BPMN 2.0 Tutorial - Kompakte Einführung in die BPMN 2.0

iTransparent
BUSINESS PROCESS ARCHITECTS

1 Ziele

Die Benennung von Zielen ist ein elementarer Schritt bei der Entwicklung eines Standards oder – in viel geringerem Umfang – bei der Erstellung eines Tutorials. Formulierte Ziele dienen als ständige Maßgabe und beeinflussen verschiedene Entscheidungen. Um dem Leser frühzeitig einen Eindruck davon zu vermitteln, ob die BPMN 2.0 (Business Process Modeling and Notation) oder dieses Tutorial für ihn von Interesse sein könnten, wird mit einer Vorstellung der Ziele beider Dokumente begonnen.

1.1 Ziele von BPMN 2.0

Die BPMN 2.0 definiert eine standardisierte Sprache für die Erstellung von Geschäftsprozessen. Die grafischen Modelle sollen dabei für alle Beteiligten leicht verständlich sein, welche an der Erstellung mitwirken. Dies trifft sowohl auf Geschäftsexperten zu, welche bisher keinen oder wenig Kontakt zum Business Process Management (BPM) hatten, als auch auf die verschiedenen Rollen auf Seiten der BPM-Experten. Die unterstützten Rollen reichen dabei vom Business Analyst bis hin zum technischen Entwickler. BPMN 2.0 will dabei als standardisierte Brücke zwischen dem Geschäftsprozessdesign und der Umsetzung von Geschäftsprozessen agieren.

Um einen Geschäftsprozess aus verschiedenen Perspektiven betrachten zu können, werden verschiedene Diagrammarten angeboten. Jede Diagrammart betont bestimmte Aspekte eines Geschäftsprozesses und vernachlässigt andere. Dadurch wird eine Rundumbetrachtung erlaubt ohne durch eine Masse an Details zu verwirren.

Ein weiteres Ziel betrifft die Ausführung von Geschäftsprozessen. Die BPMN 2.0 will auf der einen Seite als Visualisierung für XML-basierte Ausführungssprachen wie WSBPEL (Web Services Business Process Execution Language) fungieren. Auf der anderen Seite erlauben ein überarbeitetes Metamodell (überarbeitet im Vergleich zu Vorversionen) und viele zusätzlich spezifizierbare Details auch die direkte Ausführung von BPMN 2.0-Prozessmodellen.

1.2 Zielsetzung dieses Tutorials

BPMN 2.0 bietet gegenüber seinen Vorgängerversionen eine deutlich größere Detailfülle. Dies zeigt sich gerade in den vielen zusätzlichen Attributen, die für viele Modellierungselemente definiert werden können. Diese gestiegene Anzahl ist vor allem auf das angestrebte Ziel der direkten Ausführbarkeit zurückzuführen. Ob sich BPMN 2.0 auch als Ausführungssprache durchsetzen wird, bleibt abzuwarten. Es gibt verschiedene Gründe die dafür oder dagegen sprechen. Eine Diskussion soll an dieser Stelle jedoch nicht erfolgen.

In diesem Tutorial wird BPMN 2.0 ausschließlich als Modellierungssprache gesehen und behandelt. Als solche haben sich bereits die Vorgängerversionen durchgesetzt und sind stark verbreitet. Eine Ausführung – egal in welcher Form – wird als weiterer möglicher Schritt akzeptiert, aber nicht behandelt. Wenn dennoch von Ausführung die Rede ist, dann soll – ohne auf technische Details einzugehen - ein Verhalten zur Laufzeit eines Prozesses beschrieben werden. Auf die Unterstützung durch ein Business Process Management System (BPMS)

wird dabei nicht eingegangen. Somit ist auch unwichtig, in welcher Form BPMN 2.0 ausgeführt wird. Als Folge können viele technische Details der BPMN 2.0 ignoriert werden. Dies trägt in hohem Maße dazu bei, dieses Tutorial inhaltlich hochwertig, aber in einer kompakten Form anbieten zu können. Sollte an verschiedenen Stellen der Wunsch nach mehr Details bestehen, so kann als Lektüre das Buch „BPMN 2.0 – Business Process Modeling and Notation“ von Thomas Allweyer oder als ultimative Referenz die Spezifikation selbst empfohlen werden. Auf beiden Referenzen basiert auch dieses Tutorial.

2 Überblick über die Diagrammarten

BPMN 2.0 bietet unterschiedliche Diagrammarten an, um verschiedene Aspekte eines Geschäftsprozesses modellieren zu können. Dabei sind in der neuen Version einige Diagrammarten hinzugekommen. Als Einstieg wird daher ein kurzer Überblick über die Diagrammarten der BPMN 2.0 gegeben, bevor im weiteren Verlauf dieses Tutorials die einzelnen Diagrammarten und ihre Bestandteile genauer betrachtet werden.

2.1 Prozessdiagramme

Prozessdiagramme – häufig auch Orchestrierungsdiagramme genannt - basieren in einem hohen Maße auf der bereits in BPMN 1.2 spezifizierten Diagrammform. Auf den ersten Blick sind dabei auch wenige Unterschiede zu erkennen. Die Weiterentwicklungen liegen im Detail und werden bei der eingehenden Betrachtung der Prozessdiagramme diskutiert.

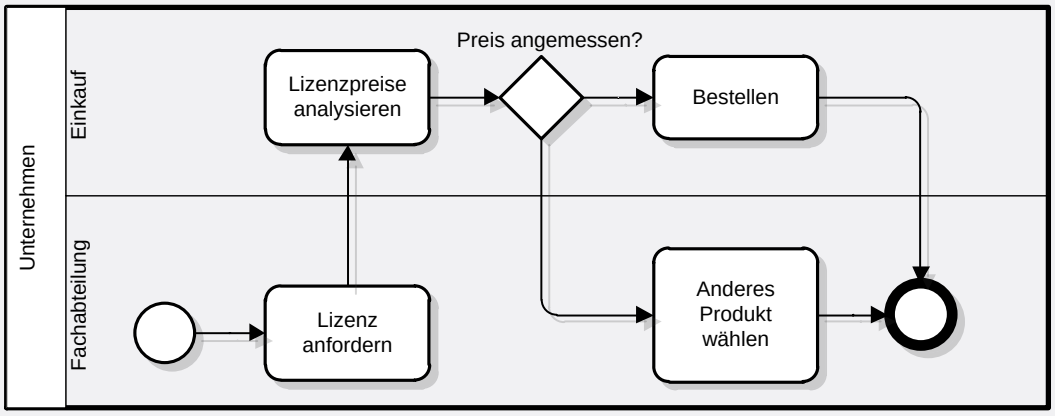


Abbildung 1: Prozessdiagrammbeispiel "Lizenz erwerben"

Das Ziel dieser Diagrammart ist es, Abläufe zur Durchführung einer Aufgabe mit ihren einzelnen Schritten zu definieren und darzustellen. Dabei beschreiben die Knoten im Prozessgraphen einzelne Tätigkeiten oder Ereignisse. Im Prozessdiagrammbeispiel in Abbildung 1 ist der Knoten "Lizenz anfordern" eine Tätigkeit. Die verbindenden gerichteten Kanten – wie beispielsweise zwischen "Lizenz anfordern" und "Lizenzpreise analysieren" dargestellt - definieren eine Reihenfolge zwischen den einzelnen Tätigkeiten. Sie sind somit ein Hauptbestandteil bei der Definition eines Kontrollflusses. Komplexere Sachverhalte bezüglich der Reihenfolge wie parallele oder bedingte Ausführung werden durch weitere Knoten modelliert, welche den Kontrollfluss aufspalten

oder wieder zusammenführen. Beispielhaft dargestellt ist dies bei der Entscheidung, ob ein Preis für eine Lizenz angemessen ist oder nicht. Zur Angabe von Verantwortlichkeiten für die einzelnen Tätigkeiten werden Pools als umschließende Rechtecke verwendet. So ist beispielsweise der Einkauf in einem Unternehmen dafür verantwortlich, Lizenzpreise zu analysieren und nicht die Fachabteilung.

Zusammengefasst zeigen Prozessdiagramme also die einzelnen Schritte größerer Abläufe und schaffen so eine Strukturierung für Prozesse.

2.2 Choreographiediagramme

Choreographiediagramme werden in der BPMN 2.0 neu eingeführt. Sie betrachten nicht einzelne Schritte oder Abläufe, sondern stellen den Nachrichtenaustausch von Prozessbeteiligten in den Vordergrund. Sie modellieren somit die Kommunikation zwischen verschiedenen Verantwortlichen (im Prozessdiagramm durch Pools dargestellt) und können – je nach Detaillierungsgrad und Vollständigkeit – als Schnittstellenspezifikation zwischen Prozessbeteiligten dienen.

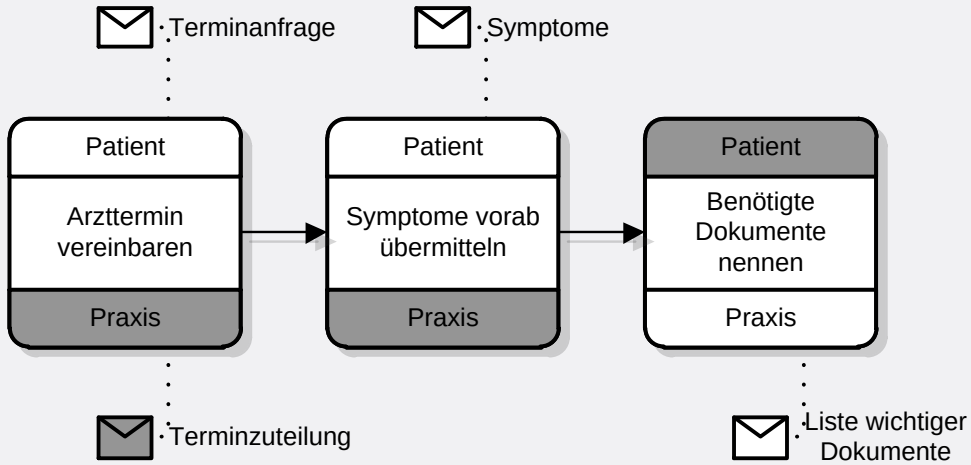


Abbildung 2 Choreographiediagrammbeispiel "Arzttermin vereinbaren"

Als Inhalt eines Choreographiediagramms werden der Nachrichtenaustausch selbst sowie Sender und Empfänger von Nachrichten modelliert. Im Beispiel in Abbildung 2 wird die Vereinbarung eines Arzttermins dargestellt. Dabei fragt ein Patient zuerst nach einem Termin in der Arztpraxis. Daraufhin bekommt er von der Praxis einen Termin zugeteilt. Anschließend übersendet der Patient eine Liste seiner Krankheitssymptome, soweit er sie beurteilen kann. Daraufhin erhält er von der Arztpraxis eine Liste der Dokumente, die er mitzubringen hat und welche eine Diagnose beschleunigen. Dies kann beispielsweise ein Impfpass oder existierende Röntgenbilder sein, welche Rückschlüsse auf die Krankheitsursache zulassen.

2.3 Kollaborationsdiagramme

Kollaborationsdiagramme stellen ebenso wie Choreographiediagramme den Austausch von Nachrichten zwischen Prozessbeteiligten dar. Zusätzlich wird hier

jedoch eine stärkere Verbindung zu den Prozessdiagrammen geschaffen und die Aufgaben der einzelnen Prozessbeteiligten werden mehr betont. Je nachdem, was im Vordergrund eines Diagramms stehen soll, können im Kollaborationsdiagramm verschiedene weitere Diagrammarten eingesetzt werden. Dies gilt sowohl für Prozessdiagramme als auch Choreographiediagramme. Auch eine Kombination der beiden Diagrammarten in einem Kollaborationsdiagramm ist möglich.

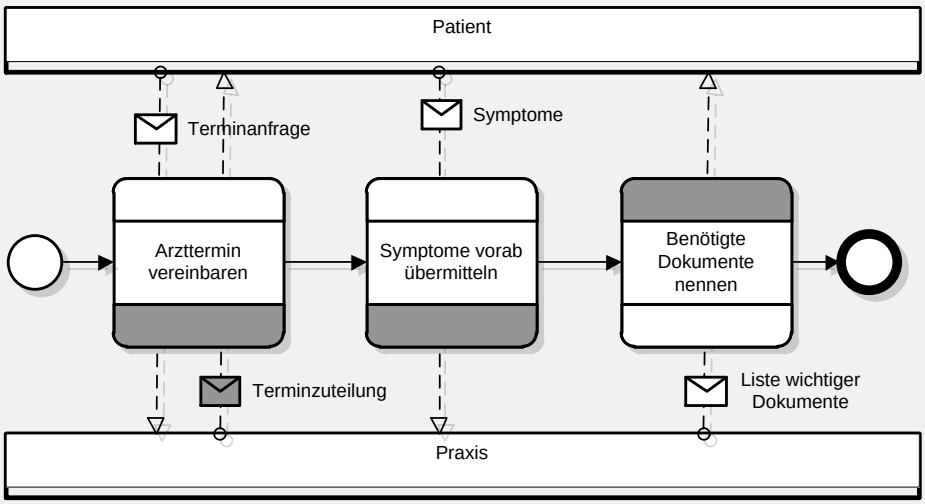


Abbildung 3 Kollaborationsdiagrammbeispiel "Arzttermin vereinbaren"

Im Beispieldiagramm in Abbildung 3 wird das bereits in Kapitel 2.2 eingeführte Choreographiediagramm in einem Kollaborationsdiagramm verwendet. Die Prozessbeteiligten werden dabei durch Pools dargestellt. Als weitere Detaillierung könnten beispielsweise die Prozessdiagramme der Beteiligten (Patient, Praxis) in den Pools dargestellt werden. Umgekehrt könnte das Choreographiediagramm durch eine einfachere Abbildung der Nachrichten ersetzt werden. Worauf das Augenmerk in einem Kollaborationsdiagramm gerichtet wird, obliegt allein dem Modellierer.

