

Transformation zwischen den ausgewählten Modelliersprachen EPK und BPMN

Bachelorarbeit



Vorgelegt von:	Christopher Partsch
Studienbereich:	Wirtschaftsinformatik
Gutachter:	Professor Dr. Hans-Knud Arndt
Betreuer:	Professor Dr. Hans-Knud Arndt

Selbstständigkeitserklärung

Ich, Christopher Partsch, versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema

Transformation zwischen den ausgewählten Modelliersprachen EPK und BPMN

selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate also gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mir ist bekannt, dass ich meine Bachelorarbeit zusammen mit dieser Erklärung fristgemäß nach Vergabe des Themas in zweifacher Ausfertigung und gebunden im Prüfungsamt der Otto-von-Guericke-Universität Magdeburg abzugeben oder spätestens mit dem Poststempel des Tages, an dem die Frist abläuft, zu senden habe.

Magdeburg, den 24.02.2018

Christopher Partsch

Inhaltsverzeichnis

Inhaltsverzeichnis	1
1 Einleitung.....	3
1.1 Motivation	3
1.2 Aufgabenstellung und Zielsetzung	5
1.3 Aufbau der Arbeit	6
2 Prozessmodellierung	8
2.1 Begriffserläuterungen	8
2.1.1 Modell.....	8
2.1.2 Prozesse.....	9
2.2 Grundsätze ordnungsgemäßer Modellierung	11
2.3 Basiselement in der Prozessdarstellung.....	13
2.4 Warum Prozessmodellierung?	14
3 Die Modelliersprachen EPK und BPMN.....	17
3.1 Einordnung der EPK und der BPMN in das ARIS Konzept	20
3.2 Die Object Management Group (OMG)	17
3.3 Ereignisgesteuerter Prozessketten (EPK)	17
3.3.1 Ursprung und Zweck.....	21
3.3.2 Notationselemente der EPK	22
3.4 Business Process Model and Notation (BPMN).....	25
3.4.1 Ursprung und Zweck.....	25
3.4.2 Notationselemente der BPMN	26
3.5 Bewertungsmethoden für die Eignung zur Prozessdarstellung	29
3.5.1 Workflow Pattern Initiative	29
3.5.2 Eignung von EPK zur Prozessdarstellung	32
3.5.3 Eignung von BPMN zur Prozessdarstellung.....	34
3.6 Fazit	36
4 Transformation von EPK in BPMN.....	38
4.1 Erste Überlegungen.....	38
4.2 Direkte Transformation von EPK zu BPMN	38
5 Verifizierung und Validierung.....	48
5.1 Begriffserklärung Verifizierung und Validierung	48
5.2 Beispieltransformation von EPK zur BPMN	49
Verifizierung	53
Validierung	53
6 Zusammenfassung und Ausblick	54

7	Tabellenverzeichnis	55
8	Abbildungsverzeichnis	56
9	Literaturverzeichnis	58
I	Anhang	61
	Notationselemente der BPMN	61
	Basiselemente	61
	Erweiterte Elemente der BPMN	66
	Notationselemente der EPK	75
	Workflow Pattern	78
	Datenmuster (Data Patterns)	78
	Ressourcenmuster (Resource Patterns)	83

1 Einleitung

Die vorliegende Arbeit beschäftigt sich mit der Transformation der ausgewählten Modelliersprachen ereignisgesteuerter Prozessketten (EPK) und der Business Process Modeling Notation (BPMN). In diesem einleitenden Kapitel wird die Motivation für diese Arbeit nähergebracht sowie die Problem- und Zielstellung beschrieben. Außerdem erfolgt ein kurzer Überblick über den weiteren Aufbau der Arbeit.

1.1 Motivation

Mit der stetig wachsenden Globalisierung und Internationalisierung der Märkte, gestiegenen Ansprüchen der Kunden und gesättigte Absatzmärkte stehen Unternehmen zunehmend und einem immer höheren Maße neuen Herausforderungen gegenüber.[1, S. 2] Diese führen zu „steigende[m] Wettbewerbsdruck bezüglich Zeit, Kosten und Qualität“ [2, S. 1] und erfordern auf die neuen Anforderungen flexibel zu reagieren und die erforderlichen Anpassungen vornehmen zu können. Dabei hat sich das Geschäftsprozessmanagement, welches auch als Business Process Management (BPM) bezeichnet wird [1, S. 5], als ein geeignetes und bewährtes Konzept herausgestellt. [1, S. 2]

Aufgrund der genannten der Anforderungen hat die BPM in den letzten Jahren an Popularität gewonnen und ist der Bereich mit der höchsten Priorität für die meisten Unternehmen geworden. [1, S. 3, 3, S. 1] Der gleichen Auffassung sind auch Ko et al. [4, S. 744ff] und heben dabei den Bedarf nach einem schnellen Informationsaustausch, kurzen Entscheidungswegen, die Notwendigkeit, sich bei Nachfrage sofort anzupassen, sowie mehr internationale Konkurrenz hervor. „Trotz der hohen Misserfolgsquoten und den kritischen Beurteilungen gibt es einen Konsens darüber, dass BPM-Projekte, wenn sie richtig und mit einem effektiven Einsatz von Informationstechnologie umgesetzt werden, bedeutende Fortschritte bei der Leistung von Organisationen tragen.“ [3, S. 2] Die BPM ist ein Gebiet, welches die Menschen, deren Arbeitsweise miteinander, die verwendete Technologie und die Zielvorgaben mit Bezug auf die Geschäftsprozesse behandelt. [3, S. 2f, 5, S. 1]

Mit dem Konzept der Architektur integrierter Informationssysteme (ARIS) ist es möglich, ein solches BPM zu unterstützen. Das ARIS-Konzept 1992 von August-Wilhelm Scheer und seinen Mitarbeitern entwickelt. Es dient der Modellierung von Geschäftsprozessen in unterschiedlichen Sichten und auf verschiedenen Ebenen und gilt als „Rahmenwerk zur Beschreibung von Unternehmen und betriebswirtschaftlichen Anwendungssystemen“. [2, S. 12]

