

BPMN - Business Process Modeling Notation - Version 1.2

Basiselemente [B]			
Ablaufknoten	Ereignis	Aktivität	Gateway
Verbindungsobjekte	Sequenzfluss	Nachrichtenfluss	Assoziation
Artefakte	Datenobjekt	Kommentar	Gruppe
Schwimmbahnen	Pools und Lanes		

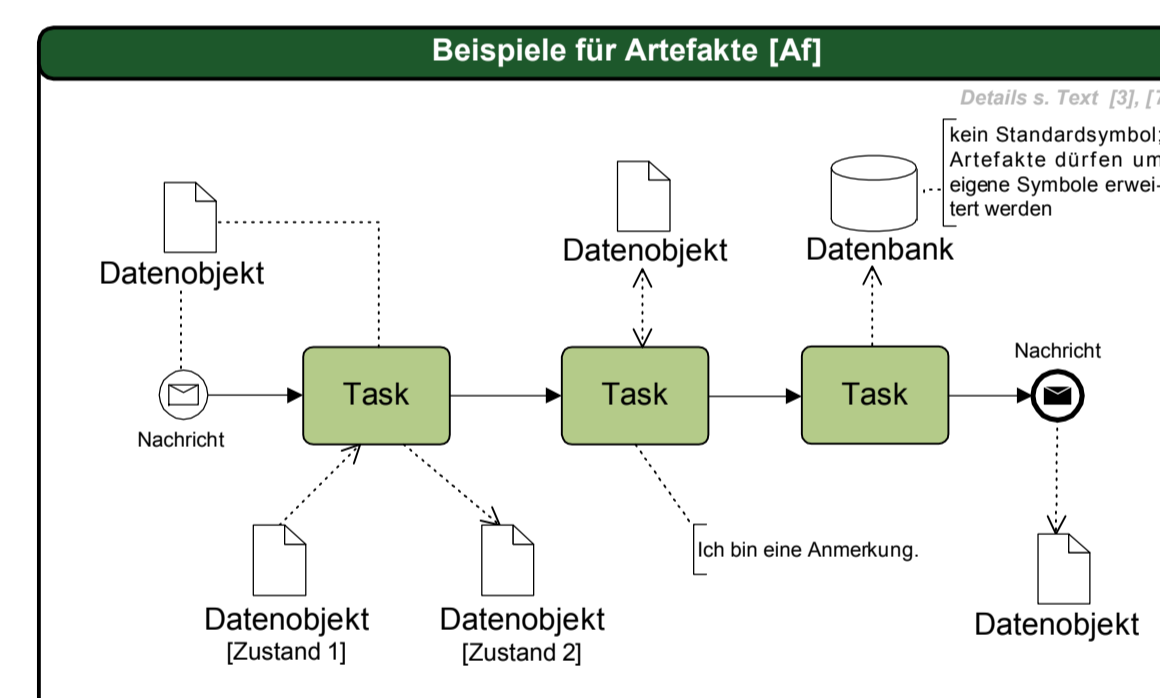
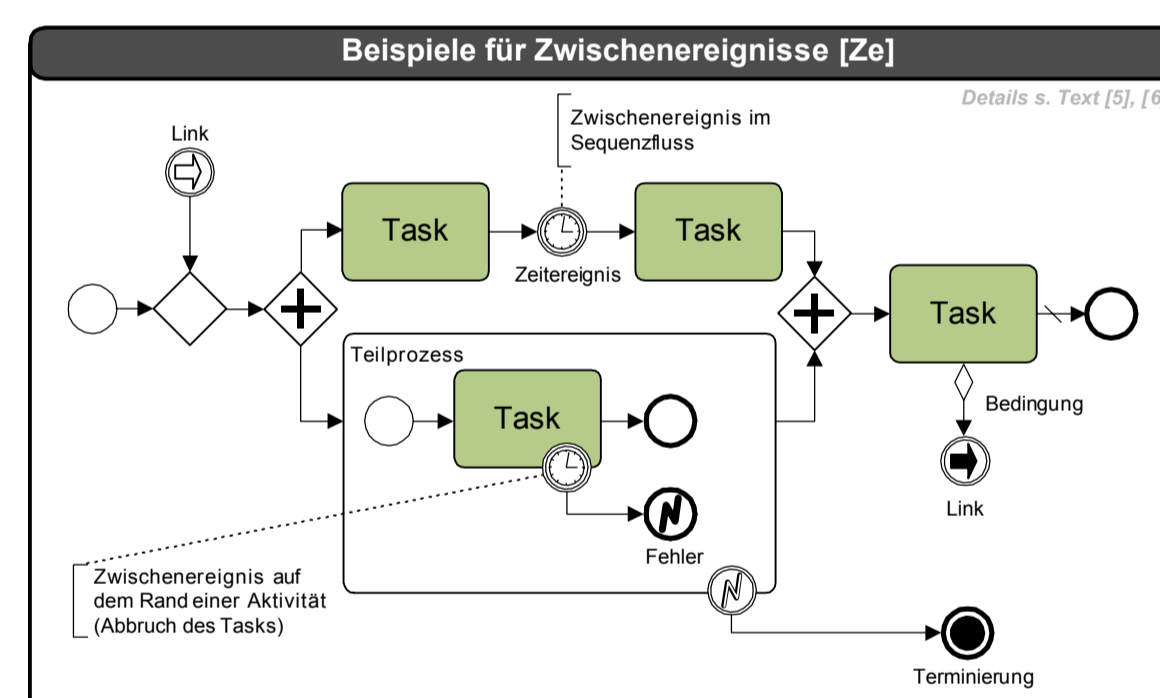
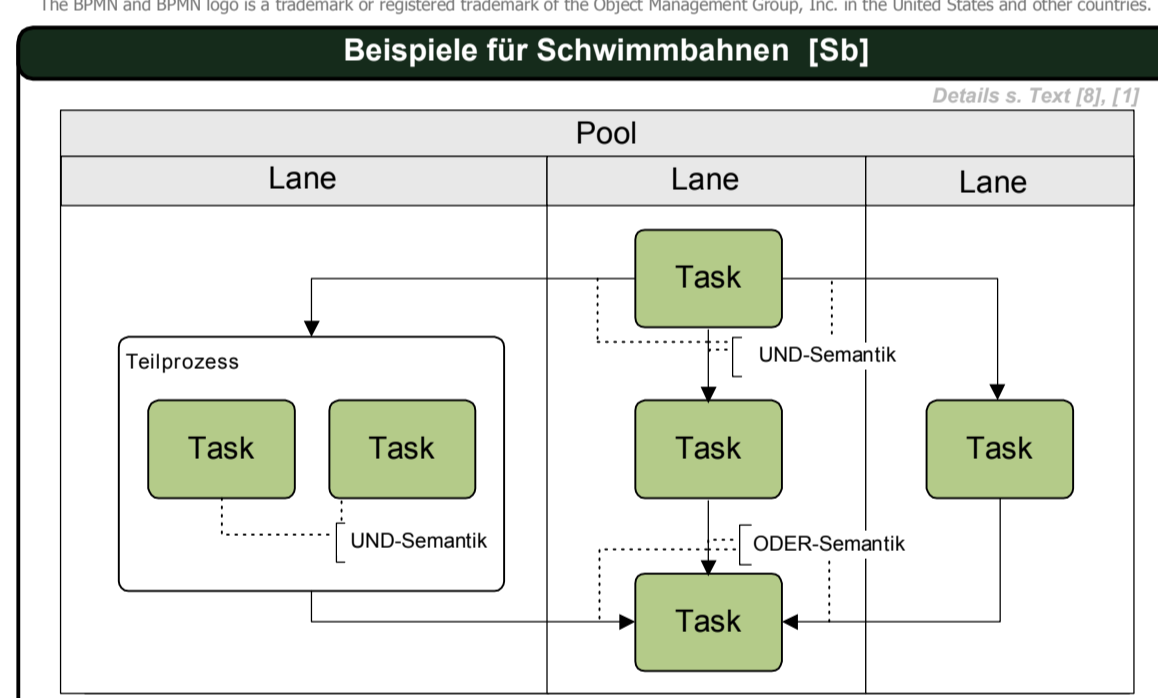
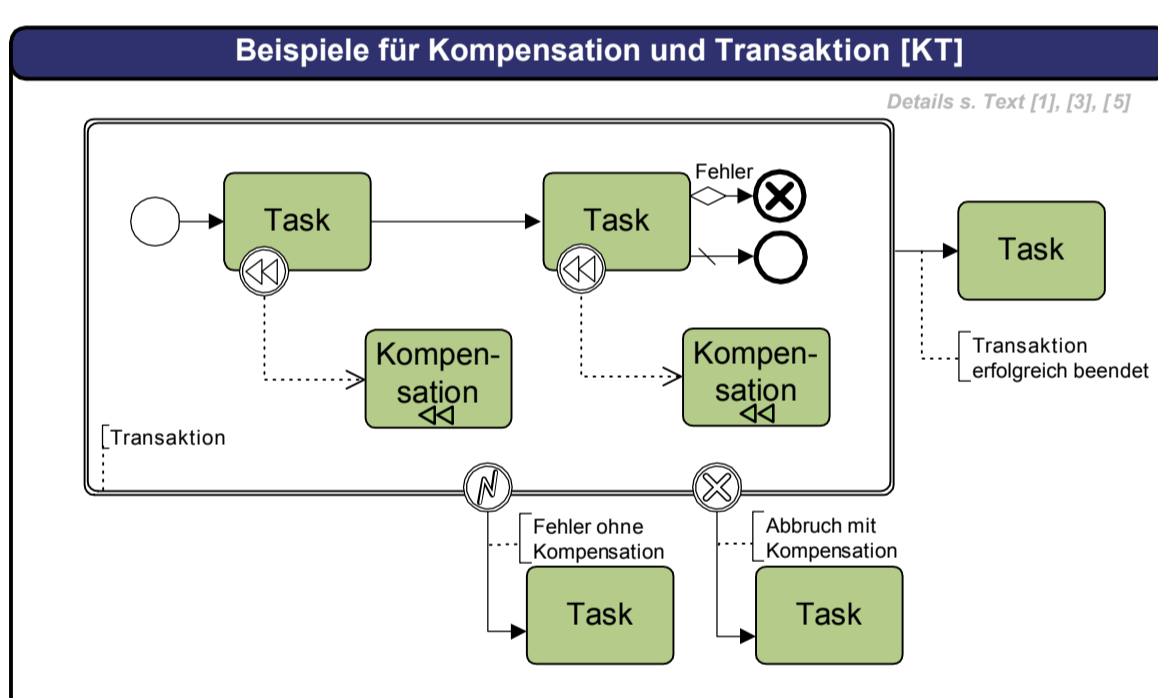
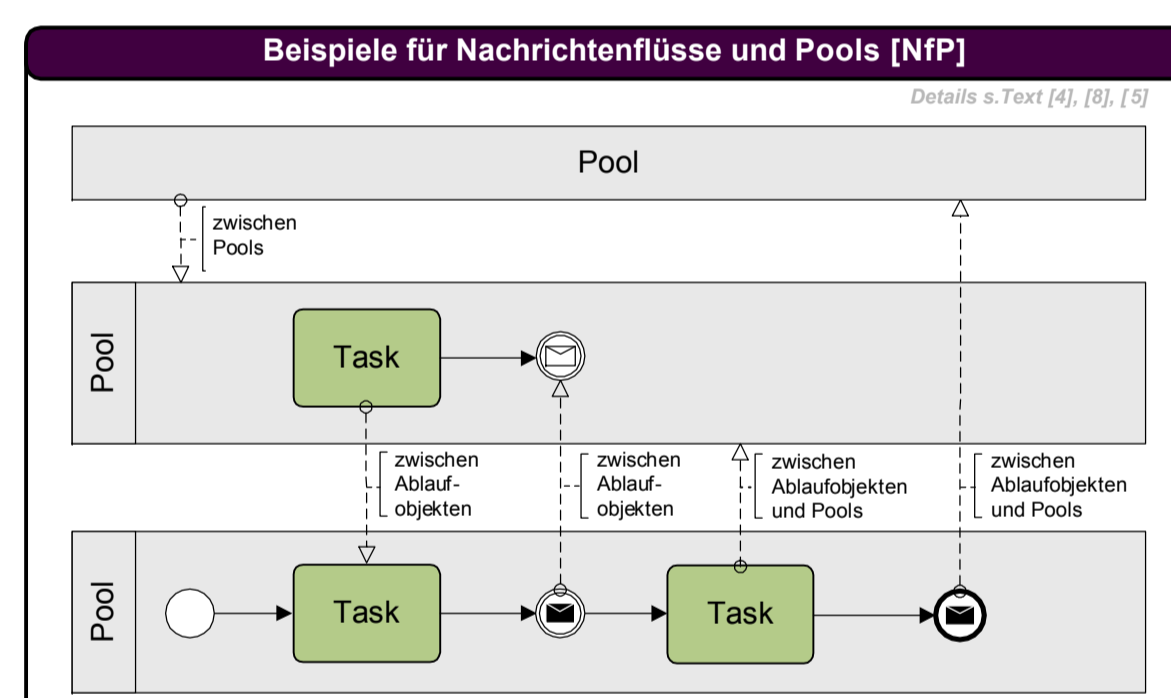
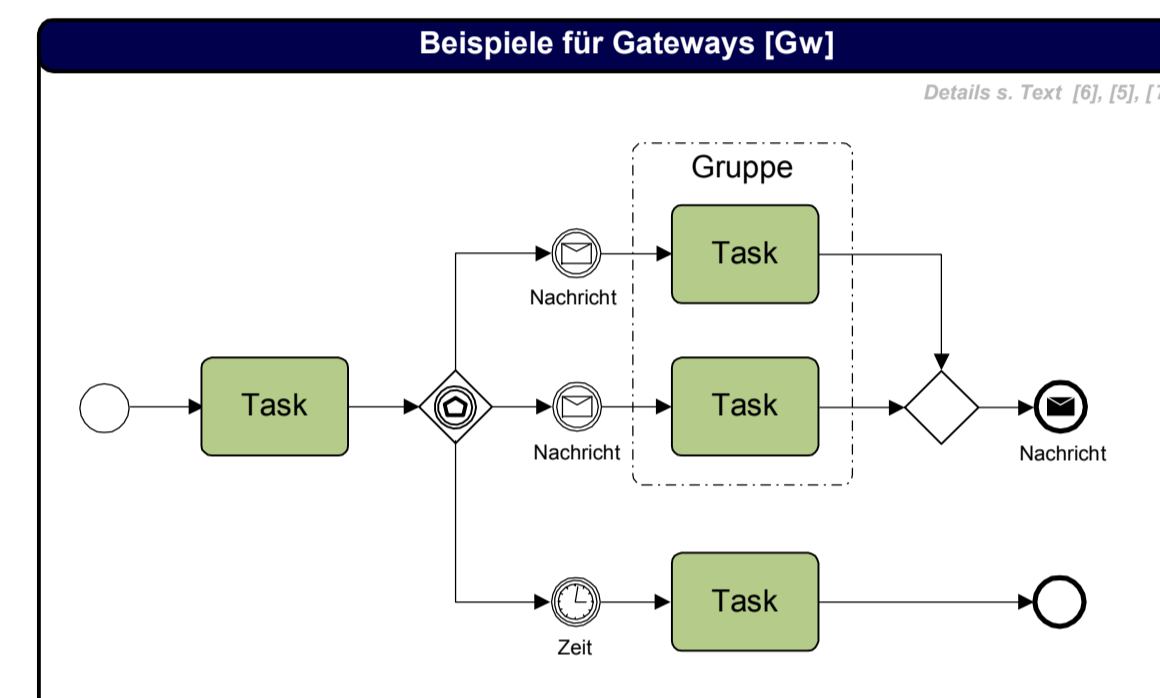