2.4 Konversationsdiagramme

Auch Konversationsdiagramme werden in der BPMN 2.0 neu eingeführt. Sie sollen vor allem einen abstrahierten Überblick über die Kommunikation zwischen verschiedenen Beteiligten schaffen. Dazu werden einzelne Nachrichten und deren Austausch nach inhaltlichen Gesichtspunkten zu Conversations zusammengefasst. Dies ermöglicht einen Blick aus einer "Vogelperspektive" auf das Kommunikationsverhalten der verschiedenen Prozessbeteiligten. Daraus lassen sich Abhängigkeiten zwischen den Prozessdiagrammen der Prozessbeteiligten ableiten. Dies gibt bei Änderungen einzelner Prozessdiagramme Hinweise darauf, welche Kommunikationsschnittstellen unter Umständen geändert oder mit angepasst werden müssen.

Ein Beispiel für ein Konversationsdiagramm ist in Abbildung 4 zu sehen. Dabei übergibt eine Filiale einem zentralen Lager die Gesamtbestellung. Sind einzelne Posten nicht mehr vorhanden oder sollen auf Vorrat bestellt werden, so sind die einzelnen Bestellungen mit verschiedenen Lieferanten abzuklären.

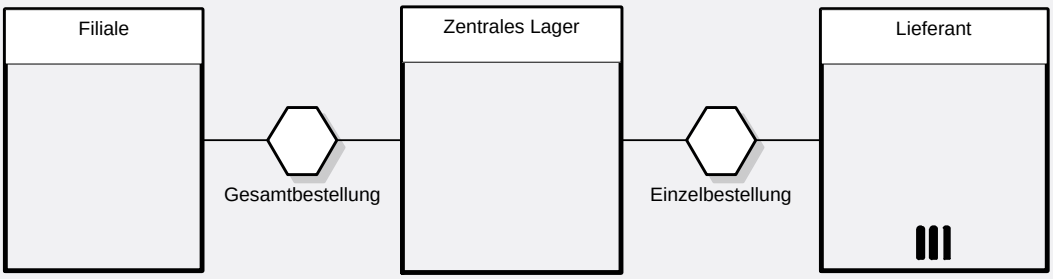


Abbildung 4 Konversationsdiagrammbeispiel "Lieferung"

3 Modellierung mit BPMN 2.0

Nach dem kurzen Überblick über die verschiedenen Diagrammarten der BPMN 2.0 werden diese nun detaillierter mit ihren jeweiligen Modellierungselementen betrachtet. Dabei steht nun nicht mehr die Frage im Mittelpunkt, was ein Diagramm aussagen soll, sondern wie es modelliert wird. Obwohl in diesem Kapitel viele Möglichkeiten der BPMN 2.0 diskutiert werden, wird kein Anspruch auf Vollständigkeit erhoben. Das Ziel ist eine kompakte Einführung, welche alle wesentlichen Elemente einer Diagrammart umfasst. Insbesondere für Detailfragen wird wiederum auf die Spezifikation verwiesen.

3.1 Prozessdiagramme

Prozessdiagramme stellen die verschiedenen Abläufe meist eines Verantwortlichen oder eines Verantwortlichkeitsbereichs wie einer Abteilung dar. Die Fokussierung auf einen Bereich ergibt sich dabei aus dem Wunsch, übersichtliche und lesbare Prozessdiagramme zu erstellen. Ein mit Information überladenes Prozessdiagramm verfehlt das Ziel, Abläufe übersichtlich darzustellen.

3.1.1 Allgemeines

Prozessdiagramme werden in private und öffentliche Prozesse unterteilt. Öffentliche Prozesse stellen dabei häufig nur eine vereinfachte Version eines privaten Prozesses dar, welche allgemein zugänglich ist und das Kommunikationsverhalten eines Prozesses mit anderen Beteiligten modelliert. Private Prozesse hingegen konzentrieren sich darauf, einen Ablauf innerhalb eines Verantwortlichkeitsbereichs so fein wie gewünscht zu modellieren. Die Kommunikation mit anderen Prozessbeteiligten kann also modelliert werden, sie ist aber nicht das primäre Ziel.

Aus diesem Grund beginnt diese Einführung in Prozessdiagramme mit der Modellierung von Abläufen und der privaten Prozessdiagramme. Erst im Anschluss daran wird die Kommunikation eines Prozesses mit weiteren Prozessbeteiligten betrachtet.

3.1.2 Grundelemente privater Prozessdiagramme

BPMN 2.0 definiert verschiedene Grundelemente für die Modellierung von Prozessdiagrammen, welche durch zahlreiche Spezialisierungen weiter mit Semantik angereichert werden. Als Einführung in Prozessdiagramme wird zuerst

auf die Grundelemente eingegangen. Eine Diskussion der Spezialisierungen erfolgt im weiteren Verlauf dieses Kapitels.

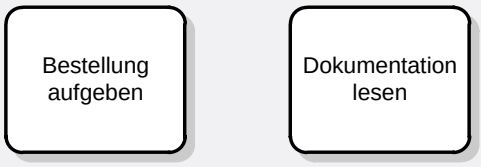


Abbildung 5 Tasks

Ein zentrales Modellierungskonzept in der BPMN 2.0 sind Aktivitäten. Die am häufigsten verwendete Aktivität ist das Modellierungselement Task. Ein Task modelliert eine einzelne Aufgabe, also einen Schritt in einem Prozess oder Ablauf. Setzt ein Prozess also eine Aufgabe um, so sind Tasks die einzelnen kleinen Schritte, welche zur Umsetzung dieser Aufgabe notwendig sind. Wie in Abbildung 5 zu erkennen ist, werden Tasks als Rechtecke mit abgerundeten Kanten dargestellt. Die narrative Beschreibung innerhalb der Rechtecke gibt Aufschluss über die Aufgabe, welche bei der Ausführung eines Tasks umgesetzt werden muss.

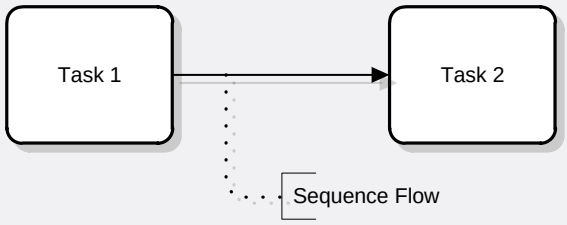


Abbildung 6 Sequence Flow

Um für Aktivitäten oder andere Elemente eine bestimmte Reihenfolge anzugeben, existiert der Sequence Flow (Abbildung 6). Ein Sequence Flow ist eine gerichtete Kante und drückt eine Abhängigkeit in der Ausführung aus. Das Modellierungselement, welches das Ziel eines Sequence Flows ist, darf erst ausgeführt werden, nachdem das Element ausgeführt wurde, von welchem der Sequence Flow ausgeht. In Abbildung 6 darf Task 2 also erst ausgeführt werden, nachdem Task 1 ausgeführt wurde. Durch die Angabe einer solchen Reihenfolge und somit Abhängigkeiten zwischen Aktivitäten oder weiteren Elementen erhält ein Prozess eine Ablaufstruktur.

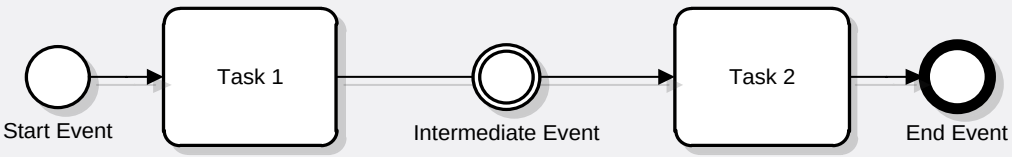


Abbildung 7 Events

Im Verlauf eines Prozesses können verschiedene Formen von Ereignissen (Events) auftreten. BPMN 2.0 unterscheidet generell drei Arten von Events (Abbildung 7): Start Events, End Events und Intermediate Events. Start Events werden mit einer dünnen Linie dargestellt. Sie stehen am Anfang eines

Prozesses oder Prozessabschnitts und haben keinen eingehenden Sequence Flow. End Events stehen am Ende eines Prozesses oder Prozessabschnitts und beenden diesen. Sie werden mit einer dicken Linie dargestellt und haben keinen ausgehenden Sequence Flow. Intermediate Events schließlich stehen für Ereignisse, die sich im Verlauf einer Prozessausführung ereignen. Sie werden mit einer doppelten dünnen Linie dargestellt.

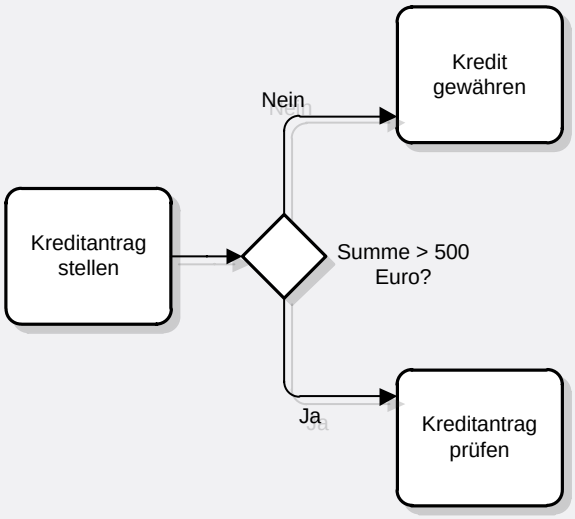


Abbildung 8 Gateways

Ein Kontrollfluss verläuft nicht immer sequenziell. Eher selten reicht das Modellierungselement Sequence Flow aus, um einen Kontrollfluss in einem Prozess vollständig zu modellieren. Als Element, das eine Aufspaltung und Zusammenführung des Kontrollflusses erlaubt, existiert daher das Gateway. Ein sehr einfaches Beispiel dazu ist in Abbildung 8 gegeben. Dabei wird ein Kreditantrag gestellt und es muss entschieden werden, ob eine tiefergehende Prüfung stattfindet. Ab einer Summe von 500 Euro ist dies nötig. Hier ist also eine XOR-Entscheidung modelliert, bei der entweder der eine oder der andere Pfad ausgeführt wird. Weitere Semantiken wie parallele (unabhängige) Ausführung existieren und werden durch Spezialisierungen eines Gateways definiert.

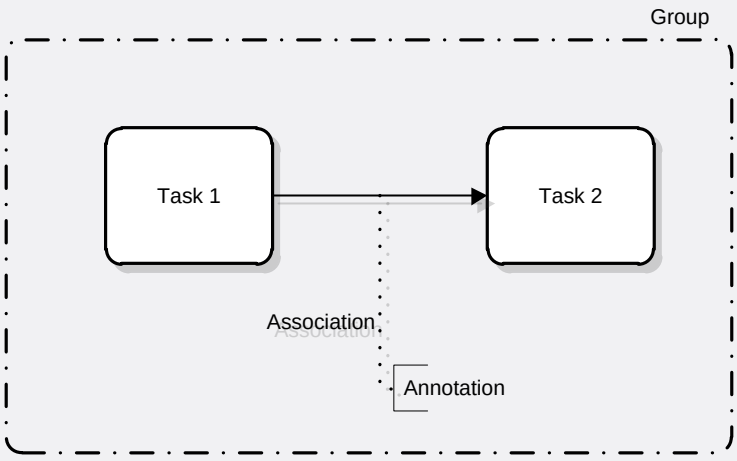


Abbildung 9 Kommentierungen

Obwohl Prozessmodelle an sich eine gewisse Aussagekraft besitzen, sind sie nicht immer vollständig selbsterklärend. Deswegen gibt es verschiedene Elemente zur weiteren Kommentierung und Erklärung eines Sachverhalts oder einer Modellierung (Abbildung 9). Um einzelne Modellierungsgegebenheiten zu kommentieren, verwendet man eine Annotation. In der eckigen Klammer kann beliebiger freier Text eingefügt werden. Um die Zuordnung des Kommentars zu einer bestimmten Stelle im Modell zu schaffen, wird eine Association verwendet. Sie ist eine simple Verknüpfung ohne weitere Semantik im Modell. Um mehrere Elemente zu gruppieren und so ihre – vielleicht andernfalls weniger offensichtliche – Zusammengehörigkeit aufzuzeigen, wird das Group-Konstrukt verwendet. Allen Kommentierungselementen ist gemeinsam, dass sie rein der Verdeutlichung der visuellen Modelldarstellung dienen. Sie haben keine definierten Auswirkungen auf eine gedachte Ausführung oder auf den Ablauf des modellierten Prozesses.

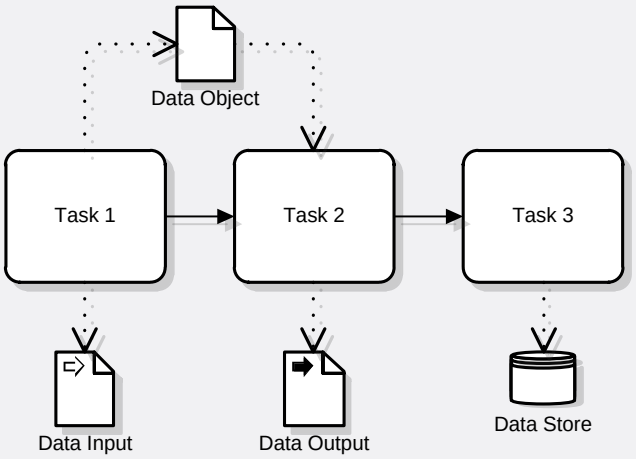


Abbildung 10 Daten und Datenfluss

Nicht nur Tätigkeiten und Abläufe werden für ein aussagekräftiges Prozessmodell benötigt. In den meisten Fällen soll die Übergabe von Informationen in Form von Daten ebenfalls dargestellt werden. Insbesondere soll erkennbar sein, welche Daten bei einer Tätigkeit produziert werden und welche Daten im Verlauf einer Tätigkeit benötigt und somit bereitgestellt werden müssen. Um diesen Anforderungen gerecht zu werden, stellt BPMN 2.0 verschiedene Modellierungselemente zur Verfügung (Abbildung 10). Zur Darstellung eines Datums (ohne tiefer gehende Spezifikation) dienen Data Objects. Dabei wird nur über den Namen ausgedrückt, um welches Datum es sich hierbei handelt. Um einen Fluss eines Datums auszudrücken, kann ein Data Object über eine Data Association verknüpft werden. In Abbildung 10 lässt beispielsweise eine Data Association ein Data Object von Task 1 nach Task 2 fließen. Anders ausgedrückt wird dieses Data Object also in Task 1 produziert und in Task 2 konsumiert.