Bei ARIS wird zwischen folgenden fünf Sichten unterschieden: Funktionssicht, Datensicht, Organisationssicht, Steuerungs- beziehungsweise Prozesssicht und Leistungssicht. In der Funktionssicht werden fachliche Aufgaben und Tätigkeiten an einem Objekt beschrieben, die eines oder mehrere Ziele eines Unternehmens unterstützen. [6, S. 21] In der Datensicht hingegen werden reale oder abstrakte Dinge als Datenstruktur beschrieben, die für eine Aufgabe von Interesse sind (beispielsweise ein Lieferant mit Namen oder eine Bestellung mit Bestellnummer). Außerdem werden die Beziehungen zwischen den Datenstrukturen untereinander dargestellt. [6, S. 23] Mit der Aufbau- und Ablauforganisation, also konkret mit den Hierarchien in dem Unternehmen befasst sich die Organisationssicht. [6, S. 25] Das Ziel dieser drei Sichten ist es, die Komplexität eines Prozesses zu reduzieren; allerdings gehen bei dieser Zerlegung die Beziehungen zwischen den einzelnen Komponenten verloren. [6, S. 20] Mittels der Prozesssicht werden die Zusammenhänge zwischen den anderen Sichten wiederhergestellt. In der fünften Sicht, der Leistungssicht, werden materielle und immaterielle Leistungen behandelt, die von Prozessen benötigt oder erzeugt werden. [6, S. 20] Die Abbildung 1.1 zeigt das sogenannte ARIS-Haus aus den vorher beschriebenen Sichten bestehend.

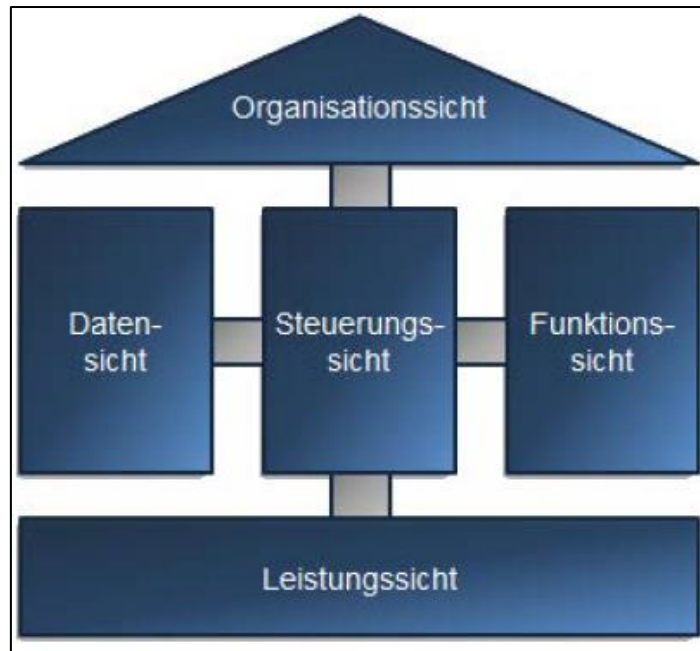


Abbildung 1.1: ARIS-Haus

Im Rahmen dieser Arbeit liegt der Fokus auf der Prozesssicht, weil in ihr die Beziehungen zwischen den einzelnen Komponenten aus den anderen Sichten zusammengefasst werden und ein gesamtheitliches Bild des betrachteten Prozesses hergestellt wird. [2, S. 21]

Seit den späten Achtzigerjahren des vergangenen Jahrhunderts haben sich diverse Standards für das BPM entwickelt, um Prozesse darstellen und beschreiben zu können. Ko et al. [4, S. 751] führen einige auf, wie die BPEL (Business Process Execution Language), BPSS (Business Process Specification Schema), Petri-Netze, Pi-Calculus, Rosetta-Netze, UML Aktivitäts-diagramme. [4, S. 753]

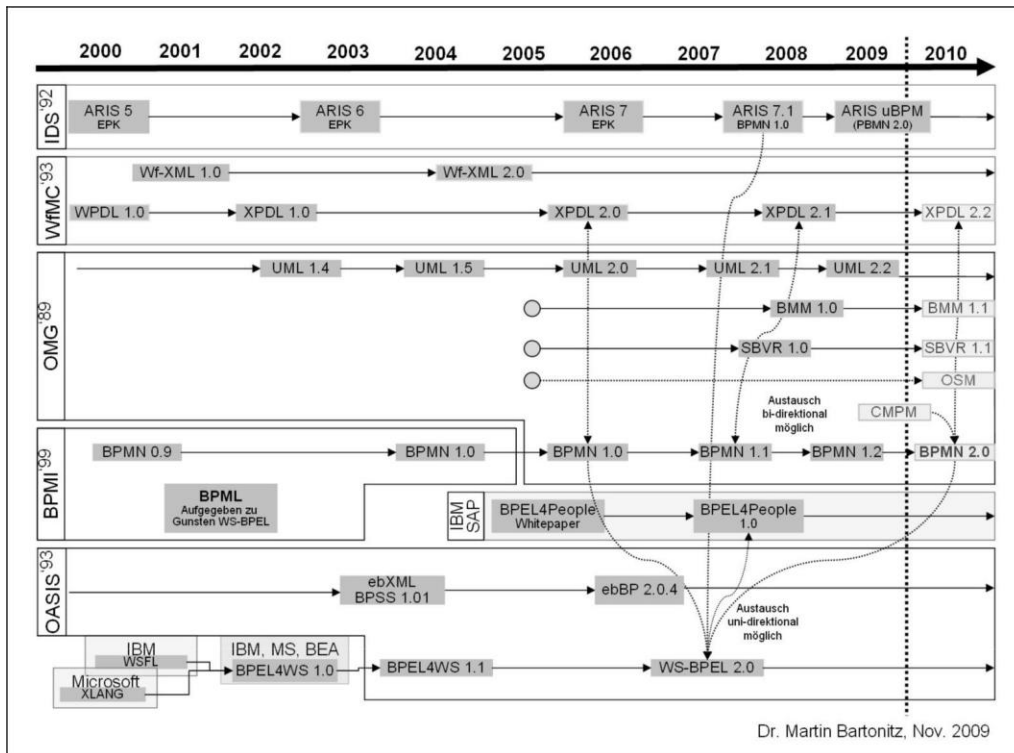


Abbildung 1.2: historische Entwicklung wichtiger Standards in der Prozessmodellierung

