

Geschäftsprozessmodellierung: Die „Killer-Applikation“ für Petrinetze

W. M. P. van der Aalst

Einleitung

Parallel zum kontinuierlichen Ausbau der Petrinetz-Theorie gab es einen bemerkenswerten Trend weg von datenzentrierten Informationssystemen, hin zu prozessorientierten Informationssystemen [4, 6].

Um Geschäftsprozesse zu unterstützen muss ein Informationssystem diese Prozesse und ihre organisierte Umgebung gut kennen. Während sich klassische Informationssysteme aus zentralen Datenbank-Systemen heraus entwickelt haben, sind moderne Systeme verteilt und prozessorientiert. Die zunehmende Bedeutung des Geschäftsprozess-Managements (Business Process Management, BPM) illustriert diesen Trend [2, 5, 9]. Der Begriff BPM umfasst Methoden, Techniken und Werkzeuge, um reale Geschäftsprozesse zu entwerfen, implementieren, verwalten und zu analysieren. BPM kann man als Erweiterung klassischer Workflow-Management-Systeme und -Konzepte auffassen.

Die meisten etablierten BPM-Notationen und -Systeme verwenden Token-basierte, von Petrinetzen entlehnte Beschreibungen. Petrinetze waren der erste Formalismus zur Modellierung nebenläufiger, verteilter, paralleler Abläufe. Solche Abläufe sind grundlegend für BPM, denn in Geschäftsprozessen geschieht oft Vieles zugleich und unabhängig voneinander. Ein BPM-System muss tausende Aktionen zugleich verwalten, und selbst innerhalb einer Aktion kann Mehreres zugleich passieren. Deshalb müssen Notationen, Techniken und Systeme zum Umgang mit BPM verteilte Abläufe als grundlegendes Konzept enthalten.

Abbildung 1 zeigt ein Petrinetz-Modell eines Geschäftsprozesses, der neun Aktivitäten enthält. Transitionen des Modells sind entsprechend be-

schriftet. So beschreibt die Inschrift „a“ von t_1 beispielsweise die Registrierung einer Bitte um Kompensation. Zwei Transitionen, t_2 und t_3 verlangen zusätzliche Informationen. t_4 und t_{13} , ohne Inschrift, repräsentieren interne, „stille“ Aktivitäten, deren Ausführung von außen nicht sichtbar ist. Das Netz in Abb. 1 ist seiner Struktur nach ein *Workflow-Netz* (WF-Netz, [3]). Es hat einen eindeutigen *Start-Platz* (ohne eingehende Kanten) und einen eindeutigen *End-Platz* (ohne ausgehende Kanten). Alle anderen Knoten des Netzes liegen auf Wegen zwischen diesen beiden Plätzen. Ein *Workflow-Netz* modelliert den Lebenszyklus einer Prozess-Instanz (übliche Bezeichnung: *case*). Die Marke auf dem Start-Platz repräsentiert eine solche Instanz. In einem Ablauf eines WF-Netzes gelangen zwischenzeitlich gegebenenfalls mehrfach Marken auf denselben Platz, beispielsweise auf c_1 und c_4 , nachdem die eingegangene Bitte um Kompensation registriert (a) und begutachtet (e) wurde. Marken verschiedener Instanzen vermischen sich in einem WF-Netz niemals: Ein WF-Netz beschreibt den Lebenszyklus verschiedener Instanzen immer voneinander isoliert. Diesem Prinzip folgen auch andere Modellierungstechniken für Geschäftsprozesse, beispielsweise BPMN, UML-Aktivitätsdiagramme, BPEL, EPCs etc. Das WF-Netz aus Abb. 1 ist *sound* [3], weil es niemals in einen Deadlock gerät, immer den Endzustand erreichen kann und jede Transition aktivierbar ist.

DOI 10.1007/s00287-013-0756-2
© Springer-Verlag Berlin Heidelberg 2014

W. M. P. van der Aalst
Eindhoven University of Technology,
Department of Mathematics and Computer Science,
Eindhoven, The Netherlands
E-Mail: w.m.p.v.d.aalst@tue.nl

Zusammenfassung

Seit ihrem Entwurf im Jahr 1962 sind Petrinetze in ganz unterschiedlichen Bereichen eingesetzt worden. Obwohl sie graphisch dargestellt werden und intuitiv einfach verständlich sind, haben Petrinetze eine formal eindeutige Semantik mit einer Vielzahl mathematischer Analysetechniken. Sie reichen vom Model Checking und der Strukturellen Analyse über das Process Mining bis zur Performanz-Analyse.