Als Spezialisierungen der Data Objects existieren Data Inputs und Data Outputs. In der Darstellung fügen sie einem Data Object noch einen ausgefüllten oder nicht ausgefüllten Pfeil hinzu. Ein Data Input symbolisiert ein Datum, welches zur Ausführung eines Tasks benötigt wird. Ein Data Output symbolisiert ein Datum, das bei der Ausführung eines Tasks erstellt wird. Sowohl Data Inputs als auch

Data Outputs werden wie Data Objects durch eine Data Association mit anderen Elementen verknüpft. Data Inputs und Data Outputs sind nicht Bestandteile eines Datenflusses. Sie stellen nur eingehende und ausgehende Daten dar. Woher diese kommen und wohin diese versandt werden, wird nicht weiter spezifiziert.

Als letztes Element zur Datenmodellierung in BPMN 2.0 existiert der Data Store. Er symbolisiert eine Datenquelle oder Datensenke und wird durch seinen Namen näher spezifiziert. Ein Beispiel für einen Data Store könnte eine Datenbank mit den Daten aller Mitarbeiter sein, aus der zu Anfang eines Prozesses zur Änderung des Nachnamens gelesen wird und in welche zum Ende dieses Prozesses geschrieben wird. Ein Datenspeicher wird oft auch über mehrere Prozesse hinweg als konstant angenommen und über seinen Namen assoziiert.

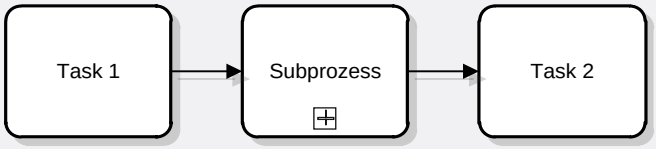


Abbildung 11 Subprozess

Neben Tasks gibt es noch weitere Aktivitäten, die bei der Erstellung größerer oder realer Prozessmodelle zum Einsatz kommen. Wichtig für die Strukturierung ist insbesondere der Subprozess. Ein Subprozess wird wie ein Task dargestellt, welcher zentral unten noch ein "+" zum Öffnen des Subprozesses aufweist (Abbildung 11). Ein Subprozess kann wiederum ein Prozessdiagramm mit allen bisher vorgestellten Elementen enthalten. Dieses enthaltene Prozessdiagramm beschreibt, was bei der Ausführung dieses Subprozesses geschieht. Somit hat ein Subprozess keine eigentliche Bedeutung für die Ausführung, sondern ermöglicht eine hierarchische Strukturierung von Prozessmodellen.

Abbildung 12 Call Aktivität

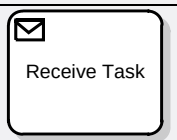
Eine weitere Aktivität, welche insbesondere Wiederverwendung ermöglicht, ist die Call Aktivität. Sie stellt einen Aufruf eines global definierten Tasks dar, der an einer bestimmten Stelle in einem Prozessmodell erfolgen soll. Dieses Element ermöglicht es, einen Task einmalig zu definieren und dann an verschiedenen Stellen im Prozessmodell zu verwenden. So muss ein Task bei Bedarf beispielsweise nur einmal geändert werden. Zur Unterscheidung einer Call Aktivität von einem Task wird diese mit einer dickeren Außenlinie dargestellt.

3.1.3 Aktivitäten im Detail

Aktivitäten bieten eine Fülle von Spezialisierungen, welche ihre Semantik weiter verfeinern. Dadurch verstärkt sich die Aussagekraft und die "Natur" beispielsweise eines Tasks wird klarer. So kann damit ausgedrückt werden, ob ein Task automatisiert ausgeführt wird oder ob ein Mensch diese Aufgabe

übernimmt. Die Möglichkeiten der Spezialisierungen für Tasks sind in Tabelle 1 aufgeführt. Die Art des spezialisierten Tasks wird dabei grafisch durch ein Symbol in der rechten oberen Ecke ausgedrückt.

Tabelle 1 Spezielle Tasks

Bezeichnung	Erklärung	Symbol
Abstract Task	Definiert die allgemeinste Form eine Tasks. Für die Ausführung ist es eine Taskspezifikation ohne weitere Angaben.	
Service Task	Symbolisiert einen synchronen Aufruf eines Web Services (default) oder den Aufruf einer anderen Applikation. Es wird auf eine Antwort gewartet, bevor der Prozess weiterläuft.	
Send Task	Symbolisiert das asynchrone Senden einer Nachricht an einen Web Service oder eine andere Applikation. Es wird nicht auf eine Antwort gewartet.	
Receive Task	Symbolisiert das Warten auf eine Nachricht (beispielsweise von einem Web Service). Der Prozess wartet so lange, bis diese Nachricht eintrifft. Erst danach erfolgt die weitere Ausführung. Ein Receive-Task kann einen Prozess starten.	
User Task	Symbolisiert einen Task, der von einem BPMS unterstützt durch einen Menschen ausgeführt wird. Ein solcher Task wird beispielsweise in einer Taskliste verwaltet.	
Manual Task	Symbolisiert einen Task, welcher von einem Menschen ohne jegliche Unterstützung durch ein BPMS ausgeführt wird.	
Business Rule Task	Symbolisiert einen Aufruf an eine Business Rules Engine. Diese werden im Zusammenhang mit BPMN 2.0 vor allem zur Auswertung von Daten auf Basis deklarativer Regeln verwendet.	

Script Task	Symbolisiert die Auswertung eines externen Skripts.	

Bereits im Kapitel 3.1.2 wurden die Call Aktivitäten eingeführt. Diese können vor allem Globale Tasks aufrufen. Globale Tasks werden einmal definiert und stehen dann überall im Prozess zur Verfügung.

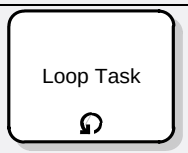
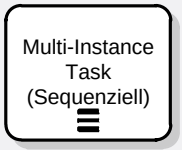
Als Globale Tasks können

- User Tasks
- Manual Tasks
- Business Rule Tasks und
- Script Tasks

definiert werden.

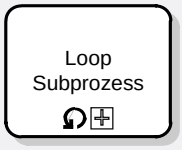
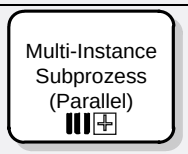
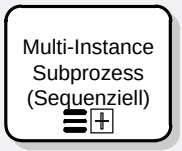
Zu allen Spezialisierungen von Tasks existiert eine Menge an Markern, welche den Kontrollfluss beeinflussen. Diese Marker werden jeweils unten in der Mitte eines Tasks eingefügt. In BPMN 2.0 stehen die folgenden Marker für Tasks zur Verfügung (Tabelle 2):

Tabelle 2 Marker für Tasks

Bezeichnung	Erklärung	Symbol
Loop	Ein Task mit einem Loop Marker wird solange wiederholt ausgeführt, bis eine definierte Bool'sche Bedingung nicht mehr erfüllt ist.	
Multi-Instance (Parallel)	Von einem Task werden basierend auf Eingabewerten oder modellierten Definitionen zur Laufzeit parallel mehrere Instanzen gebildet. Diese werden gleichzeitig ausgeführt. Der Task gilt als beendet, sobald eine weitere zu definierende Bedingung erfüllt ist.	
Multi-Instance (Sequenziell)	Erzwingt eine sequenzielle statt einer parallelen Ausführung der gebildeten Instanzen. Die restlichen Eigenschaften sind analog zum Multi-Instance (Parallel) Marker.	
Compensation	Bereits durchgeführte Tasks können durch ein Compensation Event rückgängig gemacht werden. Diese Wiederherstellung eines vorhergehenden Zustands (beispielsweise in externen Systemen) erfolgt durch einen Compensation Task, der mit dem Compensation Event verknüpft ist.	

Neben Tasks können auch Subprozesse mit verschiedenen Markern versehen werden. Die meisten Marker sind gleich und haben eine ähnliche Bedeutung. Speziell für Subprozesse kommt aber noch ein weiterer Marker hinzu. Eine Übersicht der Marker für Subprozesse liefert Tabelle 3.

Tabelle 3 Marker für Subprozesse

Bezeichnung	Erklärung	Symbol
Loop	Ein Subprozess wird in einer Schleife ausgeführt, bis eine vorher definierte Bedingung nicht mehr gilt.	
Multi-Instance (Parallel)	Es werden mehrere Instanzen eines Subprozesses erstellt und parallel ausgeführt.	
Multi-Instance (Sequenziell)	Es werden mehrere Instanzen eines Subprozesses erstellt und nacheinander ausgeführt.	
Compensation	Der Subprozess enthält die Schritte, die zu einer Compensation für einen anderen Task oder Subprozess notwendig sind.	
Ad-Hoc	Ad-Hoc-Subprozesse ermöglichen eine sehr freie Modellierung. Sie enthalten lediglich eine Ansammlung von Tasks oder weiteren Subprozessen. Zwischen diesen ist aber kein Sequence Flow oder Ähnliches modelliert. Sie können in beliebiger Reihenfolge zu beliebigen Zeitpunkten ausgeführt werden.	

Über die Marker hinaus bieten Subprozesse noch weitere Spezialisierungen und Einsatzmöglichkeiten. Dabei gilt als Voraussetzung, dass jeder Subprozess einen Scope definiert, welcher beispielsweise die Sichtbarkeit von Variablen oder Tasks, das Exception Handling oder eine Compensation umfasst.

Eine Spezialform eines Subprozesses ist der Event-Subprozess. Dieser hat keinen eingehenden oder ausgehenden Kontrollfluss und kann in einem Prozess oder einem anderen Subprozess platziert werden. Gestartet wird er durch ein eintreffendes Event. Ein Event-Subprozess wird von einer gepunkteten Linie umgeben auf der Ebene (Prozess oder Subprozess) dargestellt, auf welcher ein eingehendes Event die Ausführung des Event-Subprozesses starten soll.

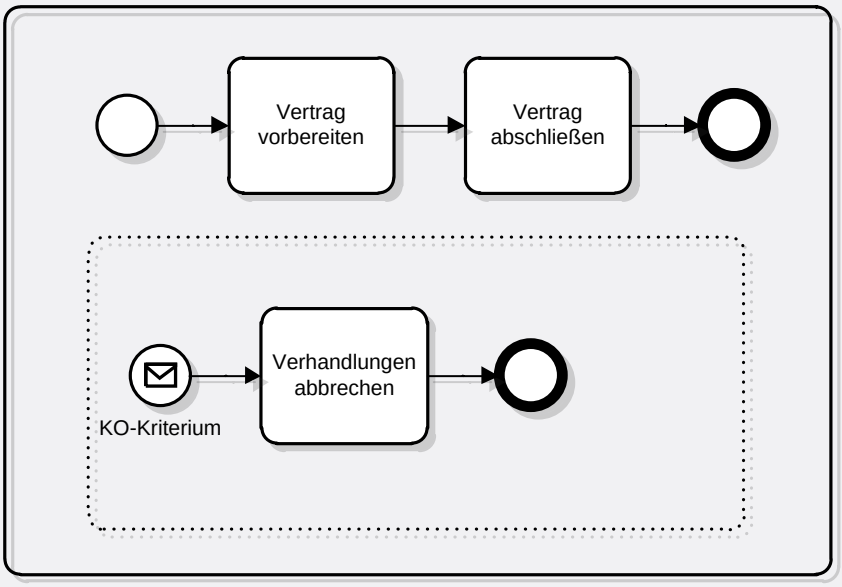


Abbildung 13 Beispiel für einen Event-Subprozess

Ein Beispiel ist in Abbildung 13 dargestellt. Dabei wird stark abstrahiert ein Prozess einer Vertragsverhandlung modelliert. Während ein Vertrag vorbereitet wird, ist davon auszugehen, dass noch weitere Erkundigungen über den Vertragspartner angestellt werden. Werden dabei Informationen gefunden, welche ein KO-Kriterium darstellen wie beispielsweise mangelnde Bonität, so werden die Verhandlungen sofort abgebrochen und der Prozess wird beendet.

Abbildung 14 Subprozess mit Transaktion

Eine weitere Form eines Subprozesses sind Transaktionen. Transaktionen stellen inhaltlich einen Subprozess dar, dessen Ausführung durch ein Transaktionsprotokoll wie WS-Transaction abgesichert ist. Dadurch werden beispielsweise die aus dem Datenbankbereich bekannten ACID-Eigenschaften auf einen Subprozess übertragen. Dargestellt wird ein Subprozess mit Transaktion in BPMN 2.0 durch eine doppelte Außenlinie, welche den Subprozess umgibt (Abbildung 14).