In Abbildung 1.2 ist eine historische Entwicklung wichtiger Standards bis zum Jahr 2009 dargestellt. Am linken Rand sind fünf Organisationen aufgeführt, die BPM-Standards entwickeln und pflegen. Darunter befinden sich beispielsweise die IDS Scheer AG, seit 2010 Software AG, mit den Ereignisgesteuerten Prozessketten (EPK) und die Object Management Group (OMG), welche sich aktuell um die Unified Modeling Language (UML) sowie die Business Process Model and Notation (BPMN) kümmert.

Aus der Abbildung 1.2 lässt sich erkennen, dass bestimmte Standards wie der ARIS EPK-Standard, der UML-Standard sowie der BPMN-Standard kontinuierlich weiterentwickelt werden. Die Pfeile mit durchgehender Linie beschreiben hierbei eine Transformation zwischen den Standards. Hier ist gut zu erkennen, dass dies nur zwischen Vor- und Nachfolger eines Standards möglich ist. Die Pfeile mit gepunkteter Linie stellen Transformationen zwischen unterschiedlichen Standards dar und ist nur selten möglich. Der Transformationsweg zwischen BPMN 2.0 und WS-BPEL von Omar [7] lässt sich hier als Beispiel aufführen. Dieser ist nur eine Richtung möglich. Da fast keine Transformationswege zwischen den wichtigen Standards vorhanden sind, ergibt sich das Problem, dass die Modelle nicht ineinander überführt werden können.

Als Lösungsansatz bietet sich an dieser Stelle eine manuelle Transformation der Prozesse in den bekannten Modellierungsstandard durch eine kleine Gruppe geschulter Mitarbeiter an. Jedoch ist das eine kostenintensive Variante, da solche Experten teuer sind und eine manuelle Transformation zeitintensiv ist. Die zeitliche Verzögerung steigt dabei proportional zu der wachsenden Zahl an zu transformierenden Prozessen an. Außerdem können weitere Probleme durch Unstimmigkeiten entstehen, wenn die Prozesse ineinander überführt werden, da es keinen einheitlichen Transformationsstandard gibt. So kann es passieren, dass zwei verschiedene Experten auf komplett unterschiedliche Art und Weise eine Transformation der Modelle durchführen. Außerdem ist es zum Teil nicht möglich, ein Gegenstück für ein Element aus dem Quell-Modellierungsstandard im Ziel-Modellierungsstandard zu finden. Dies führt zu Fehlern und Verlusten im transformierten Prozess und kann Folgefehlern bei der Entwicklung von Komponenten führen, weil nicht alle benötigten Informationen (korrekt) vorhanden sind.

1.2 Aufgabenstellung und Zielsetzung

Ein einheitlicher Standard für die automatische Transformation zwischen Modellierungsstandards wie dem ARIS EPK-Standard und dem BPMN-Standard wäre wünschenswert, existiert jedoch zum aktuellen Zeitpunkt nur eingeschränkt. Mit dieser Arbeit soll ein Teil der wissenschaftlichen Lücke geschlossen werden und aus den gewonnenen Erkenntnissen ein Regelwerk für die Transformation der EPK in die BPMN erstellt werden.

EPK und BPMN gehören zu den bekanntesten Standards für die Prozessmodellierung. [4, S. 754]. Das spiegelt auch die Abbildung 1.3 wieder, welche die Statistik zur Popolarität von Prozessnotationen im BPM-Netzwerk zeigt. Das Netzwerk verbindet mehr als 7000 BPM-Professionals aus dem deutschsprachigen Raum. [2, S. XIIIff] Auf der linken Seite der Graphik ist die Anzahl der Mitglieder festgehalten, die entweder Interesse an oder bereits Erfahrung mit dem jeweiligen Modellierungsstandard hat. Am unteren Rand sind die drei Modellierungsstandards EPK, UML und BPMN aufgeführt. In dunkelblau ist dargestellt, wie viele Mitglieder Interesse an dem jeweiligen Standard haben, in hellblau ist dargestellt, wie viele Mitglieder bereits Praxiserfahrung mit dem Standard haben. Es ist deutlich zu erkennen, dass für die EPK die meiste Praxiserfahrung vorhanden ist, aber auch, dass das Interesse BPMN hoch ist. Schmelzer und Sesselmann beobachten ebenfalls das gesteigerte Interesse an der BPMN. Im Jahr 2007 war die Verbreitung der BPMN und der UML bei der

Prozessmodellierung etwa noch gleich war und die der EPK bei circa 60% lag, hat die BPMN die EPK im Jahr 2009 bereits überholt. [1, S. 417f]