Im Lauf der Zeit haben Petrinetze solide Grundlagen für die Forschung zum Geschäftsprozess-Management (BPM) beigetragen. Sie umfassen Methoden, Techniken und Werkzeuge um Geschäftsprozesse zu entwerfen, implementieren, verwalten und zu analysieren. Die etablierten Modellierungsmethoden und Workflow-Managementsysteme verwenden Token-basierte, von Petrinetzen entlehnte Beschreibungen. Nutzer moderner BPM-Analysetechniken wissen oft gar nicht, dass ihre Geschäftsprozesse intern als Petrinetze repräsentiert werden.

Dieser Beitrag zeigt die grundlegende Rolle von Petrinetzen im BPM.

Abbildung 1 modelliert den Kontrollfluss des Geschäftsprozesses. Mit komplexeren Varianten von Petrinetzen können Ressourcen, organisatorische Zusammenhänge, Daten, die Interaktion und Kommunikation mit der Umgebung, sowie zeitliche Aspekte modelliert werden [4, 6]. In Abb. 1 wird das

elementare Konzept „schwarzer, zeitloser“ Marken verwendet. Komplexere Varianten erweitern Petrinetze um gefärbte Marken, Zeit, und hierarchische Aspekte [4, 7].

Im Folgenden diskutieren wir drei gute Gründe, Petrinetze im BPM zu verwenden.

– Formal eindeutige Semantik trotz grafischer Darstellung:

Einerseits sind Petrinetze ein grafisches, intuitives Ausdrucksmittel; schon einfache elementare Netze reichen aus um die elementaren Workflow-Komponenten zu modellieren [6]. Andererseits haben Petrinetze (einschließlich der meisten Verallgemeinerungen) eine mathematisch eindeutig formal beschriebene Bedeutung. Im Gegensatz dazu verwenden viele aktuelle BPM-Systeme und -Notationen Ad-Hoc-Konstrukte. Das führt schnell zu unübersichtlichen und verwirrenden Darstellungen, insbesondere wenn Nichtdeterminismus und Verteiltheit aufeinander treffen. Deshalb sollte man lieber eine etablierte Entwurfssprache mit einer formalen Semantik verwenden; und Petrinetze sind eine Basis dafür. Das heißt nun keineswegs, Geschäftsprozesse unbedingt als Petrinetze zu visualisieren! Abstraktere, grafisch reichhaltigere Darstellungen sind oft knapper und intuitiver. Nur sollten sie (zur Analyse und zur Beschreibung oder Semantik) auf eine solide Basis abgebildet werden können.

– Zustandsbasiert statt ereignisbasiert:

Im Unterschied zu einigen anderen Modellierungstechniken für Prozesse können die Zustände einer Instanz in einem Petrinetz explizit model-

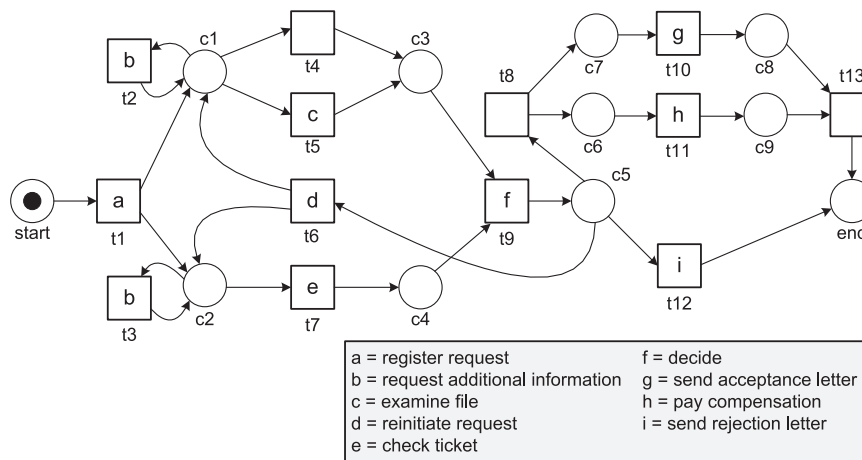


Abb. 1 Ein sound workflow net (WF-net), das den Lebenszyklus einer Bitte um Kompensation beschreibt. Jede Transition ist entweder mit einer Aktivität beschriftet oder sie ist „still“

liert werden. Andere Techniken, von informalen Darstellungen wie beispielsweise Datenfluss-Diagramme bis hin zu Prozess-Algebren sind *ereignisbasiert*, das heißt die einzelnen Schritte werden explizit modelliert, aber Zustände ergeben sich dabei nur implizit. Gängige BPM-Systeme sind vorwiegend ereignisbasiert, mit explizit dargestellten Aktivitäten, aber ohne sichtbare Zustände. Intern haben Abläufe und Systeme aber durchaus Zustände. Die Zustände sind überaus wichtig, wenn Systeme ausgeführt oder analysiert werden. Meilensteine, zeitlich verzögerte Entscheidungen und andere Workflow-Pattern können nur mit expliziten Zuständen angemessen modelliert werden [6]. Die explizite Darstellung von Zuständen gehört also zum Kern der Modellierung von Geschäftsprozessen.