3.1.4 Events im Detail

Prinzipiell gibt es die drei bereits eingeführten Arten von Events: Start Events, Intermediate Events und End Events. Von diesen Event-Typen gibt es noch diverse Spezialisierungen, welche die Aussagekraft verstärken oder Voraussetzung für die Verwendung an einer bestimmten Stelle in einem Prozess sind. Obwohl einige Formen der Spezialisierung auf mehrere Arten von Events angewendet werden können, werden sie im Folgenden jeweils im Einzelnen behandelt. Je nach Event-Art unterscheidet sich die Bedeutung teilweise zu stark, um eine allgemeine Beschreibung des Verhaltens angeben zu können.

Start Events finden an verschiedenen Stellen in einem Prozess Anwendung. Sie können einen allgemeinen Prozessbeginn markieren oder auch den Startpunkt in einem Subprozess. Durch die Verwendung in einem Event-Subprozess können sie sogar weitere Startpunkte in einem Prozess oder Subprozess definieren, welche unabhängig vom eigentlichen Kontrollfluss sind. Die Spezialisierungen der Start Events sind in Tabelle 4 aufgeführt:

Tabelle 4 Spezialisierungen von Start Events

Bezeichnung	Beschreibung	Symbol
None	Ein allgemeines Start-Event ohne weitere Spezifizierung. Wird meist in Subprozessen zur Visualisierung verwendet. Der Start eines Subprozesses zur Laufzeit erfolgt dabei durch einen Aufruf aus dem Vaterprozess.	
Message	Eine Nachricht trifft von einem externen System ein und startet damit einen Prozess oder einen Event-Subprozess. Als externes System werden dabei vor allem weitere Prozesse oder Web Services verwendet.	
Timer	Ermöglicht die Definition eines bestimmten Zeitpunkts, zu dem ein Prozess starten soll. Wird vor allem bei regelmäßig auszuführenden Prozessen ("Jeden Montag um 9.00 Uhr") verwendet.	
Conditional	Startet einen Prozess auf Basis einer Bedingung, die erfüllt ist. Diese Bedingung darf dabei nicht auf den Daten einer Prozessinstanz basieren, da es vor dem Start noch keine Instanz gibt.	
Signal	Ein Signal, welches von einem anderen Prozess im "Broadcast-Verfahren" ausgesandt wurde, startet einen Prozess. Im Unterschied zu einem Message-Event wird bei diesem Verfahren nicht direkt ein Empfänger adressiert und es können mehrere Prozesse gleichzeitig durch ein Signal gestartet werden.	
Multiple	Definiert mehrere Trigger, um einen Prozess zu starten. Es wird nur ein Trigger zum Prozessstart benötigt.	

Parallel Multiple	Es werden mehrere Trigger für einen Prozessstart definiert. Alle diese Trigger müssen gefeuert sein, bevor ein Prozess gestartet wird.	
Error	Startet einen Event-Subprozess indem ein geworfener Fehler gefangen und der Subprozess gestartet wird.	
Escalation	Startet einen Event-Subprozess, weil bestimmte vordefinierte Werte bezüglich des Fortschritts eines Prozesses oder Subprozesses nicht eingehalten wurden.	
Compensation	Startet einen Event-Subprozess, welcher die Compensation für einen Prozess oder Subprozess durchführt.	

Nach den Start Events werden nun die Spezialisierungen der End Events betrachtet. Allgemein beendet ein End Event einen Prozess oder Subprozess. Daher hat kein End Event einen ausgehenden Sequence Flow. Von End Events gibt es die folgenden Spezialisierungen (Tabelle 5):

Tabelle 5 Spezialisierungen von End Events

Bezeichnung	Beschreibung	Symbol
None	Ein allgemeines End Event ohne weitere Aussagekraft. Es wird hauptsächlich am Ende von Subprozessen verwendet.	
Message	Sendet als Ende eines Prozesses eine Nachricht an einen anderen Prozess oder einen Web Service. Häufig wird an die Adresse gesendet, von welcher die Prozessinstanz gestartet wurde.	
Error	Wirft einen spezifizierten Error und beendet einen Subprozess mit allen aktiven Threads. Es wird davon ausgegangen, dass der Error an einer bestimmten Stelle im Prozess gefangen und behandelt wird.	

Escalation	Wirft eine Escalation in einem Subprozess und beendet diesen. Die aktuell aktiven Threads werden jedoch nicht beendet. Parallel dazu wird die Escalation gefangen und die modellierten Aktionen werden ausgeführt.	
Cancel	Wird in einem Subprozess mit Transaktion verwendet und beendet diesen. Außerdem wird ein Cancel Intermediate Event aufgerufen, welches das weitere Vorgehen modelliert.	
Compensation	Beendet einen Subprozess und zeigt gleichzeitig an, dass eine Compensation notwendig ist. Das modellierte Compensation-Verhalten wird daraufhin angestoßen.	
Signal	Bei Beendigung eines Prozesses oder Subprozesses wird ein Signal im Broadcast-Verfahren ausgelöst. Es hat keinen definierten Empfänger wie eine Message.	
Terminate	Beendet einen Prozess und alle Subprozesse oder Tasks unmittelbar. Dies schließt auch Multiple Instance-Tasks mit ein. Es wird keine weitere Compensation oder ein Event Handling durchgeführt.	
Multiple	Beim Beenden eines Prozesses werden verschiedene Aktionen ausgelöst. Beispielsweise können mehrere Messages gesendet werden.	

Ein Intermediate Event modelliert Ereignisse, die während eines Prozessverlaufs eintreffen. Je nach Spezialisierung kann auf das Eintreffen eines Events gewartet werden oder es wird ein Event geworfen. Die im Folgenden vorgestellten spezialisierten Events werden durch Sequence Flows direkt in den "normalen" Kontrollfluss eingebunden (Tabelle 6):

Tabelle 6 Intermediate Events im Kontrollfluss

Bezeichnung	Beschreibung	Symbol
None	Modelliert das Aussenden eines Events ohne nähere Spezifikation. Wird beispielsweise verwendet, um eine Statusänderung zu propagieren.	
Message	Wird zur Modellierung eines Nachrichtenempfangs oder –versands verwendet. Beim Senden einer Nachricht muss der jeweilige Prozessbeteiligte oder das System definiert werden.	Senden  Empfangen 
Timer	Lässt einen Prozess auf einen bestimmten Zeitpunkt oder für eine gewisse Zeitspanne warten.	
Escalation	Wirft ein Escalation-Event. Wird häufig beim Eintreten unerwünschter, aber vorsehbarer Umstände verwendet, um den Effekten gegenzusteuern.	
Compensation	Wirft ein Compensation Event. Dies stößt einen Compensation-Vorgang auf der aktuellen Ebene an. Die Ebene kann dabei beispielsweise der Subprozess sein, in welchem das Compensation Event modelliert wird.	
Conditional	Empfängt ein Event, welches besagt, dass eine vordefinierte Bedingung erfüllt ist.	

Link	Modelliert ein Verbindungselement für verschiedene Prozessabschnitte. Link Events können als Sprünge im Prozessmodell verwendet werden, um beispielsweise lange Sequence Flow-Linien quer durch ein Prozessmodell zu verhindern. Für den Ausdruck von Prozessmodellen auf Papier können sie den Seitenübergang verdeutlichen.	Quelle  Ziel 
Signal	Sendet ein Signal ohne dedizierten Empfänger im Broadcast-Verfahren oder beobachtet die Auslösung eines Signals und reagiert darauf. Dieser Mechanismus funktioniert auch über Prozessgrenzen hinweg.	Signal aussenden  Signal empfangen 
Multiple	Beim Auslösen wird eine Vielzahl von Triggern ausgelöst. Beim Empfang reicht das Erkennen eines der vielen definierten Trigger aus, um mit dem Kontrollfluss fortzufahren.	Trigger auslösen  Trigger empfangen 
Parallel Multiple	Alle spezifizierten Trigger müssen vorhanden sein, um dieses Event im Kontrollfluss zu aktivieren und mit der Ausführung an der modellierten Stelle fortzufahren.	

Von den Intermediate Events existiert noch eine weitere Form, welche Boundary Events genannt werden. Boundary Events werden nicht direkt im Kontrollfluss durch Verknüpfungen mit Sequence Flows modelliert, sondern werden direkt an einen Task oder Subprozess angehängt. Trifft ein Event ein, während sich der Task oder Subprozess in Ausführung befindet, so wird diese Ausführung entweder unterbrochen oder es wird parallel dazu ein weiterer Ausführungspfad ausgehend vom Boundary Event gestartet.

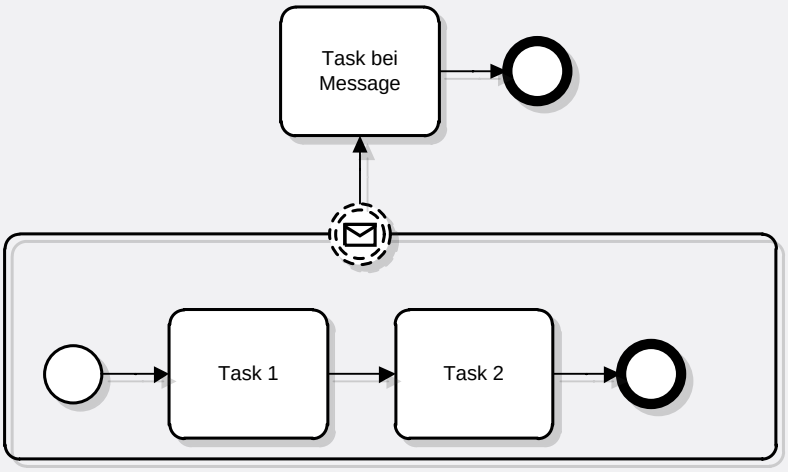


Abbildung 15 Beispiel für ein nicht unterbrechendes Boundary Message Event

In Abbildung 15 ist beispielsweise ein nicht unterbrechendes Boundary Message Event an einen Subprozess angebunden. Trifft die spezifizierte Nachricht ein, während sich Task 1 oder Task 2 in Ausführung befinden, so wird zusätzlich parallel der "Task bei Message" ausgeführt. Die Eigenschaft, dass der Subprozess nicht unterbrochen wird, ist an der gestrichelten Außenlinie des Intermediate Events erkennbar.

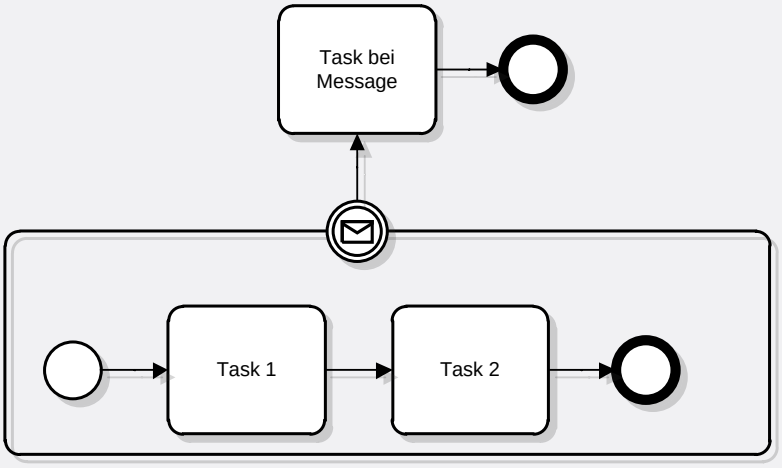


Abbildung 16 Beispiel für ein unterbrechendes Boundary Event

In Abbildung 16 ist ein unterbrechendes Boundary Message Event an einem Subprozess modelliert. Wenn die spezifizierte Nachricht in diesem Fall eintrifft, dann wird die Ausführung des Subprozesses unterbrochen. Ob sie nach

Ausführung des Event-Pfades wieder aufgenommen wird, hängt von Attributen des Boundary Events ab. Bei bestimmten Boundary Event Typen wie einem Error oder einem Cancel kann die Ausführung des Subprozesses in keinem Fall wieder aufgenommen werden, weil dies der Semantik der Event-Spezialisierung widerspricht. Die folgenden spezialisierten Events können als Intermediate Boundary Event verwendet werden (Tabelle 7):

Tabelle 7 Intermediate Boundary Events

Bezeichnung	Beschreibung	Symbol
Message	Eine eingehende Nachricht startet den Ablauf, welcher durch einen vom Event ausgehenden Sequence Flow modelliert ist. Ein Boundary Message Event kann sowohl unterbrechend als auch nicht unterbrechend verwendet werden.	Unterbrechend  Nicht unterbrechend 
Timer	Wird zu einem gewissen Zeitpunkt ausgelöst. Dieser kann einmalig sein (beispielsweise 01.03.2011, 0.00 Uhr) oder wiederkehrend (jeden Montag, 9.00 Uhr).	Unterbrechend  Nicht unterbrechend 
Escalation	Fängt ein definiertes Escalation Event und führt die definierte Behandlung unterbrechend oder nicht unterbrechend durch. Von der Intention her sollte ein Escalation Event einen Task oder Subprozess nicht komplett beenden. Durch Setzen von Parametern kann dies aber dennoch erreicht werden.	Unterbrechend  Nicht unterbrechend 

Error	Fängt ein Error Event und führt die Fehlerbehandlung durch. Es existiert keine nicht unterbrechende Version und der entsprechende Task oder Subprozess wird in jedem Fall beendet.	
Cancel	Fängt ein Cancel Event, welches in einem Transaktions-Subprozess geworfen wird. Daraufhin kann beispielsweise ein kontrollierter Rollback modelliert werden.	
Compensation	Fängt ein Compensation Event und führt die assoziierte Compensation Aktivität wie einen Task oder Subprozess aus.	
Conditional	Wird ausgelöst, wenn eine vordefinierte Bedingung als wahr evaluiert wird. Der modellierte Pfad, welche vom Event ausgeht, wird unterbrechend oder nicht unterbrechend durchgeführt.	Unterbrechend  Nicht unterbrechend 
Signal	Wenn ein Signal empfangen wird, wird der modellierte Pfad gestartet. Ein Signal ist hierbei allgemeiner als beispielsweise ein Error Event, da nicht nur Fehler der Auslöser sein können. Auch die Beendigung eines Tasks kann beispielsweise ein Auslöser sein.	Unterbrechend  Nicht unterbrechend 

Multiple	Wenn einer der spezifizierten Trigger eingeht, dann wird der modellierte Event-Pfad gestartet.	Unterbrechend  Nicht unterbrechend 
Parallel Multiple	Nur bei Eingehen aller definierten Trigger wird dieses Event ausgelöst und der modellierte Event-Pfad ausgeführt.	Unterbrechend  Nicht unterbrechend 

3.1.5 Gateways im Detail

Wie bei den Events in Kapitel 3.1.4 beschrieben gibt es auch bei den Gateways verschiedene Spezialisierungen, welche die Semantik auseinander- oder zusammenfließender Sequence Flows genau definieren. Dieses Kapitel bietet eine Übersicht über die verschiedenen Spezialisierungen der Gateways und ihre Bedeutung für den Kontrollfluss.