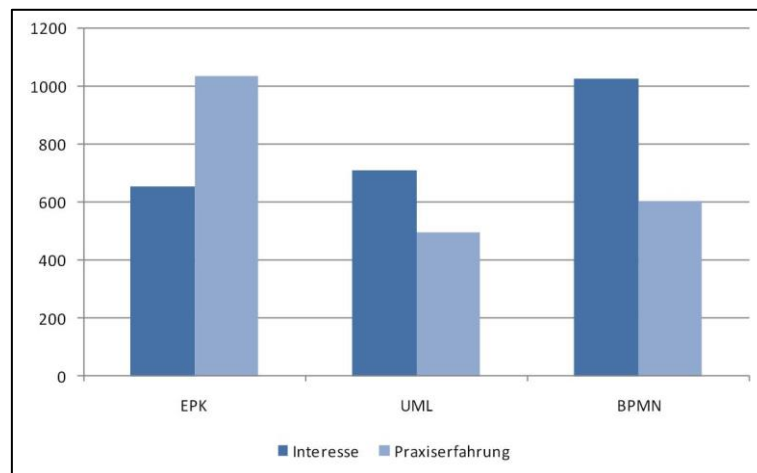


Abbildung 1.3: Popularität von Prozessnotationen

In dieser Arbeit werden die Modelliersprachen BPMN und EPK bearbeitet, weil sie wie zuvor beschrieben zu den prominentesten Modellierungsstandards gehören. Die EPK ist weit verbreiteter Standard und das Interesse an der BPMN ist groß und sie wird immer öfter für die Prozessmodellierung verwendet. [1, S. 416ff, 8, S. XIII]

Das Ziel dieser Bachelorarbeit ist es, einen ersten Transformationsweg zwischen ausgewählten Diagrammtypen der Modellsprachen EPK und BPMN für die Prozessdarstellung zu beschreiben.

Die Forschungsfrage lautet daher: „Wie kann ein Transformationsweg in Hinblick auf die Prozessdarstellung zwischen den Modellsprachen EPK und BPMN aussehen?“ Diese Frage wird am Beispiel von ausgewählten Diagrammtypen der beiden Modellsprachen untersucht. Während der Bearbeitung wird analysiert, in wie weit sich die ausgewählten Diagrammtypen wirklich für die Prozessdarstellung eignen und welche Einschränkungen in Kauf genommen werden müssen. Weiterhin wird ein Regelwerk für die Transformation erarbeitet.

1.3 Aufbau der Arbeit

In Kapitel 2 werden zunächst einige wichtige Begriffe in dieser Arbeit geklärt. Zusätzlich wird betrachtet, welche Elemente für die Prozessdarstellung elementar sind und demnach von Modellierungsstandards bedient werden müssen. Anschließend erfolgt ein kurzer Einblick in die Geschichte und die Bedeutung von Prozessen für Unternehmen.

Danach folgt in Kapitel 3 eine kurze Vorstellung der Object Management Group und des ARIS-Konzepts sowie eine Einführung in die ausgewählten Modellsprachen EPK und BPMN. Dabei wird einerseits deren Ursprung und Zweck erläutert, andererseits erfolgt je eine kritische Auseinandersetzung in wie weit sie geeignet sind, Prozesse darstellen zu können. Im Anschluss daran werden die einzelnen Notationselemente kurz in der Übersicht vorgestellt.

In Kapitel 4 wird ein erster Transformationsweg erarbeitet. Dazu wird betrachtet, welche Elemente und Konstrukte sich ineinander überführen lassen und wie dies konkret möglich ist. Im Anschluss wird ein erstes Regelwerk für die Transformation erstellt.

In Kapitel 5 erfolgt eine Zusammenfassung der Arbeit sowie eine abschließende kritische Betrachtung der erzielten Ergebnisse. Zum Schluss wird ein Ausblick auf eine zukünftige Fortführung des Projekts gegeben.

2 Prozessmodellierung

In diesem Kapitel werden grundlegende Begriffe der Prozessmodellierung, wie Modell und Prozess, und deren Bedeutung für Unternehmen erläutert. Außerdem werden die Grundsätze ordnungsgemäßer Modellierung vorgestellt und erklärt.

2.1 Begriffserläuterungen

2.1.1 Modell

Es existieren einige unterschiedliche Definitionen für ein Modell. Im Brockhaus lässt sich folgende Definition finden: Ein Modell ist ein „materielles Objekt oder theoret[isches] Konstrukt, das einem Untersuchungsgegenstand in bestimmten Eigenschaften oder Relationen entspricht (Struktur-Funktions- oder Verhaltensanalogie) und für sonst nicht mögl[iche] oder zu aufwendige experimentelle Untersuchungen, math[ematische] Berechnungen, Erklärungs- oder Demonstrationzwecke oder zur Optimierung des Originals verwendet wird.“ [9, S. 4871f] Mit dieser Definition sind auch materielle Modelle wie das des Ottomotors eingeschlossen.

Jedoch ist dies nicht die einzige Definition eines Modells. Becker et al. erklärt den Begriff wie folgt: „Ein Modell wird verstanden als ein immaterielles Abbild der Realwelt (des Objektsystems) für Zwecke eines Subjekts. Modelle werden als Hilfsmittel zur Erklärung und Gestaltung realer Systeme eingesetzt. Erkenntnisse über Zusammenhänge und Sachverhalte bei realen Problemen können mit Hilfe von Modellen aufgrund von Ähnlichkeiten gewonnen werden, die zwischen dem realen betrieblichen System und dem Modell als Abbild dieses Systems bestehen.“ [10, S. 435]

Nach der Definition von Becker et al. gibt ein Modell einen realen Sachverhalt wieder, ist aber selbst kein greifbares Objekt. Dies widerspricht jedoch der ersten Definition, welche die Existenz von materiellen Modellen wie dem Modell eines Ottomotors oder des Sonnensystems mit einbezieht.