– *Eine Fülle von Analysetechniken:*

Für Petrinetze gibt es eine Vielzahl effizienter Analysetechniken. Mit ihrer prägnanten operationellen Semantik sind allgemeine Analysetechniken wie Model Checking und Simulation unmittelbar anwendbar. Darüber hinaus gibt es Petrinetz-spezifische Techniken wie Fallen, Co-Fallen, Platz-Invarianten, Transitions-Invarianten, Überdeckungsgraphen, Regionen und verteilte Abläufe [8]. Process-Mining-Algorithmen verwenden solche Techniken bei der Suche nach einem geeigneten Prozessmodell und beim Vergleich von modelliertem mit beobachtetem Verhalten [1].

Im Rest dieses Beitrags geht es um drei Themen: Zunächst, im Abschnitt „Das ‚Markenspiel‘ im Geschäftsprozess-Management“, diskutieren wir die Rolle von Petrinetzen im BPM-Lebenszyklus.

Der Abschnitt „Der Einfluss von Petrinetzen auf Sprachen und Systeme“ beschreibt den Einfluss von Petrinetzen auf das Gebiet der BPM.

Schließlich philosophieren wir im Abschnitt „Die natürliche Struktur von Geschäftsprozessen“ über den Bezug zwischen Modell und Wirklichkeit vor dem Hintergrund der Maxime von Petri, dass Prozessmodelle mit den Gesetzen der Physik übereinstimmen sollen.

Das „Markenspiel“ im Geschäftsprozess-Management

Der BPM-Lebenszyklus [2, 4] in Abb. 2 verdeutlicht die Rolle von Petrinetzen im BPM. In der Phase des (Um-)Planens wird ein Modell entwor-

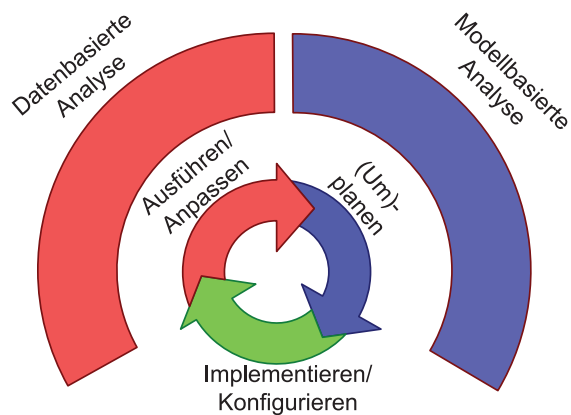


Abb. 2 Der BPM-Lebenszyklus mit seinen drei Phasen: (1) (Um-)planen, (2) Implementieren/Konfigurieren und (3) Ausführen und Anpassen

fen. Dieses Modell wird in der Implementierungs-/Konfigurationsphase in ein lauffähiges System transformiert. Wenn das Modell in einer lauffähigen Form vorliegt und ein funktionierendes WFM- oder BPM-System bereitsteht, ist diese Phase sehr kurz. Falls jedoch nur ein informelles Modell vorliegt, das noch in einer klassischen Programmiersprache kodiert werden muss, kann diese Phase lange dauern. Danach beginnt die Ausführungs- und Korrekturphase. In dieser Phase werden die beteiligten Prozesse ausgeführt und gegebenenfalls angepasst. Dabei werden sie aber nicht komplett neu entworfen und es wird auch keine neue Software geschrieben. Vielmehr werden nur vordefinierte Tests durchgeführt und kleine Änderungen vorgenommen. Abbildung 2 zeigt zwei Formen systematischer Analyse: Modell-basierte und datenbasierte Analyse. Während das System läuft, werden Ereignis-Daten gesammelt und analysiert, beispielsweise werden Flaschenhälse, ungenutzte Ressourcen oder Abweichungen vom erwarteten Verhalten entdeckt. Diese Erkenntnisse werden in der (Um-)Planungsphase verwendet. In dieser Phase werden Prozessmodelle analysiert. Beispielsweise werden „What-if“-Fragen mit der Analyse simulierten Verhaltens beantwortet, und die Korrektheit eines neu geplanten Modells mit Model Checking nachgewiesen.

In allen drei Phasen (Abb. 2) werden Petrinetze verwendet, allerdings oft im Hintergrund. Im Rest dieses Abschnitts zeigen wir die Rolle von Petrinetzen beim *Modellieren*, *Analysieren*, und *Realisieren* (Instanzieren von Geschäftsprozessen).