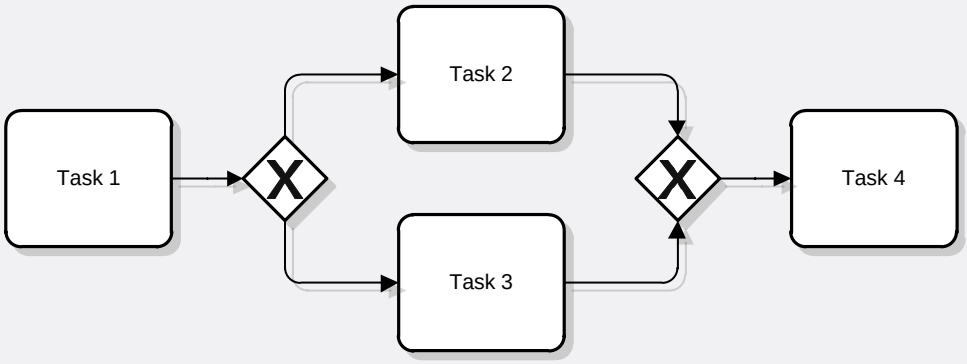


Abbildung 17 Exclusive Gateway

Das Standard-Gateway ist dabei das Exclusive-Gateway (Abbildung 17), da ein Gateway ohne weitere Spezifizierung implizit als Exclusive Gateway behandelt

wird. Explizit wird ein Exclusive Gateway durch ein "X" im Gateway-Symbol modelliert. Ein Exclusive-Gateway setzt eine XOR-Semantik um. Es wird also, basierend auf einer vordefinierten Bedingung, genau einer der ausgehenden Pfade des Gateways ausgeführt, welches den Kontrollfluss aufspaltet. Im Beispiel wird also nach der Ausführung von Task 1 entweder Task 2 oder Task 3 ausgeführt. Auf keinen Fall werden beide oder kein Task ausgeführt. Nach der Ausführung genau eines der beiden Tasks beginnt die Ausführung von Task 4.

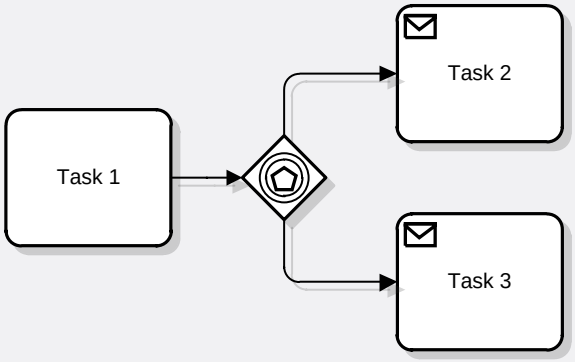


Abbildung 18 Event-Based Gateway

Abbildung 18 stellt beispielhaft das Event-Based Gateway für eine Aufspaltung des Kontrollflusses dar. Auch bei dieser Variante wird genau ein Pfad ausgeführt. Die Entscheidung wird aber nicht aufgrund einer vordefinierten Bedingung getroffen, sondern aufgrund eines eintreffenden Events. Am Beginn jedes ausgehenden Pfades muss bei einem Event-Based Gateway ein Modellierungselement stehen, welches auf das Eintreffen eines Events wartet. Dies kann ein Receive-Task wie im dargestellten Beispiel sein oder auch ein Timer-Event. Derjenige Pfad, dessen Event zuerst eintritt, wird ausgeführt. Danach eintreffende Events haben keine Bedeutung mehr für den Kontrollfluss oder die Pfadauswahl. Im Beispiel in Abbildung 18 wird also nach der Ausführung von Task 1 gewartet, welche Message zuerst eintrifft. Ist es die Message, auf welche Task 2 wartet, so wird dieser Pfad ausgeführt. Dasselbe gilt analog für Task 3 und seinen Pfad.

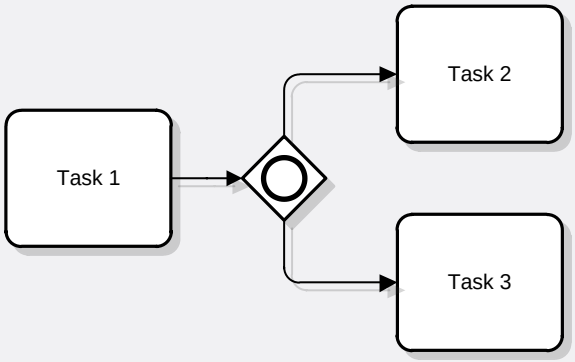


Abbildung 19 Beispiel für ein Inclusive Gateway

Eine weitere Gateway-Spezialisierung ist in Abbildung 19 in Form des Inclusive Gateways dargestellt. Ein Inclusive Gateway hat keine Limitierung hinsichtlich

der Anzahl der ausführbaren Pfade. Jeder ausgehende Pfad eines Inclusive Gateways ist mit einer eigenen Bedingung versehen. Ist die Bedingung erfüllt, wird der zugehörige Pfad unabhängig von der Ausführung der anderen Pfade ausgeführt. Es kann also durchaus sein, dass im vorliegenden Beispiel nach der Ausführung von Task 1 sowohl Task 2 als auch Task 3 ausgeführt werden. Ebenso kann einer der beiden Tasks ausgeführt werden. Es gibt außerdem die Möglichkeit, einen Default-Pfad zu definieren. Dieser wird nur ausgeführt, wenn keine Bedingung der anderen Pfade erfüllt ist.

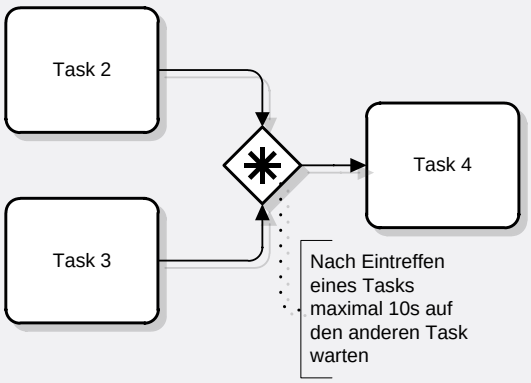


Abbildung 20 Beispiel für ein Complex Gateway

Eine sehr freie Form der Gateway-Modellierung bietet das Complex Gateway (Abbildung 20). Mit einem Complex Gateway können beliebige Semantiken umgesetzt werden, welche durch andere Gateways nicht erreicht werden können. Diese Semantiken werden durch Kommentare oder narrative Beschreibungen definiert. In Abbildung 20 wird beispielsweise nach der Beendigung eines Pfades (also Task 2 oder Task 3) maximal 10s darauf gewartet, ob auch der jeweils andere Pfad beendet wird. Dies könnte beispielsweise für den weiteren Verlauf in Task 4 noch zusätzliche Erkenntnisse bringen, welche gewinnbringend, aber nicht notwendig sind. Ein weiteres gängiges Beispiel für ein Complex Gateway ist die Modellierung eines 2-aus-3-Gateways. Wenn in einem zusammenführenden Gateway drei Pfade eingehen, so reicht die beendete Ausführung von zwei der drei Pfade, um mit dem Prozess nach dem Gateway fortzufahren. Allgemein gilt, dass bei Verwendung des Complex Gateways aufgrund seiner prinzipiell nicht vorhandenen Restriktionen immer eine möglichst genaue Beschreibung seiner Semantik gegeben werden sollte. Andernfalls sind Verständigungsschwierigkeiten zwischen den verschiedenen Personen zu erwarten, welche an der Modellierung eines Prozesses beteiligt sind.

Die letzte mögliche Gateway-Spezialisierung stellt das Parallel Gateway dar (Abbildung 21). Ein Parallel Gateway benötigt keine weiteren Bedingungen. Alle ausgehenden Pfade eines Parallel Gateways werden unabhängig voneinander ausgeführt. Das bedeutet, dass alle ausgehenden Pfade gleichzeitig oder in einer beliebigen Reihenfolge ausgeführt werden. Die Ausführung eines Prozesses nach einem zusammenfließend Parallel Gateway wird erst fortgesetzt, nachdem alle zusammenfließenden Pfade beendet wurden. In Abbildung 21 werden also nach der Ausführung von Task 1 sowohl Task 2 als auch Task 3 ausführbar. Erst

wenn beide Tasks beendet sind, wird mit der Ausführung von Task 4 fortgefahren.

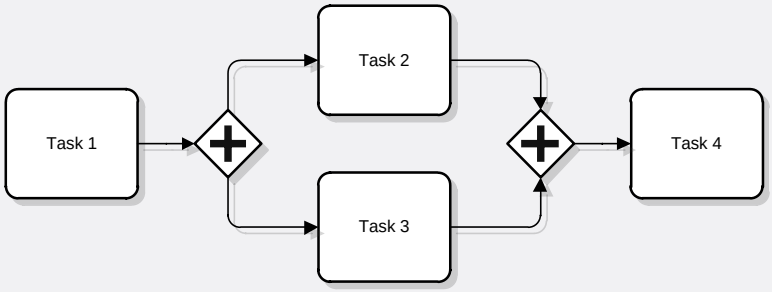


Abbildung 21 Beispiel für ein Parallel Gateway

3.1.6 Pools, Lanes und Messages

Soll in einem Prozessmodell erkennbar gemacht werden, wer für die Ausführung von Tasks oder Subprozessen verantwortlich ist, so bietet die BPMN 2.0 die Modellierungselemente Pool und Lane an. Ein Pool (Abbildung 22) stellt dabei einen Prozessbeteiligten dar. Dies kann beispielsweise ein Mensch, eine Abteilung oder auch ein Softwaresystem sein. Ein Pool kann dabei einen Prozess mit allen Elementen beinhalten, welche im Verlauf dieses Kapitels vorgestellt wurden. Bei Verwendung von Pools in einem Prozessidagramm dürfen Sequence Flows nur innerhalb eines Pools fließen.

Abbildung 22 Darstellung eines Pools

Weiterhin gibt es die Möglichkeit, Pools zu strukturieren und weiter zu untergliedern. Dafür werden Lanes verwendet (Abbildung 23). So kann beispielsweise ein Unternehmen (Pool) mit verschiedenen Abteilungen (Lanes) modelliert werden. In einem Prozessdiagramm können dann die Bestandteile wie Tasks und Subprozesse auf die jeweiligen Abteilungen in einem Unternehmen verteilt werden.

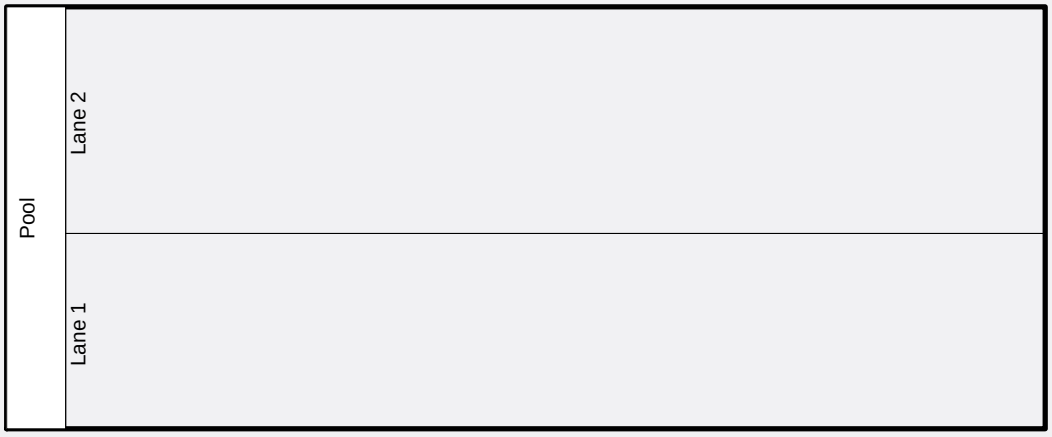


Abbildung 23 Lanes in einem Pool

Zwischen Pools darf in der BPMN 2.0 kein Sequence Flow modelliert werden. Zugrunde liegt die Vorstellung von unterschiedlichen Prozessbeteiligten, welche sich gegenseitig Nachrichten schicken. Dies wird durch Messages (Nachrichten) und einen Message Flow dargestellt. Abbildung 24 zeigt ein abgewandeltes Prozessdiagramm aus dem eingangs angegeben Buch von Thomas Allweyer. Darin wird ein Pool mit einem kompletten Prozess aufgezeigt (Zeitungsverlag) und ein Pool (Kunde), dessen interne Prozesse nicht weiter ausmodelliert wurden. Beide Varianten sind in BPMN 2.0 möglich. Zwischen diesen beiden Pools fließen Message Flows, welche die Kommunikation zwischen den beiden Prozessbeteiligten modellieren. Beim Zeitungsverlag werden Messages dabei direkt von Events oder Tasks gesendet oder empfangen, während die Message Flows beim Kunden nur in den Pool eingehen. Eine narrative Beschreibung neben einem Message Flow gibt dabei den Inhalt einer Message modellierungstechnisch wieder. Da der Prozess des Zeitungsverlags voll ausmodelliert in dieses Beispiel übernommen wurde, wird es als privates Prozessdiagramm bezeichnet.