Zieht man beide Definitionen heran, so geht aus ihnen hervor, dass ein zu untersuchender Gegenstand zu einem Objekt oder Konstrukt wird. Dieses Objekt/Konstrukt weist Ähnlichkeiten zu dem real existierenden Gegenstand auf, wodurch (neue) Erkenntnisse gewonnen werden können. Außerdem lässt sich aus beiden Definitionen erkennen, dass sich mittels eines Modells aufgrund von Ähnlichkeiten und Analogien Zusammenhänge und Sachverhalte erkennen und wiedergeben lassen. Jedoch erfolgen diese aus einer abstrahierten Sichtweise. Wichtig dabei ist der Schritt der Modellerstellung, welche auch als Modellierung bezeichnet wird. [11, S. 7] Bei dem Modellierungsvorgang wird ein realweltliches Objekt oder ein realweltlicher Vorgang abstrahiert und als ein vereinfachtes Abbild der Realität dargestellt. In dieser Arbeit werden ausschließlich Modelle von Prozessen betrachtet.

Weitere wichtige Begriffe sind die Modellierungssprache (Modellsprache) und der Modellierungsstandard, welche im Rahmen dieser Arbeit als Synonym für einander verwendet werden. Die Modellsprachen werden dazu verwendet, ein System in Form von Abstraktion zu beschreiben. [12, S. 666] Durch den Einsatz von Modellersprachen lässt sich ein System „spezifizieren, visualisieren, konstruieren und dokumentieren“. Jeder Modellierungsstandard besitzt eine eigene festgelegte Syntax und Semantik, um ein realweltliches Objekt zu abstrahieren. Das Ziel bei jeder Modellierung ist es, einen komplexen Sachverhalt ohne wesentlichen Informationsverlust leicht verständlich darzustellen.

Die Modelltransformation ist ein weiterer wichtiger Begriff in dieser Arbeit. Dabei setzt sich das Wort Modelltransformation aus zwei Ausdrücken zusammen: Das schon erläuterte Modell und dem Begriff Transformation. Der Wortursprung der Transformation liegt im lateinischen und bedeutet „Umformung, Umgestaltung, Umwandlung“. [13, S. 7626] Bei der Transformation handelt es sich nach Brockhaus um eine Umformung einer Struktur in eine andere unter Beibehaltung der Grundbedeutung. [13, S. 7626] In vielen anderen Gebieten wird ebenfalls von Transformation gesprochen, beispielsweise in der Mathematik oder Physik, wo von Ähnlichkeitstransformationen die Rede ist, der Medizin, wo normale Zellen durch Tumorzellen sich in Tumorzellen umwandeln oder auch in Wirtschaft und Politik, wo von Systemtransformation und Transformationsgesellschaften gesprochen wird. [13, S. 7626] Im Rahmen dieser Arbeit wird unter einer Transformation immer die Transformation von einem Modell zu einem anderen Modell verstanden. Mit anderen Worten: ein Modell wird mit Hilfe von festgelegten Regeln in ein anderes Modell einer anderen Modellersprache überführt (M-to-M-Transformation).

2.1.2 Prozesse

Das Wort „Prozess“ hat seinen Ursprung im Lateinischen und bedeutet „Vorgang, Verlauf oder Entwicklung“. [14, S. 5920] Weiterhin versteht man unter einem Prozess die Gesamtheit aufeinander einwirkender Vorgänge innerhalb eines Systems. So werden mittels Prozessen Materialien, Energien oder auch Informationen zu neuen Formen transformiert, gespeichert oder aber allererst transportiert.[15] Dies hat jedoch nichts mit der Art der Transformation zu tun, die im vorigen Abschnitt erläutert wurde.

Im DIN EN ISO 9000:2005 wird der Prozess als ein „Satz von in Wechselbeziehung oder Wechselwirkung stehenden Tätigkeiten, der Eingaben in Ergebnisse umwandelt“ definiert. In der Prozessmodellierung respektive Prozessdarstellung ist der Prozess das wichtigste Element.

Nach der Definition existieren zunächst Eingaben, welche während eines Prozesses in ein oder mehrere Ergebnisse umgewandelt werden. In der Abbildung 2.1 wird der grundsätzliche Aufbau eines Prozesses verdeutlicht. [16, S. 1]

Sowohl materielle als auch immaterielle Eingaben und Ergebnisse können verwendet werden, wobei die materiellen Eingaben nach DIN beispielsweise Ausrüstungen, Bauelemente oder Materialien sein können. Immaterielle Eingaben sind unter anderem Informationen oder Energie. Gleiches gilt nach DIN für die Ergebnisse. [16, S. 3]



Abbildung 2.1: Grundsätzlicher Prozess

Die Ergebnisse können dabei beabsichtigter und unbeabsichtigter Natur sein. Als unerwünschte Ergebnisse sind nach DIN Abfall und Umweltverschmutzung aufgeführt. [17, S. 3]

Außerdem können die Ergebnisse eines Prozesses häufig die Eingaben für andere Prozesse sein und sind somit mit dem Gesamtnetzwerk oder -system verknüpft. Was die DIN 9000:2005 damit ausdrücken will, ist in der Abbildung 2.2 dargestellt.

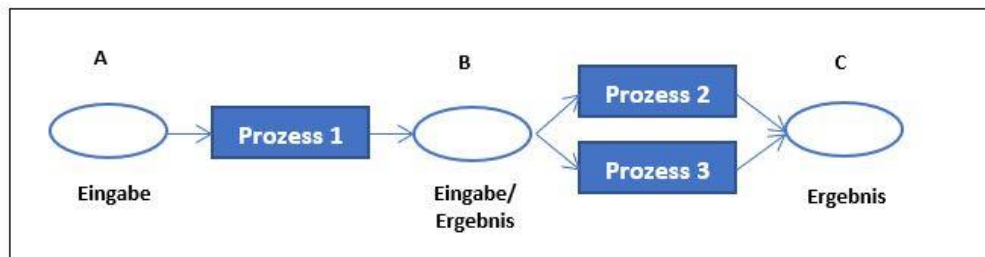


Abbildung 2.2: Mögliche Prozessreihenfolge

Nach erfolgreichem Durchlauf von Prozess 1 ist ein Ergebnis (B) entstanden. Dieses Ergebnis bzw. die Ergebnisse aus dem Prozess 1 sind gleichzeitig auch die Eingaben (B) für die Prozesse 2 und 3, welche wiederum ein Ergebnis (C) zur Folge haben.

