möglichen Ausführungspfaden geben kann. Zeising et al. (2014) diskutieren eine gemeinsame Plattform zu Ausführung von imperativen und deklarativen Prozessmodellen. Ausgegangen wird dabei von der *Declarative Process*

Intermediate Language (DPIL), einer deklarativen Prozessmodellierungssprache auf Basis der Prädikatenlogik erster Stufe. Da CMMN auf DPIL abgebildet werden kann und andere deklarative Prozessmodellierungssprachen verschiedene Nachteile aufweisen (Zeising et al. 2014), wird für die nachfolgenden Erläuterungen eine vereinfachte Version von DPIL als Beispielsprache verwendet. Der konkrete Aufbau der Sprache kann in Zeising et al. (2014) nachvollzogen werden. Anstelle der Spezifikation des Aufbaus sollen folgende Beispielprozessmodelle der Veranschaulichung dienen:

Prozessmodell G_1 : Aktivitäten = $\{A, B, C\}$ Regeln = { $r_1: 1 \leq \text{count}(\text{start_of}(A)) \leq 2$ $r_2: \text{count}(\text{start_of}(B)) = 1$ $r_3: \text{count}(\text{start_of}(C)) \leq 1$ $r_4: \text{start_of}(B) \xrightarrow{t>} \text{complete_of}(A)$ $r_5: \text{complete_of}(C) \xrightarrow{t<} \text{complete_of}(B)$	Prozessmodell G_2 : Aktivitäten = $\{A', B', C'\}$ Regeln = { $r_1: \text{count}(\text{start_of}(A')) = 2$ $r_2: \text{count}(\text{start_of}(B')) = 1$ $r_3: \text{count}(\text{start_of}(C')) = 1$ $r_4: \text{complete_of}(A') \xrightarrow{t<} \text{complete_of}(B')$ $r_5: \text{start_of}(B') \xrightarrow{t>} \text{complete_of}(C') \wedge$ $\text{complete_of}(C') \xrightarrow{t<} \text{complete_of}(B')$
---	---

Bild 1: Beispiel für deklarative Prozessmodelle. Als Beispielabbildung dient später $M: x \mapsto x'$.

Die deklarativen Prozessmodelle umfassen sowohl die Arbeitsschritte des jeweiligen Prozesses (Aktivitäten) als auch Bedingungen (Regeln), welche zur Ausführungszeit die Menge der möglichen Kombinationen von Arbeitsschritten einschränkt. Die an DPIL angelehnte Pseudomodellierungssprache ist ereignisbasiert. Hierbei werden zwei Ereignistypen unterschieden: (i) Startereignisse (*start_of*) und (ii) Endereignisse (*complete_of*). Das Auftreten eines solchen Events im Ausführungspfad trifft demnach eine Aussage über den aktuellen Zustand des betroffenen Arbeitsschrittes. Die Regeln 1 bis 3 beider Prozessmodelle limitieren damit, wie oft ein bestimmter Prozessschritt durchgeführt werden darf. Die Implikationsregeln ($G_1: r_{4/5}$, $G_2: r_4$) legen fest, dass das Auftreten des Ereignisses auf der linken Seite der Implikation fordert, dass später auch das Ereignis auf der rechten Seite auftritt ($t<$) respektive bereits aufgetreten ist ($t>$). Die UND-Verknüpfung aus G_2 (r_5) legt folgerichtig fest, dass bei Auftreten eines der beiden verknüpften Ereignisse das jeweils andere ebenfalls im Ausführungspfad auftreten muss – und zwar so, dass der Prozessschritt C' immer vor Schritt B' abgeschlossen wurde.

3 Stand der Wissenschaft

Für imperative Prozessmodelle werden in der Literatur zahlreiche Möglichkeiten für einen Modellabgleich vorgeschlagen. Zusammenfassungen und Auswertungen verschiedener Ansätze finden sich beispielsweise in Dijkman et al. (2011). Die Methoden reichen dabei von strukturellen, graphbasierten Ansätzen über labelbasierte Ansätze bis hin zu verhaltensbasierten Ansätzen (Baumann et al 2015, Dijkman et al. 2009). Informationen, die verwendete Dokumente, zugewiesene Agenten und Rollen oder zu benutzende Werkzeuge betreffen, können ebenfalls für einen Ähnlichkeitsabgleich herangezogen werden (Baumann M et al. 2014, Baumann MH et al. 2014, Weidlich et al. 2010). Vorgegangen wird bei den Methoden meist nach dem gleichen Schema. Zunächst wird eine Abbildung M festgelegt, die Aktivitäten des ersten Prozessmodells G_1 Aktivitäten des zweiten Prozessmodells G_2 zuordnet. Diese Zuordnung kann 1:1 erfolgen, was bedeutet, dass eine Aktivität aus G_1 einer aus G_2 zugeordnet wird (partiell injektiv (Dijkman et al. 2011)). Besitzen G_1 und G_2 unterschiedlich viele Aktivitäten, werden mit Sicherheit nicht alle