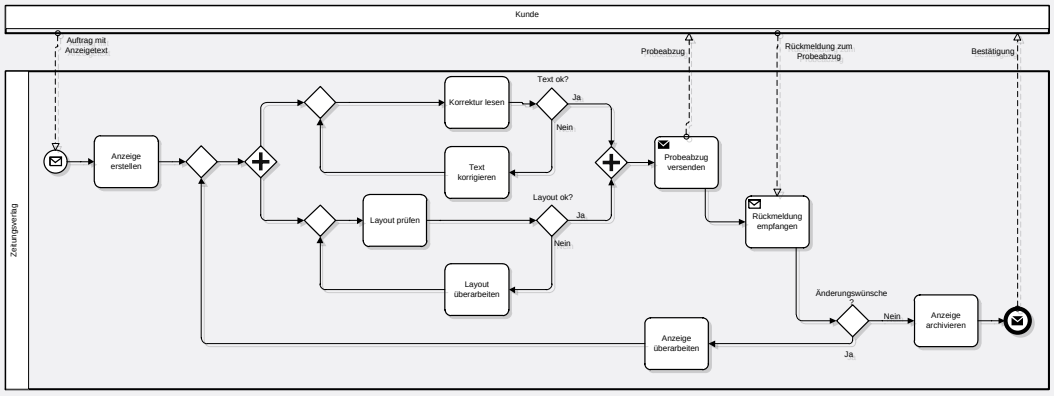


Abbildung 24 Beispiel eines Prozesses mit zwei Pools

3.1.7 Öffentliche Prozessdiagramme

Die bisher beispielhaft eingeführten Prozessdiagramme werden auch als private Prozessdiagrame bezeichnet. Sie modellieren einen Vorgang mit all seinen Schritten als Prozessmodell und geben so einen Überblick über Struktur und Abläufe. Insbesondere beim Austausch von Prozessmodellen zur Festlegung von Schnittstellen ist es aber nicht immer gewünscht, dass ein anderer Prozessteilnehmer alle internen Vorgänge in einem Prozessmodell sieht. Dies ist beispielsweise der Fall, wenn ein Prozess über zwei Unternehmen hinweg abläuft. Jedes Unternehmen ist als ein Pool modelliert. Das jeweils andere Unternehmen benötigt in einem solchen Szenario nicht das komplette Detailwissen hinsichtlich des eigenen Prozesses. Die Spezifikation der Messages und ihrer Reihenfolge ist prinzipiell ausreichend. In einem solchen Fall finden öffentliche Prozessdiagramme Anwendung.

Ein öffentliches Prozessdiagramm verfügt über die gleiche Auswahl an Modellierungselementen wie ein privates Prozessmodell. Häufig wird es aber vereinfacht und es werden nur intern relevante Schritte weggelassen, soweit sie den Message Flow nicht betreffen. Dies betrifft beispielsweise nur für den Sequence Flow relevante Gateways oder Tasks für die interne Verwaltung. Ein öffentliches Prozessdiagramm kann somit als eine schnittstellenorientierte Abstraktion eines privaten Prozessidagramms gesehen werden.

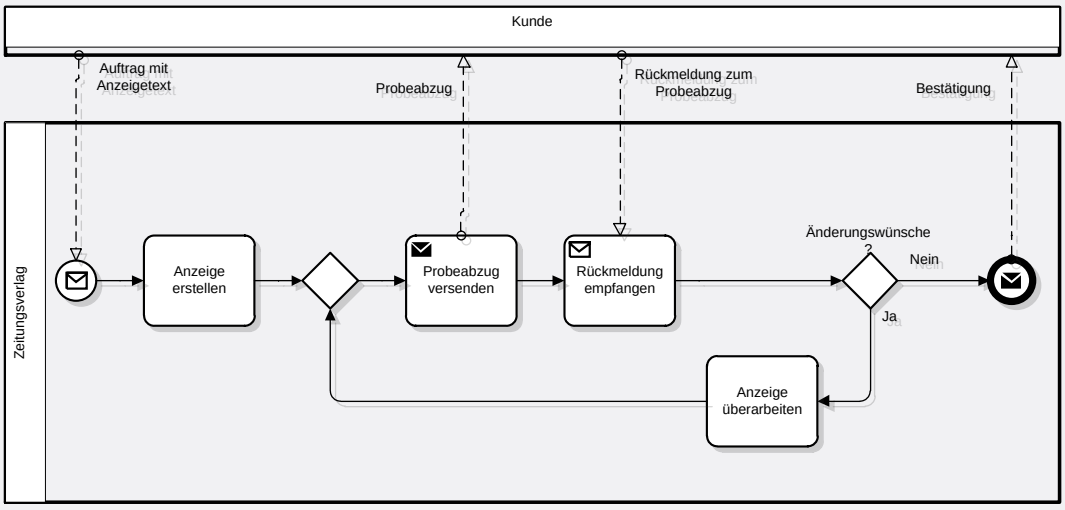


Abbildung 25 Öffentlicher Prozess

In Abbildung 25 ist der Prozess des Zeitungsverlags aus Abbildung 24 als öffentlicher Prozess dargestellt. Die interne Prüfung des Anzeigetexts wird dabei beispielsweise weggelassen. Sie beinhaltet keinen Message Flow und es ist für einen Kunden nicht notwendig zu wissen, welche internen Prüfungen eines Textes stattfinden.

Obwohl verständlicherweise häufig so wenige interne Details wie möglich Preis gegeben werden sollen, ist darauf zu achten, auch bei einem öffentlichen Prozessdiagramm noch einen nachvollziehbaren Prozess zu modellieren. Das Verstecken elementarer interner Schritte kann diesem Ziel abträglich sein und

ein unverständlicher Prozess erschwert die Definition einer Schnittstelle zwischen Prozessbeteiligten. Deshalb ist in jedem Einzelfall sorgfältig abzuwägen, welche Schritte in einem öffentlichen Prozess enthalten sein sollen und welche Schritte versteckt werden können.

3.2 Choreographiediagramme

Choreographiediagramme sind eine Diagrammart, welche in der BPMN 2.0 neu eingeführt wird. Der Fokus liegt dabei auf der Modellierung des Nachrichtenaustauschs zwischen zwei oder mehr Prozessbeteiligten. Die internen, privaten Prozesse der Prozessbeteiligten sind hingegen nur von untergeordneter Bedeutung.

3.2.1 Grundlagen der Choreographiediagramme

Der Grundgedanke bei Choreographiediagrammen ist die Modellierung von Message Exchange Patterns (MEPs). MEPs klassifizieren den Nachrichtenaustausch zwischen zwei Beteiligten in allgemeingültige Patterns. Die wichtigsten Patterns sind wohl das einfache Senden einer Nachricht (Single Message) oder das synchrone Senden einer Nachricht und das Warten auf eine Antwort (Request / Response). Weitere Patterns sind insbesondere in Zusammenhang mit Error Messages denkbar.

Zur Modellierung von MEPs führt die BPMN 2.0 das Modellierungselement Choreographietask ein. Ein Choreographietask modelliert exakt ein MEP in einem Nachrichtenaustausch zwischen zwei Beteiligten. Ein Choreographiediagramm erlaubt nun die strukturierte Modellierung mehrerer Choreographietasks, um einen Nachrichtenaustausch in Teilen oder seinem ganzen Umfang kompakt darzustellen.

3.2.2 Einführung des Choreographietasks

Choreographietasks sind ein neues Modellierungselement in der BPMN 2.0. Ein Choreographietask modelliert dabei ein MEP in einem Nachrichtenaustausch. Die gängigsten MEPs (Single Message, Receive / Reply) sind als Choreographietasks in Abbildung 26 dargestellt.

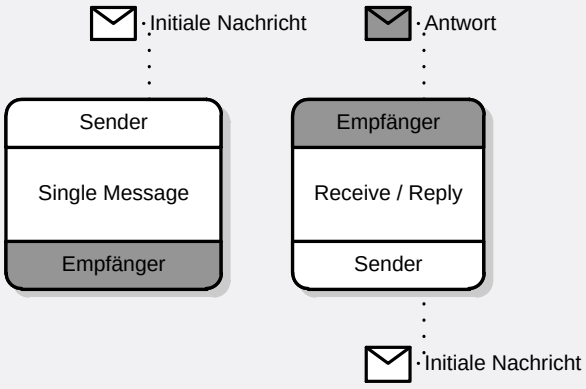


Abbildung 26 Choreographietasks

Da Choreographietasks einen Nachrichtenaustausch modellieren, existieren sie nur außerhalb von Pools. Sie bestehen üblicherweise aus 3 Komponenten:

- Einem zentralen Feld, welches den Namen des Choreographietasks enthält
- Jeweils einem Feld oberhalb und unterhalb des Namensfeldes, welches die Beteiligten am Nachrichtenaustausch beschreibt

Ob ein Beteiligter der Sender einer Nachricht oder der Empfänger ist, zeigt sich an der Farbgebung. Der Sender steht immer in einem weißen Feld, das Feld des Empfängers ist grau hinterlegt. Eine Nachricht, welche von einem Beteiligten gesendet wird, kann grafisch durch das Symbol eines Briefumschlags dargestellt und mit diesem assoziiert werden. Im Fall einer Single Message kann also maximal eine Nachricht dargestellt werden. Beim Receive / Reply können sowohl die initiale Nachricht als auch die Antwort darauf dargestellt werden.

Auch Choreographietasks können mit verschiedenen Markern versehen werden, welche Einfluss auf das Verhalten bei der Ausführung haben. In Abbildung 27 ist ein Choreographietask mit einem Loop Marker dargestellt. Ein Sender sendet also wiederholt eine Nachricht, solange eine bestimmte Bedingung erfüllt ist. Diese könnte in einem Diagramm beispielsweise durch einen Kommentar angegeben werden.

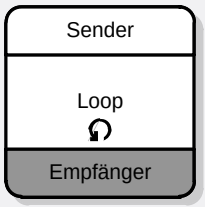


Abbildung 27 Choreographietask mit Loop Marker

In Abbildung 28 ist ein Choreographietask mit Multi-Instance Marker (Parallel) abgebildet. Dabei wird gleichzeitig eine Nachricht viele Male von einem Sender an einen Empfänger gesendet.

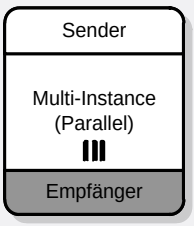


Abbildung 28 Choreographietask mit Multi-Instance Marker (Parallel)

Es mag wenig sinnvoll erscheinen, die gleiche Nachricht viele Male an einen Empfänger zu senden. Allerdings kann auch ein Empfänger als Multi-Instance deklariert werden (Abbildung 29). In diesem Fall wird eine Nachricht an verschiedene Empfänger gesendet, welche sich durch einen Oberbegriff zusammenfassen lassen. Als Beispiel dafür kann die Einholung eines Angebots

genannt werden. Ein potentieller Kunde kann beispielsweise eine Nachricht mit der Bitte um ein Angebot an verschiedene Anbieter schicken.

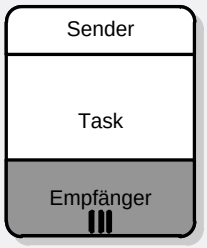


Abbildung 29 Multi-Instance Empfänger

Neben dem Choreographietask mit dem parallelen Multi-Instance Marker gibt es auch noch die Möglichkeit, Nachrichten sequenziell zu senden. Die entsprechende Darstellung ist in Abbildung 30 zu sehen.

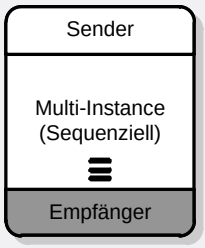


Abbildung 30 Choreographietask mit sequentiellem Multi-Instance Marker

Analog zu Subprozessen können auch Choreographietasks in verschiedene Subebenen untergliedert werden, um die Übersicht zu gewährleisten. Eine Subchoreographie (Abbildung 31) kann weitere Choreographiediagramme beinhalten und schafft so eine hierarchische Struktur zwischen den Choreographien. Erkennbar ist eine Subchoreographie an ihrem "+" im Namensfeld. Darüber hinaus gleicht die Darstellung im einfachsten Fall einem Choreographietask.

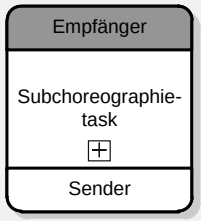


Abbildung 31 Subchoreographie

Zusätzlich zum Choreographietask besteht bei Subchoreographien die Möglichkeit, mehrere Sender oder Empfänger anzugeben. Da eine Subchoreographie mehrere Choreographietasks enthalten kann, mag eine Angabe mehrerer Beteiligter sinnvoll sein. Die Anzahl der anzugebenden Sender und Empfänger ist dabei prinzipiell nach oben hin unbeschränkt. Es besteht aber

keine Pflicht, alle Beteiligten an einer Subchoreographie anzugeben. Die Entscheidung über die Menge obliegt allein dem Modellierer.

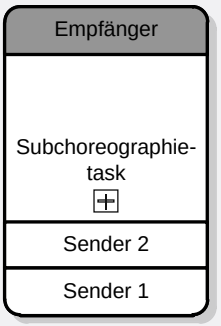


Abbildung 32 Subchoreographie mit zwei Sendern

In einer Subchoreographie können die gleichen Marker verwendet werden, welche bereits im Zusammenhang mit dem Choreographietask vorgestellt wurden. So kann beispielsweise auch eine Subchoreographie in einer Schleife laufen.