Ein Prozess in einem Unternehmen läuft nicht nur einmalig ab, sondern muss wiederholbar sein und immer wieder das gleiche Ergebnis in gewünschter Variation liefern. Ein gutes Beispiel dafür ist die Produktion von Fahrzeugen in der Automobilindustrie. Jeder Automobilkonzern wird aufgrund von Unwirtschaftlichkeit und unzureichender Rentabilität nicht nur ein Fahrzeug produzieren, mit Ausnahme von Prototypen und Designstudien. Durch eine gewünschte Variation besteht die Möglichkeit ohne Änderung der grundlegenden Gestaltung oder Funktionsweise beispielsweise die farbliche Optik oder die Motorisierung zu verändern. Ein weiteres Beispiel ist die Produktion von Desktop Computern. Dort ist es ebenfalls nicht rentabel nur einen Computer in einer Variation zu produzieren. Mit Hilfe von Variationen lassen sich so individuelle Computer durch Ändern der CPU, der Arbeitsspeichergröße oder auch der Festplattengröße erstellen. Dabei bleibt die grundlegende Funktionsweise des Computers bestehen. Um immer wieder die gleichen Ergebnisse zu erhalten, müssen die Arbeitsschritte geplant und in der gleichen Reihenfolge ausgeführt werden.

2.2 Grundsätze ordnungsgemäßer Modellierung

Wie im vorigen Abschnitt erläutert, stellt ein Modell einen komplexen Ablauf vereinfacht dar. Um eine angemessene Qualität dieser vereinfachten Darstellung zu erhalten, wurden von Becker et al. sechs Grundsätze ordnungsgemäßer Modellierung (GoM) zusammengetragen. Diese wurden unterteilt in notwendige und ergänzende Grundsätze. Inwieweit die Qualität der Modelle gesteigert werden kann, hängt von den vorher definierten Modellierungszielen ab. [10, S. 437, 18, S. 39] Die sechs GoM sind Richtigkeit, Relevanz, Wirtschaftlichkeit, Klarheit, Vergleichbarkeit und systematischer Aufbau. Die Abbildung 2.3 stellt diese noch einmal in der unterteilten Form vor. [18, S. 40]

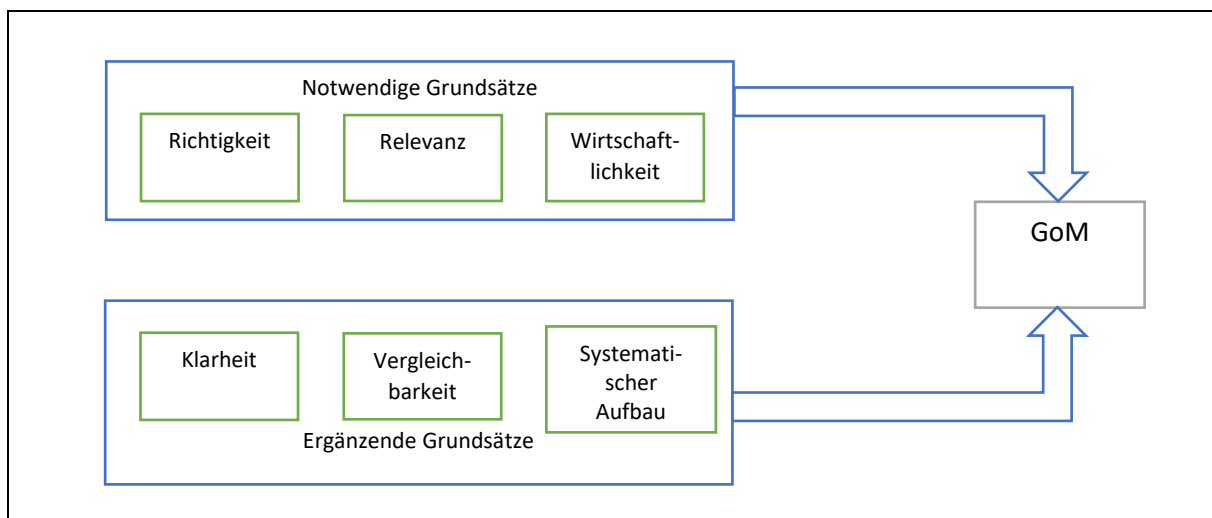


Abbildung 2.3: Grundsätze ordnungsgemäßer Modellierung

Der *Grundsatz der Richtigkeit* wird in zwei Arten unterschieden: die syntaktische Richtigkeit und die semantische Richtigkeit. Bei der syntaktischen Richtigkeit geht es um die formale Korrektheit eines Modells. Diese ist gegeben, wenn das Modell alle von der Modellierungssprache vorgegebenen Regeln einhält. Bei der semantischen Richtigkeit geht es um die Korrektheit des im Modell abgebildeten Sachverhaltes. Diese ist gegeben, wenn im Diskurs der Gutwilligen und Sachkundigen eine Einigung erzielt wurde. Mit anderen Worten: die Projektbeteiligten müssen sich darüber einigen, dass der modellierte Sachverhalt zutreffend wiedergegeben ist. [10, S. 437f, 18, S. 40]

Der *Grundsatz der Relevanz* beschreibt die Richtlinie, dass nur die Sachverhalte modelliert werden, die für den Zweck der Modellierung relevant sind. Um beurteilen zu können, welche Sachverhalte relevant für die Modellierung sind, ist es wichtig vorher entsprechende Modellierungsziele zu definieren. Die formulierten Ziele beschreiben dann eindeutig, was das Modell darstellen soll. Durch unterschiedliche Zielgruppen des Modells können sich die Modellierungsziele ändern. Damit einhergehend ändert sich auch die Relevanz einzelner Elemente der verwendeten Modellersprache [18, S. 41].