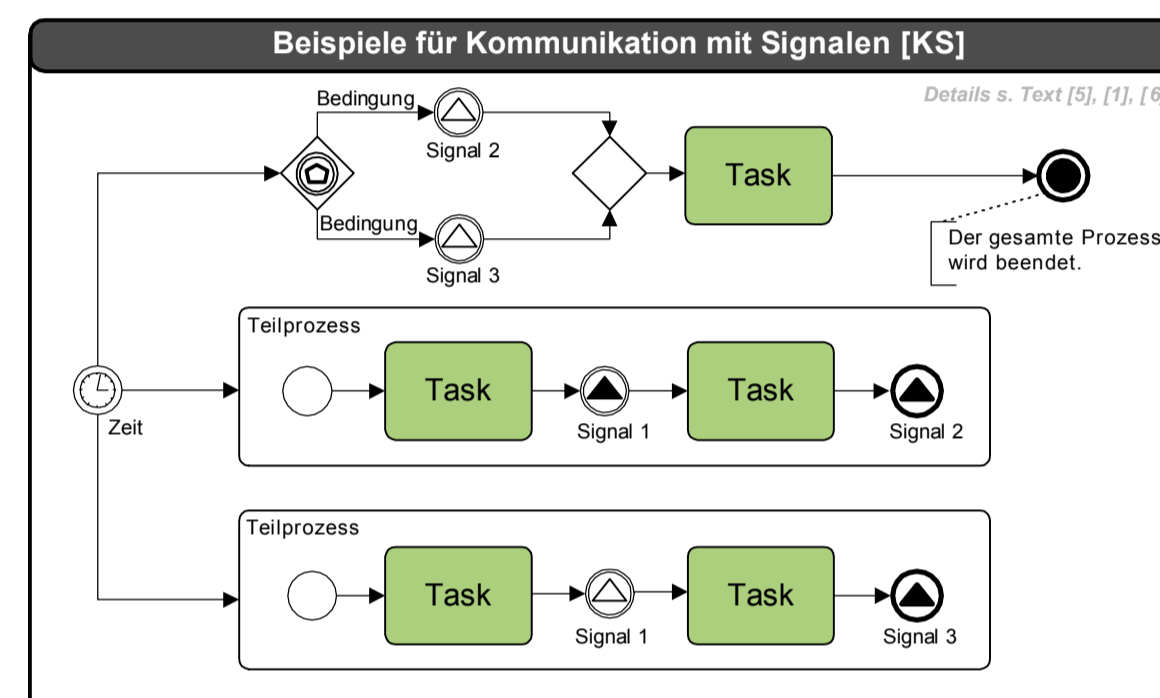
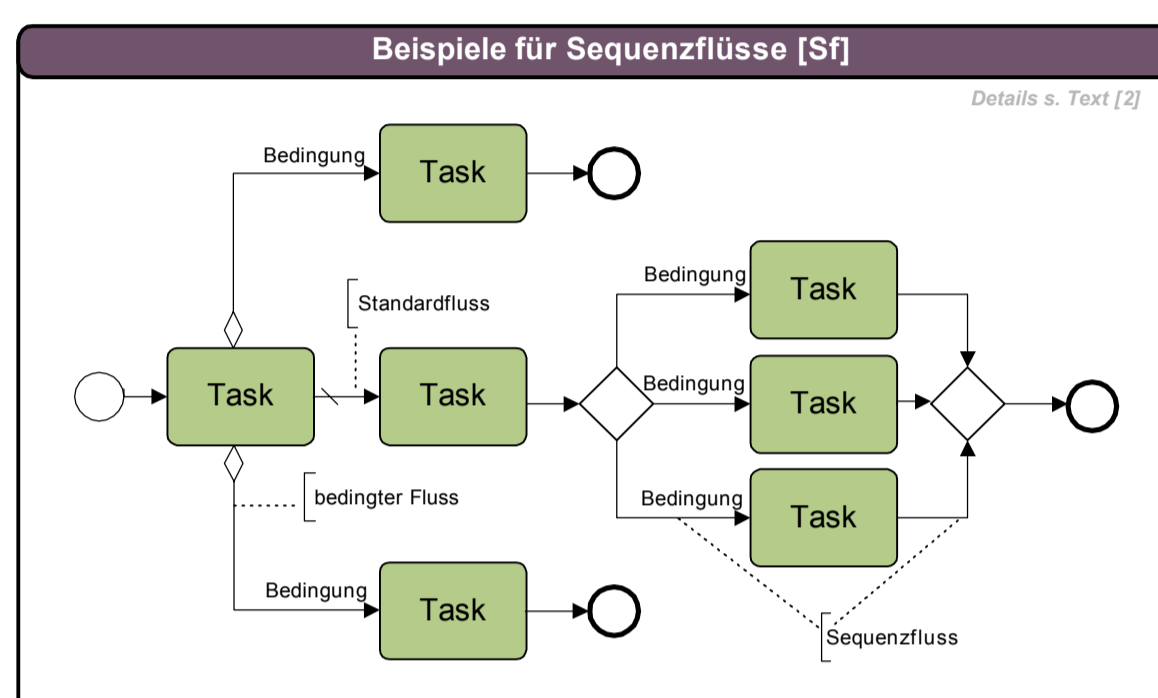
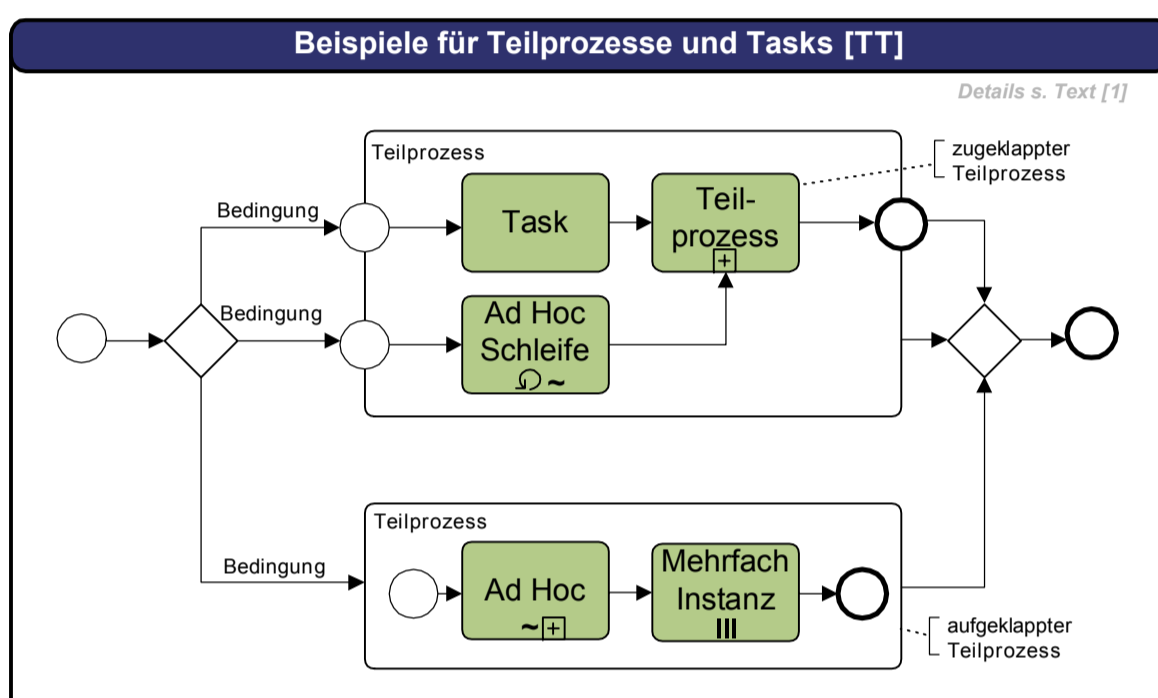
Aktivitäten [A]		
	Task	Teilprozess
	Schleife	
	Mehrfach-Instanz	
	Kompensation	
	Ad hoc-Teilprozess	
	Transaktion	

