

Mithilfe von Modellierungssprachen wird festgelegt, mit welchen Konzepten ein Ausschnitt der menschlich wahrgenommenen Realität beschrieben werden kann und wie diese zueinander in Beziehung gesetzt werden können. Modellierungssprachen stellen also das Vokabular und die Grammatik zur Verfügung, die benötigt werden, um Sachverhalte der menschlich wahrgenommenen Realität in Modellen abbilden zu können. Sie werden in der Folge strukturiert betrachtet.

Die Verwendung einer bestimmten Modellierungssprache schafft einen definierten Ausgangspunkt für die Modellbildung, auf den sich alle beteiligten Akteure – seien sie aktiv an der Erstellung oder passiv als Konsumenten beteiligt – beziehen können. Akteure können hier einerseits Personen sein, denen durch die Modellierungssprache ein gemeinsames, definiertes Vokabular zur Verfügung gestellt wird, um sich ein gemeinsames Verständnis über den Modellgegenstand zu bilden. Andererseits können Akteure auch Computersysteme sein, denen durch eine in der Modellierungssprache exakt spezifizierte Semantik von Modellelementen die Gelegenheit geboten wird, Modelle automationsgestützt weiter zu verarbeiten, um sie etwa als Grundlage für Workflow-Unterstützung einzusetzen.

Von der Wahl der Modellierungssprache hängt also ab, was in einem Modell überhaupt abgebildet werden kann, und was nicht sichtbar werden kann, weil die Modellierungssprache keine Konzepte dafür anbietet. Die Wahl der Modellierungssprache beeinflusst auch die Verwendbarkeit des Modells für unterschiedliche Zielgruppen: Während manche Modellierungssprachen eher zur Unterstützung der Kommunikation menschlicher Akteure konzipiert wurden, und dem entsprechend manchmal auch eine „vage“ Abbildung von Sachverhalten ermöglichen, gibt es andere Modellierungssprachen, die für eine exakte Spezifikation von Sachverhalten konzipiert sind und deren Anwendungsfeld eher in der Aufbereitung von Modellen für die Verwendung in IT-Systemen liegt. Die Wahl einer Modellierungssprache ist also abhängig

von der jeweiligen Zielsetzung der Modellbildung und damit ein wesentlicher Schritt hin zu einer erfolgreichen Unterstützung jener Aktivitäten, in denen die Modellierung eingebettet ist.

Im vorliegenden Abschnitt wollen wir die Grundlage für die sach- und personengerechte Auswahl einer passenden Modellierungssprache schaffen und stellen dazu unterschiedliche Sprachen mit deren jeweiligen Zielsetzungen und Sprachelementen vor. Der Fokus liegt hier auf der Abbildung von Geschäftsprozessen, weshalb alle im der Folge behandelten Sprachen im weiteren Sinne die Abbildung von Verhalten von Akteuren in Organisationen ermöglichen. Im Verständnis darüber, was nun ein Akteur sein kann und wie dieses Verhalten beschrieben wird, unterscheiden sich die Sprachen fundamental. Dies ist deren Entstehungsgeschichte und den jeweiligen Zielsetzungen geschuldet. Die Anordnung der einzelnen Ansätze folgt einer historischen Perspektive, um die Zusammenhänge zwischen den Sprachen und deren Entstehung deutlich hervortreten zu lassen. Für jede der Sprachen betrachten wir abschließend, wie sich diese zur Abbildung von Prozessen entsprechend unserer Definition eignen und wo ihre Abbildungsschwerpunkte bzw. Lücken liegen¹.

3.1 Überblick

Flowcharts sind die älteste bis heute gebräuchliche Repräsentationsform von Modellen des Kontrollflusses in einem Computerprogramm. Aufgrund der Generalität ihrer Sprachelemente werden sie auch für Zwecke der Modellierung von Abläufen in Organisationen eingesetzt und sind hier deshalb als Einstieg in die grafische Prozessmodellierung zu betrachten. Sie weisen das Konzept der Verzweigung im Kontrollfluss zur Abbildung von alternativen Abläufen auf.

eEPKs („erweiterte Ereignisgesteuerte Prozessketten“) stellen in Europa lange Zeit einen de-facto Industriestandard für die Geschäftsprozessmodellierung dar. Sie umfassen neben der Modellierung von Abläufen auch Elemente, mit denen Verantwortlichkeiten, Daten oder Leistungen modelliert werden können und sorgen damit dafür, dass Geschäftsprozesse gemeinsam mit ihrem organisationalen Kontext in Modellen abgebildet werden können. Darüber hinaus erlauben sie die explizite Modellierung von parallelen Aktivitäten und gehen damit bezüglich der Abbildung des Arbeitsablaufs über die Ausdrucksstärke von Flowcharts hinaus.

UML Aktivitätsdiagramme sind historisch gesehen eine Weiterentwicklung von Flowcharts und stellen heute als Teil der Unified Modeling Language (UML) den de-facto Standard zur Abbildung von Kontrollflüssen in Software dar. Auch sie ermöglichen die

¹Die fachliche Grundlage zu diesem Kapitel findet sich in folgenden Quellen; www.iso.org, [1, 2].

Abbildung von parallelen Abläufen. Anhand von Aktivitätsdiagrammen führen wir die Strukturierung von Modellen durch Partitionierung und Schachtelung ein und zeigen damit, wie Modelle auch anhand von Verantwortlichkeiten und nicht nur ausschließlich anhand des Arbeitsablaufs strukturiert werden können.

BPMN (Business Process Modelling and Notation) ist der heute meist eingesetzte Standard zur Abbildung von Geschäftsprozessen. Ausgehend von mehreren unterschiedlichen Modellierungssprachen – unter anderem von Aktivitätsdiagrammen – wurde eine Sprache definiert, die explizit zur Abbildung von Geschäftsprozessen geeignet sein sollte und mit deren Hilfe Modelle für unterschiedliche Zielsetzungen – von der Kommunikationsunterstützung bis hin zur Ausführung in Workflowmanagement-Systemen – erstellt werden können sollten. Aus Sicht der Sprachkonzeption ist die BPMN vor allem hinsichtlich ihrer Möglichkeiten zur kompakten Abbildung von komplexen Abläufen (etwa Ausnahmebehandlungen) interessant.

S-BPM (Subject-oriented Business Process Modeling) ist ein Modellierungsansatz, der die in einem Geschäftsprozess involvierten Akteure und deren Interaktionsverhalten ins Zentrum der Modellbildung stellt. Die sich daraus ergebende Modellierungssprache zeichnet sich durch einen geringen Umfang von Sprachelementen bei gleichzeitig umfassender Ausdrucksstärke zur Abbildung von Geschäftsprozessen aus. Sie wird hier einerseits als Vertreterin eines nicht vorrangig ablauforientierten Herangehens an die Geschäftsprozessmodellierung vorgestellt und bildet außerdem in der Sprachkonzeption eine Alternative zur BPMN und ihrem sehr umfangreichen Satz an Modellierungselementen.

3.2 Flowcharts

Flowcharts (oder deutsch: *Flussdiagramme* bzw. ursprünglich *Programmablaufpläne*) ermöglichen es, einfache, sequenzielle Prozesse abzubilden. Ein sequenzieller Prozess zeichnet sich dadurch aus, dass zu keinem Zeitpunkt mehr als eine Aktivität gleichzeitig durchgeführt wird – parallele Abläufe können also nicht abgebildet werden. Erstmals beschrieben wurden Flowcharts im Rahmen der industriellen Produktionsplanung in den 1920er-Jahren. Ende der 1940er-Jahre wurden sie für die Beschreibung von Prozessen in der aufkommenden Informationstechnologie adaptiert. Seit Mitte der 1960er-Jahre stellen sie eine standardisierte Repräsentationsform für Programmabläufe dar. Bis heute sind sie ein Mittel der Wahl, um Abläufe in Computerprogrammen oder auch Prozesse in Organisationen darzustellen, solange deren Komplexität die Ausdrucksstärke eines Flowchart nicht übersteigt.

Die Einschränkung der Ausdrucksstärke von Flowcharts ist deren historischen Entwicklung geschuldet. Sowohl in der industriellen Produktionsplanung als auch in frühen Computersystemen war es nicht notwendig, parallele Abläufe abbilden zu können.

Durch die eingeschränkten zur Verfügung stehenden Ressourcen (in Computern: nur eine CPU bzw. nur ein Prozessorkern) war es schlicht nicht notwendig, Sprachelemente zur Abbildung etwa von parallelen Abläufen zur Verfügung zu stellen.

3.2.1 Notationselemente

Flowcharts kommen heute in vielen unterschiedlichen Varianten vor, die sich vor allem in der verwendeten Notation (d. h. der graphischen Darstellung) unterscheiden. Die Semantik der Sprachelemente ist aber allen gemein: Grundsätzlich wird eine beliebige Anzahl von *Operationen* (also Aktivitäten, dargestellt als Rechtecke) definiert. Diese werden mittels gerichteten Verbindungen (dargestellt als Pfeile) in die Reihenfolge gebracht, in der sie ausgeführt werden sollen. *Terminierungselemente* (dargestellt als abgerundete Rechtecke) kennzeichnen den Beginn und das Ende eines Prozesses (siehe Abb. 3.1).

Ein wesentliches Mittel der Beschreibung von Prozessen – sowohl in einem Computerprogramm als auch in Organisationen – stellt die Abbildung von alternativen Operationen dar. Die Auswahl einer Alternative ist üblicherweise abhängig von einer Bedingung, die geprüft werden kann – in einem Computerprogramm könnte dies das Überschreiten eines Grenzwertes in einer Variablen sein, in einer Organisation etwa das Vorhandensein eines bestimmten Dokuments oder die Entscheidung einer für die Ausführung des Ablaufs verantwortlichen Person. Alternativen werden in Flowcharts durch Verzweigungen abgebildet (dargestellt als Rauten), die durch einen eingehenden Pfeil von der vorhergehenden Operation und mit zwei ausgehenden Pfeilen mit den alternativ ausgeführten nachfolgenden Operationen verbunden sind.

Die Einschränkung auf genau zwei (und nicht mehrere) nachfolgende Operationen ist ebenfalls der Herkunft von Flowcharts aus der Abbildung von Computerprogrammen geschuldet, da diese eine Bedingung üblicherweise ausschließlich binär (d. h. als wahr oder falsch) auswerten können. Soll eine Bedingung auf mehr als zwei Ausprägungen geprüft werden, so müssen in Flowcharts mehrere Verzweigungen kaskadiert eingesetzt werden. Die ausgehenden Verbindungen werden mit der jeweiligen Ausprägung beschriftet, bei der nach der Prüfung der Bedingung das Programm fortgesetzt wird. Eine wiederholte Ausführung von Operationen (etwa solange noch Dokumente vorhanden sind, die bearbeitet werden müssen) wird abgebildet, in dem mit einer ausgehenden Verbindung zu einer früheren Operation zurückgesprungen wird. Die andere ausgehende



Abb. 3.1 Notationselemente von Flowcharts

Verbindung leitet dann zu jenem Ablaufteil über, der ausgeführt wird, sobald die wiederholte Ausführung abgeschlossen ist.

Neben diesen Grundelementen bieten Flowcharts in manchen Varianten noch spezielle Sprachelemente, die vor allem der Abbildung von spezifischen Operationen im Anwendungsgebiet dienen (bei Computersystemen etwa Ein- oder Ausgabeoperationen). Für das konzeptionelle Verständnis von Flowcharts und vor allem deren Anwendung in der Geschäftsprozessmodellierung sind diese jedoch nicht relevant.

Im Folgenden betrachten wir die Verwendung der Notationselemente anhand von Beispielen aus der Geschäftsprozessmodellierung. Die verwendete Notation orientiert sich an den durch das ANSI bzw. die DIN in den 1960er-Jahren festgelegten Symbolik.

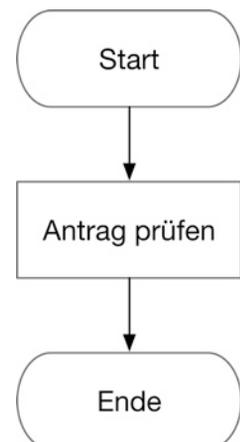
3.2.2 Beispiele

Dieses Beispiel (siehe Abb. 3.2) zeigt einen Prozess mit einer einzelnen Operation, in der ein Antrag (über den wir hier nicht mehr wissen) bearbeitet wird. Der Prozess endet, nachdem die Bearbeitung des Antrags abgeschlossen wurde.

Im obenstehenden Beispiel (siehe Abb. 3.3) ist der Prozess um eine Entscheidung erweitert. Der Antrag wird geprüft, das Ergebnis dieser Prüfung ermöglicht das Treffen einer Entscheidung über dessen Bestätigung oder Ablehnung. Das Bestätigen bzw. Ablehnen selbst sind wiederum Operationen. Die ausgehenden Verbindungen werden zusammengeführt, nachdem die alternativen Zweige abgeschlossen sind.

Ist es nun notwendig, eine Entscheidung zu treffen, die mehr als zwei mögliche Ausgänge hat, muss dieser Sachverhalt in Flowcharts mittels kaskadierter Entscheidungselemente dargestellt werden. Dies ist im obenstehenden Beispiel (siehe Abb. 3.4) zu sehen. Offensichtlich handelt es sich bei unseren Anträgen um Investitionsbegehren. Die erste Entscheidung prüft nun, ob die Investition weniger als EUR 1000,- kostet. Ist dies der Fall, wird der Antrag direkt bestätigt. Ist dies nicht der Fall, kommt es zu einer

Abb. 3.2 Einfaches Flowchart



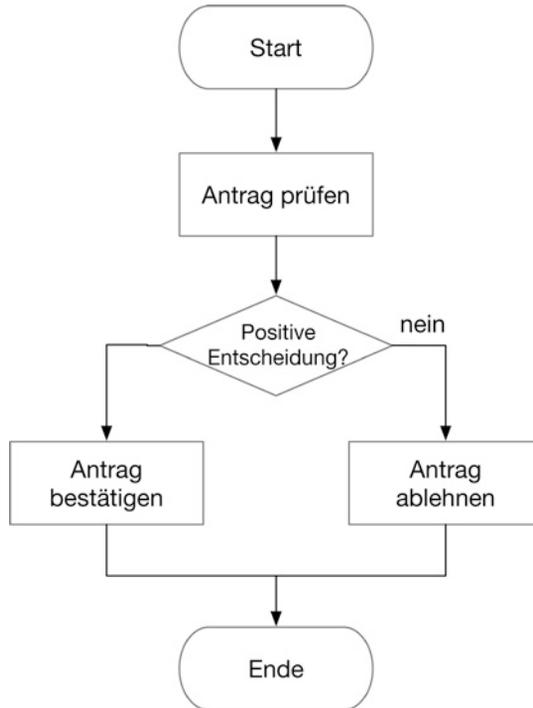


Abb. 3.3 Flowchart mit Entscheidung

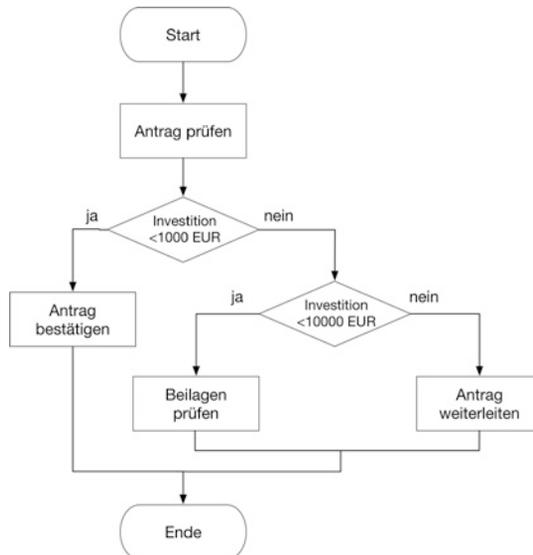


Abb. 3.4 Flowchart mit kaskadierten Entscheidungen

zweiten Entscheidung. Diese prüft, ob die Antragssumme kleiner als EUR 10.000,- ist. Wenn dies der Fall ist, liegt die Antragssumme zwischen EUR 1000,- und EUR 9999,-, was zu einer Prüfung der Beilagen zum Antrag führt. Wenn die Antragssumme nicht kleiner als EUR 10.000,- ist, also EUR 10.000,- oder mehr beträgt, wird der Antrag weitergeleitet. Wir erfahren hier nichts über das Ziel der Weiterleitung, da Flowcharts dafür keine Notationselemente anbieten.

Wie bereits erwähnt, können Verzweigungen auch genutzt werden, um Teile eines Prozesses wiederholt abzuarbeiten. Dazu wird die Entscheidung am Ende des zu wiederholenden Teils eingefügt und ein ausgehender Zweig vor die erste Operation des zu wiederholenden Teils zurückgeführt. Der andere ausgehende Zweig führt den Prozess nach der wiederholten Abarbeitung weiter. Im obenstehenden Beispiel (siehe Abb. 3.5) werden also solange Anträge bearbeitet, solange noch weitere Anträge vorhanden sind.

Wichtig ist hier zu verstehen, dass zumindest ein Antrag bearbeitet werden muss, bevor es zu einer Entscheidungsprüfung kommt. Soll der Prozess enden können, wenn gar keine Anträge vorliegen, müsste eine zusätzliche Entscheidung am Beginn des Prozesses eingefügt werden („Anträge vorhanden?“), von der eine ausgehende Verbindung („nein“) direkt zum Ende des Prozesses führt. Die andere ausgehende Kante („ja“) würde zum bereits abgebildeten Prozessablauf weiterführen.

3.2.3 Einordnung

Flowcharts bieten eine einfache Möglichkeit, Geschäftsprozesse hinsichtlich der logischen Abfolge der enthaltenen Aktivitäten abzubilden. Andere Aspekte eines Geschäftsprozesses, wie Daten oder Verantwortlichkeiten, sind im Sprachumfang nicht vorgesehen und können deshalb nicht abgebildet werden.

Die Abbildung von parallelen Abläufen ist ebenfalls nicht möglich. Dies ist ein maßgeblicher Grund dafür, dass Flowcharts zum Teil von aktuelleren Sprachen wie UML



Abb. 3.5 Flowchart mit Schleife

Aktivitätsdiagrammen oder BPMN abgelöst werden, die dafür Konstrukte anbieten. Da Flowcharts keine Verantwortlichkeiten abbilden lassen, ist auch die Abbildung von Kommunikationsvorgängen nicht möglich – diese Einschränkung wurde ebenfalls durch modernere Sprachen adressiert, die wir in den folgenden Abschnitten behandeln.

3.3 Ereignisgesteuerte Prozessketten

Ereignisgesteuerte Prozessketten (EPKs) waren lange Zeit in Europa der de-facto Standard zur Abbildung in Geschäftsprozessen in der industriellen Praxis. Sie wurden als Teil des in Abschn. 2.5.1.4 bereits vorgestellten ARIS-Konzeptes spezifiziert und dienen dort als Mittel zur Abbildung von Modellen der Steuerungssicht einer Organisation – also jener Sicht, die sich mit den Abläufen in einer Organisation und der dort stattfindenden Verknüpfung zwischen deren unterschiedlichen Ressourcen beschäftigt. Ressourcen können hier im Speziellen aus aktiver Sicht die handelnden Organisationsmitglieder bzw. -einheiten, sowie aus passiver Sicht die benötigten und manipulierten Daten sein.

Die EPK verknüpft die Funktionen, die eine Organisation ausführen kann, auf Basis von auftretenden Ereignissen miteinander. Das grundlegende Abbildungsprinzip ist dabei, dass eine Funktion immer durch ein Ereignis ausgelöst wird – dass also vor einer Funktion immer ein Ereignis modelliert werden muss, um feststellen zu können, ob mit der Ausführung einer Funktion begonnen werden kann. In der umfassenderen Variante „erweiterte EPK“ (eEPK) sind den Funktionen die für deren Ausführung relevanten Elemente der anderen ARIS-Sichten zugeordnet. Insbesondere können aus der Organisationssicht die ausführenden Akteure, Rollen oder Organisationseinheiten zugeordnet werden, aus der Datensicht die relevanten Dokumente oder Datenobjekte. Wenn eine Funktion eine abrechenbare Leistung erbringt, so kann diese durch Elemente der Leistungssicht abgebildet werden.