3.2.3 Sequence Flows in Choreographiediagrammen

Über den vorgestellten Choreographietask und Subchoreographien hinaus bietet BPMN 2.0 verschiedene weitere Elemente an, welche in einem Choreographiediagramm verwendet werden können. Diese wurden bereits in Zusammenhang mit den Prozessdiagrammen eingeführt und haben in einem Choreographiediagramm gleiche oder ähnliche Bedeutung.

Ein zentrales Element ist dabei der Sequence Flow. Dieser modelliert Abhängigkeiten zwischen Choreographietasks, Gateways und Events und schafft somit eine Struktur hinsichtlich der Reihenfolge (Abbildung 33).

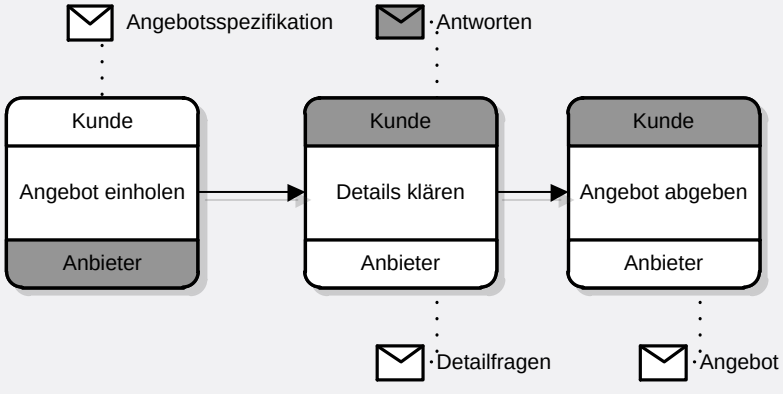


Abbildung 33 Sequence Flows in einem Choreographiediagramm

Als Grundregel für eine sinnvolle Modellierung von Choreographiediagrammen sollte dabei beachtet werden, dass ein Sender wissen muss, wann er eine Nachricht zu senden hat. Nur der initiale Sender kann seine Nachricht frei senden. Alle anderen Beteiligten sollten bereits durch den Empfang einer

Nachricht in eine Choreographie eingebunden worden sein, bevor das Senden einer Nachricht ausgelöst wird.

3.2.4 Events in Choreographiediagrammen

Auch Events können in einem Choreographiediagramm Verwendung finden. Sie haben dabei dieselbe Bedeutung wie in einem Prozessdiagramm. In Abbildung 34 werden beispielsweise ein None Start und ein End Event verwendet.

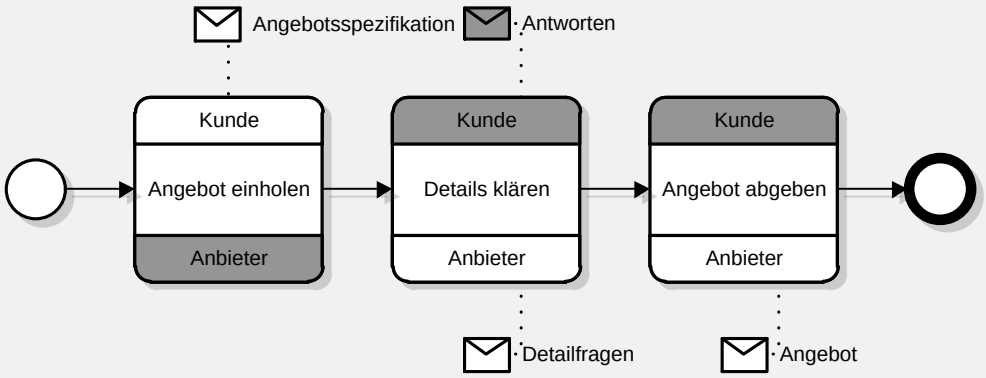


Abbildung 34 Events in einem Choreographiediagramm

In Choreographiediagrammen sind nicht alle Spezialisierungen von Events erlaubt. Die folgenden Start Events dürfen in einem Choreographiediagramm verwendet werden:

- Ein **None Start Event** wird als allgemeines Start Event ohne weitere Spezifikation verwendet.
- Ein **Timer Start Event** startet einen Nachrichtenaustausch zu einem bestimmten Zeitpunkt. Es wird dabei vorausgesetzt, dass alle Beteiligten sich auf eine einheitliche Uhrzeit geeinigt haben.
- Mit einem **Conditional Start Event** wird modelliert, dass der initiale Sender von einem Conditional Start Event zum Start des Nachrichtenaustauschs angestoßen wird.
- Ein **Signal Start Event** kann einen Nachrichtenaustausch starten, wenn alle Beteiligten das Signal empfangen können. Eine Signalquelle muss dabei nicht zwangsweise modelliert werden.
- Mit einem **Multiple Start Event** können verschiedene Timer oder Signal Start Events definiert werden, von welchen ein eintreffendes Event ausreicht, um eine Choreographie zu starten.

Weiterhin sind verschiedene Intermediate Events zur Verwendung in einem Choreographiediagramm vorgesehen:

- Ein **None Intermediate Event** stellt rein grafisch einen bestimmten Zeitpunkt in einer Choreographie dar, welcher hervorgehoben werden

soll. Er hat keine weiteren Auswirkungen auf das Verhalten einer Choreographie.

- Ein **Message Intermediate Event** kann nur als Boundary Event in einem Choreographietask verwendet werden. Dort kann es am Empfängerfeld angehängt werden und so den Empfang einer unplanmäßigen Nachricht modellieren.
- Ein **Timer Intermediate Event** kann sowohl im normalen Sequence Flow als auch als Boundary Event verwendet werden. Je nach Verwendung mit Bestimmung eines absoluten Zeitpunkts oder Definition einer relativen Zeit nach einer vorangegangenen Aktion müssen nicht alle Beteiligten über eine synchronisierte Uhrzeit verfügen oder den Timer berücksichtigen.
- Ein **Cancel Intermediate Event** kann als unterbrechendes Boundary Event an einem Sender oder Empfänger modelliert werden. Es fängt ein potentielles Cancel, welches in einem privaten Prozess geworfen wird.
- Ein **Compensation Intermediate Event** kann an einem der Beteiligten modelliert werden. Es fängt ein entsprechendes Event und reagiert darauf.
- Ein **Conditional Intermediate Event** kann sowohl im normalen Sequence Flow als auch als Boundary Event modelliert werden. Es symbolisiert das Warten auf eine Änderung von Daten. Diese Daten müssen einem Choreographietask bekannt sein; folglich müssen sie auf einer bereits verschickten Nachricht basieren.
- Ein **Link Intermediate Event** kann im normalen Sequence Flow verwendet werden und modelliert Sprünge zwischen verschiedenen Prozessteilen.
- Ein **Signal Intermediate Event** kann sowohl im normalen Sequence Flow als auch als Boundary Element in Form eines fangenden Events verwendet werden. Es kann also kein Signal gesendet werden, aber es kann auf das Eintreffen eines Signals gewartet werden, das an anderer Stelle gesendet wird.
- Ein **Multiple Intermediate Event** kann wiederum verschiedene Trigger definieren, von denen einer ausreicht, um ein Event zu starten.

Insbesondere von den End Events gibt es nur sehr wenige erlaubte Spezialisierungen in einem Choreographiediagramm:

- Ein **None End Event** verdeutlicht grafisch das Ende einer Choreographie. Es hat keine Auswirkungen, da die Choreographie eigentlich schon mit dem Versand der letzten Nachricht beendet wurde.
- Obwohl ein **Terminate End Event** keine wirklichen Folgen in einer Choreographie hat, kann es einer Verdeutlichung dienen. Allerdings müssen die notwendigen Daten für das Beenden bereits mit der letzten vorhergehenden Nachricht übermittelt werden, da eine Choreographie keinerlei Möglichkeiten hat, ein Prozessdiagramm zu beenden.

3.2.5 Gateways in Choreographiediagrammen

Die bekannten Spezialisierungen von Gateways können auch in einem Choreographiediagramm Anwendung finden. Da in einer Choreographie – im Gegensatz zu einem Prozessdiagramm – jedoch nicht immer alle Daten zugänglich sind, müssen bei ihrer Verwendung verschiedene Dinge berücksichtigt werden.

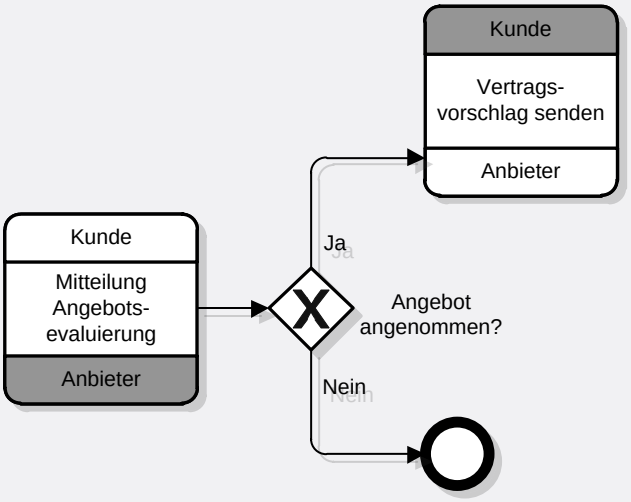


Abbildung 35 Ein Exclusive Gateway in einem Choreographiediagramm

Ein **Exclusive Gateway** modelliert wie in einem Prozessdiagramm eine Aufspaltung des Kontrollflusses, bei der genau ein Pfad weiter ausgeführt wird (Abbildung 35). Die Daten, auf welcher die Entscheidung hinsichtlich der Pfadauswahl beruht, müssen bei Verwendung in einem Choreographiediagramm allerdings Bestandteil einer vorher gesendeten Nachricht sein. Meist beruht eine Entscheidung auf Inhalten der zuletzt gesendeten Nachricht.

Das **Event-Based Gateway** verhält sich ebenfalls ähnlich wie in Prozessdiagrammen. Es wird ein Pfad auf Basis des ersten eintreffenden Events ausgewählt. Jeder mögliche Pfad muss dabei mit einem Timer Intermediate Event oder einem Signal Intermediate Event beginnen. Weitere Eventtypen sind nicht erlaubt.

Inclusive Gateways haben auch in Choreographiediagrammen die Eigenschaft, dass jeder Pfad ausgeführt wird, dessen Bedingung zutrifft. Auch hier kommt die Restriktion hinzu, dass Bedingungen auf Daten aufbauen müssen, welche vorher in Messages versendet wurden. Außerdem gilt die Bedingung, dass der Sender der ersten Nachricht in einem Pfad gleichzeitig der Empfänger der letzten Nachricht vor dem Gateway sein muss. Er muss den Start eines Pfades kundtun. Die anderen Beteiligten können von der Entscheidung nichts wissen, da ihnen die Daten fehlen.

Parallel Gateways können wie in Prozessdiagrammen verwendet werden. Da sie nicht von Bedingungen abhängen, sondern alle Pfade ausgeführt werden, muss auch keine Rücksicht auf Daten und Messages genommen werden.

Complex Gateways sind auch in Choreographiediagrammen als sehr freie Form eines Gateways zu sehen. Sollte eine Bedingung auf Daten basieren, ist jedoch auch hier zu beachten, dass diese bekannt sein müssen.

3.2.6 Weitere Elemente in Choreographiediagrammen

Über die vorgestellten Elemente hinaus können noch die aus Prozessdiagrammen bekannten Elemente zur Kommentierung eines Prozessmodells verwendet werden. Hierbei sind insbesondere Annotations und Groups zu nennen.

Nicht verwendet werden können Data Objects oder Repositories. Da konzeptuell keine zentrale Datenhaltung in Choreographien vorgesehen ist, sind derartige Mechanismen obsolet.

Abschließend sei noch angemerkt, dass ein Choreographiediagramm einen Nachrichtenaustausch zwischen Beteiligten nicht zwangsweise vollständig modellieren muss. Es können prinzipiell auch Teile ignoriert werden. Ob ein Choreographiediagramm vollständig sein soll, wird durch ein entsprechendes Attribut des Diagramms definiert.

3.3 Kollaborationsdiagramme

Auch Kollaborationsdiagramme sind eine Diagrammart, welche in der BPMN 2.0 neu eingeführt werden. Kollaborationsdiagramme kombinieren im Wesentlichen die Elemente von Prozessdiagrammen und Choreographiediagrammen. Während Prozessdiagramme den Fokus auf einen Prozess und seine Abläufe richten, betonen Choreographiediagramme den Nachrichtenaustausch zwischen verschiedenen Beteiligten. Kollaborationsdiagramme schließlich integrieren beide Sichten.

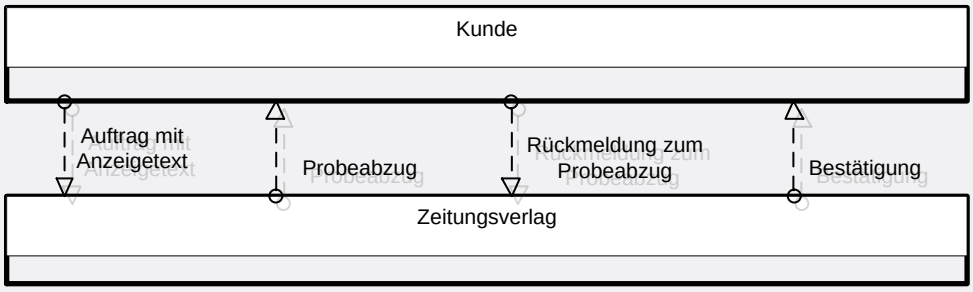


Abbildung 36 Kollaborationsdiagramm mit Black Box-Pools

Das Ziel ist dabei die Darstellung der Kommunikation zwischen zwei oder mehreren Beteiligten. Die Beteiligten werden dabei meist als Pools modelliert. Die Pools können dabei als Black Box (Abbildung 36) oder mit einem privaten oder öffentlichen Prozessdiagramm dargestellt werden. In Abbildung 36 wird die Kommunikation durch Message Flows dargestellt. Über den Inhalt der einzelnen Nachrichten gibt dabei ein kurzer Kommentar Aufschluss. Sender und Empfänger sind durch die Bindung an die Pools klar definiert.