Modellierung

Der alte Spruch „Ein Bild sagt mehr als 1000 Worte“ erklärt prägnant den überragenden Nutzen von Petrinetzen beim Entwurf von Geschäftsprozessen. Das einfache Workflow-Netz aus Abb. 1 kann als Basis der Diskussion verschiedener Varianten der (Um-)Planung und Neu-Strukturierung verwendet werden. Ganz wichtig ist dabei, mit dem „Markenspiel“ die Möglichkeit, Abläufe zu analysieren und verschiedene Szenarien durchzuspielen. Die „Regel“ des Markenspiels versteht jedermann sofort.

Wie schon erwähnt, modelliert Abb. 1 nur den *Kontrollfluss*, also die Ordnung der Aktivitäten eines Geschäftsprozesses. Oft wollen wir aber auch die Verwendung von *Ressourcen* modellieren, beispielsweise den Beitrag einzelner Mitarbeiter, ganzer Abteilungen oder beteiligter Maschinen. Man unterscheidet Ressourcen für funktionale Aufgaben von Ressourcen für organisatorische Aufgaben.

Die *Daten-Perspektive* diskutiert die Herstellung und die Verwendung von Daten. Jeder einzelnen Instanz eines Workflows sind Kontrolldaten zugeordnet, die das Routing steuern. Produktionsdaten (beispielsweise Dokumente, Formblätter, Tabellen) sind Informationsobjekte, deren Gestalt und Verwendung nicht nur vom Routing abhängt. Aktivitäten brauchen oft ganz spezielle Eingabedaten und produzieren spezielle Ausgabedaten. Beispielsweise bekommt ein Antragsteller ein in Teilen ausgefülltes Formblatt, das er vervollständigt. Zudem hängen die meisten Entscheidungen in einem Prozess von Daten ab.

Die *Interaktions-Perspektive* betrifft die Zusammenhänge zwischen Prozessen und Instanzen. Beispielsweise hängen Einzelbestellungen, Sammelbestellungen und Lieferungen untereinander zusammen. Sie können und sollen aber nicht einen einzigen Workflow bilden, egal ob als WF-Netz, BPMN-Graph, EPK oder UML-Diagramm. Zudem müssen Prozesse über Organisationsgrenzen hinweg interagieren können.

Die *zeitliche Perspektive* betrachtet die Dauer von Objekt- und Datenflüssen, Fristen und ihre Überschreitung, Wartezeiten, Betriebszeiten, Antwortzeiten etc. Beispielsweise soll ein Anspruch abgewiesen werden, wenn er nicht innerhalb zweier Wochen angemeldet wurde. Man kann auch eine *durchschnittliche* Bearbeitungsdauer modellieren wollen.

Welche der genannten Perspektiven eines Prozesses in welchen Einzelheiten modelliert werden, hängt vom *Zweck* des Modells ab. Wenn ein Modell vorwiegend „What-if-Szenarien“ simulieren soll, müssen vor allem Bedienzeiten und Ressourcen genau modelliert werden. Daten sind dabei weniger wichtig. Wenn das Modell primär zur Ausführung verwendet wird, kann man auf Bedienzeiten verzichten (sie ergeben sich ja ohnehin), entscheidend sind aber Ein- und Ausgabedaten der Aktivitäten.

Das WF-Netz in Abb. 1 hat nur „schwarze“, ununterscheidbare Marken ohne Dateninhalt. Um alle genannten Perspektiven angemessen zu modellieren, werden Petrinetze mit „farbigen“ (Daten tragenden) Marken, Zeit und Hierarchie ausgestattet [4, 7]. Um alle modellierten Prozesse zu analysieren, können solche Modelle wiederum zu elementaren Netzen abstrahiert werden.

Analyse

Informelle Modelle sind für die Analyse von Prozessen nicht verwendbar. Für Petrinetze gibt es jedoch vielfältige Analysetechniken. Abbildung 3 klassifiziert diese Techniken in zwei Dimensionen: Zum einen kann man ein Ad-hoc-Modell bilden und analysieren, oder aktuelle Ereignis-Daten verwenden, im Sinne datenbasierter Analyse nach Abb. 2. Zum anderen kann man sich auf funktionale Eigenschaften beschränken, oder auch nicht-funktionale Eigenschaften mit einbeziehen.