Neben der Abbildung von Entscheidungen im abgebildeten Ablauf ermöglicht die EPK auch die Abbildung von parallelen Abläufen in einen Geschäftsprozess. Dafür stellt sie zusätzliche Notationselemente zur Verfügung, die sich an den Operatoren der Bool'schen Logik orientieren – mittels UND-Konnektoren können parallel ablaufende Prozesszweige abgebildet werden, der XOR-Konnektor dient dazu, Entscheidungen abzubilden, bei denen genau eine Alternative gewählt werden muss (dies entspricht dem Entscheidungselement bei Flowcharts). Mit dem ODER-Konnektor können Prozesse abgebildet werden, bei denen aus einer oder mehreren Alternativen gewählt werden kann.

3.3.1 Notationselemente der EPK

Das Grundelement zur Beschreibung von Geschäftsprozessen ist bei EPKs, ähnlich wie bei Flowcharts, die Funktion (bei Flowcharts: Operation). Die Abfolge der Funktionen in einem Prozess wird aber nicht ausschließlich über Verbindungspfeile festgelegt. Durch

den Einsatz von Ereignissen wird der Ablauf hier genauer spezifiziert. Jede Funktion wird durch ein Ereignis ausgelöst und erzeugt selbst ein oder mehrere Ereignisse. Ein Prozess wird also durch eine Folge von Ereignissen und Funktionen dargestellt, wobei sich Ereignisse und Funktionen immer abwechseln. Bei der Benennung muss hier darauf geachtet werden, dass Funktionen einen Vorgang beschreiben (etwa „Antrag prüfen“), während Ereignisse einen Zustand beschreiben (etwa „Antrag bestätigt“ oder „Antrag abgelehnt“).

Eine EPK beginnt und endet immer mit einem Ereignis. Ereignisse, die einen Prozess auslösen, sind Start-Ereignisse. Ereignisse, die den Abschluss eines Prozesses beschreiben, sind Ende-Ereignisse. Folgeprozesse können durch Ende-Ereignisse eines vorangegangenen Prozesses ausgelöst werden, d. h. ein Ende-Ereignis kann in einem anderen Prozess ein auslösendes Start-Ereignis darstellen.

Durch den Einsatz unterschiedlicher Konnektoren können nun in Kombination mit Ereignissen verschiedene Ablaufvarianten eines Prozesses innerhalb eines Modells abgebildet oder auch die Möglichkeit zur parallelen Ausführung von Funktionen (wenn diese nicht voneinander abhängig sind) dargestellt werden (siehe Abb. 3.6). Es lassen sich der UND-, ODER- und XOR-Konnektor unterscheiden.

Wird ein *UND-Konnektor* mit mehreren ausgehenden Verbindungen verwendet, so bedeutet dies, dass alle ausgehenden Pfade parallel durchlaufen werden. Diese werden dann in der Regel an einer zeitlich bzw. kausal nachgelagerten Stelle wieder mit einem weiteren UND-Konnektor zusammengeführt. Die darauf folgende Funktion wird erst ausgeführt, wenn alle der zusammengeführten Pfade abgeschlossen sind.

Ein *ODER-Konnektor* mit mehreren ausgehenden Kanten zeigt an, dass ein oder mehrere der folgenden Pfade parallel durchlaufen werden. Diese Pfade werden zumeist an einer späteren Stelle wiederum mit einem ODER-Konnektor zusammengeführt, wobei der dann folgende Prozessschritt erst zu einem Zeitpunkt durchgeführt wird, wenn genau die Pfade, die bei der entsprechenden ODER-Verzweigung ausgewählt wurden, abgearbeitet sind. Wichtig ist hier, dass die zu aktivierenden Pfade zum Zeitpunkt des Eintreffens beim Konnektor ausgewählt werden müssen. Hinter jedem ODER-Konnektor muss sich auf jedem Pfad ein Ereignis befinden, das durch die vorangegangene Funktion ausgelöst werden könnte. Es werden jene Pfade aktiviert, deren erste Ereignisse tatsächlich eingetreten sind.

Ein *XOR-Konnektor* steht schließlich für ein „exklusives Oder“ bzw. ein „entweder, oder“. Die Verwendung eines XOR-Konnektors mit mehreren ausgehenden Kanten bedeutet, dass bei der Durchführung des Prozesses genau einer der folgenden Pfade



Abb. 3.6 Notationselemente von EPKs

ausgewählt wird. Er eignet sich also zur Abbildung von einander ausschließenden Alternativen in der Prozessabarbeitung. Bei der Zusammenführung dieser Pfade mit einem XOR-Konnektor wird der folgende Schritt dann ausgeführt, wenn der ausgewählte Pfad fertig durchlaufen wurde. Auch bei einem XOR-Konnektor muss sich auf jedem Pfad ein Ereignis befinden, das durch die vorangegangene Funktion ausgelöst werden könnte. Diese Ereignisse müssen einander ausschließen. Es wird jener Pfade aktiviert, dessen erstes Ereignis tatsächlich eingetreten ist. Im Gegensatz zu Flowcharts ist es hier möglich, auch mehr als zwei Alternativen zu beschreiben, solange die eingesetzten Ereignisse einander ausschließen.

3.3.2 Beispiele zur EPK

Wir verwenden hier vorerst die gleichen Beispiele, wie sie für Flowcharts angegeben wurden, um die Unterschiede in der Notation zu visualisieren.

Dieses Beispiel (siehe Abb. 3.7) zeigt einen Prozess mit einer einzelnen Funktion, in der ein Antrag (über den wir hier nicht mehr wissen) bearbeitet wird. Der Prozess endet, nachdem der Antrag geprüft wurde.

Im obenstehenden Beispiel (siehe Abb. 3.8) ist der Prozess um eine Entscheidung erweitert. Der Antrag wird geprüft, das Ergebnis dieser Prüfung ermöglicht das Treffen einer Entscheidung auf Basis dessen positiver oder negativer Beurteilung. Das Bestätigen bzw. Ablehnen selbst sind ebenfalls Funktionen. Die ausgehenden Verbindungen werden mit einem XOR-Konnektor zusammengeführt, nachdem die alternativen Zweige abgeschlossen sind.

Die Darstellung von Entscheidungen, die mehr als zwei mögliche Ausgänge haben, ist hier nun einfacher möglich als bei Flowcharts (siehe Abb. 3.9). Dem XOR-Konnektor werden in diesem Fall drei Ereignisse nachgelagert, die sich alle auf die Investitionssumme beziehen und einander ausschließen.

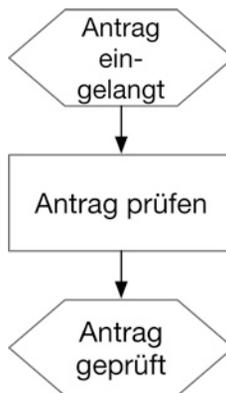


Abb. 3.7 Einfache EPK

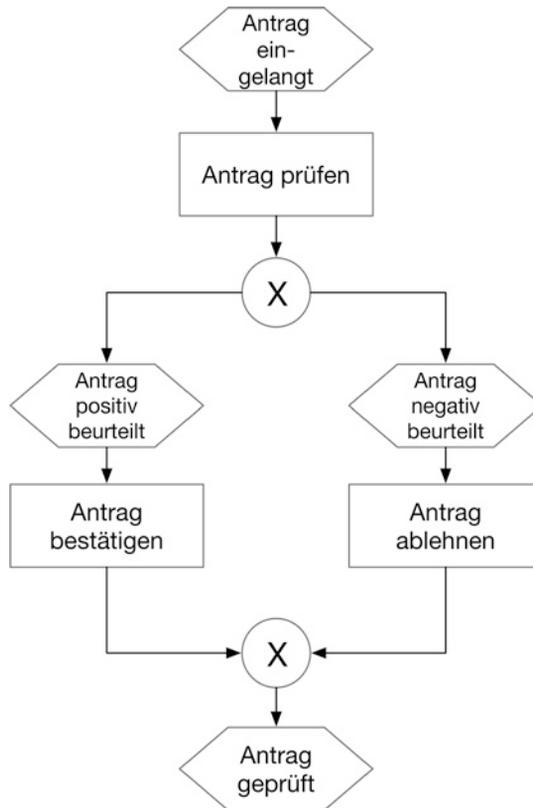


Abb. 3.8 EPK mit Verzweigung

Der XOR-Konnektor kann auch genutzt werden, um Teile eines Prozesses wiederholt abzuarbeiten. Dazu wird der Konnektor am Ende des zu wiederholenden Teils eingefügt und ein ausgehender Zweig auf das Ereignis zurückgeführt, das den zu wiederholenden Teil auslöst. Der andere ausgehende Zweig führt den Prozess nach der wiederholten Abarbeitung weiter und muss dementsprechend zu einem Ereignis führen, das den Abbruch der Wiederholung auslöst. Im obenstehenden Beispiel (siehe Abb. 3.10) werden also solange Anträge bearbeitet als Anträge vorhanden sind.

Der UND-Konnektor kann zum Einsatz kommen, wenn zwei Funktionen unabhängig voneinander ausgeführt werden können. Der im obigen Beispiel (siehe Abb. 3.11) modellierte Prozess ist also nur korrekt, wenn die Beilagen tatsächlich unabhängig vom Antrag geprüft werden können. Ist dies nicht der Fall, müssten die beiden Funktionen sequenziell angeordnet werden. Aus Sicht der Modellerstellung ist hier anzumerken, dass dem UND-Konnektor im Gegensatz zu den anderen Konnektoren nicht unmittelbar Ereignisse folgen müssen, da ja keine Entscheidung getroffen wird, sondern in jedem Fall alle ausgehenden Zweige aktiviert werden.

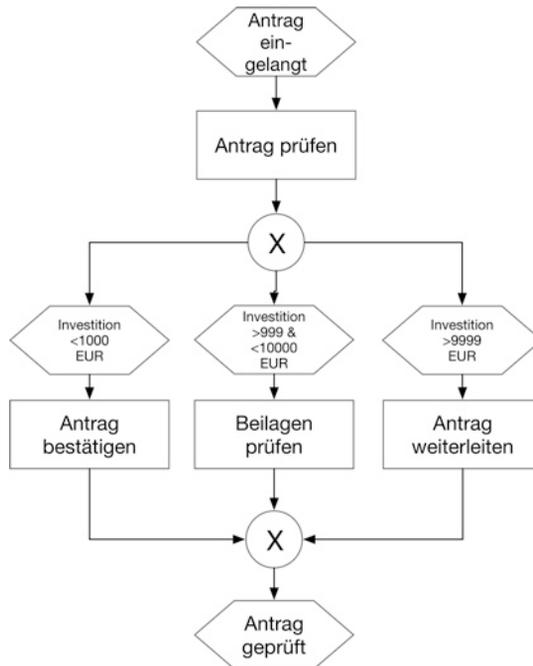


Abb. 3.9 EPK mit Verzweigung in drei Alternativen



Abb. 3.10 EPK mit Schleife

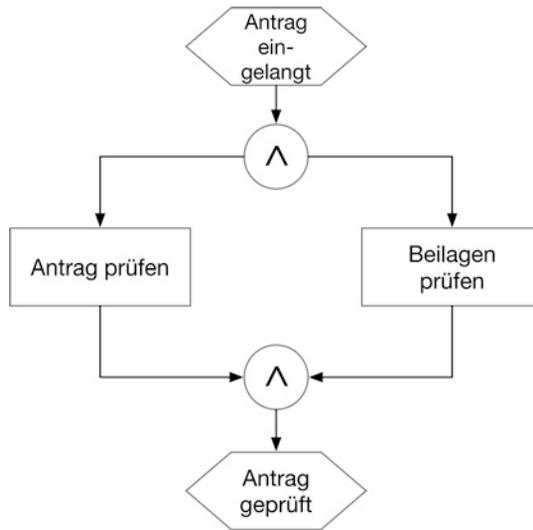


Abb. 3.11 EPK mit UND-Konnektor und parallel ablaufenden Zweigen

Die obige Abbildung (siehe Abb. 3.12) zeigt ein Beispiel für den Einsatz des ODER-Konnektors. Hier gehen wir davon aus, dass ein Antrag Angebote oder Stellungnahmen oder beides enthalten kann (aber zumindest Angebote oder Stellungnahmen enthalten muss). Wären Angebote und Stellungnahmen vollkommen optional (könnten also beide fehlen), würde deren grundsätzliche Notwendigkeit mit einem vorgelagerten XOR-Konnektor und entsprechenden Ereignissen zu prüfen sein. Alternativ könnte nach dem ODER-Konnektor ein zusätzlicher Zweig mit einem Ereignis „Antrag alleine genügt“ eingefügt werden. In beiden Fällen muss darauf geachtet werden, dass die Bedingung, dass sich Ereignisse und Funktionen auf allen Pfaden durch den Prozess abwechseln, nicht verletzt wird. Falls nicht anders möglich, muss dies durch eine „Dummy“-Funktion, die keine Aktivität verursacht, gewährleistet werden.

3.3.3 Ergänzende Notationselemente der eEPK

Die eEPK ergänzt nun den in einer EPK abgebildeten Geschäftsprozess um Information über dessen Ausführungskontext. Insbesondere werden hier den Funktionen Verantwortlichkeiten und Ressourcenbedarf zugeordnet. Die grundsätzlichen Regeln einer EPK bleiben wie oben beschrieben in Kraft. Die zusätzlichen Elemente können Funktionen zugeordnet werden. Ereignisse sind nicht betroffen. Wir verzichten an dieser Stelle auf eine vollständige Beschreibung der möglichen Elemente und führen nur die gängigsten an (siehe Abb. 3.13).

Verantwortlichkeiten werden durch *Organisationale Einheiten* abgebildet. Solche Einheiten sind üblicherweise nicht konkrete Personen, sondern werden abstrakt durch

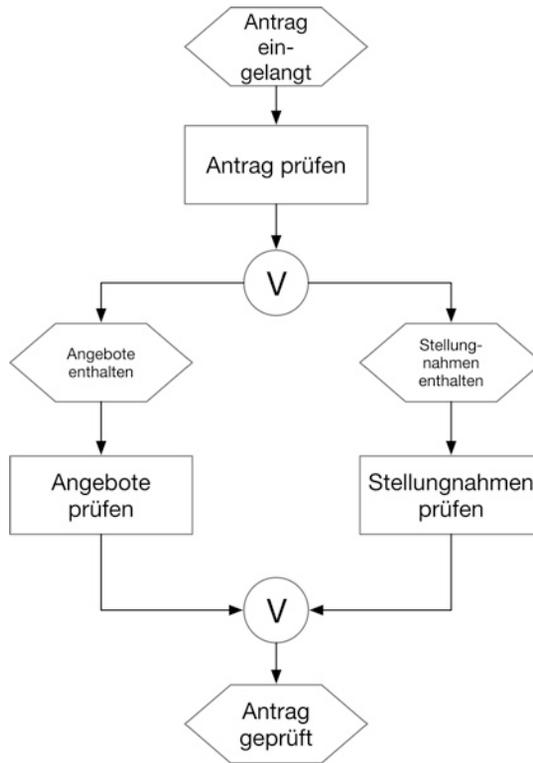


Abb. 3.12 EPK mit ODER-Konnektor und optional parallel ablaufenden Zweigen

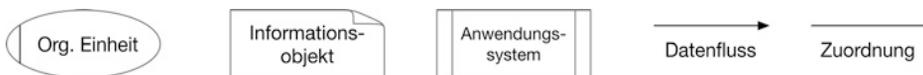


Abb. 3.13 Zusätzliche Notationselemente in eEPKs

die Benennung von Rollen (etwa: Geschäftsführung) oder Abteilungen (etwa: Finanzbuchhaltung) angeführt. Dadurch wird gewährleistet, dass die Spezifikation eines Prozesses von der Verfügbarkeit konkreter Personalressourcen unabhängig ist und erst zur Durchführungszeit konkrete Personen zugewiesen werden müssen. Die Zuordnung zu einer Funktion erfolgt durch ungerichtete Linien. Eine organisationale Einheit kann so mehreren Funktionen zugeordnet werden. Auch ist es möglich, organisationale Einheiten mehrfach anzuführen, wenn die Prozessdarstellung dadurch übersichtlicher wird.

In ähnlicher Form werden *Anwendungssysteme* modelliert. Sie kennzeichnen die Notwendigkeit, bei der Ausführung einer Funktion ein bestimmtes IT-System einzusetzen (z. B. ein ERP-System oder eine Datenbank). Sie werden Funktionen ebenfalls mit ungerichteten Linien zugeordnet.

Informationsobjekte werden dazu verwendet, um die Datenverarbeitung in einem Geschäftsprozess darzustellen. Ein Informationsobjekt kann beliebig umfassend sein (also ein einzelner Wert ebenso wie ein vollständiges Dokument) und wird einer Funktion immer mittels einem gerichteten Pfeil zugeordnet, der den Datenfluss beschreibt. Endet der Pfeil bei der Funktion, bedeutet dies, dass das Informationsobjekt zur Durchführung der Funktion benötigt wird. Endet der Pfeil beim Informationsobjekt, so bedeutet dies, dass es durch die Funktion erzeugt oder verändert wird. Ein Informationsobjekt kann dabei mehrere ein- und ausgehende Verbindungen haben, wodurch sowohl dessen Entstehung als auch dessen Verwendung in einem Geschäftsprozess beschrieben werden kann.

3.3.4 Beispiele zur eEPK

Zur Illustration von eEPKs ergänzen wir eines der oben angeführten Beispiele um den stattfindenden Datenfluss sowie die benötigten Personal-Ressourcen. Anwendungssysteme würden analog zur Verwendung von organisationalen Einheiten modelliert werden (siehe Abb. 3.14).

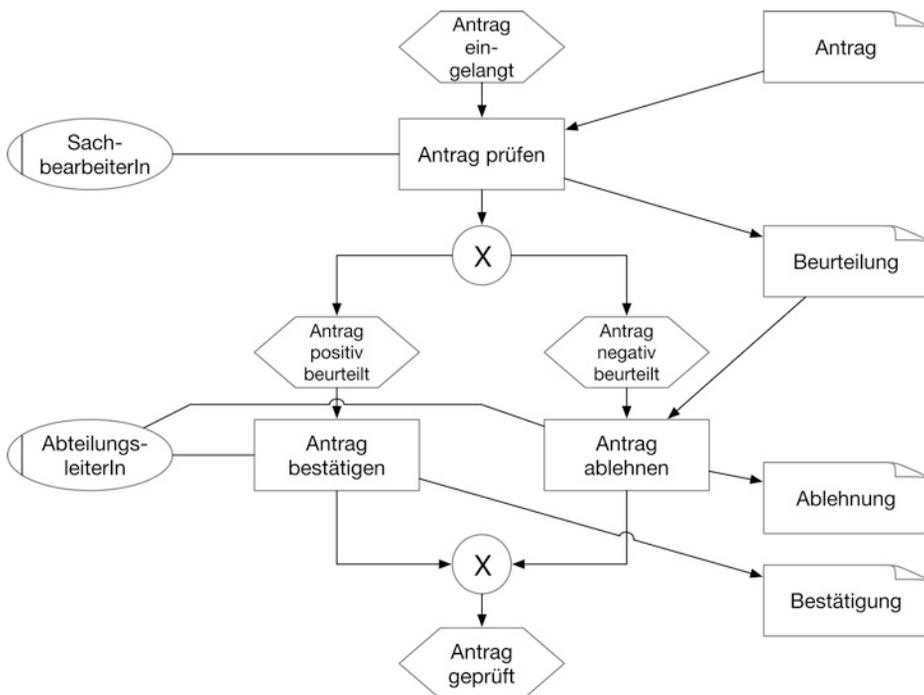


Abb. 3.14 Beispiel für eine eEPK

Hinsichtlich des Datenflusses erkennen wir nun, dass zum Prüfen des Antrags der eigentliche Antrag vorhanden sein muss. Diese Prüfung führt nicht nur zur positiven oder negativen Beurteilung eines Antrags im Prozessablauf, sondern auch zu einem Informationsobjekt, in dem die Beurteilung gespeichert ist. Im Falle einer negativen Beurteilung wird dieses Informationsobjekt benötigt, um die Ablehnung zu erstellen (wir können also annehmen, dass die Ablehnung eine inhaltliche Begründung enthält). Im Falle der Bestätigung des Antrags wird das Datenobjekt „Beurteilung“ nicht mehr benötigt – wir können also annehmen, dass in diesem Fall keine weitere Begründung erfolgt.