Aktivitäten abgebildet. Die Abbildung kann auch M:N erfolgen, dann werden Mengen von Aktivitäten aus G_1 Mengen von Aktivitäten aus G_2 zugeordnet (bijektiv (Baumann MH et al. 2014)). In beiden Fällen können Aktivitäten auch gelöscht werden. Ist eine Abbildung M zwischen G_1 und G_2 festgelegt ($M: G_1 \rightarrow G_2$), wird die Ähnlichkeit der einzelnen Abbildungselemente anhand von einem oder mehreren der bereits genannten Kriterien (Struktur, Label, Verhalten etc.) bestimmt. Ein Abbildungselement ist ein Paar aus Aktivitäten (A, B) mit $A \in G_1$ und $B = M(A) \in G_2$ oder ein Paar aus einer Menge von Aktivitäten (X, X') , $X \subset G_1$ und $X' = M(X) \subset G_2$. Die einzelnen Ähnlichkeitswerte werden dann für die gesamte Abbildung, d. h. für alle Abbildungselemente, gemittelt und so ein Ähnlichkeitswert, abhängig von M , errechnet. Auch die gelöschten Aktivitäten werden hierbei berücksichtigt. Um die Ähnlichkeit unabhängig von einer zufällig gewählten Abbildung zu definieren, wird der Ähnlichkeitswert über alle möglichen Abbildungen maximiert. Man erhält so die beste Abbildung M^* zwischen den zwei Prozessmodellen G_1 und G_2 und einen Ähnlichkeitswert für die beiden Modelle unter M^* . Dieser Wert ist so bestimmt, dass er aus dem Intervall $[0,1]$ ist, wobei 1 volle Ähnlichkeit und 0 keine Ähnlichkeit bedeutet.

Alle Perspektiven, die weder Struktur noch Verhalten betreffen, z. B. Label, können für deklarative wie für imperative Modelle gleich behandelt werden. Folglich sind im Weiteren vor allem Verhalten und Struktur von Interessen. Für die Ähnlichkeitsbestimmung zweier deklarativer Prozessmodelle können zwei Ansätze der Literatur entnommen werden. Zum einen ist es möglich, deklarative Prozessmodelle in imperative zu überführen (Prescher et al. 2014), für die dann die oben erläuterten Methoden angewendet werden können. Diese Lösung funktioniert gut, solange alle Regeln des deklarativen Modells in der imperativen Repräsentation vorhanden sind. Soweit die Autoren wissen, gibt es jedoch keine imperative Modellierungssprache, die z. B. die perspektivenübergreifenden Regeln, die DPIL erlaubt, abbildet. Wie in Giannakopoulou und Havelund (2001) gezeigt, ist es auch möglich, bestimmte deklarative Prozessmodelle auf Basis von LTL auf endliche Zustandsautomaten abzubilden. Zustandsautomaten sind eine bestimmte Form von Graphen, für die ein Editierabstand (Graph-Edit-Distance) berechnet werden kann (Wombacher 2006). Da in einem Zustandsautomat jedoch Zustände, und nicht Aktivitäten, die Knoten darstellen, ist eine Abwandlung der Abbildung von Aktivitäten auf eine Abbildung von Zuständen erforderlich. Auch ist zu beachten, dass deklarative Modelle, die Variablen benutzen, nicht zwingend in einen endlichen Automaten überführt werden können (Zeising et al. 2014). Es müssen also Ansätze auch für solche Prozessmodelle gefunden werden. Für die Beispielmodelle aus Kapitel 2 kann man sich überlegen, dass der Automat für G_1 19 Strukturelemente und der für G_2 14 besitzt. Es sind 5 (Einfüge- bzw. Lösch-)Operationen nötig, um die Modelle ineinander zu überführen. Dies führt zu einem Ähnlichkeitswert von: 1 –

$$\frac{\min\{\# \text{Operationen}\}}{\max\{\# \text{Elemente Automat}_{G_1}, \# \text{Elemente Automat}_{G_2}\}} = 14/19$$