Verbindungsregeln für Sequenzflüsse [VfS]						
Von / Nach	○	□	◇	○	○	○
○	→	→	→	→	→	→
□	→	→	→	→	→	→
◇	→	→	→	→	→	→
○	→	→	→	→	→	→

Ereignisse [E]			
	Start empfangend	Zwischenereignis empfangend	Ende auslösend
Zeitereignis	🕒	🕒	🕒
Nachricht	✉️	✉️	✉️
Signal	📡	📡	📡
Fehler	⚠️	⚠️	⚠️
Abbruch	✖️	✖️	✖️
Kompensierung	↩️	↩️	↩️
Bedingung	📄	📄	📄
Link	🔗	🔗	🔗
Mehrfachtyp	📄	📄	📄
Terminierung	🛑	🛑	🛑

Gateways [G]		
datenbasiertes exklusives Gateway	⊗	◇
ereignisbasiertes Gateway	⊕	◇
paralleles Gateway	⊕	◇
einschließendes Gateway	⊕	◇
komplexes Gateway	⊕	◇

Verbindungsregeln für Nachrichtenflüsse [VNF]						
Von / Nach	✉️	✉️ (Pool)	✉️	✉️	✉️	✉️
✉️	→	→	→	→	→	→
✉️ (Pool)	→	→	→	→	→	→
✉️	→	→	→	→	→	→
✉️	→	→	→	→	→	→



The BPMN and BPMN logo is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries.

[1] Aktivitäten

Eine **Aktivität (Activity)** beschreibt einen Arbeitsschritt in einem Prozess, wie z.B. das Schreiben einer E-Mail. Hat eine Aktivität mehrere eingehende Sequenzflüsse, so wird die Aktivität für jeden eingehenden Token^{*} einmal ausgeführt (Oder-Semantik). Hat eine Aktivität mehrere ausgehende Sequenzflüsse, erhält jeder Sequenzfluss einen Token^{*} (Und-Semantik) (s. Abb. [Sb]).

Ein **Task (Task)** ist eine atomare Aktivität innerhalb eines Prozesses, d.h. sie wird nicht weiter detailliert.

Ein **Teilprozess (Subprocess)** ist eine Aktivität, die mit Hilfe von weiteren Aktivitäten konkretisiert wird. Es gibt zwei Darstellungsformen: Den zugeklappten Teilprozess, der die Details verbirgt, und den aufgeklappten Teilprozess, der den Sequenzfluss darstellt und die Details des aufgeklappten Teilprozesses (nicht mit den inneren Elementen) verbunden werden. Die inneren Elemente können jedoch direkt über Nachrichtenflüsse miteinander kommunizieren (s. Abb. [TT]).

Eine **Schleife (Activity Looping)** wiederholt die Aktivität solange, bis die Schleifenbedingung erfüllt ist. Zulässige Attribute der Schleife sind: Schleifenbedingung, Schleifenanzahl, Maximum und Abfrage (before oder after).

Eine **Mehrfach-Instanz (Multi Instance)** wiederholt – parallel oder sequentiell – alle enthaltenen Aktivitäten mit unterschiedlichen Daten (z.B. für jede Position eines Auftrags) bis die zugehörige Schleifenbedingung erfüllt ist.