Hinsichtlich der Verantwortlichkeiten erkennen wir nun, dass mehrere organisationale Einheiten am Prozess beteiligt sind. Während die Antragsprüfung durch einen Sachbearbeiter erfolgt, ist für die endgültige Bestätigung oder Ablehnung der Abteilungsleiter zuständig. Wichtig ist hier, dass die Ereignisse, auf Basis derer die Entscheidung getroffen wird, durch die Funktion „Antrag prüfen“ ausgelöst werden, für die der Sachbearbeiter zuständig ist.

3.3.5 Einordnung

Die EPK bietet umfassendere Möglichkeiten zur Abbildungen von Geschäftsprozessen als Flowcharts. Gemein ist ihnen die Orientierung an den Abläufen innerhalb einer Organisation als primäres Strukturierungsmerkmal des Geschäftsprozesses (d. h. alle im Modell abgebildete Information ist an der Beschreibung des Prozessablaufs verankert). Dies ist zwar naheliegend, wenn ein organisationaler Prozess beschrieben wird, aber – wie wir später sehen werden – nicht unbedingt die einzige Möglichkeit. Andere Modellierungssprachen nutzen Akteure oder Daten als primäre Strukturierungsmerkmale, an denen alle anderen Informationen verankert werden und machen so Aspekte eines Prozesses sichtbar, die in (e)EPKs nur implizit abgebildet werden können (wie etwa der Übergang von Verantwortlichkeiten im Prozess und der dabei notwendigen Kommunikation zwischen den Akteuren).

Die Notwendigkeit, Funktionen und Ereignisse einander immer abwechseln zu lassen, führt in EPKs zu sehr umfangreichen und zum Teil nur schwer verständlichen Modellen. Außerdem birgt sie das Risiko, Modellierende bei der Erstellung der Modelle dazu zu verleiten, Trivial-Ereignisse zu formulieren, die dem Modell keine Information hinzufügen (etwa: Funktion: „Aufgabe ausführen“, Ereignis: „Aufgabe ausgeführt“). Korrekt eingesetzt, bietet diese Systematik aber Vorteile: Einerseits können Prozesse exakter beschrieben und abgegrenzt werden als etwa mit Flowcharts, andererseits erlaubt die EPK eine Verknüpfung zwischen der Sicht auf die Fähigkeiten einer Organisation (ihrer Funktionen) und der Sicht darauf, wie sie mithilfe ihrer Fähigkeiten auf externe Reize oder Ereignisse innerhalb der Organisation selbst reagiert. So können organisationale Fähigkeiten generisch beschrieben und mehrfach in Prozessen eingesetzt werden, wodurch Ineffizienzen durch Replikation vermieden werden.

Aus pragmatischer Sicht hat sich in der Praxis jedoch gezeigt, dass sowohl die Spezifikation generischer Funktionen als auch prozessspezifischer Ereignisse nicht immer

in der notwendigen Gesamtheit umsetzbar ist. Modernere Ansätze, wie die im folgenden behandelten Aktivitätsdiagramme oder die BPMN, kennen deshalb zwar nach wie vor das Konzept von Ereignissen, setzen diese aber nur ein, wenn tatsächlich auf einen externen Reiz (wie eine eingehende Nachricht, einen Fehler, oder einen Zeitablauf) eingegangen werden soll.

In Abgrenzung zu den Modellierungssprachen mit technisch orientierter Entstehungsgeschichte (wie Flowcharts oder die im nächsten Abschnitt beschriebenen Aktivitätsdiagramme) verfolgen die eEPK und das umgebende ARIS-Framework als ein ursprünglich aus der Betriebswirtschaftslehre stammendes Konzept einen umfassenderen Ansatz zur Beschreibung von Geschäftsprozessen. Die Berücksichtigung von Daten, Verantwortlichkeiten, aber auch Zielen oder Leistungen (die hier nicht diskutiert wurden) ermöglicht insgesamt eine umfassende Modellierung von Geschäftsprozessen, die bis heute Einfluss auf die Gestaltung von Modellierungssprachen zur Abbildung organisationaler Phänomene (wie Geschäftsprozessen oder Unternehmensarchitekturen) hat.

3.4 UML Aktivitätsdiagramme

Das Aktivitätsdiagramm wurde als Teil der UML (Unified Modeling Language) definiert, die eine Sammlung von Diagrammen enthält, die zur Spezifikation von Softwaresystemen geeignet sind. Das Aktivitätsdiagramm dient dabei analog zum Flowchart zur Abbildung des Verhaltens eines Softwaresystems, stellt aber ob seiner jüngeren Entstehungsgeschichte auch Elemente zur Abbildung verteilter und paralleler Prozessabläufe zur Verfügung. Wie das Flowchart ist auch das Aktivitätsdiagramm zur Abbildung organisationaler Abläufe, also von Geschäftsprozessen, geeignet. Während dieses bis heute zu diesem Zweck eingesetzt wird, hat sich der Fokus im Bereich der Geschäftsprozessmodellierung stark hin zur BPMN (Business Process Modeling and Notation) verschoben. Diese wurde vom gleichen Standardisierungsgremium wie die UML spezifiziert und hat viele Elemente des Aktivitätsdiagramms übernommen. Die BPMN fokussiert explizit auf die Anforderungen der Geschäftsprozessmodellierung und der dort abzubildenden organisationalen Aspekte, die wir bereits im Rahmen der EPKs diskutiert haben.

3.4.1 Notationselemente

Ein Aktivitätsdiagramm beschreibt per Definition immer eine Aktivität, die sich aus einzelnen *Aktionen* zusammensetzt („Aktivität“ wird hier also analog zu „Prozess“ verwendet). Eine Aktion entspricht einer Operation bei Flowcharts oder einer Funktion bei EPKs (siehe Abb. 3.15).

Eine Aktivität beginnt üblicherweise mit einem Startknoten und endet mit einem *Endknoten* (analog zu den Terminierungselementen bei Flowcharts). Zwischen diesen Knoten werden die enthaltenen Aktionen angegeben und durch Ablaufpfeile in die

Abb. 3.15 Notationselemente von Aktivitätsdiagrammen

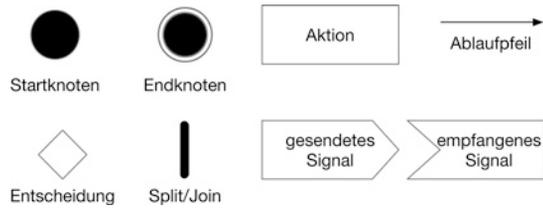


Abb. 3.16 Notationselemente zur Abbildung von Verantwortlichkeiten und Datenflüssen in Aktivitätsdiagrammen



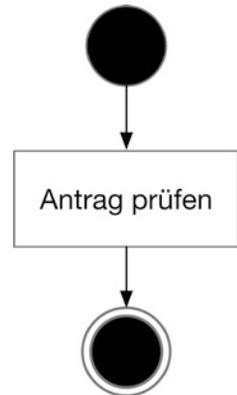
durchzuführende Reihenfolge gebracht. Zur Beeinflussung des Ablaufs ist es möglich, *Entscheidungselemente* einzufügen. Entscheidungen können beliebig viele ausgehende Zweige haben, deren Aktivierungsbedingungen sich einander ausschließen müssen. Die Bedingungen werden an den ausgehenden Verbindungen angeführt. Die Semantik des Entscheidungssymbols entspricht dem XOR in der EPK – für den ODER-Konnektor gibt es in Aktivitätsdiagrammen keine Entsprechung.

Um voneinander unabhängig parallel ausführbare Prozessteile abzubilden, bietet das Aktivitätsdiagramm das *Split/Join*-Element. Zum Aufspalten des Ablaufs eingesetzt, kann es beliebig viele ausgehende Verbindungen haben, die alle gleichzeitig aktiviert werden. Die so angelegten Zweige sollten durch ein Join wieder zusammengeführt werden. Der Ablauf wird erst fortgesetzt, sobald alle Zweige abgearbeitet sind.

Signale dienen der Kommunikation zwischen Prozessteilen in unterschiedlichen Aktivitäten (also in unterschiedlichen Diagrammen) oder innerhalb einer Aktivität, wenn Information für spätere Prozessteile bereitgestellt werden soll. Sie werden wie Aktionen in den Kontrollfluss eingebaut. Modelle müssen nicht immer vollständige Signalkette (also gesendete und empfangene Signale) enthalten, sondern können auch Signale für nicht abgebildete Prozesse bereitstellen (also nur ein gesendetes Signal enthalten), oder dieses von einem nicht abgebildeten Prozess entgegennehmen (also nur ein empfangenes Signal enthalten). Empfangene Signale können außerdem eine Aktivität auslösen und damit den Startknoten in einem Diagramm ersetzen.

Das Aktivitätsdiagramm bietet auch Elemente, um Verantwortlichkeiten und Datenflüsse abzubilden (siehe Abb. 3.16). Verantwortlichkeiten werden mittels *Partitionen* abgebildet. Partitionen sind Elemente, die Teile eines Aktivitätsdiagramms umschließen und damit festlegen, dass alle Elemente, insbesondere Aktionen, die von ihnen umschlossen sind, in die Verantwortlichkeit der angeführten organisationalen Einheit (oder bei Softwaresystemen der Systemkomponente) fällt. Wenn Partitionen eingesetzt

Abb. 3.17 Einfaches Aktivitätsdiagramm



werden (was nicht verpflichtend ist), so sollten alle Elemente des Aktivitätsdiagramms von genau einer der abgebildeten Partitionen umschlossen werden. Überlappungen sind nicht erlaubt, Aktionen außerhalb einer Partition sollten vermieden werden.

Datenobjekte werden in Aktivitätsdiagrammen direkt im Kontrollfluss, und zwar zwischen Aktionen modelliert. Sie können damit (nur) dafür eingesetzt werden, den Informationsfluss zwischen zwei aufeinanderfolgenden Aktionen abzubilden. Wird ein Datenobjekt erst später im Prozessablauf wieder benötigt, so müsste dieses im Kontrollfluss über alle dazwischenliegenden Aktionen weitergegeben oder mittels eines Signals übergeben werden.

3.4.2 Beispiele

Um die Unterschiede und Gemeinsamkeiten zu den zuvor diskutierten Modellierungssprachen herauszuarbeiten, bedienen wir uns hier wieder der bereits verwendeten Beispiele.

Dieses Beispiel (siehe Abb. 3.17) zeigt eine Aktivität mit einer einzelnen Aktion, in der ein Antrag (über den wir hier nicht mehr wissen) bearbeitet wird. Die Aktivität endet, nachdem die Bearbeitung des Antrags abgeschlossen wurde.

Im obenstehenden Beispiel (siehe Abb. 3.18) ist der Prozess um eine Entscheidung mit drei möglichen Ausgängen erweitert, die einander ausschließen. Der Antrag wird geprüft, das Ergebnis dieser Prüfung ermöglicht das Treffen einer Entscheidung über die weitere Bearbeitung.

Auch hier können wir Verzweigungen nutzen, um Teile eines Prozesses wiederholt abzarbeiten (siehe Abb. 3.19). Dazu wird die Entscheidung am Ende des zu wiederholenden Teils eingefügt und ein ausgehender Zweig zu einer schließenden Entscheidung vor die erste Operation des zu wiederholenden Teils zurückgeführt (eine schließende Entscheidung dient der Zusammenführung und hat mehrere eingehende und

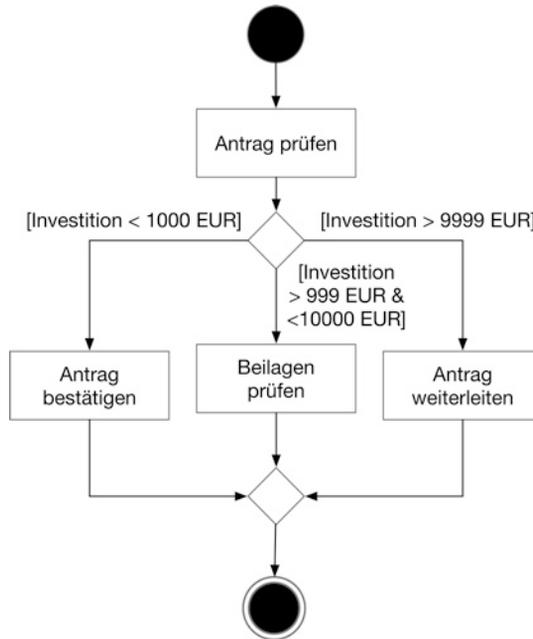


Abb. 3.18 Aktivitätsdiagramm mit Verzweigung (drei Zweige)

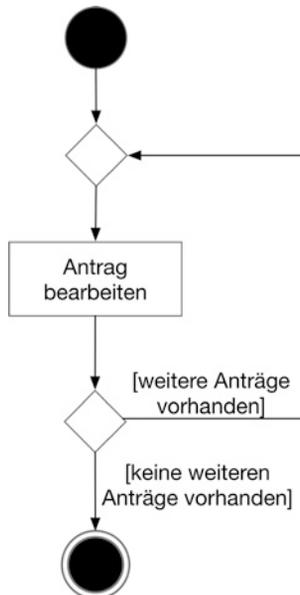


Abb. 3.19 Aktivitätsdiagramm mit Schleife

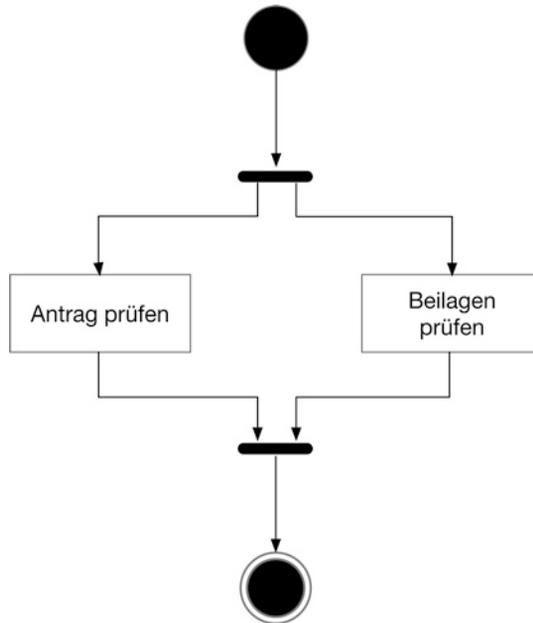


Abb. 3.20 Aktivitätsdiagramm mit parallel ablaufenden Zweigen

nur eine ausgehende Verbindung). Der andere ausgehende Zweig der öffnenden Verzweigung führt den Prozess nach der wiederholten Abarbeitung weiter.

Das Split/Join-Element kann zum Einsatz kommen, wenn zwei Aktionssequenzen unabhängig voneinander ausgeführt werden können. Der im obigen Beispiel (siehe Abb. 3.20) modellierte Prozess ist also nur korrekt, wenn die Beilagen tatsächlich unabhängig vom Antrag geprüft werden können. Ist dies nicht der Fall, müssten die beiden Funktionen sequenziell angeordnet werden. Beim zusammenführenden Join-Element wird auf den Abschluss beider eingehender Zweige gewartet, bevor der Prozess fortgesetzt wird.

Das obige Beispiel (siehe Abb. 3.21) zeigt den Einsatz von Partitionen, Datenobjekten und Signalen. Mittels Partitionen werden die Verantwortlichkeiten im Prozess abgebildet. Dessen Auslösung durch ein empfangenes Signal bildet explizit ab, dass die Ausführung erst dann startet, wenn ein Antrag einlangt (das hatten wir bei Flowcharts und den anderen Beispielen zum Aktivitätsdiagramm bislang immer implizit angenommen, aber im Gegensatz zu EPKs nie im Modell abbilden können). Gesendete Signale werden eingesetzt, um die Bestätigung oder Ablehnung an den Empfänger (der hier nicht abgebildet ist) zu übermitteln. Die Beurteilung des Antrags wird nur im Falle einer negativen Beurteilung als Datenobjekt zur Aktion „Antrag ablehnen“ übergeben – wir können folglich annehmen, dass die Ablehnung eine inhaltliche Begründung enthält. Im Falle der Bestätigung des Antrags wird das Datenobjekt „Beurteilung“ nicht mehr benötigt – wir können also annehmen, dass in diesem Fall keine weitere Begründung erfolgt.

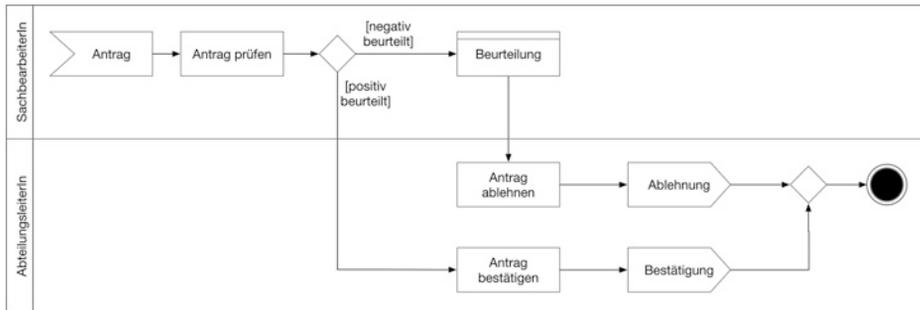


Abb. 3.21 Aktivitätsdiagramm mit Partitionen und Datenobjekten und Signalen

3.4.3 Einordnung

Aktivitätsdiagramme vereinen mit gewissen Einschränkungen die Einfachheit der Flowchart-Notation mit der Ausdrucksstärke von EPKs. Sie erlauben es, die Behandlung von Daten im Prozess darzustellen und führen mit Partitionen ein Mittel zur übersichtlichen Abbildung von Verantwortlichkeiten ein. Die Verfügbarkeit von Signalen erlaubt im Gegensatz zu den bisher behandelten Sprachen erstmals eine Abbildung von Kommunikationsvorgängen zwischen Prozessbeteiligten oder mit der Umwelt des abgebildeten Prozesses.

Das Fehlen eines Elements, das dem ODER-Konnektor in der EPK entspricht, stellt zwar eine Einschränkung dar, die allerdings in der Praxis selten schlagend wird, da im realen Umfeld zumeist einander ausschließende Alternativen oder vollständig voneinander unabhängige Ausführungszweige vorkommen. Insgesamt stellen Aktivitätsdiagramme also ein geeignetes Mittel zur Abbildung von Geschäftsprozessen dar, vor allem wenn die Zielgruppe für die Modellverwendung einen informationstechnischen Hintergrund hat und mit der Notation bereits vertraut ist. Für andere Zielgruppen ist aufgrund der flexibleren Einsetzbarkeit und der höheren Ausdrucksstärke für Geschäftsprozessmodellierung die BPMN vorzuziehen, die wir im nächsten Abschnitt behandeln werden.

3.5 BPMN

Die BPMN – Business Process Modeling (and) Notation – wurde 2002 bei IBM entwickelt und nachfolgend von der BPMI (Business Process Management Initiative) veröffentlicht. Ziel war es, der Vielzahl an Prozessmodellierungssprachen, die im akademischen Bereich und in der industriellen Praxis eingesetzt wurden, einen universell verwendbaren Standard entgegen zu setzen. Dieser sollte die wesentlichen Eigenschaften der gängigsten Sprachen übernehmen und es ermöglichen, neben der Dokumentation von Geschäftsprozessen auch Modelle zu erstellen, die unmittelbar zur IT-unterstützten

Ausführung geeignet sind. Die BPMN wiederum wurde 2005 von der OMG (Object Management Group) übernommen bzw. fusionierte mit dieser. Dadurch wurde BPMN zum OMG-Standard und ergänzt damit die bereits erwähnte UML (Unified Modelling Language).

Im dritten Quartal 2010 wurde der neue Standard BPMN 2.0 veröffentlicht. Dieser Standard umfasst unterschiedliche Diagrammtypen: das Choreografie-, das Konversations- und das Kollaborationsdiagramm. Im Folgenden betrachten wir die grundlegenden Elemente der BPMN 2.0, die die Abbildung von Geschäftsprozessen auf fachlicher Ebene ermöglichen.

Die BPMN konzentriert sich auf Geschäftsprozesse, welche sie als eine zeitlich logische Abfolge von Aktivitäten (Aufgaben) darstellt und hinsichtlich der organisationalen Verantwortlichkeiten strukturiert. Die Darstellung von Daten ist nur ansatzweise und im Kontext von Prozessabläufen vorgesehen.