Traditionell zielt die Forschung über Petrinetze auf die *Modellbasierte Analyse*, und in diesem Rahmen auf *funktionale Eigenschaften*. Mit generischen Techniken wie dem Model Checking kann man spezielle Eigenschaften nachweisen, beispielsweise die Deadlockfreiheit. Petrinetzspezifische Notationen wie Fallen, Co-Fallen, Platzinvarianten, Transitionsinvarianten, und Überdeckungsgraphen werden oft zum Nachweis funktionaler Eigenschaften verwendet, beispielsweise Lebendigkeits- oder Sicherheits-Eigenschaften [8]. Für Workflows interessant ist die Eigenschaft der *Soundness* von WF-Netzen [3]: Ein WF-Netz ist genau dann *sound* wenn es folgende drei Eigenschaften hat: (1) *Es kann immer terminieren*: Von jeder erreichbaren Markierung aus ist eine Markierung mit einer Marke auf dem *end*-Platz erreichbar. (2) *Es terminiert korrekt*: Wenn der *end*-Platz eine Marke trägt, gibt es im Netz keine weiteren Marken. (3) *Es hat nur aktivierbare Transitionen*: Zu jeder Transition gibt

	Modellbasierte Analyse	Datenbasierte Analyse
funktionale Eigenschaften (z.B. Deadlocks)	<i>Verifikation, Model checking, Soundness checking, etc</i>	<i>Process mining (z.B. process discovery und Konformanz-Prüfung)</i>
nicht-funktionale Eigenschaften (z.B. Durchsatz)	<i>Simulation, Markovketten, Optimierung</i>	<i>Process mining (z.B. model extension und model prediction)</i>

Abb. 3 Elementare Charakterisierung prozessbasierter Analysetechniken

es eine erreichbare Markierung, die diese Transition aktiviert. Das WF-Netz aus Abb. 1 ist *sound*. Also kann keine Instanz des Netzes in einen Deadlock geraten oder nicht ausführbare Komponenten enthalten. Nun gilt ein Theorem, mit dem man die Frage nach der Soundness eines WF-Netzes auf die klassischen Eigenschaften der Lebendigkeit und Beschränktheit zurückführen kann, für deren Nachweis seit Langem effiziente Verfahren bekannt sind. Ein WF-Netz ist genau dann *sound*, wenn das zugehörige „kurzgeschlossene“ Petrinetz lebendig und beschränkt ist. Neben der Soundness gibt es zahlreiche weitere interessante Fragestellungen. Manche verlangen zu ihrer Beantwortung komplexe Techniken, beispielsweise Fragen der Art „Kann dieselbe Ressource die beiden Aktivitäten c und f ausführen, wenn die Kompensation einen Transatlantik-Flug erfordert?“

Auch nicht-funktionale Eigenschaften, beispielsweise Ausführungsgeschwindigkeit, Reaktionszeiten, Kosten, Risiken, Nutzbarkeit, Fehlertoleranz können modellbasiert analysiert werden. Derartige Eigenschaften sind für BPM äußerst wichtig. Spezielle Klassen von Petrinetzen kann man mit Markov-Ketten analysieren. Beispielsweise können stochastische Petrinetze mit negativ exponentieller Verzögerung in Markov-Ketten übersetzt werden. Damit können mittlere Ausführungsgeschwindigkeiten, Wahrscheinlichkeiten bei Alternativen etc. analysiert werden. Für kompliziertere Prozessmodelle und Fragestellungen bleibt oft nur noch die Simulation. Deshalb unterstützen die meisten BPM-Werkzeuge einige Simulationsverfahren [4].

In den vergangenen Jahren wuchs das Interesse an der „rechten Hälfte“ von Abb. 3, beflügelt von der Verfügbarkeit entsprechender Daten und dem Interesse von Organisationen an faktenbasierten Analyse-Ergebnissen („evidence based BPM“). Der Begriff *Process Mining* bezeichnet dabei Techniken, die Wissen aus event logs ableiten [1].

Solche Process Mining-Techniken gehören zu den *A-posteriori*-Analysetechniken, die Informationen aus audit trails, transaction logs, Datenbankabfragen etc. ableiten. Zum Process Mining gehören ganz unterschiedliche Verfahren: automatisch Prozessmodelle aus event logs ableiten (vgl. Abb. 4c, d), Konformanzprüfung (beispielsweise Inkonsistenzen zwischen Modellen und logs erkennen), soziale Netzwerke/Prozesse beobachten, automatisch Simulationsmodelle erzeugen, Modelle reparieren, Instanzen vorhersagen, und Empfehlungen aufgrund bisherigen Verhaltens ableiten.

Die wachsende Bedeutung des Process Mining sieht man an folgender Beobachtung: Eine typische Festplatte von 2010 hat eine Kapazität von ca. einem Terabyte (1 TB) = 10^{12} bytes. Damals betrug das gesamte „Digitale Universum“ ca 1,2 Zentabyte ($1,2 \times 10^{21}$ bytes)¹.

Nach der Wachstumsrate von Hard Disks und nach Moores Gesetz passt diese Datenmenge in ca. 50 Jahren auf eine einzige Hard Disk. Diese Überschlagsrechnung zeigt die unglaubliche Zunahme von Ereignisdaten in den nächsten Jahrzehnten. Geschäftsprozesse werden zunehmend anhand der Daten analysiert, die sie produzieren. Transaktionsdaten und Sensordaten, beispielsweise von RFID-Tags, werden zukünftig traditionelle, auf handgefertigten Modellen basierende Analysetechniken durch völlig neuartige Techniken des Process Mining ersetzen.