Die Aktivitäten innerhalb einer **Ad Hoc-Aktivität (Ad Hoc Activity)** werden in beliebiger Reihenfolge ausgeführt.

Mehrere zusammengefasste Aktivitäten, die gemeinsam ein unterteilbares Ganzes ergeben, bezeichnet man als eine **Transaktion (Transaction)**. Eine Transaktion kann auf drei Arten beendet werden:

- (1) Erfolgreich
- (2) Abbruch mit einer anschließenden Kompensation (Abbruchereignis)
- (3) Fehlerhaft ohne Kompensation (Fehlerereignis)

Bricht die Transaktion ab, benachrichtigt das Transaktionsprotokoll alle Beteiligten. Diese können auf den Abbruch ggf. mit einer Kompensation reagieren (s. Abb. [KT]).

Eine **Kompensation (Compensation)** beschreibt die Schritte, die bei einer abgebrochenen Transaktion rückgängig gemacht werden müssen. Eine Kompensation darf keinen eingehenden oder ausgehenden Sequenzfluss besitzen und liegt innerhalb der Transaktion. Eine gerichtete Assoziation verbindet das Kompensationsereignis auf dem Rand der Aktivität mit der Kompensation.

Ein **Sequenzfluss (Sequence Flow)** verbindet die Ablaufknoten (Aktivitäten, Gateways und Ereignisse) miteinander und beschreibt somit deren Reihenfolge im Prozess. Bei einem konkreten Ablauf führt ein Token^{*} entlang des Sequenzflusses von Ablaufknoten zu Ablaufknoten.

Der **bedingte Sequenzfluss (Conditional Sequence Flow)** ist eine verkürzte Schreibweise des einschließenden Gateways (s. Abb. [Sf]) und darf nur direkt an eine Aktivität modelliert werden. Der Token läuft entlang des bedingten Sequenzflusses, wenn dessen Bedingung zutrifft. Hat eine Aktivität mehrere bedingte Sequenzflüsse, so erhält jeder Sequenzfluss einen Token^{*}, dessen Bedingung erfüllt ist.

Der **Standardsequenzfluss (Default Sequence Flow)** wird eingesetzt, um bei einer Verzweigung sicherzustellen, dass der Ablauf bzw. Token^{*} nicht hängen bleibt. Ist keine der zugehörigen Verzweigungsbedingungen erfüllt, läuft der Ablauf entlang des Standardsequenzflusses (s. Abb. [Sf]). Dieser kann direkt nach einem exklusiven Gateway, einem einschließenden Gateway oder einer Aktivität modelliert werden.

Die **Assoziation (Association)** verbindet Anmerkungen oder Datenobjekte mit anderen Diagrammelementen. Die Assoziation kann ungerichtet, gerichtet oder beidseitig gerichtet sein (s. Abb. [Af]) und hat keine Einflüsse auf den Sequenzfluss. Eine Assoziation vom Typ **Kompensation** ist eine gerichtete Assoziation, die eine Aktivität mit der zugehörigen Kompensation verbindet.

[3] Assoziationen

Eine **Assoziation (Association)** verbindet Anmerkungen oder Datenobjekte mit anderen Diagrammelementen. Die Assoziation kann ungerichtet, gerichtet oder beidseitig gerichtet sein (s. Abb. [Af]) und hat keine Einflüsse auf den Sequenzfluss. Eine Assoziation vom Typ **Kompensation** ist eine gerichtete Assoziation, die eine Aktivität mit der zugehörigen Kompensation verbindet.

[4] Nachrichtenflüsse

Der **Nachrichtenfluss (Message Flow)** beschreibt die Kommunikation zwischen zwei Prozessbeteiligten, die in unterschiedlichen Pools liegen (s. Abb. [VNF]). Nachrichtenflüsse dürfen ausschließlich zur Kommunikation zwischen Pools eingesetzt werden. Die Reihenfolge des Nachrichtenflusses ergibt sich implizit aus dem modellierten Sequenzfluss. Der Nachrichtenfluss hat jedoch **keinen** Einfluss auf den Sequenzfluss.

Zulässig sind (s. Abb. [NFP]) Nachrichtenflüsse zwischen:

- (1) zwei separaten Pools,
- (2) zwischen zwei Ablaufknoten in separaten Pools, und
- (3) zwischen einem Pool und Ablaufknoten eines anderen Pools.

Nachrichtenflüsse zwischen Objekten innerhalb desselben Pools sind **nicht zulässig!**

[5] Ereignisse

Ein **Ereignis (Event)** kann einen Prozess starten, verzögern, unterbrechen oder beenden. Es gibt drei Typen von Ereignissen: Startereignis, Zwischenereignis und Endereignis.

Ein **Startereignis (Start Event)** markiert einen Prozessbeginn. Wird der Prozess gestartet, erzeugt das Startereignis einen Token^{*}, der entlang von Sequenzflüssen und Ablaufknoten läuft. Ein Prozess kann mehrere Startereignisse enthalten.

Ein **Endereignis (End Event)** markiert, wo der Prozess – oder genauer, wo ein Pfad des Prozesses – beendet wird. Jedes Endereignis konsumiert einen Token^{*} und kann ein Ergebnis erzeugen. Ein Prozess kann mehrere Endereignisse enthalten.

Start- und Endereignisse sind optional. Hat ein Prozess kein Startereignis, starten alle Aktivitäten, die keinen eingehenden Sequenzfluss besitzen. Bei einem Prozess ohne Endereignis konsumieren alle Aktivitäten ohne ausgehenden Sequenzfluss einen Token^{*}. Wenn Startereignisse modelliert werden, so müssen auch Endereignisse modelliert werden und umgekehrt.

Ein **Zwischenereignis (Intermediate Event)** tritt zwischen dem Start und Ende eines Prozesses auf.

Ein **Signal (Signal)** wird von einem Teilnehmer gesendet und kann von mehreren Teilnehmern empfangen werden (broadcast). Signale ermöglichen eine Kommunikation über Poolgrenzen und Prozessebenen hinweg, z.B. zwischen zwei Teilprozessen oder zwischen Vater- und Teilprozess. Da der Sequenzfluss den Rand eines Teilprozesses nicht kreuzen darf, können Signale auch genutzt werden, um die Kommunikation zwischen zwei Teilprozessen zu synchronisieren (s. Abb. [KS]). Signale müssen nicht paarweise benutzt werden.

In einer **Aktivität** kann ein **Fehlerereignis (Error Event)** z. B. durch eine Applikation oder durch das Ergebnis eines Teilprozesses ausgelöst werden. Der Fehler führt gewöhnlich zum Abbruch oder zur Korrektur des Ablaufs (s. Abb. [Ze]).