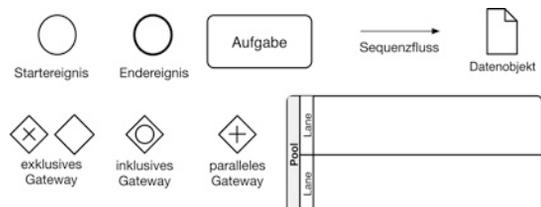
3.5.1 Notationselemente zur Modellierung von Abläufen

Prozessdiagramme, die mit der BPMN erstellt wurden, werden „Business Process Diagrams (BPD)“ genannt. Das BPD orientiert sich in seinen Kernelementen an Aktivitätsdiagrammen, die um Elemente ergänzt sind, die eine potenziell komplexe Ablaufsteuerung in Geschäftsprozesse abbildbar machen (siehe Abb. 3.22).

Im Prinzip müssen in einem Prozess bestimmte Dinge getan werden (*Aufgaben*), möglicherweise aber nur unter bestimmten Bedingungen (*Gateways*), und es können Dinge passieren (*Ereignisse*). Diese drei Flussobjekte werden über Sequenzflüsse miteinander verbunden, jedoch nur innerhalb eines Pool bzw. einer Lane. Pools und Lanes sind Konstrukte, um Verantwortlichkeiten in verteilten Geschäftsprozessen darzustellen. Sie werden im Folgenden genauer betrachtet. Falls eine Verbindung über Pool-Grenzen hinweg erfolgt, wird diese mittels Nachrichtenflüssen modelliert, die wir später detaillieren werden.

Ein Prozess besteht aus *Aufgaben* (Tasks). Nach dem Start eines Prozesses (durch ein Ereignis) folgt eine Aufgabe der anderen bis der Prozess endet (durch ein Ereignis). Aufgaben können dabei atomar sein (also nicht weiter verfeinert sein), oder als Subprozess vorliegen. In diesem Fall wird eine Aufgabe durch ein eingebettetes weiteres BPD verfeinert, d. h. dessen detaillierter Ablauf dargestellt. Dieser detaillierte Ablauf

Abb. 3.22 Grundlegende Notationselemente in BPMN



kann „versteckt“ werden und wird durch ein „+“-Symbol am unteren Rand der Aufgabe repräsentiert.

Ein Prozess beginnt mit einem *Start-Ereignis* und endet in einem *End-Ereignis*. Die BPMN bietet eine Vielzahl an Möglichkeiten, Ereignisse zu definieren, die einen Prozess auslösen, abschließen oder dessen Verlauf beeinflussen können. Diese werden wir später näher behandeln.

An dieser Stelle ist wichtig zu betonen, dass ein Prozess mit einem oder mehreren Start-Ereignissen beginnen kann und auf jedem Pfad durch den Prozess (siehe Sequenzfluss und Gateways weiter unten) mit einem oder mehreren End-Ereignissen abgeschlossen werden kann. Von jedem Startereignis muss es einen durchgängigen Sequenzfluss zu mindestens einem Endereignis geben. Aufgaben, Gateways oder Zwischen-Ereignisse dürfen keine Endpunkte im Prozess sein, benötigen also auch immer mindestens einen ausgehenden Sequenzfluss.

Ein Gateway ist eine Abzweigung im Kontrollfluss. Das *exklusive (oder XOR-)Gateway* benötigt zu jedem ausgehenden Kontrollfluss eine Bedingung, die sich lt. Standard immer auf das Ergebnis einer unmittelbar vorangehenden Aufgabe beziehen muss.

Das *parallele (oder AND-)Gateway* verfolgt alle ausgehenden Kontrollflüsse unabhängig voneinander und parallel weiter. Die verzweigten Kontrollflüsse können separat mit Endereignissen beendet oder wieder explizit mit einem weiteren parallelen Gateway zusammengeführt werden. Der Kontrollfluss setzt nach dieser Zusammenführung erst fort, wenn alle eingehenden Kontrollflüsse abgeschlossen sind (entsprechend dem Split/Join-Konzept bei Aktivitätsdiagrammen).

Das *inklusive (oder OR-)Gateway* kann einen oder mehrere Pfade weiterverfolgen, wobei zur Pfadauswahl jeweils (wie beim exklusiven Gateway) eine Bedingung angeführt werden muss. Diese Bedingung muss zum Zeitpunkt der Entscheidung bereits prüfbar sein, die notwendigen Daten müssen also in einer der vorangegangenen Aufgaben erzeugt worden sein.

Für Entscheidungen, die nicht auf Basis bereits existierender Daten getroffen werden können, bietet sich die Verwendung des *ereignisbasierten Gateways* an. Dieses benötigt in jedem ausgehenden Zweig unmittelbar nach dem Gateway ein Ereignis (z. B. eintreffende Nachricht oder Timer). Es wird dann jener Zweig (und nur dieser) aktiviert, dessen Ereignis zuerst eintritt. Dieses werden wir genauer behandeln, wenn wir die Verwendung von Ereignissen diskutieren.

3.5.2 Beispiele zur Modellierung von Abläufen

Um die Unterschiede und Gemeinsamkeiten zu den zuvor diskutierten Modellierungssprachen herauszuarbeiten, bedienen wir uns hier wieder der bereits verwendeten Beispiele.

Dieses Beispiel (siehe Abb. 3.23) zeigt einen Prozess mit einer einzelnen Aufgabe, in der ein Antrag (über den wir hier nicht mehr wissen) bearbeitet wird. Der Prozess endet, nachdem die Bearbeitung des Antrags abgeschlossen wurde.

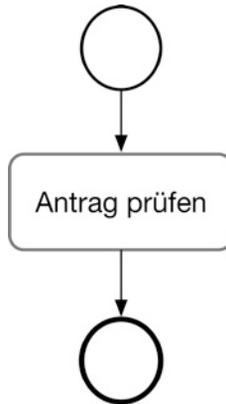


Abb. 3.23 Einfaches BPMN-Modell

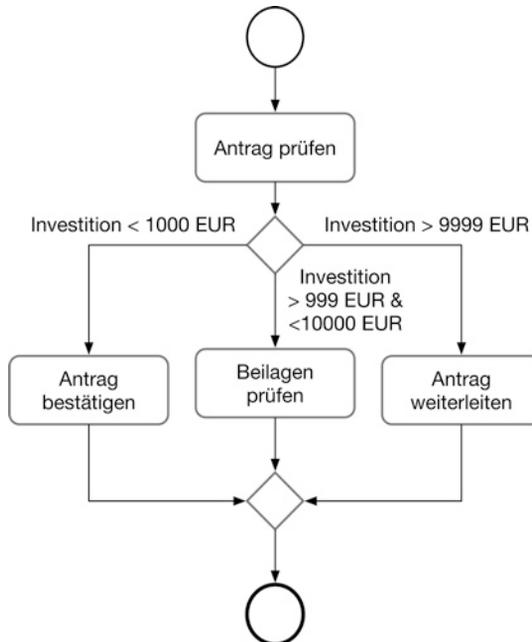


Abb. 3.24 BPMN-Modell mit Verzweigung (3 Zweige)

Im obenstehenden Beispiel (siehe Abb. 3.24) ist der Prozess um eine Entscheidung mit drei möglichen Ausgängen erweitert, die einander ausschließen. Der Antrag wird geprüft, das Ergebnis dieser Prüfung ermöglicht das Treffen einer Entscheidung über die weitere Bearbeitung. In der BPMN ist wichtig, dass die Daten, die als Grundlage einer

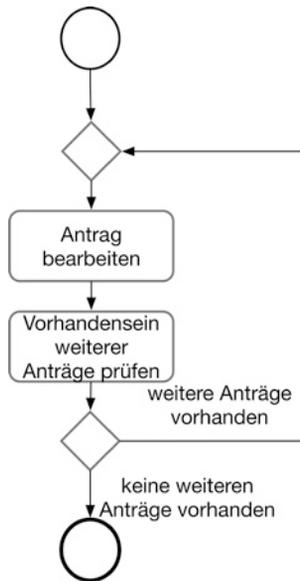


Abb. 3.25 BPMN-Modell mit Schleife

Entscheidung verwendet werden, vor dem Gateway explizit erzeugt oder empfangen werden.

Auch hier können wir Verzweigungen nutzen, um Teile eines Prozesses wiederholt abzarbeiten (siehe Abb. 3.25). Dazu wird die Entscheidung am Ende des zu wiederholenden Teils eingefügt und ein ausgehender Zweig zu einer schließenden Entscheidung vor die erste Operation des zu wiederholenden Teils zurückgeführt (eine schließende Entscheidung dient der Zusammenführung und hat mehrere eingehende und nur eine ausgehende Verbindung). Der andere ausgehende Zweig der öffnenden Verzweigung führt den Prozess nach der wiederholten Abarbeitung weiter. Die Anforderung, die Entscheidungsgrundlage explizit vor dem Gateway zu schaffen, ist hier durch die zusätzliche Aufgabe abgebildet, das Vorhandensein weiterer Anträge zu prüfen.

Das parallele Gateway kann zum Einsatz kommen, wenn zwei Aufgaben-Sequenzen unabhängig voneinander ausgeführt werden können. Der im obigen Beispiel (siehe Abb. 3.26) modellierte Prozess ist also nur korrekt, wenn die Beilagen tatsächlich unabhängig vom Antrag geprüft werden können. Ist dies nicht der Fall, müssten die beiden Funktionen sequenziell angeordnet werden. Beim zusammenführenden Gateway wird auf den Abschluss beider eingehenden Zweige gewartet, bevor der Prozess fortgesetzt wird.

Das obige Beispiel (siehe Abb. 3.27) zeigt den Einsatz von Pools, Lanes und Datenobjekten. Mittels Lanes werden die Verantwortlichkeiten im Prozess abgebildet. Mit den bislang eingeführten Elementen der BPMN können wir Kommunikationsvorgänge nicht abbilden. Die in Aktivitätsdiagrammen verfügbaren Signale können deshalb vorerst nicht

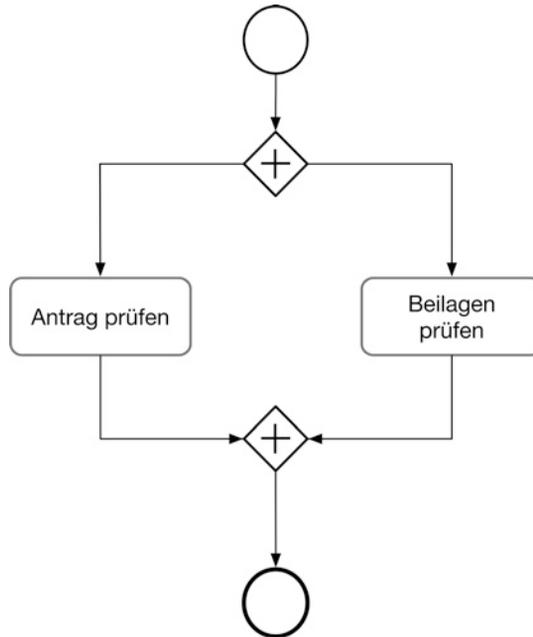


Abb. 3.26 BPMN-Modell mit parallel ablaufenden Zweigen

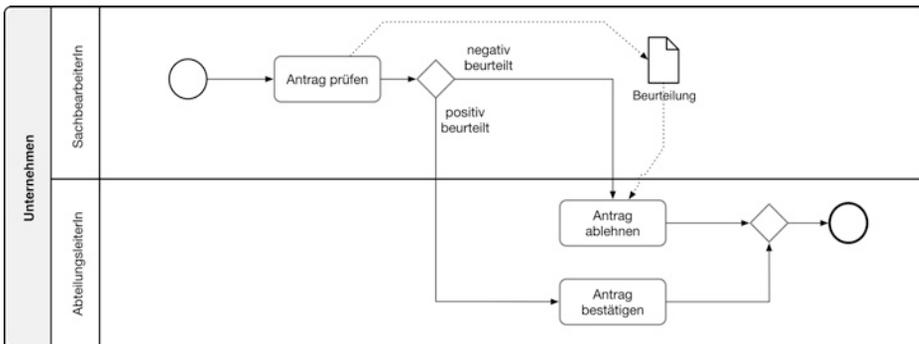


Abb. 3.27 BPMN-Modell mit Pools, Lanes und Datenobjekten

abgebildet werden, weswegen das Beispiel hier erneut unspezifischer wird. Der Einsatz von Nachrichten-Ereignissen, die wir im übernächsten Abschnitt einführen werden, wird diesen Mangel aber wieder beheben. Die Beurteilung des Antrags wird nur im Falle einer negativen Beurteilung als Datenobjekt zur Aufgabe „Antrag ablehnen“ übergeben – wir können also annehmen, dass die Ablehnung eine inhaltliche Begründung enthält. Im Fall der Bestätigung des Antrags, wird das Datenobjekt „Beurteilung“ nicht mehr benötigt – wir können also annehmen, dass in diesem Fall keine weitere Begründung erfolgt.

3.5.3 Notationselemente zur Ablaufsteuerung mit Ereignissen

Ein Unterscheidungsmerkmal von BPMN zu allen bislang behandelten Sprachen ist das hier sehr detailliert und umfassend ausgeführte Ereignis-Konzepte, das eine umfassende Kontrolle des Prozessablaufs ermöglicht.

Ereignisse geben an, dass etwas geschehen ist und sind damit Zeitpunkte im Gegensatz zu Aufgaben, die einen gewissen Zeitraum in Anspruch nehmen. Bis jetzt haben wir nur Start- und Endereignisse eingeführt, im Folgenden werden Start-, Zwischen- und Endereignisse noch einmal ausführlich beschrieben (siehe Abb. 3.28).

Ereignisse werden immer mit einem Kreis und üblicherweise einem Symbol dargestellt. Einfache Kreise kennzeichnen Startereignisse, doppelte Kreislinien Zwischenereignisse und dicke Kreislinien Endereignisse. Ist kein Symbol angegeben, handelt es sich um ein untypisiertes (Blanko-)Ereignis, das in der Regel nur am Beginn oder Ende eines Prozesses oder als auslösendes Zwischenereignis zu finden ist.

3.5.3.1 Startereignisse

Startereignisse geben den Auslöser für einen Prozess oder Teilprozess an. Häufig wird das unbestimmte (Blanko-)Ereignis verwendet, wenn sich der Auslöser aus dem Zusammenhang ohnehin ergibt oder wenn der Auslöser nicht näher bekannt ist.

Wird ein Prozess aufgrund eines Zeitpunkts oder einer Zeitspanne oder eines periodisch stattfindenden Ereignisses ausgelöst, wird zusätzlich das Symbol der Uhr verwendet. Ein Brief-Symbol wird verwendet, wenn eine Nachricht den Prozess auslöst (siehe Abb. 3.29).

Darüber hinaus gibt es noch Symbole für Bedingungen – der Prozess wird nur ausgeführt, wenn die angegebene Bedingung erfüllt ist. Ein Signal ist ein Zeichen, durch das der Prozess gestartet wird. Das Fünfeck als Symbol kennzeichnet mehrere mögliche Startereignisse, wobei nur eines der Ereignisse eintreten muss, um den Prozess zu starten (siehe Abb. 3.30).

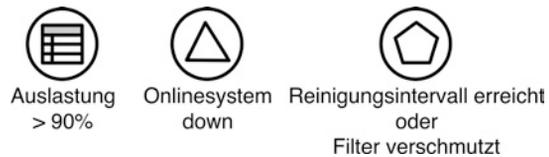
Ein Prozess muss kein einzelnes Startereignis haben, Prozesse können auch mehrere alternative Startereignisse haben.

Abb. 3.28 Beispiele für Start-, Zwischen- und Endereignisse



Abb. 3.29 Grundlegende BPMN-Startereignisse



Abb. 3.30 Komplexe BPMN-Startereignisse**Abb. 3.31** Grundlegende BPMN-Endereignisse

3.5.3.2 Endereignisse

Mit einem Endereignis werden Prozesse beendet, wobei es mit Ausnahme des zeitbezogenen Symbols die Bedingung und den parallelen mehrfachen Auslöser die gleichen Symbole gibt wie bei Startereignissen.

Zusätzlich zu den verschiedenen Arten von Endereignissen gibt es ein Terminierungs-Endereignis (schwarz gefüllter Kreis mit dicker schwarzer Umrandung), das den gesamten Prozess sofort beendet, d. h. die gesamte Prozessinstanz beendet, unabhängig davon, ob andere Sequenzen innerhalb des Prozesses zur gleichen Zeit noch durchlaufen werden oder nicht. Ein Standard-Endereignis beendet immer nur jenen Prozesszweig, in dem es eingefügt ist. Eventuell weitere, noch laufende Prozesszweige werden weiter ausgeführt (siehe Abb. 3.31).

Prozesse können, wie bereits bei den Startereignissen erklärt, mehrere Endereignisse haben. Ein Prozess ohne Endereignis ist unvollständig.

3.5.3.3 Zwischenereignisse & das ereignisbasierte Gateway

Zwischenereignisse können an irgendeiner Stelle in einem Prozess verwendet werden und werden durch einen Kreis mit doppelter Umrandung dargestellt. Sie werden modelliert, wenn in einem Prozess ein für andere (Prozesse) relevantes Zwischenergebnis erreicht wird, oder innerhalb eines Prozesses auf ein Ereignis reagiert wird, etwa auf eine eingehende Nachricht oder den Ablauf eines bestimmten Zeitraums.

Gateways können auch ereignisbasiert sein, wenn ein oder mehrere Ereignisse zum Durchlaufen von verschiedenen Pfaden führen. Dabei kann die Modellierung mit einem ereignisbasierten Gateway erfolgen.

Die folgende Abbildung (siehe Abb. 3.32) zeigt die Prozessmodellierung eines Bewerbungsvorgangs mit Ereignissen. Abhängig davon, ob eine Einladung oder eine Absage eintrifft oder eine Deadline von 2 Wochen abläuft, werden unterschiedliche Wege im Prozess beschritten. Das ereignisbasierte Gateway ist damit das einzige Gateway, bei dem zum Zeitpunkt der Prüfung des Gateways die dafür notwendigen Daten noch nicht vorliegen müssen. Ein ereignisbasiertes Gateway blockiert den Kontrollfluss so lange, bis eines der unmittelbar nachgelagerten Ereignisse eintritt und verfolgt dann den jeweiligen Zweig weiter.

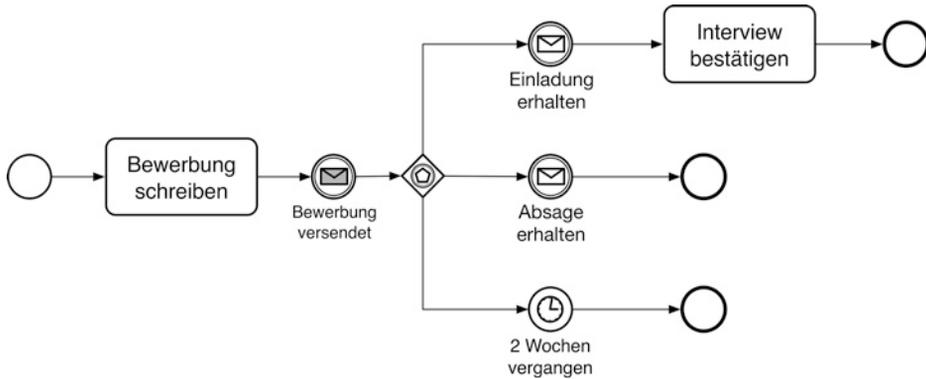


Abb. 3.32 Beispiel für den Einsatz des ereignisgesteuerten Gateways

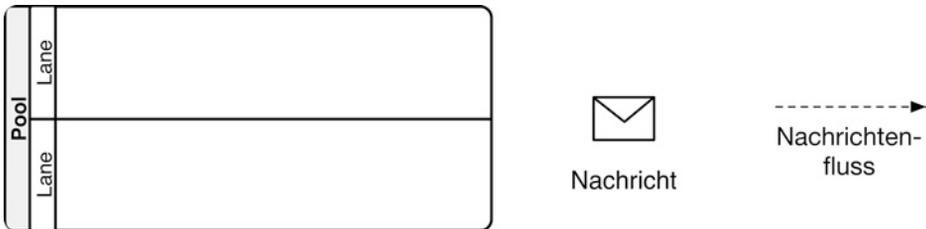


Abb. 3.33 Notationselemente zur Abbildung von Kommunikation in BPMN

3.5.4 Notationselemente zur Modellierung von Kommunikation

Die BPMN unterstützt auch die Modellierung verteilter Geschäftsprozesse. Die BPMN rückt zwar klar den Prozess bei der Modellierung in den Fokus (wie etwa auch ein Flowchart, eine EPK oder ein Aktivitätsdiagramm), aber auch den ausführenden bzw. beteiligten Personen eines Prozesses wird Beachtung geschenkt (siehe Abb. 3.33). Die dafür zur Verfügung stehenden Modellierungselemente werden in diesem Abschnitt betrachtet.