In Abbildung 37 ist derselbe Prozess in einem Kollaborationsdiagramm dargestellt, welcher den Nachrichtenaustausch durch eine Choreographie modelliert. Weiterhin wird der öffentliche Prozess des Zeitungsverlags dargestellt und der Message Flow durch die direkte Bindung an Tasks und Events genauer spezifiziert. In einem Kollaborationsdiagramm sind sogar Multi-Instance-Pools erlaubt, welche die Kommunikation mit mehreren Instanzen eines Beteiligten modellieren.

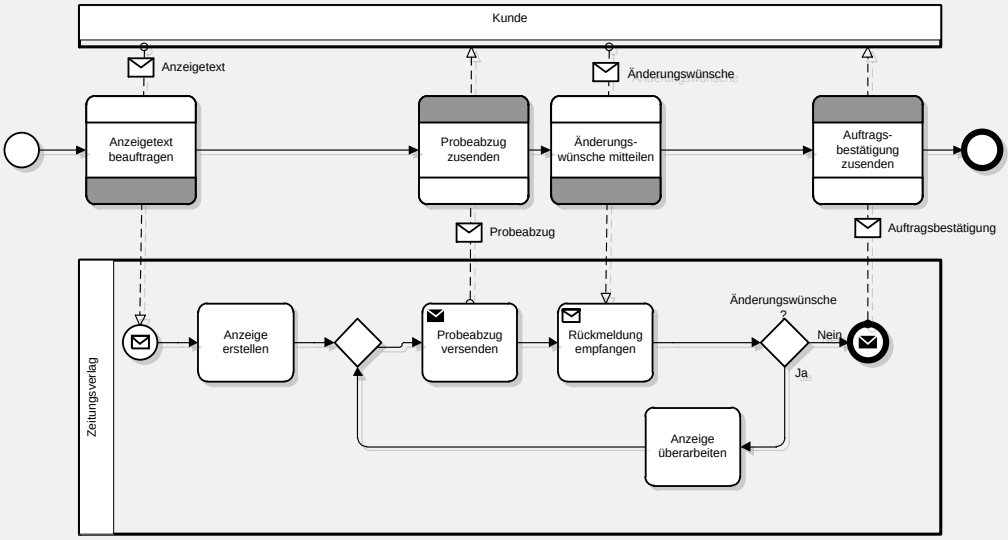


Abbildung 37 Kollaborationsdiagramm mit Choreographie und öffentlichem Prozess

Zusammengefasst dienen Kollaborationsdiagramme also dazu, sich einen Überblick auf verschiedenen Ebenen zu verschaffen. Je nachdem, welcher Aspekt eines Prozesses besonders betont werden soll, werden verschiedene Details hervorgehoben oder versteckt. Durch die gegebenen Abstraktionsmöglichkeiten können Kollaborationsdiagramme somit auf sehr hohem oder sehr niedrigem Detaillevel verwendet werden.

3.4 Konversationsdiagramme

Die letzte noch fehlende Diagrammart in BPMN 2.0 sind die Konversationsdiagramme. Auch diese Diagrammart wurde in der BPMN 2.0 neu eingeführt. Ein Konversationsdiagramm bietet einen stark abstrahierten Überblick über die Kommunikationsvorgänge in einer Domäne. Ziel ist es dabei nicht mehr, die einzelnen Nachrichten zu modellieren, sondern die übergeordneten Zusammenhänge, welche sich aus dem Nachrichtenaustausch ergeben.

Conversation Node

Abbildung 38 Conversation Node

Als neue Modellierungselemente führt das Konversationsdiagramm die Conversation Node (Abbildung 38) und den Conversation Link ein. Eine Conversation Node fasst eine Menge von Messages und Message Flows zusammen, welche einem bestimmten Ziel dienen. Conversation Links verknüpfen Conversation Nodes mit den Beteiligten (modelliert durch Pools), welche an einem Nachrichtenaustausch beteiligt sind.

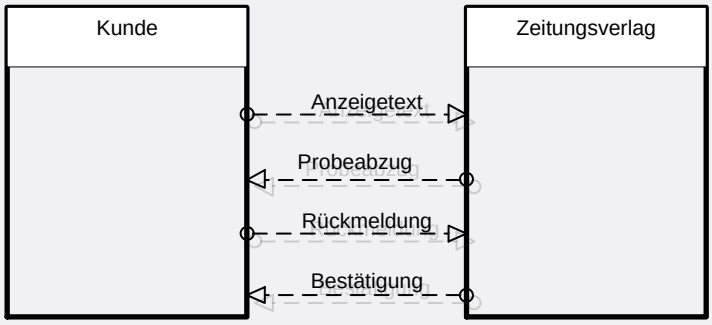


Abbildung 39 Beispielhafter Nachrichtenaustausch

Der in Abbildung 39 dargestellte Nachrichtenaustausch wird beispielsweise in Abbildung 40 durch eine Conversation Node dargestellt. Es gibt prinzipiell keine Begrenzung für eine Anzahl von Message Flows, die zu einer Conversation Node zusammengefasst werden. Entscheidend ist das Kriterium, dass sie einem gemeinsamen, formulierbaren Ziel dienen.

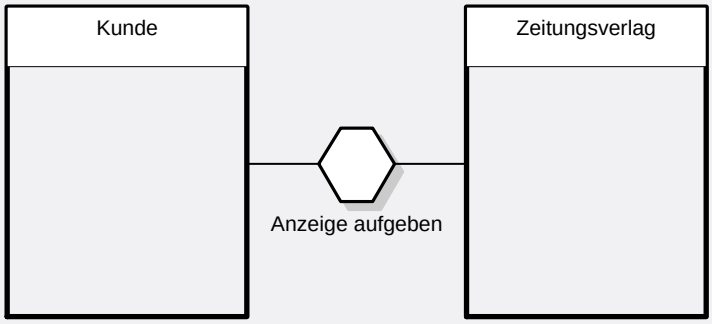


Abbildung 40 Verwendung einer Conversation Node

Abbildung 41 zeigen die Verwendung von Conversation Nodes in einem größeren Beispiel. Es ist leicht vorstellbar, dass die Modellierung der einzelnen Message Flows der Conversation Nodes dieses Diagramm unübersichtlich macht und die Nachvollziehbarkeit senkt.

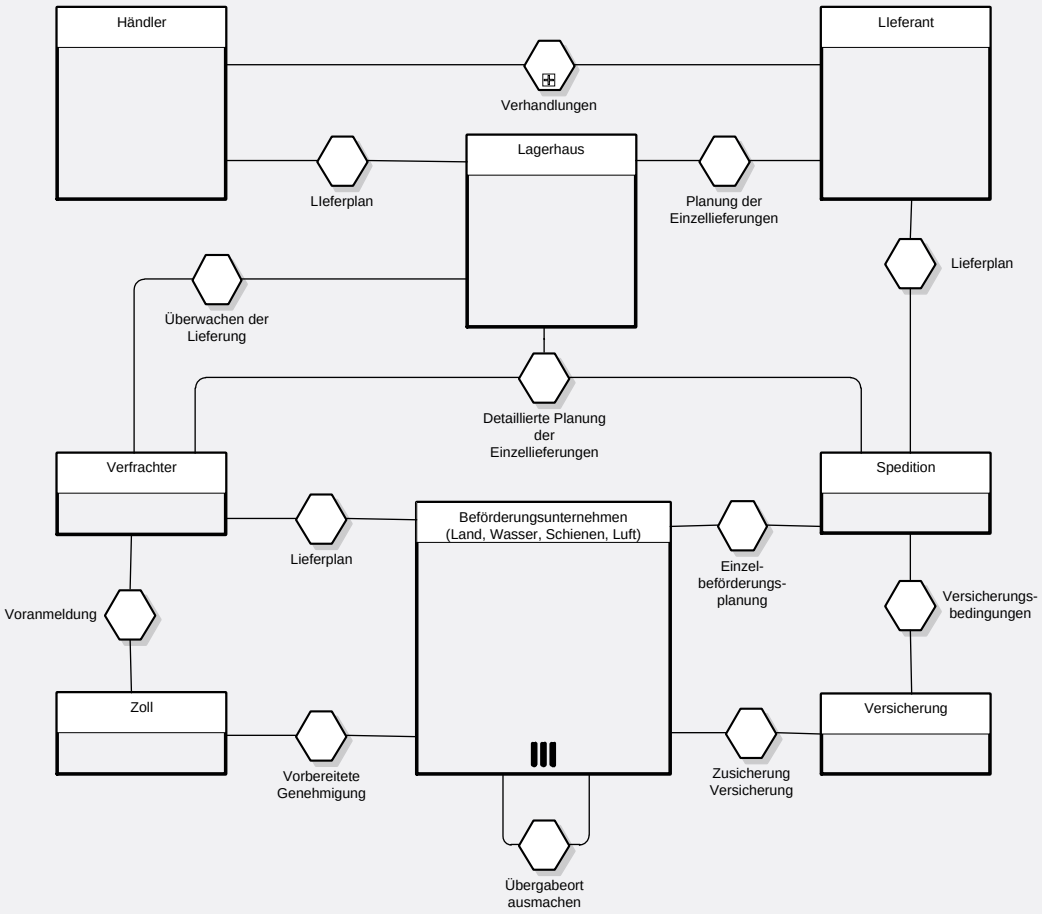


Abbildung 41 Conversation Nodes an einem größeren Beispiel

Auch Conversation Nodes können untergliedert und hierarchisch strukturiert werden. Dazu bietet BPMN 2.0 die Möglichkeit der Sub-Conversation. Eine solche Sub-Conversation ist in Abbildung 41 bei den Verhandlungen zwischen einem Händler und einem Lieferanten dargestellt.

4 Zusammenfassung

Als Fazit zur BPMN 2.0 kann gezogen werden, dass die Möglichkeiten im Vergleich zur BPMN 1.2 deutlich erweitert wurden. Viele Erweiterungen werden in diesem Tutorial nicht einmal behandelt, da sie vor allem auf die direkte Ausführbarkeit von BPMN 2.0 abzielen. Hinsichtlich der Modellierung kamen vor allem neue Diagrammarten und die damit einhergehenden Modellierungselemente hinzu. Diese bieten das Potential zu abstrakteren Modellen und machen die BPMN vor allem auch für eine Modellierung in großen Anwendungen nutzbar. Dennoch hat bereits die Erfahrung mit der UML gezeigt, dass nicht alle Diagrammarten gleich stark genutzt werden. Welche Diagrammarten also wirklich Einzug in die aktive Prozessmodellierung finden, muss sich erst noch zeigen.

Weiterhin bleibt noch abzuwarten, ob sich die BPMN 2.0 als Modellierungssprache selbst durchsetzen kann. Vieles spricht dafür. Vor allem große Hersteller von Tools bieten Unterstützung und integrieren die BPMN 2.0 bereits in ihre BPMS. Auch die erweiterten Möglichkeiten beheben viele Kritikpunkte, welche an die Vorgänger gerichtet wurden. Allerdings ist mit diesen Möglichkeiten auch die Einfachheit in der Modellierung verloren gegangen. Die BPMN ist nun nicht mehr so schnell und einfach zu verstehen. Es gibt beispielsweise ein Message Event, welches eine Message empfängt, und einen Receive Task. Beide setzen dasselbe Konzept um. Dies könnte auch zu Verwirrung und schließlich zur Ablehnung bei den Business Experten führen. Vielleicht setzt sich eine Teilmenge der BPMN 2.0-Modellierungselemente und –diagramme in der Geschäftswelt als inoffizieller Standard durch?

Obwohl es nicht Bestandteil dieses Tutorials war, ist die direkte Ausführbarkeit von Prozessmodellen auf den ersten Blick sicherlich ein Pluspunkt. Allerdings müssen dafür extrem technische Attribute gesetzt werden, was ein Business Experte im Normalfall nicht umsetzen kann. Weiterhin ist nicht garantiert, dass die BPMN 2.0 auf allen Systemen gleich ausgeführt wird. Im Gegenteil, es wird von Experten wie Bruce Silver sogar erwartet, dass jeder Hersteller ein eigenes Subset umsetzt: "No engine vendor is going to support every possible BPMN 2.0 element and attribute called out in the metamodel. And I'm not saying just in the first release. Not ever. In that sense, BPMN 2.0 is not a self-contained execution language like BPEL is." (http://www.brsilver.com/2009/11/19/bpmn-vs-bpel-are-we-still-debating-this/) Damit wäre die BPMN 2.0 zwar hinsichtlich der Modellierung ein Standard. In der Ausführung wäre man aber wiederum an einen Hersteller gebunden.

Insgesamt bietet die BPMN 2.0 dennoch enorme Chancen. Viele Sachverhalte können modelliert werden und die saubere Definition der Modellierungselemente schafft in vielen Fällen eine semantische Klarheit, welche die Vorgänger nicht bieten konnten. Somit wird sich die BPMN 2.0 wohl in der einen oder anderen Form als "Standard in Anwendung" etablieren und durchsetzen.

5 Kontakt

iTransparent GmbH | Business Process Architects

Bad Brückenauer Str. 23
90427 Nürnberg

Tel.:+49(0)911 93754499
www.iTransparent.de