[6] Gateways

Innerhalb einer Transaktion bricht ein Endereignis vom Typ **Abbruch (Cancel)** die gesamte Transaktion ab. Hierdurch wird das Abbruch-Zwischenereignis auf dem Rand der Transaktion sowie das Transaktionsprotokoll gestoppt.

Ein Ereignis vom Typ **Bedingung (Conditional)** definiert Regeln, die erfüllt sein müssen, damit der Prozess beginnen bzw. fortfahren kann.

Ein Ereignis vom Typ **Link (Link)** beschreibt eine Sprungmarke. Sie wird eingesetzt, um lange Sequenzflüsse zu verkürzen oder um einen Ablauf auf der nächsten Seite fortzuführen (s. Abb. [Ze]).

Ein Ereignis vom Typ **Mehrfach (Multiple)** erlaubt verschiedene Kombinationen der Auslöser: Nachricht, Zeit, Fehler, Bedingung oder Signal. Sobald einer der Auslöser eintritt oder empfangen wird, reagiert der Prozess.

Bei einem Ereignis vom Typ **Terminierung (Terminate)** werden alle Aktivitäten augenblicklich beendet und somit der gesamte Prozess beendet. Ein Teilprozess kann seine übergeordneten Prozesse (Elternteilprozesse) durch ein Terminierungsereignis nicht beenden (s. Abb. [Ze]).

Ein **auslösendes Zwischenereignis (Throwing Intermediate Event)** erzeugt ein Ereignis und der Ablauf wird unmittelbar fortgesetzt. Zum Beispiel wird eine Nachricht versendet und der Prozess läuft direkt weiter.

Das **empfangende Zwischenereignis (Catching Intermediate Event)** unterbricht den Ablauf solange, bis das Ereignis eintrifft. Beispielsweise wird ein Ablauf solange unterbrochen bis eine Nachricht empfangen wird.

Ein **Zwischenereignis** kann an zwei Stellen modelliert werden: Im Sequenzfluss oder auf dem Rand einer Aktivität.

- (1) Ein **Zwischenereignis im Sequenzfluss** kann ein Ereignis auslösen oder auf ein empfangendes Ereignis warten (s. Abb. [Ze]). Zwischenereignisse im Sequenzfluss dürfen nur genau einen eingehenden und einen ausgehenden Sequenzfluss besitzen.
- (2) Ein **Zwischenereignis auf dem Rand einer Aktivität** wartet auf ein eintrittendes Ereignis (Auslöser) und bricht bei Eintritt die Aktivität ab (s. Abb. [Ze]).

Mit Hilfe eines **Auslösers (Trigger)** (z. B. einer Nachricht, Zeit etc.) kann kenntlich gemacht werden, unter welchen Umständen das Ereignis eintritt (z.B. Eingang einer Nachricht). Es gibt zehn Auslöser: Zeit, Nachricht, Signal, Fehler, Abbruch, Kompensation, Bedingung, Link, Mehrfach und Terminierung (s. Tabelle [E]).

Bei einem Ereignis vom Typ **Zeit (Time)** wird der Ablauf solange unterbrochen, bis der definierte Zeitpunkt erreicht ist. Ein Zeitauslöser kann ein spezifisches Zeitereignis (Datum, Zeit) oder ein zyklisch wiederkehrendes Ereignis (z.B. So., 8-15 Uhr) sein.

Ein Ereignis vom Typ **Nachricht (Message)**, z.B. ein Brief, kann von einem Prozess-Teilnehmer gesendet oder empfangen werden. Nachrichten können immer nur zwischen Teilnehmern aus unterschiedlichen Pools zur Kommunikation eingesetzt werden (nicht zur Kommunikation zwischen Lanes innerhalb eines Pools).

Ein **Signal (Signal)** wird von einem Teilnehmer gesendet und kann von mehreren Teilnehmern empfangen werden (broadcast). Signale ermöglichen eine Kommunikation über Poolgrenzen und Prozessebenen hinweg, z.B. zwischen zwei Teilprozessen oder zwischen Vater- und Teilprozess. Da der Sequenzfluss den Rand eines Teilprozesses nicht kreuzen darf, können Signale auch genutzt werden, um die Kommunikation zwischen zwei Teilprozessen zu synchronisieren (s. Abb. [KS]). Signale müssen nicht paarweise benutzt werden.

Ein **Gateway (Gateway)** steuert, wie der Sequenzfluss aufgeteilt oder zusammengeführt wird. Es gibt fünf Typen von Gateways: Kombinationen aus ein- und ausgehenden Flüssen in nur einem Gateway sind erlaubt.

Das **datenbasierte exklusive Gateway (Data Based Exclusive Gateway)** entscheidet abhängig von den ausgehenden Bedingungen, wie der Sequenzfluss aufgeteilt wird. Es erhält genau einer der alternativen Sequenzflüsse den Token^{*} und setzt den Ablauf fort. Wenn keine Bedingung erfüllt ist, bleibt der Token^{*} am Gateway hängen. Bei der Zusammenführung wird der Token direkt weitergeleitet (s. Abb. [Gw]).

Das **ereignisbasierte exklusive Gateway (Event Based Exclusive Gateway)** entscheidet aufgrund von eintretenden Ereignissen, wie der Sequenzfluss aufgeteilt wird. Der Token^{*} läuft entlang des Sequenzflusses, dessen Ereignis als erstes eintrifft. Die Ereignisse (bzw. Receive Tasks) werden direkt nach dem ereignisbasierten exklusiven Gateway modelliert (s. Abb. [Gw]).

Ein **paralleles Gateway (Parallel Gateway)** teilt den Sequenzfluss in zwei oder mehrere parallele Flüsse auf. Die Anzahl der erzeugten Token^{*} entspricht der Anzahl der ausgehenden Flüsse. Bei einer Zusammenführung synchronisiert das parallele Gateway alle eingehenden Sequenzflüsse und wartet bis alle Token^{*} eingetroffen sind. Deren Anzahl entspricht hierbei der Anzahl der eingehenden Sequenzflüsse. Parallele Gateways haben keine Bedingung (s. Abb. [Ze]).

Das **einschließende Gateway (Inclusive Gateway)** verhält sich ähnlich, wie das parallele Gateway. Allerdings werden nur die Sequenzflüsse bei der Aufteilung in einen parallelen Fluss mit eingeschlossen, deren Bedingung erfüllt ist. Bei der Synchronisierung wird nur auf die eingehenden Sequenzflüsse gewartet, die aktiv sind.

Beim **komplexen Gateway (Complex Gateway)** kann der Modellierer selber eine Verzweigungsbedingung definieren, die das Verhalten am Gateway festlegt. Komplexe Gateways werden immer dann eingesetzt, wenn sich das Verhalten mit einem der anderen Gatewaytypen nicht beschreiben lässt.

Die **Artefakte (Artifacts)** bieten die Möglichkeit neben der Ablaufreihenfolge zusätzliche Informationen, wie z. B. benötigte Geschäftsobjekte, festzuhalten. Es gibt drei Arten von Artefakten: Datenobjekte, Kommentare und Gruppen.