Ein *Pool* stellt eine Unternehmung oder eine Organisationseinheit in einer Unternehmung dar, wie z. B. eine Abteilung. Jede (Swim-)Lane in einem Pool repräsentiert eine prozessbeteiligte Person bzw. Rolle, die diesem Pool zugeordnet ist.

BPMN bietet die Möglichkeit, das Zusammenspiel von zwei oder mehreren Prozessen aufzuzeichnen. Zur Darstellung von Kollaborationen sind die eben erwähnten Pools und Lanes notwendig, wobei für alle an einem Prozess beteiligten Personen oder Gruppen eine eigene Lane erforderlich ist, für jeden Prozess bzw. jede Organisationseinheit, die für diesen Prozess verantwortlich ist, ein eigener Pool. In jedem Pool entstehen so eigene, individuelle Prozesse mit separaten Start- und Endereignissen. Dennoch kann es

vorkommen, dass die einzelnen Prozesse von anderen Prozessen stark beeinflusst werden. Dies kann durch Nachrichtenflüsse modelliert werden.

Nachrichtenflüsse geben an, wenn zwischen verschiedenen Prozessen Daten ausgetauscht werden. Innerhalb eines Prozesses (Pool) kann daher kein Nachrichtenfluss stattfinden. Somit ist kein Nachrichtenfluss innerhalb einer Lane und zwischen einzelnen Lanes möglich. Sequenzflüsse zeigen an, welche Aktivitäten in welcher Reihenfolge ausgeführt werden. Sie dürfen im Gegensatz zu Nachrichtenflüssen nur innerhalb eines Pool stattfinden können und nicht zwischen unterschiedlichen Prozessen (Pools).

An Nachrichtenflüsse können zusätzlich Nachrichten angebracht werden, die hier als Datenelemente eingesetzt werden und eine genauere Spezifikation der übermittelten Information enthalten.

Nachrichtenflüsse dürfen einerseits von Pools und Aktivitäten ausgehen und dort enden, andererseits können sie explizit von Sende-Ereignissen abgeschickt und durch Empfangs-Ereignisse empfangen werden. Der erstgenannte Fall ist für die beschreibende Modellierung von Geschäftsprozessen sinnvoll, bei der ein Kommunikationsvorgang zwar erfasst werden soll, aber nicht notwendigerweise exakt beschrieben werden muss. Eine von einer Aktivität ausgehende Nachricht wird irgendwann im Zuge der Aufgabenausführung gesendet – der genaue Zeitpunkt bleibt unklar. Eine an einem Pool endende Nachricht sagt nur aus, dass die repräsentierte Organisation diese Nachricht empfängt, aber nicht, welche Aktivität sie auslöst. Dies kann sinnvoll sein, wenn externe Organisationseinheiten modelliert werden sollen, deren detailliertes Verhalten unbekannt ist. Nur durch die Verwendung von expliziten Sende- und Empfangsereignissen ist eine exakte Spezifikation von Kommunikationsvorgängen möglich.

3.5.5 Beispiele zur Modellierung von Kommunikation

Im Folgenden ergänzen wir das Beispiel, das wir zur Demonstration des Einsatzes von Ereignissen verwendet haben, um den dort nicht abgebildeten Kommunikationspartner, also das Unternehmen, an das eine Bewerbung gerichtet wird.

Das obenstehende Beispiel (siehe Abb. 3.34) zeigt zwei Prozesse (je einen pro Pool), die über Nachrichtenflüsse miteinander verknüpft sind. Der Prozess des Unternehmens wird durch eine eingehende Bewerbung gestartet, die hier im ersten Nachrichtenfluss abgebildet ist. Nach der Prüfung kann die Entscheidung getroffen werden, ob eine Einladung ausgesprochen oder eine Absage erteilt werden soll. Im oberen Pool wartet der Bewerber auf eine Antwort – allerdings maximal 2 Wochen lang. Durch das ereignisbasierte Gateway wird jener Prozesszweig aktiviert, dessen Ereignis zuerst eintritt. Die jeweils zusammengehörigen Sende- und Empfangs-Ereignisse sind über einen Nachrichtenfluss gekoppelt. Wichtig ist hier, dass Nachrichtenflüsse immer 1:1-Beziehungen abbilden – dass also eine gesendete Nachricht genau einmal empfangen werden kann und dass ein empfangendes Ereignis auf genau eine Nachricht reagieren kann.

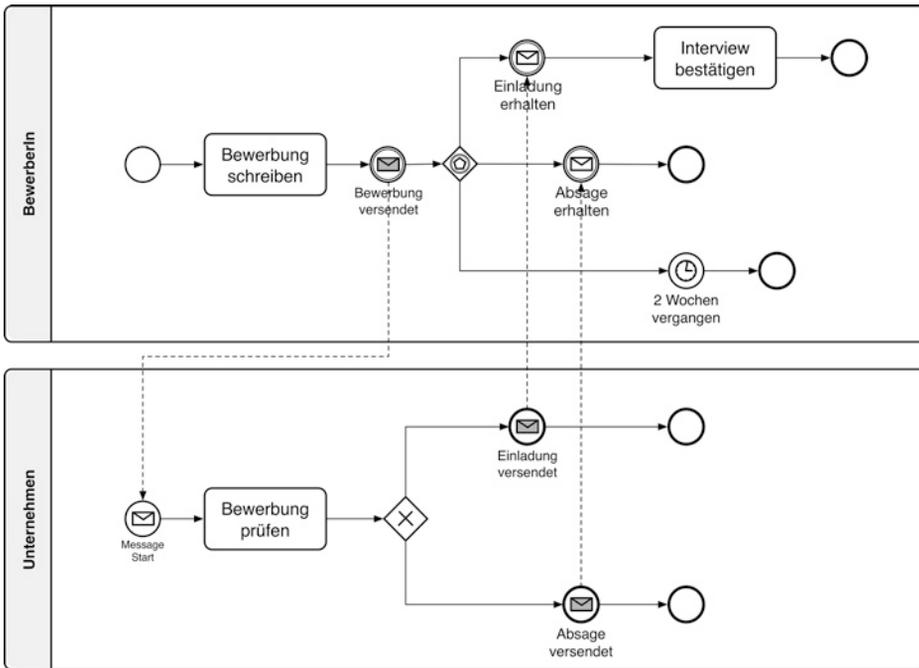


Abb. 3.34 Beispiel von verteilt ablaufenden Prozessen mit Kommunikationsvorgängen

Das obenstehende Beispiel (siehe Abb. 3.35) zeigt nun, wie die BPMN verwendet werden kann, um ausschließlich Kommunikationsvorgänge abzubilden. Die Pools werden hier als Blackbox verwendet, das in ihnen enthaltene Verhalten ist nicht abgebildet und bleibt unbekannt. Wir sehen lediglich, dass Nachrichten ausgetauscht werden, wobei die Reihenfolge der Nachrichten nicht definiert ist. Nachdem hier die näher beschreibenden Ereignisse zum Versenden und Empfangen fehlen, ergänzen wir das Modell um Nachrichten-Elemente, um die Kommunikation nachvollziehen zu können. Ein weiterer Unterschied ist der im oberen Pool angebrachte Modifikator, der eine parallele Mehrfachausführung des im oberen Pool enthaltenen Prozesses kennzeichnet. Dies bedeutet, dass der Prozess im unteren Pool mit mehreren, parallel unabhängig voneinander eintreffenden Bewerbungen umgehen können würde bzw. müsste.

Leere und gefüllte Pools können auch beliebig kombiniert werden. Wollten wir z. B. den Prozess der Bearbeitung einer eingehenden Bewerbung abbilden, könnten wir den Pool „BewerberIn“ unspezifisch belassen, da wir deren Verhalten nicht kennen (und es auch nicht relevant ist), aber wissen müssen, dass wir von dort eine Bewerbung erhalten können und dass wir unsere Antwort dann wieder dorthin adressieren.

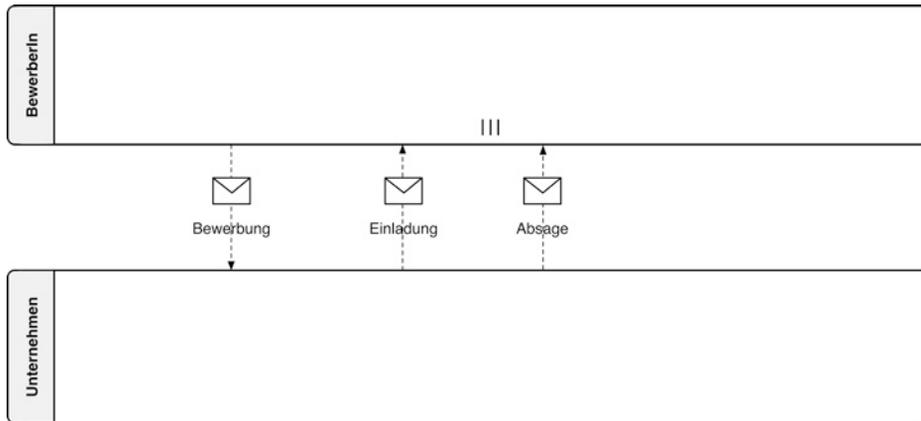


Abb. 3.35 Ausschließliche Abbildung von Kommunikationsvorgängen in BPMN

3.5.6 Notationselemente zur Modellierung komplexer Sachverhalte

Die BPMN bietet mithilfe der bislang vorgestellten Notationselemente die Möglichkeit, Geschäftsprozesse aus Sicht der durchführenden organisationalen Einheiten darzustellen. Die BPMN lässt dabei die Freiheit, Prozessmodelle vage zu halten oder Teile davon unspezifiziert zu lassen, wenn diese für die Zielsetzung der Modellierung nicht relevant erscheinen. In manchen Fällen soll aber ein Prozess möglichst exakt gefasst und in all seinen vorstellbaren Varianten und möglicherweise auftretenden Ausnahmefällen abgebildet werden. Dies ist etwa notwendig, wenn das Modell die Grundlage für die IT-basierte Unterstützung der abgebildeten Arbeitsprozesse dienen soll. Werden hier Aspekte weggelassen oder verkürzt abgebildet, ergibt sich eine Diskrepanz zwischen dem realen Arbeitsprozess und den auf dem Modell basierenden Unterstützungsmaßnahmen, was letztendlich zu nicht zufrieden stellenden Werkzeugen und Workarounds führt. Dieser Abschnitt beschreibt jene Notationselemente der BPMN, die komplexe Ablaufbeschreibungen ermöglichen. Aufgrund der Vielfalt der abbildbaren Szenarien führen wir Beispiele hier direkt bei der Beschreibung der jeweiligen Elemente an.

3.5.6.1 Varianten der Aktivitätsmodellierung

Im folgenden Abschnitt werden Besonderheiten der Modellierung von Aktivitäten sowie deren Modellierung als Subprozesse genauer erklärt.

3.5.6.1.1 Subprozesse

Prozesse können mittels Subprozessen modelliert werden. Diese Methode wird meistens verwendet, um bei der Modellierung von großen, umfangreichen Prozessen den Überblick behalten zu können. Subprozesse können dabei diagrammatisch minimiert werden. Dann ist der Prozess nur mehr als kleine Aufgabe im gesamten Prozess zu erkennen und durch ein kleines Plus gekennzeichnet.

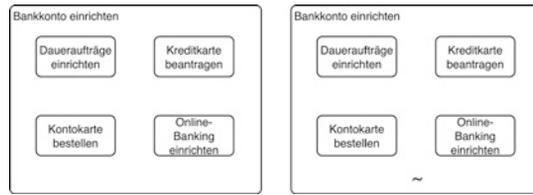


Abb. 3.36 Beispiele für Sub-Prozesse mit parallel (links) oder ad-hoc (rechts) auszuführenden Aktivitäten

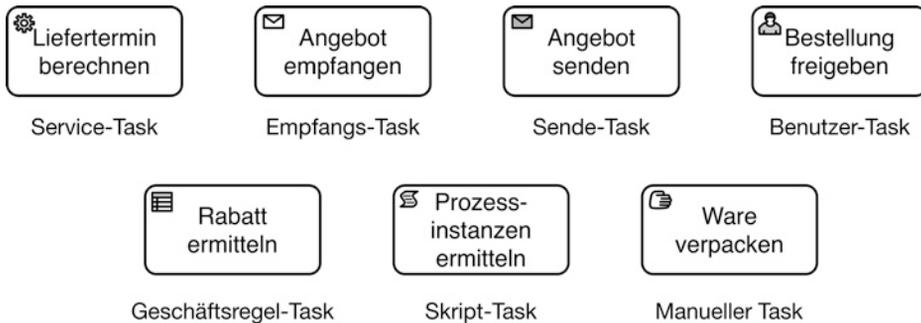


Abb. 3.37 Unterschiedliche Aktivitätstypen

Subprozesse erlauben aber auch die Zusammenfassung mehrerer Aktivitäten in einen Durchführungskontext, ohne deren genaue Reihenfolge festzulegen. Das rechte Modell (ein Ad-hoc Unterprozess) in der nachfolgenden Abbildung (siehe Abb. 3.36) gibt nur an, dass beliebig viele der eingebetteten Aktivitäten durchgeführt werden können, macht aber keine Aussage über deren Zusammenhänge. Das linke Modell legt fest, dass alle vier Subaktivitäten ausgeführt werden müssen, bevor der Subprozess abgeschlossen ist. Es trifft keine Aussage über die Reihenfolge oder sonstige Zusammenhänge – die Aktivitäten können parallel ausgeführt werden.

3.5.6.1.2 Aktivitätstypen

Aufgaben-Typen beschreiben den Charakter einer Aufgabe, also welchen Zweck eine Aufgabe erfüllt. Dabei wird unterschieden nach Service, Empfang, Sendung, Benutzer, Geschäftsregel, Skript, manuell und unbestimmt (Abbildung von links nach rechts; siehe Abb. 3.37). Diese Modifikatoren müssen nicht unbedingt verwendet werden, konkretisieren aber die Semantik eines Prozessmodells.

3.5.6.1.3 Ausführungsverhalten von Aktivitäten

Aktivitäten können unterschiedliche Markierungen haben, die das Ausführungsverhalten von Aktivitäten beschreiben. Aktivitäten oder gesamte Prozesse oder Teilprozesse können dabei als Schleife, parallel, sequentiell, ad hoc oder als Kompensation ausgeführt werden (siehe Abb. 3.38).

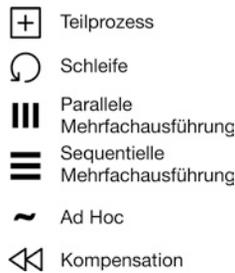


Abb. 3.38 Modifikatoren zur Abbildung unterschiedlichen Ausführungsverhaltens

Bei einer Schleife kann eine Abbruchbedingung zusätzlich zum Symbol angegeben werden. Ist die Abbruchbedingung erreicht, wird die Aktivität oder der betreffende Prozess nicht mehr länger ausgeführt und die jeweilige übergeordnete Sequenz weiter ausgeführt. Kann ein Prozess parallel ausgeführt werden, so wird dies durch drei vertikale Striche gekennzeichnet. So kann beispielsweise der Prozess „Bewerbung prüfen“ für mehrere eingelangte Bewerbungsunterlagen von mehreren Bearbeitern durchgeführt werden. Ist nur eine parallele Ausführung nicht möglich, die einzelnen Fälle aber unabhängig voneinander, so handelt es sich um eine sequenzielle Mehrfachausführung, die durch drei horizontale Striche gekennzeichnet wird.

Bei Ad-Hoc-Prozessen ist die genaue Reihenfolge der enthaltenen Aktivitäten unbekannt und ergibt sich erst während der Ausführung des Prozesses. Solche Prozesse werden über eine Tilde gekennzeichnet (wie oben bereits dargestellt). Es müssen auch nicht alle Aktivitäten abgearbeitet werden.

Kompensationsaktivitäten werden im Rahmen der Transaktionsmodellierung verwendet und werden weiter unten beschrieben.

3.5.6.2 Ereignistypen

Die BPMN bietet zur detaillierten Ablaufsteuerung eine Vielzahl von unterschiedlichen Events in verschiedenen Ausprägungen an. Während wir diese im Folgenden im Detail behandeln werden, soll hier ein Überblick gegeben und der systematische Aufbau von Events dargestellt werden (siehe Abb. 3.39).

Grundsätzlich können Events in drei unterschiedlichen Varianten vorkommen, die wir oben bereits eingeführt haben:

- *Start-Ereignisse* lösen neue Prozessinstanzen aus, starten also die Ausführung eines Prozesses. Sie sind immer „empfangend“, reagieren also auf Stimuli von außen (etwa eingehende Nachrichten, Zeitabläufe etc.).
- *End-Ereignisse* sind Ereignisse, die beim Beenden einer Prozessinstanz ausgelöst werden. Sie sind immer „sendend“, lösen also potenziell Stimuli für andere Prozesse bzw. Prozessteile aus.
- *Zwischen-Ereignisse* treten innerhalb des Sequenzflusses auf, haben also sowohl eine eingehende als auch eine ausgehende Kante (Ausnahme: Link-Event, siehe weiter unten).

	Start			Zwischen				Ende	
	Prozess-Ebene	Sub-Prozess unterbrechend	Sub-Prozess nicht-unterbrechend	Eingetreten	Angehftet unterbrechend	Angehftet nicht-unterbrechend	Ausgelöst		
Blanko									unspezifisches Ereignis
Nachricht									Kommunikationsereignis mit 1 Sender und 1 Empfänger
Timer									Periodische Ereignisse oder Zeitpunkte
Eskalation									Meldung an höhere Verantwortungsebene
Bedingung									Ereignis, dessen Eintritt durch Regeln beschrieben ist
Link									kein echtes Ereignis, nur zur Verbindung entfernter Prozesssteile
Fehler									wird verwendet, wenn in einem Subprozess ein Fehler auftreten kann, der behandelt werden muss
Abbruch									wird verwendet, wenn in einer Transaktion ein Fehler auftreten kann, der behandelt werden muss
Kompensation									wird verwendet, um das Rollback von bereits durchgeführten Tasks in einer Transaktion auszulösen
Signal									Kommunikationsereignis mit 1 Sender und mehreren Empfängern (prozessweit sichtbar)
Mehrfach									Ereignis, dass aus mehreren Einzelereignissen besteht, von denen eines eintreten muss
Mehrfach (parallel)									Ereignis, dass aus mehreren Einzelereignissen besteht, von denen alle eintreten müssen
Terminierung									Abbruch des gesamten Prozesses inkl. aller noch aktiven Abläufe etwa in verzweigten Sequenzflüssen

Abb. 3.39 Vollständiger Satz an BPMN-Ereignis-Typen

Start-Ereignisse lassen sich darüber hinaus hinsichtlich ihres Einsatzortes unterscheiden. Sie können zum Auslösen eines gesamten Prozesses oder zum Auslösen von Subprozessen verwendet werden. Der zweite Fall wird als „Ereignis-Teilprozess“ bezeichnet und kann mittels „unterbrechend“ und „nicht-unterbrechend“ spezifiziert werden. Im eher üblichen Fall wird durch ein „unterbrechendes“ Starterereignis gekennzeichnet, dass der Kontrollfluss vollständig an den Subprozess übergeben wird, innerhalb des jeweiligen Pool also alle anderen Aktivitäten unterbrochen werden. „Nicht-unterbrechende“ Ereignis-Teilprozesse werden beim Eintreffen des jeweiligen Ereignisses gestartet, ohne dass die Durchführung der aktuell innerhalb eines Pool ausgeführten Aktivitäten unterbrochen wird. Damit kann etwa auf Ereignisse reagiert werden, die nicht im Hauptprozess eines Pool behandelt werden sollen oder können, deren Auftreten aber eine Reaktion nach sich ziehen soll, ohne dass der Hauptprozess beeinträchtigt wird (etwa Kundennachfragen über die Status einer Auftragsbearbeitung, während dieser Auftrag gerade im Rahmen des Hauptprozesses bearbeitet wird).