4 Ansätze zum Ähnlichkeitsabgleich

Da in deklarativ modellierten Modellen kein expliziter Kontrollfluss gegeben ist, lassen sich struktur- und verhaltensbasierte Verfahren nicht immer übertragen. Im deklarativen Modell sind Regeln gegeben, die statt des Kontrollflusses für eine Ähnlichkeitsberechnung verwendet werden können. Wie für imperative Prozessmodelle wird auch für deklarative Prozessmodelle zunächst eine Abbildung M , die zwei Modelle G_1 und G_2 aufeinander abbildet, festgelegt. Für die im Folgenden aufgeführten Ideen zur Ähnlichkeitsberechnung wird zunächst eine 1:1-Abbildung angenommen.

Das Finden der besten Abbildung M^* erfolgt dann über einen Optimierungsansatz und wird hier nicht weiter behandelt.

4.1 Ähnlichkeitsabgleich via Ausführungspfade und Konformitätsprüfung

Die erste vorgestellte Möglichkeit geht ähnlich vor, wie es bei Process-Mining-Verfahren üblich ist, um die Fitness von Logdaten bezüglich der erzeugten Modelle zu bestimmen (Rozinat und van der Aalst 2006). Es werden simulierte, im besten Fall alle möglichen, Ausführungspfade für ein Modell verwendet und dann mit dem zweiten Modell abgeglichen, d. h. auf Ausführbarkeit im zweiten Modell getestet. Da in deklarativen Prozessmodellen jedoch prinzipiell auch unendlich bzw. beliebig lange Ausführungspfade möglich sind, kann nur über Simulation eine endliche Menge an möglichen, endlichen Ausführungspfaden erzeugt werden. Dieses Thema der *Automated Sequence Generation* oder *Automated Test Case Generation* findet sich beispielsweise in Hallé et al. (2012). Das Ergebnis der Ähnlichkeitsprüfung hängt dann sehr von der Menge an simulierten Pfaden ab. Es stellt sich die Frage, ob das Ergebnis repräsentativ ist. Andersherum kann auch verglichen werden, ob Ausführungspfade, die in einem Modell nicht erlaubt sind, im anderen auch nicht erlaubt sind. Insgesamt ergäben sich also vier Gruppen von Pfaden (erlaubt-erlaubt, erlaubt-nicht erlaubt, nicht erlaubt-erlaubt, nicht erlaubt-nicht erlaubt). Als Ähnlichkeitsmaß böte sich der Quotient aus gleichartigen und allen Pfaden an. Sind alle diese Zahlen endlich, kann das Maß einfach berechnet werden, sind die Anzahlen jedoch unendlich, existiert lediglich ein Grenzwert, der für manche Konstellationen jedoch immer aus $\{0,1\}$ ist.

Für die Beispielmodelle G_1 und G_2 aus Kapitel 2 ergeben sich folgende erlaubte Pfade: $G_1: AB, AAB, ABA, CAB, ACB, AACB, ACAB, CAAB, ACBA, CABA$ und $G_2: A'C'B', A'A'C'B', A'C'A'B', C'A'A'B', C'A'B'$. Hierfür bietet sich als Ähnlichkeit folgender Wert an, der nicht erlaubte Pfade nicht berücksichtigt: $\frac{1}{2} \left(\frac{\# \text{ erlaubte Pfade } G_1 \text{ gültig in } G_2}{\# \text{ erlaubte Pfade } G_1} + \frac{\# \text{ erlaubte Pfade } G_2 \text{ gültig in } G_1}{\# \text{ erlaubte Pfade } G_2} \right) = 3/4$