Ein **Datenobjekt (Data Object)** ist ein Eingangs- bzw. Ergebnisobjekt einer Aktivität, wie z. B. ein Auftrag und kann um Zustände erweitert werden (s. Abb. [Af]). Datenobjekte beeinflussen den Sequenz- oder Nachrichtenfluss nicht direkt.

Eine **Anmerkung (Text Annotation)** ist eine textuelle Beschreibung, die über eine Assoziation mit jedem Diagrammelement verbunden werden kann (s. Abb. [Af]).

Eine **Gruppe (Group)** fasst mehrere Elemente zusammen, die logisch zusammengehören (s. Abb. [Gw]). Der Sequenzfluss eines Prozesses wird durch eine Gruppe nicht beeinflusst, da die Gruppe nicht mit dem Nachrichten- oder Sequenzfluss verbunden werden darf. Gruppen können über Pools und Lanes laufen.

[7] Artefakte

Die **Artefakte (Artifacts)** bieten die Möglichkeit neben der Ablaufreihenfolge zusätzliche Informationen, wie z. B. benötigte Geschäftsobjekte, festzuhalten. Es gibt drei Arten von Artefakten: Datenobjekte, Kommentare und Gruppen.

Ein **Datenobjekt (Data Object)** ist ein Eingangs- bzw. Ergebnisobjekt einer Aktivität, wie z. B. ein Auftrag und kann um Zustände erweitert werden (s. Abb. [Af]). Datenobjekte beeinflussen den Sequenz- oder Nachrichtenfluss nicht direkt.

Eine **Anmerkung (Text Annotation)** ist eine textuelle Beschreibung, die über eine Assoziation mit jedem Diagrammelement verbunden werden kann (s. Abb. [Af]).

Eine **Gruppe (Group)** fasst mehrere Elemente zusammen, die logisch zusammengehören (s. Abb. [Gw]). Der Sequenzfluss eines Prozesses wird durch eine Gruppe nicht beeinflusst, da die Gruppe nicht mit dem Nachrichten- oder Sequenzfluss verbunden werden darf. Gruppen können über Pools und Lanes laufen.

Das **datenbasierte exklusive Gateway (Data Based Exclusive Gateway)** entscheidet abhängig von den ausgehenden Bedingungen, wie der Sequenzfluss aufgeteilt wird. Es erhält genau einer der alternativen Sequenzflüsse den Token^{*} und setzt den Ablauf fort. Wenn keine Bedingung erfüllt ist, bleibt der Token^{*} am Gateway hängen. Bei der Zusammenführung wird der Token direkt weitergeleitet (s. Abb. [Gw]).

Das **ereignisbasierte exklusive Gateway (Event Based Exclusive Gateway)** entscheidet aufgrund von eintretenden Ereignissen, wie der Sequenzfluss aufgeteilt wird. Der Token^{*} läuft entlang des Sequenzflusses, dessen Ereignis als erstes eintrifft. Die Ereignisse (bzw. Receive Tasks) werden direkt nach dem ereignisbasierten exklusiven Gateway modelliert (s. Abb. [Gw]).

Ein **paralleles Gateway (Parallel Gateway)** teilt den Sequenzfluss in zwei oder mehrere parallele Flüsse auf. Die Anzahl der erzeugten Token^{*} entspricht der Anzahl der ausgehenden Flüsse. Bei einer Zusammenführung synchronisiert das parallele Gateway alle eingehenden Sequenzflüsse und wartet bis alle Token^{*} eingetroffen sind. Deren Anzahl entspricht hierbei der Anzahl der eingehenden Sequenzflüsse. Parallele Gateways haben keine Bedingung (s. Abb. [Ze]).

Das **einschließende Gateway (Inclusive Gateway)** verhält sich ähnlich, wie das parallele Gateway. Allerdings werden nur die Sequenzflüsse bei der Aufteilung in einen parallelen Fluss mit eingeschlossen, deren Bedingung erfüllt ist. Bei der Synchronisierung wird nur auf die eingehenden Sequenzflüsse gewartet, die aktiv sind.

Beim **komplexen Gateway (Complex Gateway)** kann der Modellierer selber eine Verzweigungsbedingung definieren, die das Verhalten am Gateway festlegt. Komplexe Gateways werden immer dann eingesetzt, wenn sich das Verhalten mit einem der anderen Gatewaytypen nicht beschreiben lässt.

[8] Schwimmbahnen

Schwimmbahnen (Swimlane) können den Prozess beispielsweise nach Organisationen, Rollen und Verantwortlichkeiten strukturieren. Es gibt zwei Arten von Schwimmbahnen: Pools und Lanes.

Mit einem **Pool (Pool)** wird eine Organisation, wie z.B. ein Kunde, Lieferant oder Unternehmen, abgebildet. Jeder Prozess muss in einem Pool liegen. Einzige Ausnahme ist der unternehmensinterne Prozess. Hier darf der Pool, der das Unternehmen darstellt, weggelassen werden (s. Abb. [Sb]).

Ein Pool kann mit Hilfe von **Lanes (Lanes)** weiter unterteilt werden. Lanes sind z. B. interne Organisationseinheiten, Rollen oder Systeme. Lanes dürfen Lanes enthalten (s. Abb. [Sb]). Pools und Lanes können vertikal oder horizontal dargestellt werden.

Das **datenbasierte exklusive Gateway (Data Based Exclusive Gateway)** entscheidet abhängig von den ausgehenden Bedingungen, wie der Sequenzfluss aufgeteilt wird. Es erhält genau einer der alternativen Sequenzflüsse den Token^{*} und setzt den Ablauf fort. Wenn keine Bedingung erfüllt ist, bleibt der Token^{*} am Gateway hängen. Bei der Zusammenführung wird der Token direkt weitergeleitet (s. Abb. [Gw]).

Das **ereignisbasierte exklusive Gateway (Event Based Exclusive Gateway)** entscheidet aufgrund von eintretenden Ereignissen, wie der Sequenzfluss aufgeteilt wird. Der Token^{*} läuft entlang des Sequenzflusses, dessen Ereignis als erstes eintrifft. Die Ereignisse (bzw. Receive Tasks) werden direkt nach dem ereignisbasierten exklusiven Gateway modelliert (s. Abb. [Gw]).

Ein **paralleles Gateway (Parallel Gateway)** teilt den Sequenzfluss in zwei oder mehrere parallele Flüsse auf. Die Anzahl der erzeugten Token^{*} entspricht der Anzahl der ausgehenden Flüsse. Bei einer Zusammenführung synchronisiert das parallele Gateway alle eingehenden Sequenzflüsse und wartet bis alle Token^{*} eingetroffen sind. Deren Anzahl entspricht hierbei der Anzahl der eingehenden Sequenzflüsse. Parallele Gateways haben keine Bedingung (s. Abb. [Ze]).