Realisieren (Instanzieren) von Geschäftsprozessen

Petrinetze sind ausführbar; sie erzeugen Verhalten. Der Kern jedes WFM/BPM-Systems ist eine sogenannte „Workflow Engine“. Eine solche Maschine interagiert mit ihrer Umgebung, indem sie

¹ Gemäß dem jährlichen IDC Bericht „The Digital Universe Decade: Are You Ready?“, Mai 2010.

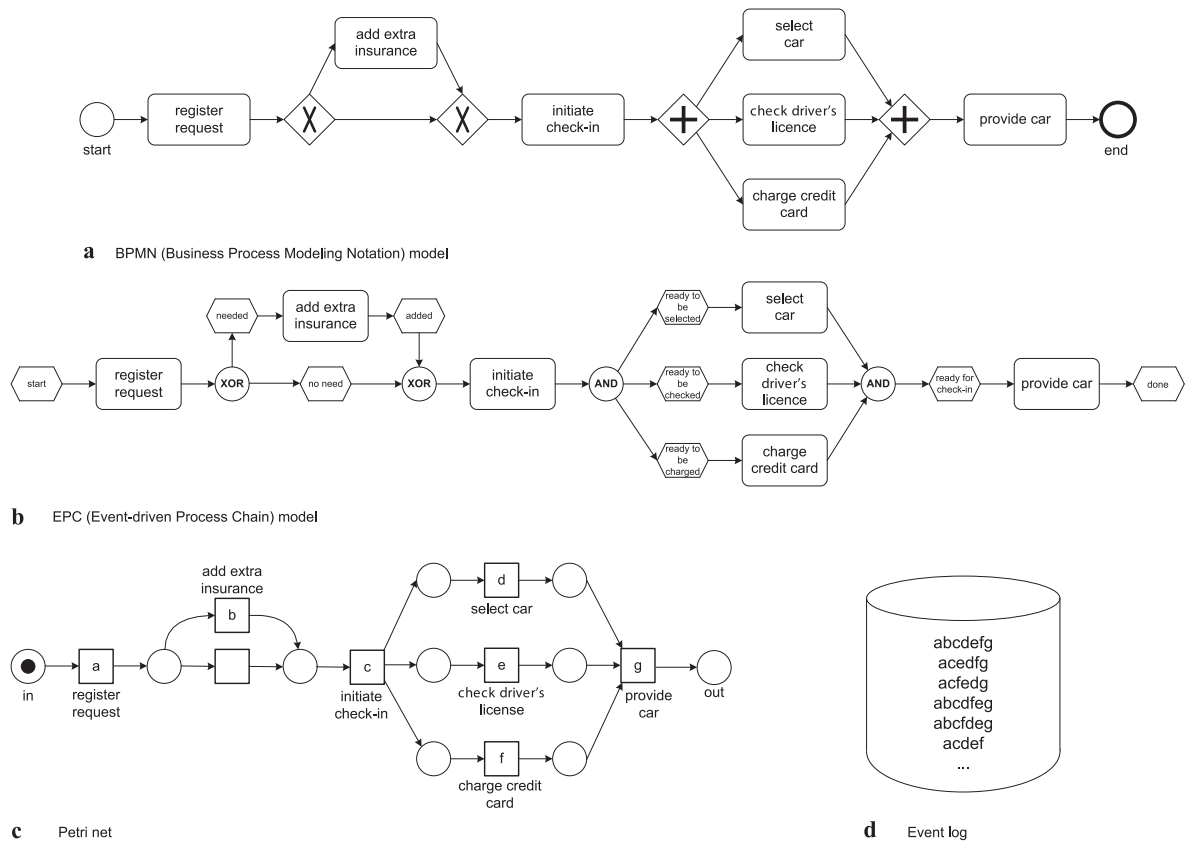


Abb. 4 Ein Geschäftsprozess in drei verschiedenen Darstellungen: (a) BPMN, (b) EPK, (c) Petrinetz. Der Event-Log (d) skizziert die Traces des Prozesses (mit den Transitionsnamen des Petrinetzes)

– anschaulich formuliert – das Markenspiel von Petrinetzen ausführt: Die meisten Workflow Engines funktionieren gemäß einer „token based“ Semantik. Immer wenn eine Engine eine Aktivität einer Instanz durchgeführt hat, aktualisiert sie den Zustand (die Markierung eines entsprechenden Petrinetzes), und neue Aktivitäten sind aktiviert, die der Umgebung angeboten werden.