Zwischenereignisse existieren grundsätzlich in einer „empfangenden“ und einer „sendenden“ Variante. Die „empfangende“ Variante (eingetretenes Ereignis) wartet auf das Eintreffen des spezifizierten Ereignisses und blockiert dabei den Sequenzfluss. Dieser kann also nicht fortgesetzt werden, bis das Ereignis eingetreten ist. Die „sendende“ Variante (ausgelöstes Ereignis) kennzeichnet das Eintreten bestimmter Ereignisse in einem Prozess (oder auch zwischen Prozessen aus unterschiedlichen Pools). Ereignisse treten bei vollständig modellierten Prozessen üblicherweise reziprok auf, d. h. dass zu einem eingetretenen Event auch ein auslösendes Event existiert.

Empfangende Zwischenereignisse existieren auch in einer „angehefteten“ Form. Diese Ereignisse werden an Aktivitäten „angeheftet“ (d. h. grafisch an deren (Unter-)Kante angebracht) und kennzeichnen, dass während der Durchführung der Aktivität auf das jeweilige Ereignis reagiert werden können soll. Die Reaktion wird durch einen aus dem angehefteten Event ausgehenden Sequenzfluss spezifiziert, der zu den jeweils durchzuführenden Aktivitäten führt. Das angeheftete Zwischenereignis existiert dabei wiederum in einer unterbrechenden und einer nicht-unterbrechenden Form. Die unterbrechende Form bricht die Ausführung der so gekennzeichneten Aktivität ab und setzt den Sequenzfluss ausschließlich über das angeheftete Event fort. Die nicht-unterbrechende Form erlaubt die weitere Ausführung der so gekennzeichneten Aktivität, parallel dazu wird aber der vom Event ausgehende Sequenzfluss ausgelöst.

Die auslösenden Ereignisse, die zum Eintreten von angehefteten Events führen, können von außen kommen (etwa von anderen Pools eintreffende Nachrichten) oder auch von innerhalb der Aktivität, sofern diese durch einen Subprozess detailliert wird. So kann ein Fehler bei der Ausführung des Subprozesses zu Aktivitäten im Hauptprozess führen, etwa das Dokumentieren des Fehlers und einer Eskalation zu Vorgesetzten.

Nachrichten- und Timer-Ereignisse haben wir bereits in der grundlegenden BPMN-Modellierung verwendet. Die übrigen Ereignistypen betrachten wir nun im Rahmen ihrer jeweiligen Einsatzgebiete.

3.5.6.3 Das Link-Event

Bei komplexeren oder umfangreicheren Abläufen wird die Nachverfolgung von Sequenzflüssen durch das Diagramm manchmal schwierig. Einander kreuzende Sequenzflüsse oder Sequenzflüsse mit vielen Richtungsänderungen erschweren die Lesbarkeit und sind der Verständlichkeit und Übersichtlichkeit abträglich (siehe Abb. 3.40).

In derartigen Fällen kann das Link-Event verwendet werden. Im Gegensatz zu den übrigen Events bildet es semantisch kein echtes Ereignis ab, sondern dient lediglich als Konnektor zwischen zwei Sequenzflüssen, die weit voneinander entfernt liegen. Die Kopplung erfolgt dabei über die Bezeichnung des auslösenden und des empfangenden Events. Es muss immer eine 1:1-Zuordnung bestehen – eine implizite Parallelisierung durch ein auslösendes und mehrere gleichnamige empfangende Link-Events sind folglich nicht zulässig.

Link-Events sollten zur Erhöhung der Übersichtlichkeit trotzdem nur in nicht anders auflösbaren Fällen genutzt werden, da das Suchen zusammengehöriger Link-Events im

Abb. 3.40 Beispiel für den Einsatz des Link-Events



Aufwand das Nachverfolgen komplexer Sequenzflüsse sogar übersteigen kann (sofern keine Werkzeugunterstützung durch Sprungfunktionalitäten oder optische Markierung zusammengehöriger Events besteht). Eine alternative Anordnung von Aktivitäten oder Lanes ist üblicherweise die bessere Wahl.

3.5.6.4 Verwendung von Signalen

Nachrichten können in BPMN ausschließlich zur Kommunikation zwischen Pools verwendet werden. Des Weiteren hat eine nachrichtenbasierte Kommunikation immer exakt zwei Endpunkte, kann also immer nur genau einen Sender mit genau einem Empfänger verbinden. Wenn eine Information global innerhalb einer Kollaboration zur Verfügung gestellt werden soll, und dies unabhängig von Poolgrenzen passieren soll, können Signale verwendet werden. Signale können (als Zwischen- oder End-Ereignisse) im Prozess ausgelöst werden und stehen dann sowohl innerhalb des Pool als auch in allen anderen Pools der gleichen Kollaboration zur Verfügung.

Signale können dazu eingesetzt werden, alle Pools einer Kollaboration etwa über den Abbruch eines der abgebildeten Prozesse zu informieren. So können alle anderen noch ausgeführten Prozesse für sich ihre Prozesse zum Abschluss bringen und es bleiben keine „Prozessleichen“ zurück, die nicht mehr abgeschlossen werden können, weil etwa aufgrund eines Prozessabbruchs eines Kommunikationspartners eine erwartete eingehende Nachricht nicht mehr ankommt.

3.5.6.5 Behandlung von Ausnahmen und Unterbrechungen

Aktivitäten, also Aufgaben und Prozesse, können durch bestimmte Ereignisse abgebrochen oder unterbrochen werden. Dies wird durch Ereignissymbole gekennzeichnet, die der jeweiligen Aktivität angeheftet sind. Wird durch das eintretende Ereignis die Aktivität abgebrochen, so wird dies durch zwei durchgehende äußere Kreislinien gekennzeichnet. Wird die Aktivität nicht unterbrochen, so wird dies durch zwei gestrichelte Kreislinien dargestellt. Im untenstehenden Beispiel reagieren wir auf das Eintreten eines Timer-Ablaufs. Wir können also modellieren, dass die Ausführung der Aufgabe nicht länger als eine bestimmte Zeitspanne dauern darf (links) oder dauern sollte (rechts), und dass bei Zeitablauf in irgendeiner Form reagiert werden muss. Die Reaktion ist in dem vom angehefteten Ereignis ausgehenden Sequenzfluss zu modellieren. Auf dieselbe Art und Weise können auch ungeplante Ereignisse wie Fehler oder Eskalationen modelliert und dargestellt werden (siehe Abb. 3.41).

3.5.6.5.1 Beispiel: Nicht-unterbrechende Timerevents

Die folgende Abbildung (siehe Abb. 3.42) stellt dar, dass der Teilprozess, der die Aktivitäten der Bearbeitung einer Bestellung in einem Fast Food Restaurant abbildet, nicht länger als 5 min dauern soll. Dauert die Bearbeitung der Bestellung länger als 5 min, soll

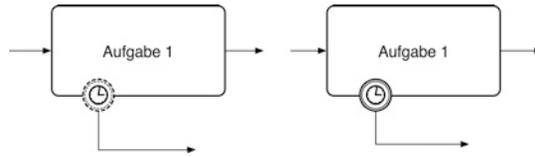


Abb. 3.41 Beispiel für angeheftete Ereignisse (links: nicht unterbrechend, rechts: unterbrechend)

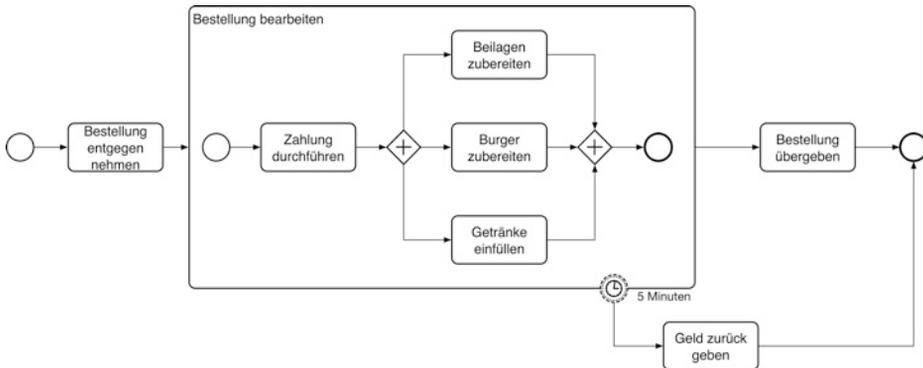


Abb. 3.42 Beispiel für die Verwendung von nicht-unterbrechenden Timer-Events

der Kunde das Geld zurückerhalten. Die Bestellung soll trotzdem noch fertig bearbeitet werden. Im Fall eines unterbrechenden Events würde der Kunde lediglich sein Geld zurückerhalten, nicht aber das bestellte Essen.

3.5.6.6 Terminierung von Prozessen

Sequenzflüsse in einem Prozess werden üblicherweise mit einem Endereignis abgeschlossen. Endereignisse beenden jedoch nur die Ausführung des jeweiligen Sequenzflusses. Falls parallel noch weitere Sequenzflüsse aktiv sind, etwa, weil sie durch eine paralleles oder inklusives Gateway geöffnet wurden oder weil sie durch angeheftete Events ausgelöst wurden, werden diese weiterhin ausgeführt. Um einen (Sub-)Prozess vollständig und sofort zu beenden (also die Ausführung in allen Zweigen zu beenden), existieren mehrere Möglichkeiten.

3.5.6.6.1 Das Terminate-Event

Das Terminate-Event bricht alle aktiven Zweige eines Prozesses innerhalb eines Pool unmittelbar und sofort ab. Prozesse in anderen Pools sind nicht betroffen und sollten deshalb ggf. vor dem Abbruch durch das Senden eines Signals vom Abbruch in Kenntnis gesetzt werden.

3.5.6.2 Das Error-Event

Das Error-Event zeigt das Auftreten eines semantischen Fehlers an und wird üblicherweise in Subprozessen eingesetzt. Es bricht die Ausführung des Subprozesses ab, der Fehlergrund kann dem Event als Parameter mitgegeben werden. Durch angeheftete empfangende Ereignisse kann auf diese Fehler im übergeordneten Prozess reagiert werden und können entsprechende Aktivitäten ausgelöst werden. Angeheftete Fehlerereignisse sind immer unterbrechend, brechen also die Ausführung des Subprozesses (inkl. aller gegebenenfalls noch aktiver Sequenzflüsse in Zweigen, in denen kein Fehler aufgetreten ist) ab. Als „schwächere“ Variante kann auch das „Eskalationsereignis“ in identischer Weise verwendet werden. Dieses existiert auch in einer nicht-unterbrechenden Form und erlaubt damit die Fortsetzung der Bearbeitung des Subprozesses, in dem das Problem aufgetreten ist.

Falls beim Abbruch von Subprozessen die Auswirkungen von bereits durchgeführten Aktivitäten rückgängig gemacht werden müssen, kann auf die im nächsten Abschnitt vorgestellte Transaktionsbehandlung wie in der BPMN vorgesehen zurückgegriffen werden.

3.5.6.7 Transaktionen

In BPMN existiert auch die Möglichkeit, Transaktionen in einem Prozess abzubilden. Von einer Transaktion spricht man, wenn eine Menge von Aufgaben eines Prozesses als Gesamtheit vollständig oder gar nicht ausgeführt werden sollen. Insbesondere sollen bei Scheitern einer Aufgabe bestimmte andere bereits abgeschlossenen Aufgaben wieder rückgängig gemacht werden. BPMN kennt das Konzept transaktionaler Subprozesse in Kombination mit Kompensierungsereignissen und -aktivitäten. Kompensierung bedeutet, dass bereits ausgeführte Prozessschritte dadurch zurückgerollt werden, indem konkrete Gegenmaßnahmen durch einen weiteren Prozessschritt eingeleitet werden.

In einem Transaktionssubprozess (in BPMN gekennzeichnet durch eine doppelte Umrandung) wird jeder Aufgabe über ein angeheftetes Kompensierungszwischenereignis (gekennzeichnet durch ein „Zurückspulen“-Symbol) eine Kompensierungsaufgabe zugeordnet. Bricht die Transaktion ab, so wird für jede Aufgabe, die bereits erfolgreich abgeschlossen wurde, die jeweilige Kompensierung ausgeführt. In diesem Zusammenhang kommt das Abbruch-Event (gekennzeichnet durch „X“) ins Spiel. Als auslösendes Endereignis in einem Transaktionssubprozess bewirkt es dessen Abbruch und damit das Auslösen der Kompensierungen. Wenn es an die Transaktion angeheftet wird, ist es ein empfangendes Zwischenereignis und bestimmt es den weiteren Ablauf des Prozesses nach dem Abbruch der Transaktion. Durch das Konzept der Kompensierung kann eine Transaktion auch dann noch zurückgerollt werden, nachdem sie eigentlich schon erfolgreich abgeschlossen wurde. Außerhalb der Transaktion kann ein auslösendes Kompensierungsereignis verwendet werden, um die Kompensierung der referenzierten Transaktion nachträglich auszulösen.

Die nachfolgende Abbildung (siehe Abb. 3.43) illustriert diese Konzepte anhand eines Reisebuchungsprozesses.

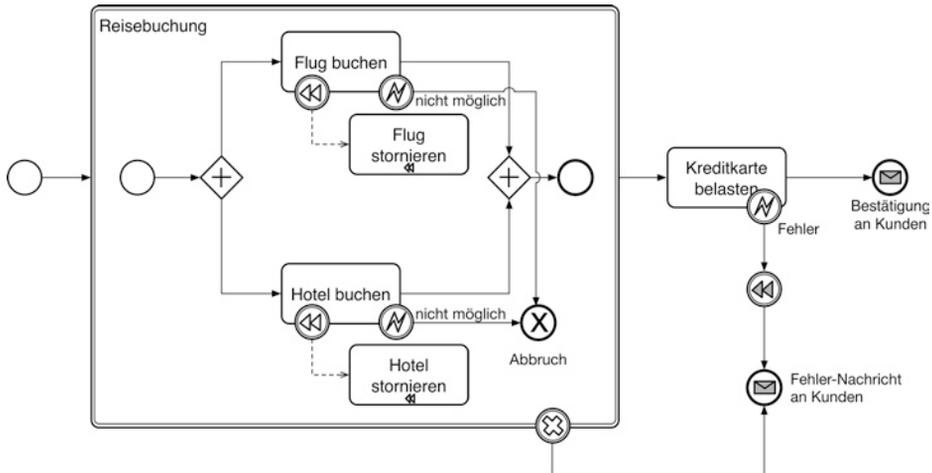


Abb. 3.43 Beispiel für einen Transaktions-Subprozess

Eine Reisebuchung besteht aus einer Flug- und einer Hotelbuchung, die potenziell parallel vorgenommen werden können. Ist eine der Buchungen nicht möglich, so muss die andere Buchung (falls sie schon ausgeführt wurde) wieder storniert werden. Ein Fehler bei einer einzelnen Buchung löst also den Abbruch der Transaktion aus und führt dazu, dass der Prozess mit einer Fehlernachricht an den Kunden reagiert. Außerhalb der eigentlichen Transaktion führt ein Fehler bei der Belastung der Kreditkarte zur Stornierung der gesamten Buchung.

3.5.6.8 Ereignisgesteuerte Sub-Prozesse

Ereignisgesteuerte Sub-Prozesse sind eine Alternative zur Behandlung von auftretenden Ereignissen in (Sub-)Prozessen. Während an Subprozessen angeheftete Events die Reaktion auf diese in den umgebenden Prozess ‚hinaustragen‘, kann diese durch den Einsatz von Ereignisgesteuerten Sub-Prozessen lokal gehalten werden.

Die folgende Abbildung (siehe Abb. 3.44) zeigt ein Beispiel für einen timer-gesteuerten nicht-unterbrechenden Subprozess. Es greift das oben bereits verwendete Szenario der Zubereitung einer Bestellung in einem Fast-Food-Restaurant wieder auf.

Die Events, mit denen ereignisgesteuerte Subprozesse ausgelöst werden können, sind identisch mit jenen, die an einen Subprozess angeheftet werden können. Beide Varianten können unterbrechend und nicht-unterbrechend sein. Semantisch unterscheiden sie sich wie erwähnt lediglich in der Art der Behandlung des Ereignisses – nämlich lokal innerhalb des Subprozesses oder außerhalb im umschließenden Prozess. Je nach Anwendungsfall kann die eine oder die andere Variante zu sinnvollen und/oder übersichtlichen Darstellungsformen führen.

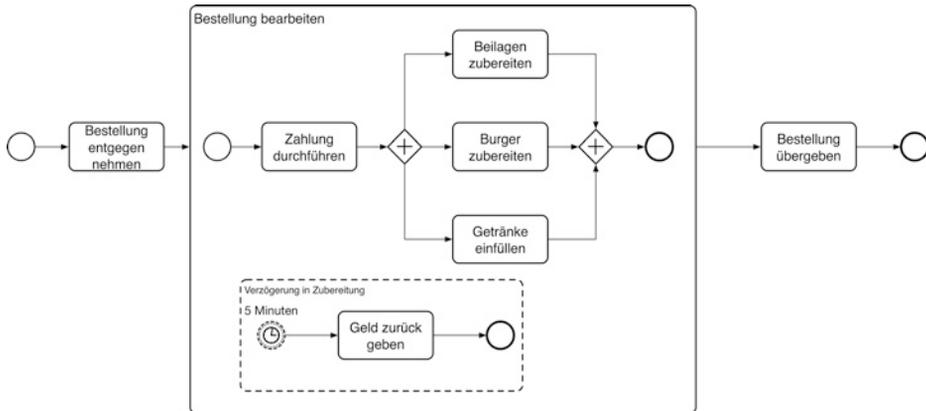


Abb. 3.44 Beispiel für den Einsatz eines nicht-unterbrechenden ereignisgesteuerten Sub-Prozesses

3.5.7 Choreografiediagramme

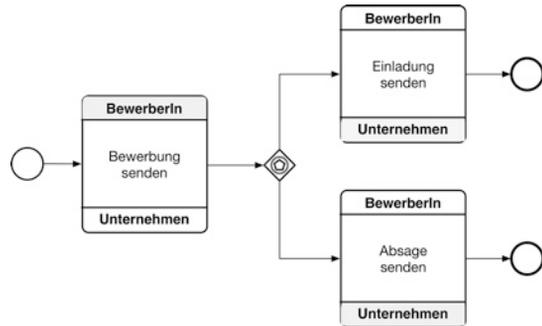
Die explizite Modellierung von Choreografien in Choreografie-Diagrammen wurde in der BPMN 2.0 neu eingeführt. Bei einer Choreografie handelt es sich um den Ablauf von Nachrichtenaustauschen zwischen unterschiedlichen Akteuren. Eine Choreografie ist damit eine andere Sicht auf eine Kollaboration, bei der die Reihenfolge der Nachrichtenübermittlung unabhängig von den Prozessen der einzelnen Akteure dargestellt wird.

Zwar kann man einer Darstellung der Kommunikation mittels zugeklappten, d. h. leeren, Pools entnehmen, welche Nachrichten zwischen welchen Partnern ausgetauscht werden, doch lassen sich die genaue Reihenfolge, bedingte Nachrichtenflüsse oder Schleifen nicht daraus ersehen. So ist im bislang verwendeten Bewerbungsbeispiel beispielsweise nicht gezeigt, dass nach dem Eintreffen eines Angebots beim Kunden entweder die Einladungs- oder die Absage-Nachricht gesendet wird. Dies lässt sich mit einem Choreografie-Diagramm visualisieren.

Die folgende Abbildung (siehe Abb. 3.45) enthält die Choreographie zu dem oben bereits als Kollaboration gezeigten Beispiel eines Bewerbungsprozesses. Hier sind die Nachrichtenaustauschvorgänge in den Mittelpunkt gerückt. Eine sogenannte Choreographie-Aktivität (Choreography Activity) repräsentiert den Austausch einer oder mehrerer Nachrichten zwischen zwei oder mehreren Partnern. Im einfachsten Fall entspricht sie dem Senden einer einzigen Nachricht von einem Partner zu einem anderen.