Das **einschließende Gateway (Inclusive Gateway)** verhält sich ähnlich, wie das parallele Gateway. Allerdings werden nur die Sequenzflüsse bei der Aufteilung in einen parallelen Fluss mit eingeschlossen, deren Bedingung erfüllt ist. Bei der Synchronisierung wird nur auf die eingehenden Sequenzflüsse gewartet, die aktiv sind.

Beim **komplexen Gateway (Complex Gateway)** kann der Modellierer selber eine Verzweigungsbedingung definieren, die das Verhalten am Gateway festlegt. Komplexe Gateways werden immer dann eingesetzt, wenn sich das Verhalten mit einem der anderen Gatewaytypen nicht beschreiben lässt.

Die **Artefakte (Artifacts)** bieten die Möglichkeit neben der Ablaufreihenfolge zusätzliche Informationen, wie z. B. benötigte Geschäftsobjekte, festzuhalten. Es gibt drei Arten von Artefakten: Datenobjekte, Kommentare und Gruppen.

Ein **Datenobjekt (Data Object)** ist ein Eingangs- bzw. Ergebnisobjekt einer Aktivität, wie z. B. ein Auftrag und kann um Zustände erweitert werden (s. Abb. [Af]). Datenobjekte beeinflussen den Sequenz- oder Nachrichtenfluss nicht direkt.

Eine **Anmerkung (Text Annotation)** ist eine textuelle Beschreibung, die über eine Assoziation mit jedem Diagrammelement verbunden werden kann (s. Abb. [Af]).

Eine **Gruppe (Group)** fasst mehrere Elemente zusammen, die logisch zusammengehören (s. Abb. [Gw]). Der Sequenzfluss eines Prozesses wird durch eine Gruppe nicht beeinflusst, da die Gruppe nicht mit dem Nachrichten- oder Sequenzfluss verbunden werden darf. Gruppen können über Pools und Lanes laufen.

[9] Geschäftsprozessmodellierung mit BPMN & Co.

Das **datenbasierte exklusive Gateway (Data Based Exclusive Gateway)** entscheidet abhängig von den ausgehenden Bedingungen, wie der Sequenzfluss aufgeteilt wird. Es erhält genau einer der alternativen Sequenzflüsse den Token^{*} und setzt den Ablauf fort. Wenn keine Bedingung erfüllt ist, bleibt der Token^{*} am Gateway hängen. Bei der Zusammenführung wird der Token direkt weitergeleitet (s. Abb. [Gw]).

Das **ereignisbasierte exklusive Gateway (Event Based Exclusive Gateway)** entscheidet aufgrund von eintretenden Ereignissen, wie der Sequenzfluss aufgeteilt wird. Der Token^{*} läuft entlang des Sequenzflusses, dessen Ereignis als erstes eintrifft. Die Ereignisse (bzw. Receive Tasks) werden direkt nach dem ereignisbasierten exklusiven Gateway modelliert (s. Abb. [Gw]).

Ein **paralleles Gateway (Parallel Gateway)** teilt den Sequenzfluss in zwei oder mehrere parallele Flüsse auf. Die Anzahl der erzeugten Token^{*} entspricht der Anzahl der ausgehenden Flüsse. Bei einer Zusammenführung synchronisiert das parallele Gateway alle eingehenden Sequenzflüsse und wartet bis alle Token^{*} eingetroffen sind. Deren Anzahl entspricht hierbei der Anzahl der eingehenden Sequenzflüsse. Parallele Gateways haben keine Bedingung (s. Abb. [Ze]).

Das **einschließende Gateway (Inclusive Gateway)** verhält sich ähnlich, wie das parallele Gateway. Allerdings werden nur die Sequenzflüsse bei der Aufteilung in einen parallelen Fluss mit eingeschlossen, deren Bedingung erfüllt ist. Bei der Synchronisierung wird nur auf die eingehenden Sequenzflüsse gewartet, die aktiv sind.

Beim **komplexen Gateway (Complex Gateway)** kann der Modellierer selber eine Verzweigungsbedingung definieren, die das Verhalten am Gateway festlegt. Komplexe Gateways werden immer dann eingesetzt, wenn sich das Verhalten mit einem der anderen Gatewaytypen nicht beschreiben lässt.

Die **Artefakte (Artifacts)** bieten die Möglichkeit neben der Ablaufreihenfolge zusätzliche Informationen, wie z. B. benötigte Geschäftsobjekte, festzuhalten. Es gibt drei Arten von Artefakten: Datenobjekte, Kommentare und Gruppen.

Ein **Datenobjekt (Data Object)** ist ein Eingangs- bzw. Ergebnisobjekt einer Aktivität, wie z. B. ein Auftrag und kann um Zustände erweitert werden (s. Abb. [Af]). Datenobjekte beeinflussen den Sequenz- oder Nachrichtenfluss nicht direkt.

Eine **Anmerkung (Text Annotation)** ist eine textuelle Beschreibung, die über eine Assoziation mit jedem Diagrammelement verbunden werden kann (s. Abb. [Af]).

Eine **Gruppe (Group)** fasst mehrere Elemente zusammen, die logisch zusammengehören (s. Abb. [Gw]). Der Sequenzfluss eines Prozesses wird durch eine Gruppe nicht beeinflusst, da die Gruppe nicht mit dem Nachrichten- oder Sequenzfluss verbunden werden darf. Gruppen können über Pools und Lanes laufen.

Das **datenbasierte exklusive Gateway (Data Based Exclusive Gateway)** entscheidet abhängig von den ausgehenden Bedingungen, wie der Sequenzfluss aufgeteilt wird. Es erhält genau einer der alternativen Sequenzflüsse den Token^{*} und setzt den Ablauf fort. Wenn keine Bedingung erfüllt ist, bleibt der Token^{*} am Gateway hängen. Bei der Zusammenführung wird der Token direkt weitergeleitet (s. Abb. [Gw]).

Das **ereignisbasierte exklusive Gateway (Event Based Exclusive Gateway)** entscheidet aufgrund von eintretenden Ereignissen, wie der Sequenzfluss aufgeteilt wird. Der Token^{*} läuft entlang des Sequenzflusses, dessen Ereignis als erstes eintrifft. Die Ereignisse (bzw. Receive Tasks) werden direkt nach dem ereignisbasierten exklusiven Gateway modelliert (s. Abb. [Gw]).



* Token: Ein Token ist einer Art virtuelle Murrel mit deren Hilfe der Ablauf eines Prozesses simuliert werden kann. Wenn der Prozess startet, wird ein Token erstellt, der von Aktivität zu Aktivität wandert.

© Copyright 2009 by oose Innovative Informatik GmbH, Hamburg. Bezug dieses Posters über www.oose.de/poster