Im *Grundsatz der Wirtschaftlichkeit* ist definiert, dass „ein gegebenes Modellierungsziel mit minimalem Aufwand erreicht werden soll oder, dass mit einem gegebenen Modellierungsaufwand ein Modell erreicht werden soll, dass dem Modellierungszweck am nächsten kommt.“ Aus dieser Definition geht die Frage hervor, wie hoch der Detaillierungsgrad eines Modells sein soll, denn „sowohl Modellerstellung als und die Nutzung [sollen] möglichst kosteneffizient ablaufen.“ [18, S. 41] Dabei wird von ausgegangen, dass jede weitere Verfeinerung des Modells zusätzliche Kosten und zusätzlichen Nutzen generieren. [18, S. 41] Ab welchem Punkt des Detaillierungsgrades ein Modell

unwirtschaftliche ist, also mehr Kosten als Nutzen verursacht, hängt vom Unternehmen und dessen Zielen ab.

Beim *Grundsatz der Klarheit* geht es um die leichte Verständlichkeit der Modelle. Dabei werden Aspekte wie Übersichtlichkeit, Lesbarkeit und Strukturiertheit des Modells vordergründig betrachtet. Weitere Teilaspekte sind die Hierarchisierung, Layoutgestaltung und Filterung, welche ebenfalls zur leichteren Verständlichkeit eines Modells beitragen. [10, S. 438f, 18, S. 42]) Jedoch stehen sich der Grundsatz der Klarheit und der Grundsatz der Richtigkeit kritische gegenüber: Ist ein Modell klar dargestellt so kann es sein, dass der Sachverhalt nicht richtig wiedergeben wird. Andersherum besteht die Möglichkeit, dass ein semantisch korrektes Modell die leichte Verständlichkeit beeinträchtigt und somit die Klarheit der Abbildung verloren geht.

Der *Grundsatz der Vergleichbarkeit* kann wie der Grundsatz der Richtigkeit in zwei Betrachtungsweisen aufgeteilt werden. Die erste Art der Betrachtung befasst sich dabei mit der Vergleichbarkeit von Modellen, die mittels unterschiedlicher Modelliersprachen erstellt wurden. Modelle, die mit unterschiedlichen Modelliersprachen abgebildet wurden, müssen konsistent und ineinander überfahrbar sein. Die zweite Betrachtungsweise befasst sich mit der Vergleichbarkeit des Inhaltes eines Modells. Beide Modelle müssen zum gleichen Ergebnis führen. [10, S. 439, 18, S. 42]

Der letzte GoM ist der *Grundsatz des systematischen Aufbaus* und befasst sich mit dem Aufbau des Modells. Dabei soll sichergestellt werden, dass die verwendeten Elemente einen „gewissen Zusammenhang in der Struktur aufweisen“. [10, S. 439, 18, S. 42] Zudem muss auch auf eine konsistente Verwendung der Elemente in den unterschiedlichen Sichten (in Anlehnung an ARIS Datensicht, Funktionssicht, Organisationssicht und Prozesssicht) geachtet werden. So müssen bereits bei der Modellierung des Organisationsmodells die Folgen für die Prozesssicht betrachtet werden. [10, S. 439, 18, S. 42]

Durch die Einhaltung der GoM wird einerseits die Modellierung vereinfacht, da durch die GoM einige Modellierungskonventionen vorgegeben sind, nach denen modelliert werden sollte. Andererseits erleichtern die GoM die Verständlichkeit und das Lesen von Modellen, wenn die Richtlinien eingehalten wurden. Außerdem sollte das erhaltene Modell nach einer M2M-Transformation die GoM ebenso erfüllen wie das ursprüngliche Modell. Insbesondere für den Grundsatz der Vergleichbarkeit ist dies Relevant. [10, S. 437, 18, S. 39ff] In der Tabelle 2.1 sind die GoM nochmal übersichtlich dargestellt.

Grundsatz	Zusammenfassung
Der Richtigkeit	Syntaktische und semantische Richtigkeit
Der Relevanz	Modellierung von relevanten Elementen
Der Wirtschaftlichkeit	Fokus auf Kosten-Nutzen-Faktor durch Beschränkung des Detaillierungsgrades des Modells
Der Klarheit	Lesbarkeit und Verständlichkeit durch klare Struktur
Der Vergleichbarkeit	Modelle in unterschiedlichen Modelliersprachen müssen zum gleichen Ergebnis führen
Des systematischen Aufbaus	Konsistente Verwendung der modellierten Elemente

Tabelle 2.1: Überblick der Grundsätze ordnungsgemäßer Modellierung

2.3 Basiselement in der Prozessdarstellung

Betrachtet man nochmals die Prozessdefinition, so lassen sich direkt verschiedene für die Prozessdarstellung notwendige Elemente ableiten. Diese müssen von einem Modellierungsstandard wie der EPK oder der BPMN umgesetzt werden. Nach Göpfert und Lindenbach sind dies folgende Elemente: [19, S. 1]

- definierter Anfang und definiertes Ende
- Verzweigungs- und Zusammenführungselemente
- Steuerungselemente
- (äußere) Ereignisse
- Zeitablauf, Ressourcenverbrauch
- Verantwortliche und Beteiligte

Der Anfang eines Prozesses liegt bei der ersten Eingabe. Die Eingabe stellt eine materielle oder immaterielle Ressource dar, welche während des Prozesses verarbeitet wird. Das Ende eines Prozesses liegt bei dem verlassenden primär relevanten Ergebnis des Prozesses. Zwischen Anfang und Ende des Prozesses können Kuppelprodukte den Prozess bereits verlassen haben.