Jede Choreografie-Aktivität wird von einem der beteiligten Partner ausgelöst, indem er die erste Nachricht sendet. Dieser auslösende Partner wird am oberen oder unteren Rand der Choreografie-Aktivität in einem hellen Feld eingetragen. Die Namen des oder der weiteren Beteiligten werden am anderen Rand in einem dunkleren Feld eingetragen. Wer oben und wer unten eingetragen wird, ist dem Modellierer freigestellt. Normalerweise wird man bei mehreren Choreografie-Aktivitäten zwischen denselben Partnern die

Abb. 3.45 Beispiel für den Einsatz von Choreographie-Aktivitäten



Anordnung beibehalten. Wenn man zusätzlich noch eine Kollaboration modelliert, ist es naheliegend, die vertikale Anordnung der Pools zugrunde zu legen. Entsprechend sind in den Choreografie-Aktivitäten der Abbildung entweder der Kunde oben und die Werbeagentur unten oder aber die Werbeagentur oben und die Grafiker unten eingezeichnet.

Choreografie-Aktivitäten mit mehr als zwei Partnern kommen in diesem Beispiel nicht vor. Hierfür kann man oben bzw. unten mehrere Partner-Felder eintragen. Dabei ist aber immer nur ein Feld hell hinterlegt, da nur einer der Partner den Nachrichtenaustausch durch eine initiale Nachricht in Gang setzen kann.

Für die Choreografie-Aktivitäten ist im Choreografie-Diagramm ein Sequenzfluss definiert. Seine Modellierung entspricht im Wesentlichen der Sequenzfluss-Modellierung von Prozessen. Allerdings sind gewisse Elemente der Prozessmodellierung im Zusammenhang mit der Choreografie-Modellierung nicht sinnvoll und daher auch nicht zulässig. So gibt es z. B. keine Nachrichten- Ereignisse innerhalb eines Sequenzflusses, da der Nachrichtenaustausch per Definition Teil der Choreografie-Aktivitäten ist. Entsprechend folgen beispielsweise auf das ereignisbasierte Gateway in der Abbildung keine Ereignisse, sondern Choreografie-Aktivitäten. Hierbei wird der Pfad gewählt, dessen Choreografie-Aktivität als erste durch die jeweilige auslösende Nachricht gestartet wird.

Will man wissen, welche Nachrichten in jeder Choreografie-Aktivität ausgetauscht werden, so können diese wiederum in Form kleiner Briefsymbole hinzugefügt und mit dem jeweiligen Partnerfeld verbunden werden. Die Briefe sind ebenso wie die beteiligten Partner farblich gekennzeichnet. Ein helles Briefsymbol steht für die Nachricht, mit der eine Choreografie-Aktivität ausgelöst wird. Die Briefsymbole der anderen Nachrichten sind dunkler dargestellt.

3.5.8 Einordnung

Die BPMN hat sich in den letzten Jahren auch in der industriellen Praxis durchgesetzt. Durch ihren umfassenden Satz an Sprachelementen eignet sie sich für viele Anwendungsbereiche von der Dokumentation bis hin zur automationsgestützten Ausführung von Geschäftsprozessen in Organisationen. Der umfangreiche Sprachschatz

wird gleichzeitig aufgrund der gesteigerten Komplexität als Manko der BPMN gesehen. Insbesondere die Vielzahl an Ereignistypen mit zum Teil nur schwer zu unterscheidenden Bedeutungen führt zu gesteigertem Aufwand beim Erlernen der Sprache.

Der mit der möglichen Komplexität der entstehenden Modelle und der damit einhergehenden erschwerten Verständlichkeit wird üblicherweise dadurch begegnet, dass in geeigneten Anwendungsfällen ein reduzierter Sprachumfang verwendet wird. Zur deskriptiven Dokumentation von Geschäftsprozessen ist es üblicherweise nicht notwendig, den vollständigen Satz an Ereignissen und komplexeren Aufgabentypen zu verwenden. Erst wenn ein Prozessmodell etwa durch Simulation validiert oder zur Ausführung gebracht werden soll, müssen die Modelle um Information zu Ausnahmefällen o. Ä. angereichert werden. In diesen Fällen kann von den einfacheren Modellen ausgehend eine Detaillierung und Ergänzung vorgenommen werden.

Die BPMN ist eine der wenigen Sprachen zur Geschäftsprozessmodellierung, die explizit auf Kommunikationsvorgänge zwischen beteiligten Akteuren eingeht und deren Abbildung ermöglicht. Bei der Definition der Sprache war der Ausgangspunkt zur Spezifikation von Kommunikationsflüssen allerdings die Kopplung technisch getrennter Informationssysteme. Die BPMN geht implizit davon aus, dass es innerhalb eines Pool (also zwischen Lanes) nicht notwendig ist, sich um die Kommunikation zwischen Akteuren zu kümmern, weil alle Zugriff auf die gleiche Informationsinfrastruktur haben. Zwischen Pools werden Nachrichtenflüsse modelliert, die im Rahmen der Ausführung dazu dienen, die Abbildung der im Ausgangspool verwendeten Datenstrukturen auf jene des Ziel-Pool zu beschreiben.

Ein Nachrichtenfluss entspricht also im Wesentlichen semantisch einer Datenübergabe von einem Informationssystem zu einem anderen und ist dem entsprechend immer ein Kommunikationsvorgang mit genau einem Sender und genau einem Empfänger – mehrere Empfänger können beispielsweise nicht mit einer einzelnen Nachricht angesprochen werden. Während dieser Mechanismus auch zur Abbildung von nicht-technischer Kommunikation eingesetzt werden kann, ist seine Ausdrucksstärke jedoch beschränkt. Insbesondere kann etwa eine Kommunikation zwischen zwei oder mehreren Akteuren ohne klar abgrenzbare Nachrichten nur auf Umwegen und mehrdeutig modelliert werden. Diese Einschränkung ist jedoch dem Anspruch der Ausführbarkeit der erstellten Prozesse geschuldet und existiert in dieser Form auch in anderen kommunikationsorientierten Ansätzen.

Die BPMN fokussiert des Weiteren auf Prozessen mit einem vollständig spezifizierbaren Kontrollfluss. Dies stößt an Grenzen, sobald Prozessteile fallspezifisch und nicht vorab im Detail beschreibbar sind. Für derartige Prozesse haben sich in den letzten Jahren unterschiedliche Ansätze gebildet, die entweder auf eine deklarative Modellierung der Ausführungsbedingungen von Prozessteilen abzielen, oder die Kommunikationsvorgänge zwischen den beteiligten Akteuren in den Mittelpunkt stellen. Als Beispiel für letztere Kategorie betrachten wir im nächsten Abschnitt die subjektorientierte Geschäftsprozessmodellierung (S-BPM).

3.6 S-BPM

Ein subjektorientiertes Prozessmodell beschreibt im Gegensatz zu anderen Modellierungsansätzen Geschäftsvorgänge primär aus Sicht von miteinander kommunizierenden Handelnden oder Systemen. Es erfasst, welche Arbeitsschritte eines Geschäftsprozesses durch wen bzw. welches System mit welchen Hilfsmitteln ausgeführt werden, welches Ergebnis dadurch erzeugt wird und für wen dieses bestimmt ist.

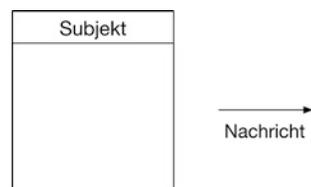
Bei der Modellierung nach dem subjektorientierten Ansatz stehen die Subjekte als Repräsentanten für an einem Prozess beteiligte Handelnde bzw. ausführende Systeme im Mittelpunkt der Betrachtung. Es wird im Wesentlichen beschrieben, wer in welcher Form miteinander kommuniziert und wie die einzelnen Akteure auf erhaltene Nachrichten reagieren. Die Beschreibung der Kommunikation erfolgt durch die Definition der Nachrichten, die zwischen den Subjekten ausgetauscht werden. Das Verhalten der Subjekte wird jeweils separat durch Zustandsdiagramme beschrieben, wobei drei unterschiedlichen Zustandstypen zum Einsatz kommen: Ein Subjekt kann auf eine Nachricht warten, eine Nachricht senden oder etwas tun, ohne mit anderen Subjekten zu kommunizieren. Der letztere Zustandstyp wird als Funktionszustand bezeichnet und wird verwendet, um das eigentliche Verhalten, also die Aktivitäten eines Subjekts zu beschreiben.

3.6.1 Notationselemente

Bei der Modellierung nach dem subjektorientierten Ansatz stehen die Subjekte als Repräsentanten für an einem Prozess beteiligte Handelnde bzw. ausführende Systeme im Mittelpunkt der Betrachtung. Die Modellierung eines Prozesses läuft im Wesentlichen in folgenden Schritten mit zunehmendem Detaillierungsgrad ab: zuerst wird das *Interaktionsdiagramm* erstellt, in dem die Subjekte und deren Nachrichtenaustausch modelliert werden (siehe Abb. 3.46). In einem weiteren Schritt wird das Verhalten jedes Subjektes in einem separaten *Verhaltensdiagramm* beschrieben.

Für das *Interaktionsdiagramm* werden zuerst die an einem Prozess beteiligten *Subjekte* festgelegt. Ein Subjekt ist eine aktive Einheit, muss aber nicht unbedingt ein menschlicher Akteur sein. Auch technische Systeme können Subjekte sein, solange sie eine aktive Rolle im Prozess einnehmen. Subjekte sind immer abstrakt zu beschreiben, d. h. nicht für konkrete Personen oder Maschinen, sondern auf Basis der notwendigen zu erfüllenden Aufgaben im Prozess festzulegen (also etwa „Antragsprüfer“, und nicht „Hr. Müller“).

Abb. 3.46 Notationselemente des S-BPM Interaktionsdiagramms



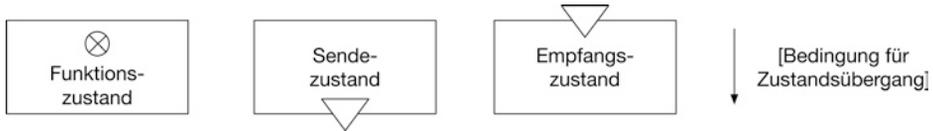


Abb. 3.47 Notationselemente des S-BPM Verhaltensdiagramms

Zwischen den Subjekten werden Nachrichten ausgetauscht. Im Interaktionsdiagramm wird nur festgelegt, welche Nachrichten existieren und wer jeweils Sender und Empfänger ist. Die Reihenfolge der Nachrichten wird hier noch nicht festgelegt (siehe Abb. 3.47).

Für jedes Subjekt wird im Verhaltensdiagramm beschrieben, in welcher Reihenfolge es Nachrichten sendet und empfängt bzw. interne Funktionen ausführt. Die einzelnen Zustände werden mit Verbindungen zueinander in Beziehung gesetzt, welche die Zustandsübergänge beschreiben. Deren Verwendung hängt vom jeweils eingesetzten Zustandstyp ab:

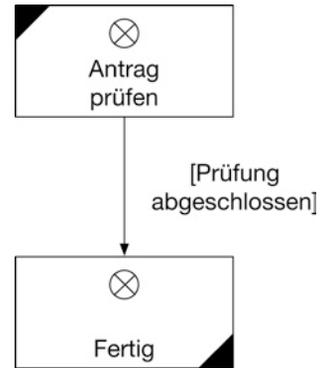
- Zu jedem *Funktionszustand* wird beschrieben, was im jeweiligen Verhaltensschritt zu tun ist. Die Endbedingungen eines Funktionszustandes entsprechen den ausgehenden Verbindungen, die vom jeweiligen Funktionszustand ausgehen. Wenn die Funktion zu unterschiedlichen Ergebnissen führen kann, können also verschiedene Folgezustände aktiviert werden. Dadurch wird die Abbildung von alternativen Verhaltensweisen möglich.
- In einem *Sendezustand* wird eine Nachricht an einen Empfänger übermittelt. In ihm wird so lange verharrt, bis der Empfänger in der Lage ist, die Nachricht entgegenzunehmen. Wer die Nachricht empfangen soll und welche Nachricht übermittelt wird, wird an der ausgehenden Kante des Sendezustands beschrieben.
- In einem *Empfangszustand* wird so lange verharrt, bis eine der Nachrichten eingetroffen ist, die der Empfangszustand entgegennehmen kann. Da auf unterschiedliche Nachrichten reagiert werden kann, kann je nach Typ der eingetroffenen Nachricht auch ein unterschiedlicher Folgezustand aktiviert werden. Dazu werden wiederum mehrere ausgehende Verbindungen verwendet, an denen beschrieben wird, welche Nachricht von welchem Absender zum entsprechenden Zustandsübergang führt. Auf diese Weise wäre es auch möglich, auf die gleiche Nachricht von unterschiedlichen Absendern unterschiedlich zu reagieren.

3.6.2 Beispiele

Um die Unterschiede und Gemeinsamkeiten zu den zuvor diskutierten Modellierungssprachen herauszuarbeiten, bedienen wir uns hier wieder der bereits verwendeten Beispiele. In den ersten Beispielen findet keine Kommunikation statt, wir fokussieren deshalb auf das Verhaltensdiagramm des jeweils einzigen beteiligten Subjektes.

Dieses Beispiel (siehe Abb. 3.48) zeigt einen Prozess mit einer einzelnen Aufgabe, in der ein Antrag (über den wir hier nicht mehr wissen) bearbeitet wird. Der Prozess endet,

Abb. 3.48 Einfacher Prozess als S-BPM Verhaltensdiagramm



nachdem die Bearbeitung des Antrags abgeschlossen wurde. Ein Verhaltensdiagramm muss immer einen Startzustand haben, der üblicherweise durch ein Dreieck in der linken oberen Ecke gekennzeichnet wird. Außerdem muss ein Endzustand existieren, der durch ein Dreieck in der rechten unteren Ecke gekennzeichnet wird. Um das Verhalten des Subjektes vollständig zu beschreiben, benötigen wir einen Zustandsübergang, der kennzeichnet, unter welchen Bedingungen wird den Zustand „Antrag prüfen“ verlassen können. Deshalb fügen wir hier einen funktionslosen Zustand „Fertig“ ein, den wir als Endzustand kennzeichnen.

Im obenstehenden Beispiel (siehe Abb. 3.49) ist der Prozess um eine Entscheidung mit drei möglichen Ausgängen erweitert, die einander ausschließen. Der Antrag wird geprüft, das Ergebnis dieser Prüfung ermöglicht das Treffen einer Entscheidung über die weitere Bearbeitung. Im Falle einer Investitionssumme ab 10.000 EUR wird der Antrag weitergeleitet. Dies kennzeichnen wir durch einen Sendezustand und spezifizieren an der ausgehenden Verbindung, wer den Antrag erhalten soll. Damit der Prozess vollständig beschrieben ist, müsste an dieser Stelle auch ein Verhaltensdiagramm für den Vorgesetzten erstellt werden.

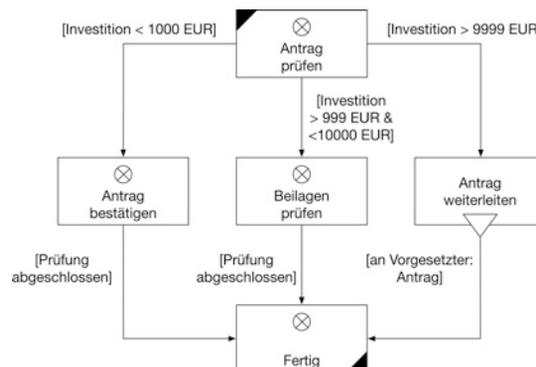


Abb. 3.49 S-BPM Verhaltensdiagramm mit Verzweigung (3 Zweige)

Auch hier können wir Teile eines Prozesses wiederholt abarbeiten (siehe Abb. 3.50). Dazu wird eine Verbindung am Ende des zu wiederholenden Teils eingefügt, mit einer Wiederholungsbedingung versehen und zum ersten Zustand des zu wiederholenden Teils zurückgeführt. Die andere ausgehende Verbindung führt den Prozess nach der wiederholten Abarbeitung weiter.

Das obige Beispiel zeigt nun einen Prozess mit zwei Subjekten. Die erste Abbildung (siehe Abb. 3.51) zeigt das Interaktionsdiagramm. Die beiden folgenden Modelle (siehe Abb. 3.52) zeigen die Verhaltensdiagramme von Sachbearbeiter und Abteilungsleiter. Die positive oder negative Beurteilung wird jeweils als Nachricht übermittelt. Die Begründung zur Beurteilung wird nur im Falle einer negativen Beurteilung als Datenobjekt zur Aufgabe „Antrag ablehnen“ übergeben. Im Fall der Bestätigung des Antrags wird das Datenobjekt „Beurteilung“ nicht mehr benötigt – wir können also annehmen, dass in diesem Fall keine weitere Begründung erfolgt.

3.6.3 Erweiterte Formen der Kommunikationsmodellierung

Durch den Fokus von S-BPM auf die Abbildung von Kommunikationsvorgängen erlaubt diese eine umfassendere und flexiblere Beschreibung derselben als sämtliche zuvor betrachteten Modellierungssprachen. Insbesondere erlaubt die S-BPM die Abbildung

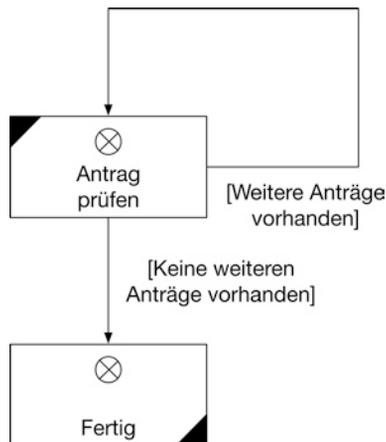
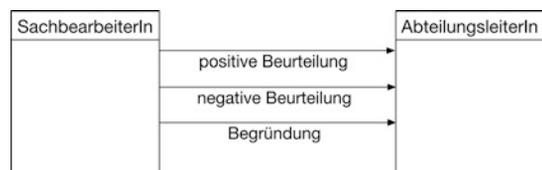


Abb. 3.50 S-BPM Verhaltensdiagramm mit Schleife

Abb. 3.51 S-BPM Interaktionsdiagramm



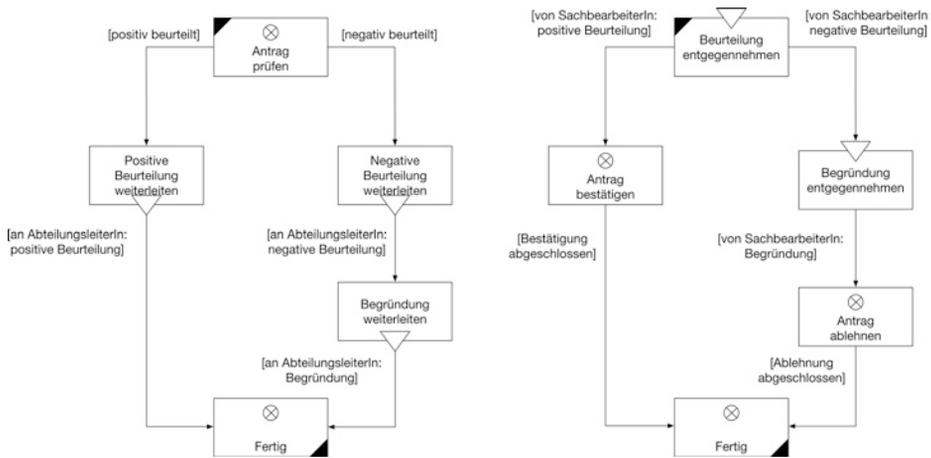


Abb. 3.52 S-BPM Verhaltensdiagramme zum obigen Interaktionsdiagramm (links: Verhalten Sachbearbeiter, rechts: Verhalten Abteilungsleiter)

von komplexen Kommunikationsszenarien durch den Einsatz von Inputpools sowie die detaillierte Beschreibung der in Nachrichten ausgetauschten Daten durch Geschäftsobjekte – wie nachfolgend erklärt.

3.6.3.1 Inputpools

Ein *Inputpool* dient einem Subjekt quasi als Postkasten, in dem eingehende Nachrichten gespeichert werden, bis sie im Verhaltensdiagramm benötigt werden. Im Gegensatz zu einem einfachen Postkasten ist ein Inputpool aber konfigurierbar. Es kann festgelegt werden, wie viele Nachrichten welchen Typs zwischengespeichert werden können. Wenn der Inputpool entsprechend seiner Konfiguration nicht in der Lage ist, eine Nachricht entgegenzunehmen, so muss der Sender im Sendezustand warten, bis die Nachricht zugestellt werden kann. Dadurch lassen sich unterschiedliche Kommunikationsszenarien abbilden.