4.2 Ähnlichkeitsabgleich via Verhaltensmuster und Wahrheitstabellen

Die zweite Möglichkeit ist eine Art Approximation der Methoden aus 4.1. Es werden zunächst für die abgebildeten Aktivitäten vorher festgelegte Verhaltensmuster bestimmt, die eine bestimmte Anzahl an Aktivitäten umfassen. Bei bis zu 3-stelligen Mustern für die Aktivitäten A und B ergeben sich beispielsweise folgende Muster, wobei nur positive Zusammenhänge berücksichtigt werden: $-, A, B, AA, AB, \dots, BBB$. Es wird dann mit Hilfe der Regelmengen beider Prozessmodelle jedem Muster ein Wahrheitswert zugewiesen, d. h., es wird geprüft, ob das Muster im jeweiligen Modell so ausführbar ist. Ein einzelnes A würde in diesem Fall bedeuten, dass A bzw. $M(A)$ irgendwann einmal ausgeführt wird und davor und danach weder A noch B (bzw. $M(A), M(B)$) ausgeführt werden. Es wird hier zunächst nicht unterschieden, ob die Ausführung direkt oder irgendwann nacheinander erfolgen muss. Sind die Wahrheitswerte für die Muster in beiden Modellen bestimmt, werden sie abgeglichen: Ist ein Muster in beiden Modellen erlaubt oder in beiden Modellen nicht erlaubt, wird ihm der Wert 1, ist ein Muster in einem Modell erlaubt, im andern nicht, der Wert 0 zugewiesen. Diese Werte werden addiert und durch die Anzahl der Muster geteilt, was einen Mittelwert im Intervall $[0,1]$ liefert. Erlauben beide Modelle die gleichen Muster, ist der Wert 1. Je weniger Muster übereinstimmen, desto mehr tendiert der Mittelwert zu 0. Solch ein Mittelwert wird für je zwei abgebildete Aktivitäten berechnet. Für eine Abbildung mit drei abgebildeten Aktivitäten (wie in unserem Beispiel) werden also drei Mittelwerte gebildet. Diese Mittelwerte werden dann noch einmal gemittelt, um einen Gesamtwert zu erhalten. Die wichtigste Frage, die sich für diese Möglichkeit stellt, ist die, welche Verhaltensmuster für den Abgleich verwendet werden sollen um

ein möglichst objektives und verallgemeinerbares Ergebnis zu erhalten. Ein Vorteil dieser Methode ist, dass Transitivitäten in den Regelmengen berücksichtigt werden.

G_1, G_2	–	x	y	xx	xy	yx	yy
$x = A, y = B$	✗,✗	✗,✗	✗,✗	✗,✗	✓,✓	✗,✗	✗,✗
$x = A, y = B$	✗,✗	✓,✗	✗,✗	✓,✗	✓,✓	✓,✓	✗,✗
$x = B, y = C$	✗,✗	✓,✗	✗,✗	✗,✗	✗,✗	✓,✓	✗,✗

Tabelle 1: Verhaltensmustervergleich für die Beispielmodelle aus Kapitel 2

Die Wahrheitstabelle für alle 2-wertigen und max. 2-stelligen, positiven Muster für das Beispiel aus Kapitel 2 ist in Tabelle 1 dargestellt. Ob ein Muster im jeweiligen Modell erlaubt (✓) oder nicht erlaubt (✗) ist, erkennt man leicht durch Vergleich der Muster mit den möglichen Pfaden (siehe Abschnitt 4.1). Es ergibt sich ein Ähnlichkeitswert von: $\frac{\# \text{gleicher Wahrheitswert}}{\# \text{Muster}} = 6/7$

4.3 Ähnlichkeitsprüfung via Prädikatenlogik und Regelähnlichkeit

Eine weitere Möglichkeit, die sich im Zusammenhang mit einem Ähnlichkeitsabgleich von zwei Prozessmodellen ergibt ist die, die Regeln der Prozessmodelle auf Ebene der zugrunde liegenden Logik direkt miteinander zu vergleichen. Die Regeln deklarativer Prozessmodelle sind üblicherweise in Prädikatenlogik erster Stufe (vgl. DPIL) oder einer äquivalenten Logik formuliert. Es stellt sich die Frage, ob aus der Logik bekannte Erfüllbarkeits- bzw. Entscheidbarkeitskriterien für eine Ähnlichkeitsbestimmung benutzt werden können, wie zum Beispiel in Bisson (1992) oder Segan und Schoenauer (1994) für statistische Lernverfahren verwendet. Nach einer ersten Einschätzung ergibt sich für die Regelmengen deklarativer Prozessmodelle, zumindest dann, wenn sie Aussagen über komplexere Zusammenhänge erlauben, dass sie unentscheidbar werden. Der Vergleich kann (gegeben entscheidbare Regelmengen) so erfolgen: Kann eine Regel aus G_1 aus den Regeln von G_2 abgeleitet werden, wird sie mit 1 bewertet, sonst mit 0, und andersherum. Dann werden diese Werte aufsummiert und durch die Anzahl der Regeln geteilt. Dabei ergeben sich folgende Schwierigkeiten: Aus der Regel $50 \leq \text{count}(\text{start_of}(A))$ lässt sich sowohl $49 \leq \text{count}(\text{start_of}(A))$ als auch $1 \leq \text{count}(\text{start_of}(A))$ ableiten. Also ergibt sich je ein Ähnlichkeitswert von 0,5, wobei die ersten beiden Regeln ähnlicher sein sollten als die erste und die dritte. Ein weiteres Problem stellt die Anzahl der Regeln dar, da Regelmengen, die zu einem gleichen Verhalten führen, nicht identisch, also vor allem nicht gleich mächtig, sein müssen. Das heißt sowohl Dividend als auch Divisor in diesem Ähnlichkeitsmaß sind nicht eindeutig bestimmt.