Der Einfluss von Petrinetzen auf Sprachen und Systeme

Kontinuierlich werden neue Notationen zur Modellierung von Prozessen vorgeschlagen. Einige davon sind grundlegend und werden seit Jahrzehnten verwendet (beispielsweise Petrinetze). Andere haben Firmen etablieren wollen, oder sie erweitern lediglich eine bewährte Notation. Viele solche Notationen verschwinden schnell wieder.

Schwerpunkte der Notationen reichen von Sprachen die den Modellen eine formale Basis geben (beispielsweise endliche Automaten, Petrinetze, und

Prozessalgebren) bis hin zu herstellerdefinierten Notationen (verschiedene proprietäre, herstellerebundene Workflow-Sprachen). Industriestandards wie BPEL (Business Process Execution Language) und BPMN (Business Process Modeling Notation) werden im Allgemeinen nur in Teilen berücksichtigt: Kommerzielle Werkzeuge unterstützen oft nur eine Teilmenge des Standards, und Anwender nutzen wiederum nur einen Bruchteil des Implementierten [6]. Ganz offensichtlich gibt es wenig Übereinstimmung über die „beste“ Modellierungssprache. Prozess-Beschreibungssprachen türmen sich wie der sprichwörtliche Turmbau zu Babel: Eine Vielzahl ähnlicher, aber in spitzfindigen Aspekten verschiedene Sprachen verhindern effektive, einheitliche, werkzeuggestützte Entwurfs- und Analysetechniken.

In dieser unübersichtlichen Welt spielen Petrinetze eine wichtige Rolle. Fast alle BP-Modellierungssprachen und BPM/WFM-Systeme verwenden eine Token-basierte Semantik, inspiriert

vom Markenspiel der Petrinetze. Oft sind Petrinetze an der Benutzeroberfläche gar nicht erkennbar, oft hantiert der Nutzer aber auch direkt mit ihnen. Ein Beispiel aus den 1990er-Jahren ist das damals führende COSA-System: Der COSA-Modellersteller, die COSA-Engine und der COSA-Simulator basieren auf Petrinetzen. Der größte SAP-Konkurrent beim ERP (Enterprise Resource Planning) in den 1990er-Jahren, *Baan*, war bekannt für seinen dynamischen „Enterprise Modeler“ (DEM). Dieses Werkzeug modelliert Prozesse als Petrinetze und ist sehr hilfreich, um das Baan ERP-System in die organisatorische Struktur einer Anwender-Firma zu integrieren. COSA und DEM haben viele spätere BPM/WFM/ERP-Systeme beeinflusst. Beispiele dafür sind die aktuelle Business Operations Platform (BOP) von Cordys, und Oracles BPM suite.

Ein weiteres bemerkenswertes Beispiel ist der *Protos Modeler* von Palas Athena. Der Protos Modeler notiert Modelle als Petrinetze. Im Jahr 2010 haben mehr als 250 der 441 niederländischen Gemeinden aktiv mit Protos ihre Verwaltung modelliert. Protos unterstützt auch das Simulieren und verwendet dafür intern das ExSpecT Simulationswerkzeug. Entwickelt wurde ExSpecT ursprünglich als Prototyping- und Simulationswerkzeug an der Technischen Universität Eindhoven. Heute ist Protos eine Komponente der *BPMone*-Plattform. Das ist eine BPM-Suite zum Suchen, Entwerfen, Ausführen und Verbessern von Geschäftsprozessen.

Neben diesen kommerziellen Systemen gibt es eine Reihe open source/akademischer BPM Systeme und Werkzeuge, die an zentraler Stelle Petrinetze verwenden. Dazu gehören *YAWL* (ein WFM-System), *WoPeD* und *Yasper* (Geschäftsprozess-Modellierung und Simulation) sowie *ProM* (process mining). Auch frühere Systeme zur Prozessautomatisierung verwenden häufig Petrinetze, beispielsweise *Officetalk* von Xerox PARC in den späten 1970ern, *SCOOP* (System zur Automatisierung von Prozessen im Büro) von Michael Zisman (späte 1970er-Jahre), *Income Workflow* von Promatis in den 1990er-Jahren, etc.

Trotz der vielen Beispiele interessanter BPM-Systeme und Werkzeuge, die ihren Benutzern explizit Petrinetze zeigen, ist der Einfluss von Petrinetzen oft hinter grafisch blumigen Darstellungen verborgen, die in der Software-Industrie so beliebt sind. Abbildung 4 zeigt denselben Prozess in drei unterschiedlichen Notationen. Die *Business Process Modeling Notation*