Wird der Platz im Inputpool für einen bestimmten Nachrichtentyp auf 0 reduziert, so muss der Sender immer warten, bis der Empfänger bereit ist, die Nachricht entgegenzunehmen. Man spricht dann von synchroner Kommunikation. Wenn der Inputpool so konfiguriert ist, dass er Nachrichten zwischenspeichern kann, muss der Sender nicht warten, bis der Empfänger in jenem Zustand ist, in dem er die Nachricht annehmen kann. Man spricht dann von asynchroner Kommunikation (dies ist die einzige Art der Kommunikation, die in der BPMN abgebildet werden kann). Darüber hinaus erlauben Inputpools, Nachrichten in beliebiger Reihenfolge entgegen zu nehmen. Die Nachrichten müssen also nicht in jener Reihenfolge abgearbeitet werden, in der sie eintreffen, sondern können entsprechend der Bedürfnisse des Empfängers verarbeitet werden.

Inputpools haben keine grafische Entsprechung in der S-BPM, sondern sind ein Konzept der Ausführungssemantik. Sie werden für jedes Subjekt textuell bzw. in einem

Konfigurationswerkzeug beschrieben. Wenn keine Inputpools definiert werden, so hat die Standardkonfiguration unbegrenzt viele Speicherstellen für beliebige Nachrichten. Das Kommunikationsverhalten entspricht also den Nachrichtenflüssen der BPMN (asynchrone Kommunikation).

3.6.3.2 Geschäftsobjekte

Geschäftsobjekte dienen der Spezifikation jener Dinge, die zur Leistungserbringung in einem Geschäftsprozess benötigt werden. Es sind also die Dinge, die in einem Prozess verwendet werden und können Daten genauso umfassen wie physische Ressourcen. Geschäftsobjekte sind passiv, d. h. sie initiieren keine Interaktionen oder Aktionen. Geschäftsobjekte werden von Subjekten bearbeitet und können Nachrichten zugeordnet werden, um diese hinsichtlich ihres Inhalts näher zu spezifizieren.

Wie für Inputpools gibt es auch für Geschäftsobjekte keine grafische Entsprechung in der Notation der Modellierungssprache und sind ebenfalls Konzepte der Ausführungsemantik und daher von der technischen Ausführungsumgebung abhängig. Sie werden deshalb üblicherweise tabellarisch angegeben. Eine Grundstruktur von Geschäftsobjekten besteht aus einem Bezeichner, aus Datenstrukturen und Datenelementen. Der Bezeichner eines Geschäftsobjektes ergibt sich aus dem Geschäftsumfeld, in dem es eingesetzt wird. Beispiele sind Dienstreisantrag, Bestellung, Lieferschein, Rechnung etc.

Geschäftsobjekte setzen sich aus Datenstrukturen zusammen, deren Komponenten einfache Datenelemente eines bestimmten Typs (z. B. Zeichenkette oder Zahl) oder selbst wieder Datenstrukturen sein können.

Um das Verständnis sicherzustellen bzw. zu erleichtern empfiehlt es sich, die Bedeutung der Datenelemente näher zu beschreiben, insbesondere dann, wenn sich diese nicht zweifelsfrei aus den Bezeichnern ableiten lässt.

Die folgende Abbildung (siehe Tab. 3.1) zeigt ein Beispiel für einen Dienstreisantrag. Dieser besteht unter anderem aus der Datenstruktur ‚Daten zum Antragsteller‘ (Mitarbeiter) mit den Datenelementen für Name, Vorname und Personalnummer und der Struktur ‚Daten zur Dienstreise‘ mit den Datenelementen für Beginn, Ende und Zweck der Reise.

In vielen Fällen ändert sich die Semantik eines Geschäftsobjekts während der Prozessausführung, etwa wenn ein Lieferschein in eine Rechnung überführt wird. Für ein Geschäftsobjekt können deshalb mehrere verschiedene Zustände definiert werden. Bei einem Wechsel des Status werden nur die Datenstrukturen bzw. Datenelemente des vorherigen Status übernommen, die der neue Status benötigt, und bei Bedarf neue Komponenten hinzugefügt oder nicht mehr benötigte entfernt. Damit ist gewährleistet, dass ein Subjekt nur diejenigen Datenelemente für seine Arbeit zur Verfügung bekommt, die es dafür wirklich benötigt. Dies erleichtert die Einhaltung von Datenschutzbestimmungen.

Im Beispiel des Dienstreisantrags kann aus dem ursprünglichen Status ‚Reiseantrag‘ des Geschäftsobjekts der Status ‚Dienstreisebuchung‘ abgeleitet werden. Dabei werden insbesondere Datenelemente mit internen Angaben wie Personalnummer, Vergütungs-

Tab. 3.1 Datenstruktur des Geschäftsobjekts ‚DR-Antrag‘ (Dienstreiseantrag)

Datenstruktur	Bedeutung	Datentyp	Kann/Muss	Wertebereich/Default
Daten zum Antragsteller				
Name	Nachname	Character	M	...
Vorname	Vorname	Character	M	...
Personalnummer	...	Integer	M	...
Organisationseinheit	K	...
Vergütungsgruppe	K	...
Daten zur Reise				
Reisebeginn	...	Date	M	innerhalb/Jahres ab akt. Datum/akt. Datum
Reiseende	...	Date	M	Reisebeginn plus/Jahr/Reisebeginn
Internationale Reise	...	Boolean	K	j/n; n
Reiseziel (Ort/Land)	...	Character	M	...
Reisegrund	...	Character	M	...
Gewünschter Vorschuss	...	Integer	K	...
Daten zur Genehmigung				
Genehmigung	Genehmigungsvermerk	Boolean	M	j/n; n
Kostenstelle	...	Integer	M	...
Gewünschter Vorschuss	...	Integer	K	...

gruppe, Reisegrund und die komplette Datenstruktur zur Genehmigung entfernt, welche beispielsweise bei der Einschaltung eines Reiseagenten für die Buchung das Unternehmen nicht verlassen sollen und auch nicht außerhalb relevant sind. Dafür wird, wie in der folgenden Abbildung (siehe Tab. 3.2) gezeigt, eine neue Datenstruktur „Daten zur Buchung“ eingefügt. Sie enthält Datenelemente, mit denen die Reiserstelle gegenüber dem Reiseagenten eine Frist für den spätesten Eingang der Buchungsbestätigung setzen und bestimmte Hotelketten vorgeben kann, mit denen beispielsweise Rahmenverträge bestehen.

3.6.4 Einordnung

Im Gegensatz zu den anderen bislang besprochenen Modellierungssprachen gibt es in der S-BPM kein einzelnes Diagramm, das einen Geschäftsprozess vollständig beschreibt. Vielmehr wird für jedes Subjekt ein separates Verhaltensdiagramm erstellt. Diese werden

Tab. 3.2 Geschäftsobjekt ‚DR-Antrag‘ im Status ‚Dienstreisebuchung‘

Datenstruktur/ Datenelement	Bedeutung	Datentyp	Kann/Muss	Wertebereich/Default
Daten zum Antragsteller				
Name	Nachname	Character	M	...
Vorname	Vorname	Character	M	...
Daten zur Reise				
Reisebeginn	...	Date	M	innerhalb/Jahres ab akt. Datum/akt. Datum
Reiseende	...	Date	M	Reisebeginn plus/Jahr/ Reisebeginn
Reiseziel (Ort/Land)	...	Character	M	...
Daten zur Buchung				
Vertragshotelketten	Genehmigungs-vermerk	Character	M	...
Frist für Buchungs- bestätigung	...	Date	K	...
Buchungsbestätigung	...	Date	M	j/n

durch ein Interaktionsdiagramm miteinander verknüpft, in dem ihr Nachrichtenaustausch beschrieben ist. Dadurch ermöglicht S-BPM eine lose Kopplung von Prozessteilen und eine einfachere Veränderbarkeit des Verhaltens eines Subjektes, solange dessen Kommunikationsschnittstelle, also der Satz an empfangenen und gesendeten Nachrichten und deren Reihenfolge, unverändert bleibt.

Die Verwendung von Zustandsdiagrammen zur Beschreibung des Verhaltens eines Subjekts stellt ebenfalls einen grundlegenden Unterschied zu den anderen bislang behandelten Sprachen dar. Ein Zustandsdiagramm beschreibt – wie im Namen bereits enthalten – den Zustand eines Systems (hier: eines Subjektes – dies kann genauso ein Mensch wie eine Maschine sein) und die Ereignisse, die zu einem Zustandsübergang führen. Ein Subjekt kann sich immer nur in genau einem Zustand befinden – es ist deshalb per Definition nicht in der Lage, Vorgänge parallel auszuführen. Vielmehr arbeiten alle Subjekte parallel und unabhängig voneinander. Dies bedingt ein Umdenken bei der Modellierung, da Konstrukte wie UND-Konnektoren (in EPKs), Split/Joins (in Aktivitätsdiagrammen) oder parallele Gateways (in BPMN) nicht zur Verfügung stehen. Gleichzeitig führt dieser Modellierungsansatz zu einfacheren, kompakteren Modellen und einem vor allem im Gegensatz zur BPMN deutlich reduzierten Sprachumfang, was der Verständlichkeit der Modelle zuträglich ist.

3.7 Vergleich und Gegenüberstellung

Die hier betrachteten Modellierungssprachen sind unterschiedlich ausdrucksstark und haben aufgrund ihrer historischen Entwicklung unterschiedliche Schwerpunkte in der Abbildung von Aspekten der Geschäftsprozessmodellierung. Dieser Abschnitt versucht, diese Unterschiede nochmals systematisch anhand der im letzten Kapitel vorgestellten Prozessdefinition darzustellen und so die Handhabbarkeit der Sprachen ihrer Ausdrucksstärke gegenüberzustellen. Dabei ziehen wir die Semantik der vorgestellten Modellierungselemente als Ausgangspunkt heran.

Der Referenzpunkt der folgenden Betrachtungen ist die Prozessdefinition aus dem letzten Kapitel, die wir hier der Einfachheit halber nochmals anführen:

1. Prozessstrategie: Prozesse haben
 - a) einen definierten Anfang und Input (Startereignis),
 - b) und weisen ein definiertes Ende mit einem Ergebnis auf,
 - c) zur Befriedigung eines Kundenbedürfnisses (und damit zur Wertschöpfung beizutragen),
2. Prozesslogik: Ein Prozess
 - a) ist die Summe von miteinander verknüpften Aktivitäten (Aufgaben),
 - b) die nach dem Startereignis von Handelnden
 - c) in sachlogischer und zeitlicher Reihenfolge
 - d) zur Bearbeitung eines Geschäftsobjekts ausgeführt werden um
 - e) das gewünschte Ergebnis zu erzeugen.
3. Prozessrealisierung: Ein Prozess wird realisiert
 - a) In dem Menschen und/oder Maschinen die Aufgaben der jeweiligen Handelnden übernehmen, und diese
 - b) mit Hilfsmitteln (Sachmittel, Information, Anwendungsprogramme etc.) ausführen

Auf Basis dieser Definition können die Konzepte identifiziert werden, die in einem Prozessmodell abbildbar sein müssen, um Prozesse vollständig (entsprechend obiger Definition) abbilden zu können. Die folgende Tabelle (siehe Tab. 3.3) zeigt diese Konzepte. Mehrfach vorkommende Konzepte werden nur beim erstmaligen Auftreten genannt – wie z. B. „Ergebnis“. Bei Konzepten, die unterschiedlich konkret angeführt sind, werden nur die konkreteren Konzepte betrachtet – z. B. „miteinander verknüpfte Aktivitäten“ als allgemeinere Formulierung von „sachlogische und zeitliche Reihenfolge“.

Ordnet man nun die Modellierungselemente der betrachteten Sprachen diesen Konzepten zu, so ergibt sich folgende Tabelle (siehe Tab. 3.4):

Zu erkennen ist, dass die konzeptionelle Abdeckung über die Sprachen hinweg variiert. Darüber hinaus zeigt sich, dass nicht alle Sprachen alle Konzepte in gleichem Umfang bzw. in der gleichen Ausdrucksstärke behandeln. Die Zuordnung der Modellierungselemente zu den Konzepten ermöglicht einen ersten Ansatzpunkt für die Einschätzung ihrer Ausdrucksstärke.

Tab. 3.3 Konzepte der Geschäftsprozessdefinition

Definitionsteil	Konzept
1a	Anfang
	Input
1b	Ende
	Ergebnis
1c	Kundenbedürfnis
2a	Aktivitäten/Aufgaben
2b	Startereignis
	Handelnder
2c	Sachlogische Reihenfolge
	Zeitliche Reihenfolge
2d	Geschäftsobjekt
3a	Mensch
	Maschine
3b	Sachmittel
	Information
	Anwendungsprogramm
	Hilfsmittel (allgemein)

Nur bedingt erkennbar werden in dieser Übersicht die unterschiedlichen Zugänge in der Abbildung der sachlogischen und zeitlichen Zusammenhänge. Dennoch unterscheiden sich in diesem Punkt die Sprachen wesentlich: Flowcharts bieten keine Möglichkeit zur Abbildung paralleler Abläufe, in EPKs ist lediglich eine starke Kopplung von parallel verlaufenden Aktivitätszweigen vorgesehen, indem diese innerhalb eines Prozesses mittels UND- bzw. ODER-Operator verknüpft werden. UML Aktivitätsdiagramme und BPMN bieten die gleichen Mechanismen (unter anderen Namen), erlauben aber auch eine lose Kopplung von Prozessen bzw. Prozessteilen mittels Signalen (bei Aktivitätsdiagrammen) bzw. Nachrichtenflüssen (bei BPMN).

Insbesondere letzterer Mechanismus ermöglicht eine detaillierte Beschreibung von Kommunikationsabläufen von grundsätzlich unabhängigen Prozessteilen. Eine Einschränkung liegt in der notwendigen Vorabfestlegung von eindeutigen Zuordnungen zwischen Sendern und Empfängern von Nachrichten. Die S-BPM bietet einen ähnlichen Kommunikationsmechanismus, ist aber in diesem Punkt flexibler (insbesondere bei Verwendung der in der grafischen Darstellung der Sprache nicht abgebildeten Inputpools). Eine Beschreibung von parallel ablaufenden Prozessteilen ist in der S-BPM nur durch die Verteilung derselben auf unterschiedliche Subjekte möglich – innerhalb eines Subjektes kann immer nur ein Funktionszustand aktiv sein, es können also ausschließlich alternative Zweige im Verhalten eines Subjektes dargestellt werden.

Tab. 3.4 Zuordnung der Konzepte zu den Elementen der betrachteten Sprachen

Definitionsteil	Konzept	Flowchart	eEPK	UML Aktivitätsdiagramm	BPMN	S-BPMN
1a	Anfang	Terminierungselement	Ereignis	Startknoten	Startereignis	Startzustand
1b	Input	–	Informationsobjekt	Datenobjekt/ empfangenes Signal	Datenobjekt, Nachricht	Geschäftsobjekt, Nachricht
1c	Ende	Terminierungselement	Ereignis	Endknoten	Endereignis	Endzustand
2a	Ergebnis	–	Informationsobjekt	Datenobjekt/_ gesendetes Signal	Datenobjekt, Nachricht	Geschäftsobjekt, Nachricht
2b	Kundenbedürfnis	–	–	–	–	–
2c	Aktivitäten/Aufgaben	Operation	Funktion	Aktion	Aufgabe (unterschiedliche Typen)	Funktionszustand
2d	Startereignis	Nur unspezifisch	Spezifisch benanntes Ereignis	Nur unspezifisch, evtl. durch empfangenes Signal	Unterschiedliche Startereignisse (Nachrichten, Timer, Regeln etc.)	Startzustand, meist via Empfang einer Nachricht
2e	Handlender	–	Org. Einheiten	Partition	Pool, Lane	Subjekt
2f	Sachlogische/zeitliche Reihenfolge	Ablaufpfeil & Entscheidung	Ablaufpfeil & Konnektoren für alternative und parallele Abläufe, Datenfluss	Ablaufpfeil, Entscheidung, Split/Join	Sequenzfluss, Gateways für alternative und parallele Abläufe, Nachrichtenfluss, Ausnahmebehandlung, Transaktionen, Choreografie	Nachrichten zwischen Subjekten, Bedingungen für Zustandsübergänge in Subjektverhalten
2g	Geschäftsobjekt	–	Informationsobjekt	Datenobjekt	Datenobjekt, Nachricht	Geschäftsobjekt, Nachricht

(Fortsetzung)

Tab. 3.4 (Fortsetzung)

Definitionsteil	Konzept	Flowchart	eEPK	UML Aktivitätsdiagramm	BPMN	S-BPMN
3a	Mensch	–	Org. Einheiten	Partition	Pool, Lane, Benutzer-Task, Manueller Task	Subjekt
	Maschine	–	–	–	Service-Task	Subjekt
3b	Sachmittel	–	–	–	–	–
	Information	–	Informationsobjekt	Datenobjekt	Datenobjekt, Nachricht, Datenspeicher	Geschäftsobjekt, Nachricht
	Anwendungsprogramm	–	Anwendungssystem	–	Service-Task	Subjekt
	Hilfsmittel (allgemein)	–	–	–	–	–

Generell bietet die BPMN die größte Flexibilität in der Wahl der Darstellungsform eines Prozesses. Durch die Vielzahl an Modellierungselementen können auch komplexe Zusammenhänge kompakt dargestellt werden, was jedoch höhere Ansprüche an das Sprachverständnis der Modellnutzenden stellt. Einen anderen Ansatz verfolgen hier Aktivitätsdiagramme oder die S-BPM, die auf einem kompakten Satz an Modellierungselementen aufbauen. Dies führt bei komplexen Zusammenhängen zu umfangreichen Modellen, die hohe Ansprüche an die Modellnutzenden hinsichtlich des Modellverständnisses stellt. Die S-BPM reduziert die unmittelbar sichtbare Komplexität von Modellen durch die Verteilung eines Prozesses auf unterschiedliche Subjekte. Während dies zwar zu einfach erfassbaren Teilmodellen führt, stellt es gleichzeitig höhere Ansprüche an Modellnutzende bei der Erfassung der Gesamtzusammenhänge innerhalb des Prozesses.

Bei der Auswahl einer für eine gegebene Aufgabenstellung und Zielgruppe geeigneten Modellierungssprache ist dem entsprechend nicht nur auf den Abbildungsgegenstand (also den betrachteten Geschäftsprozess) Bezug zu nehmen, sondern auch auf die bekannten oder vermuteten Kompetenzen der Modellierenden und Modellnutzenden. Eine grundlegende Unterscheidung kann zwischen den Sprachen getroffen werden, die den Aktivitätsfluss ins Zentrum der Betrachtung rücken (wie die FlowCharts und die EPK), und jenen, die die Handelnden eines Prozesses und deren Kommunikation in den Vordergrund stellen (wie die S-BPM).

Die BPMN und Aktivitätsdiagramme eignen sich grundsätzlich für beide Abbildungsarten, wobei die BPMN ausdrückstärkere Mittel zur Abbildung von Kommunikationsabläufen bietet. Die endgültige Auswahl einer Modellierungssprache nach Festlegung des grundlegend verfolgten Abbildungsansatzes (Aktivitäts- vs. Kommunikationsfluss) ist letztendlich abhängig von den Präferenzen der Modellierenden bzw. Modellnutzenden.

Literatur

1. Scheer, A. W. (2003). ARIS – Business Process Modeling (3rd ed.). Springer; www.omg.org
2. Fleischmann, A., Obermeier, S., Schmidt, W., Sary, C., & Börger, E. (2011). Subjekt-orientiertes Prozessmanagement. Hanser Verlag

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitung 4.0 International Lizenz (<http://creativecommons.org/licenses/by-nc-nd/4.0/deed.de>) veröffentlicht, welche die nicht-kommerzielle Nutzung, Vervielfältigung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden. Die Lizenz gibt Ihnen nicht das Recht, bearbeitete oder sonst wie umgestaltete Fassungen dieses Werkes zu verbreiten oder öffentlich wiederzugeben.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist auch für die oben aufgeführten nicht-kommerziellen Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