5 Schluss

Die vorliegende Arbeit zeigt einige Ansätze und Schwierigkeiten beim Ähnlichkeitsabgleich deklarativer Prozessmodelle auf. Da die bisher in der Wissenschaft verfügbaren Techniken nicht für alle flexiblen Prozesse verallgemeinerbar sind, werden neue Ansätze vorgestellt, die jedoch noch nicht ausgereift sind. Nach dem derzeitigen Stand der Forschung scheint die Methode „Verhaltensmustervergleich“ am praktikabelsten und vielversprechendsten.

6 Danksagung

Die Arbeit von Michael Heinrich Baumann wird durch ein Begabtenstipendium der „Hanns-Seidel-Stiftung e.V. (HSS)“ aus Mitteln des „Bundesministeriums für Bildung und Forschung (BMBF)“ gefördert.

7 Literatur

- Baumann M, Baumann MH, Jablonski S (2015) On Behavioral Process Model Similarity Matching: A Centroid-based Approach. Accepted for ICCGI
- Baumann M, Baumann MH, Schönig S, Jablonski S (2014) Resource-Aware Process Model Similarity Matching. In press for RMSOC
- Baumann MH, Baumann M, Schönig S, Jablonski S (2014) Towards Multi-perspective Process Model Similarity Matching. EOMAS, LNBIP 191:21-37
- Bisson G (1992) Learning in FOL with a Similarity Measure. In: Proceedings of the National Conference on Artificial Intelligence: 82-82
- Dijkman R, Dumas M, van Dongen B, Käärik R (2011) Similarity of business process models: Metrics and evaluation. *Information Systems* 36(2):498-516
- Dijkman R, Dumas M, García Bañuelos L, Käärik R (2009) Aligning Business Process Models. In: IEEE Int. Enterprise Distributed Object Computing Conference (EDOC): 45-53
- Dumas M, Rosa ML, Mendling J, Reijers HA (2013) Fundamentals of Business Process Management. Springer Verlag
- Fahland D, Lübke D, Mendling J, Reijers H, Weber B, Weidlich M. (2009) Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. *BPMDs*: 353-366
- Giannakopoulou D, Havelund K (2001) Automata-based verification of temporal properties on running programs. In: Int. Conf. on Automated Software Engineering (ASE): 412-416
- Hallé S, Villemaire R, Cherkaoui O, Deca R (2012) A logical approach to data-aware automated sequence generation. In: *Transactions on Computational Science XV*: 192-216
- Jablonski S (1994) MOBILE: A modular workflow model and architecture. In: Working Conference on Dynamic Modelling and Information Systems
- Pichler P, Weber B, Zugal S, Pinggera J, Mendling J, Reijers H (2012) Imperative versus declarative process modeling languages: An empirical investigation. In: *LNBIP 99*: 383-394
- Prescher J, Di Ciccio C, Mendling J (2014) From declarative processes to imperative models. In: 4th Int. Symp. on Data-Driven Process Discovery and Analysis (SIMPDA) 100
- Rozinat A, van der Aalst WMP (2006) Conformance testing: Measuring the fit and appropriateness of event logs and process models. In: *Business Process Management Workshops*:163-176
- Sebag M, Schoenauer M (1994) A Rule-Based Similarity Measure. In: *Topics in Case-Based Reasoning (LNCS)* 837:119-131
- Tka M, Ghannouchi SA (2012) Comparison of Business Process Models as Part of BPR Projects. In: Conference on ENTERprise Information Systems (CENTERIS): 427-436

van der Aalst W, Pesic M, Schonenberg H (2009) Declarative Workflows: Balancing between flexibility and support. In: Computer Science – Research and Development 23(2): 99-113

Weidlich M., Dijkman R., Mendling, J. (2010) The ICoP framework: Identification of correspondences between process models. In: Adv. Information Systems Engineering: 483-498

Wombacher A (2006) Evaluation of technical measures for workflow similarity based on a pilot study. In: CoopIS, DOA, GADA, and ODBASE: 255-272

Zeising M, Schöning S, Jablonski S (2014) Towards a common platform for the support of routine and agile business processes. CollaborateCom: 94-103