(BPMN) verwendet „Aktivitäten“, „Ereignisse“ und „Zugänge“ (Gateways genannt) zur Modellierung des Kontrollflusses. Abbildung 4a verwendet zwei Typen von Gateways: Exklusive Gateways modellieren XOR-Verzweigungen und -Zusammenführungen; parallele Gateways modellieren AND-Verzweigungen und deren Zusammenführungen. BPM unterstützt auch andere Arten von Gateways, die inklusive OR-Verzweigungen, verzögerte Auswahl etc. modellieren [5, 6, 9]. *Event-driven Process Chains* (Ereignisgesteuerte Prozessketten, EPKs) verwenden Funktionen, Ereignisse und Konnektoren um Kontrollfluss zu modellieren (Abb. 4b). Konnektoren in EPKs ähneln Gateways in BPMN. Dabei gibt es OR, XOR und AND-Konnektoren. Ereignisse in EPKs ähneln Plätzen in Petrinetzen. Und so wie Plätze und Transitionen sich in Petrinetzen abwechseln, müssen auf jedem Pfad einer EPK die Ereignisse und Funktionen immer abwechselnd vorkommen. Ein Ereignis kann jedoch höchstens einen Folgeknoten haben; damit können verzögerte Alternativen nicht dargestellt werden [6]. *UML-Aktivitätsdiagramme* (UML-Ads) – in Abb. 4 nicht dargestellt – modellieren elementare Konzepte des Kontrollflusses ähnlich wie BPMN.

BPMN, EPKs, UML-Ads und viele andere Notationen für Geschäftsprozess-Modelle verwenden alle eine tokenbasierte Semantik. Daher gibt es eine Reihe von Techniken und Werkzeugen, die BPMN, EPKs und UML-Ads in Petrinetze und zurück übersetzen. Damit werden Petrinetze indirekt verwendet, beispielsweise um Modelle zu analysieren, auszuführen, oder ihre Semantik zu klären.

Die natürliche Struktur von Geschäftsprozessen

Petri hat immer zwei Maximen betont: „Nebenläufigkeit (und damit die Lokalität von Aktionen) soll von Anfang an Bestandteil der Modellierung sein, nicht erst ein späterer Zusatz“, und „Eine Modellierungstechnik soll den elementaren Gesetzen der Physik nicht widersprechen“. Petrinetze waren das erste Systemmodell das Nebenläufigkeit adäquat gefasst hat. Nebenläufigkeit spielt selbstverständlich eine fundamentale Rolle in Geschäftsprozessen: Es gibt fast immer mehrere Aktivitäten, beispielsweise Personen und Maschinen, die nebenläufig ablaufen, und irgendwann gibt es mehrere Prozessinstanzen. Petri hat sich immer für den Zusammenhang zwi-

schen Prozessmodellen und ihrer physikalischen Entsprechung interessiert.

Bei BPM sollte man den Bezug zwischen Prozessmodellen und den tatsächlichen Charakteristika von Geschäftsprozessen sorgfältig gestalten. Geschäftsprozesse tendieren immer mehr zu hochgradiger Nebenläufigkeit und einer nichtmonolithischen, systematisch zusammengesetzten Struktur. Dafür sind sequenzielle Modelle nicht angemessen [4]. Zudem darf ein Modell nicht immer nur einzelne, isolierte Prozessinstanzen modellieren (wie man das in BPMN, EPKs etc. macht). Beispielsweise können Einzelbestellungen, Sammelbestellungen und Auslieferung in einer „many-to-many“-Beziehung stehen [1]. Die empirische Natur des Process Mining hilft Managern, Beratern und Prozess-Analysten, die „Natur realer Geschäftsprozesse“ besser zu verstehen und damit die Schwächen und Schranken konventioneller Prozess-Beschreibungssprachen

zu erkennen [1]. Hier besteht eine große Herausforderung darin, elegante und fokussierte formale Modelle wie beispielsweise Petrinetze mit der tatsächlichen beobachteten Realität zu verknüpfen.

Literatur

1. van der Aalst WMP (2011) *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, Berlin
2. van der Aalst WMP (2013) *Business Process Management: a Comprehensive Survey*. ISRN Software Engineering 2013:1–37, doi: [10.1155/2013/507984](https://doi.org/10.1155/2013/507984)
3. van der Aalst WMP, van Hee KM, ter Hofstede AHM, Sidorova N, Verbeek HMW, Voorhoeve M, Wynn MT (2011) Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects Comput* 23(3):333–363
4. van der Aalst WMP, Stahl C (2011) *Modeling Business Processes: a Petri Net Oriented Approach*. MIT press, Cambridge, MA
5. Dumas M, La Rosa M, Mendling J, Reijers H (2013) *Fundamentals of Business Process Management*. Springer, Berlin
6. ter Hofstede AHM, van der Aalst WMP, Adams M, Russell N (2010) *Modern Business Process Automation: YAWL and its Support Environment*. Springer, Berlin
7. Jensen K, Kristensen LM (2009) *Coloured Petri Nets*. Springer, Berlin
8. Reisig W (2013) *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer, Berlin
9. Weske M (2007) *Business Process Management: Concepts, Languages, Architectures*. Springer, Berlin