Viele Prozesse besitzen Unterprozesse, also einen Prozess in einem Prozess. Kommen Unterprozesse in einem Prozess vor, so erhalten diese die Ergebnisse von dem davorliegenden Unterprozess als Eingabe. Sie geben ihre eigenen Ergebnisse wiederum an nachfolgende bzw. nachlagernde Unterprozesse als Eingaben weiter. Ein Unterprozess kann dabei auch die Ergebnisse von mehreren Unterprozessen erhalten oder auch die eigenen Ergebnisse an mehrere andere Unterprozesse als Eingabe weitergeben. Eine wichtige Rolle spielen dabei die Verzweigungs- und Zusammenführungsknoten, welche eine besondere Form der Steuerungselemente darstellen.

Mit Hilfe der Steuerungselemente wird der weitere Prozessablauf beeinflusst. Die Steuerung ist dabei von verschiedenen Faktoren, wie die Ergebnisse von vorherigen Prozessen oder von externen Ereignissen, abhängig. Die zeitliche Reihenfolge der einzelnen Aktionen wird mit dem Prozessablauf dargestellt. Weiterhin besteht bei entsprechender Darstellung die Möglichkeit die Verantwortlichkeit und Beteiligung von Personen und Akteure für bestimmte Prozesse zuzuweisen. Durch die Zuweisung von Verantwortlichen und Beteiligten zu den Prozessen können Absprachen schneller getroffen werden und Fehler schneller identifiziert und beseitigt werden.

2.4 Warum Prozessmodellierung?

Durch die stetig wachsenden und zunehmenden komplexer werdenden Aufgaben versuchen die Unternehmen die Herausforderungen mit Hilfe der Prozessmodellierung zu meistern. In der Prozessmodellierung haben sich zwei wichtige Ansätze herauskristallisiert: der funktionsorientierte Ansatz und der prozessorientierte Ansatz.

Der funktionsorientierte Ansatz für die Prozessmodellierung wurde vor dem prozessorientierten Ansatz verwendet. Dabei ist jede Funktion darauf spezialisiert, eine bestimmte Aufgabe zu verrichten wodurch nur Teile der Kundenleistung bearbeitet werden. [1, S. 73] Die Prozesse werden durch die verschiedenen Abteilungsgrenzen unterbrochen und die Prozessverantwortlichen wechseln. Je autonomer die einzelnen Funktionsbereiche handeln, desto stärker steigen die Kosten für die Abstimmung und Koordination zwischen den einzelnen Bereichen im Unternehmen. [1, S. 74] Weiterhin führen die Schnittstellen zu Missverständnissen und Fehlern, verzögern die Entscheidungen, verbrauchen Zeit, erschweren die Kommunikation, führen zu Informationsverlust und mindern insgesamt die Ergebnisqualität sowie die Produktivität. [1, S. 74] „In einer Welt dauerhafter Produkte, stabiler Verbraucherbedürfnisse, klar abgegrenzter nationaler und regionaler Märkte und erkennbarer Konkurrenten war der Wettbewerb ein „Stellungskrieg“, in dem die Unternehmen bestimmte Felder wie auf einem Schachbrett einnehmen.“ Zu dieser Zeit hatte der funktionsorientierte Ansatz seine Berechtigung. Jedoch haben sich die Zeiten geändert. Um als Unternehmen Erfolg zu haben, ist es notwendig schnell auf sich verändernde Kundenbedürfnisse zu reagieren. [1, S. 75] Die Abbildung 2.4 und Abbildung 2.5 zeigen die funktionsorientierte Arbeitsweise gegenüber dem prozessorientierten Vorgehen.

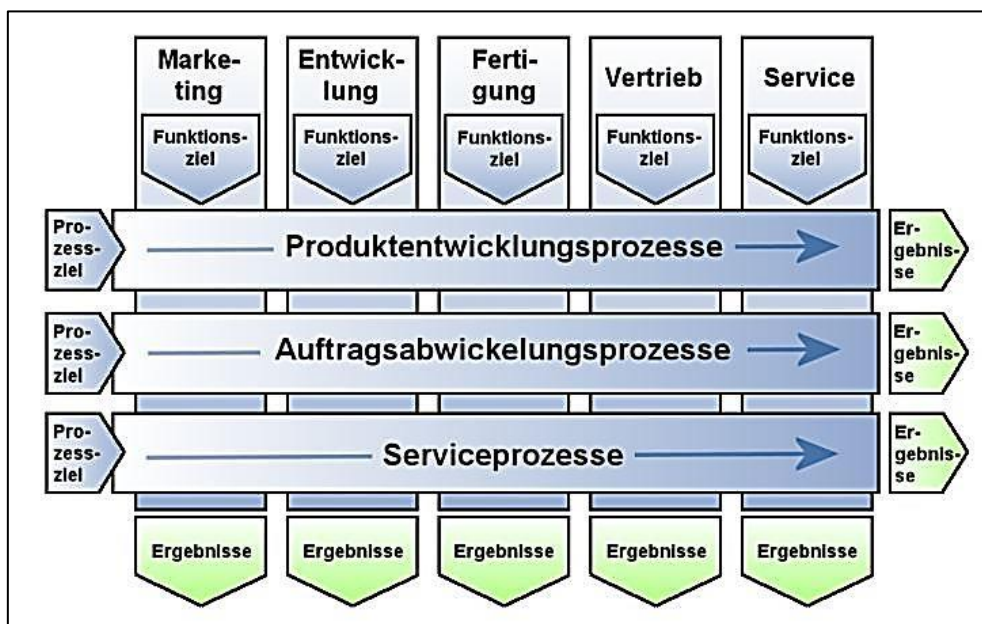


Abbildung 2.4: Funktions- versus Prozessorientierung (vgl. Schmelzer und Sesselmann [20, S. 54])

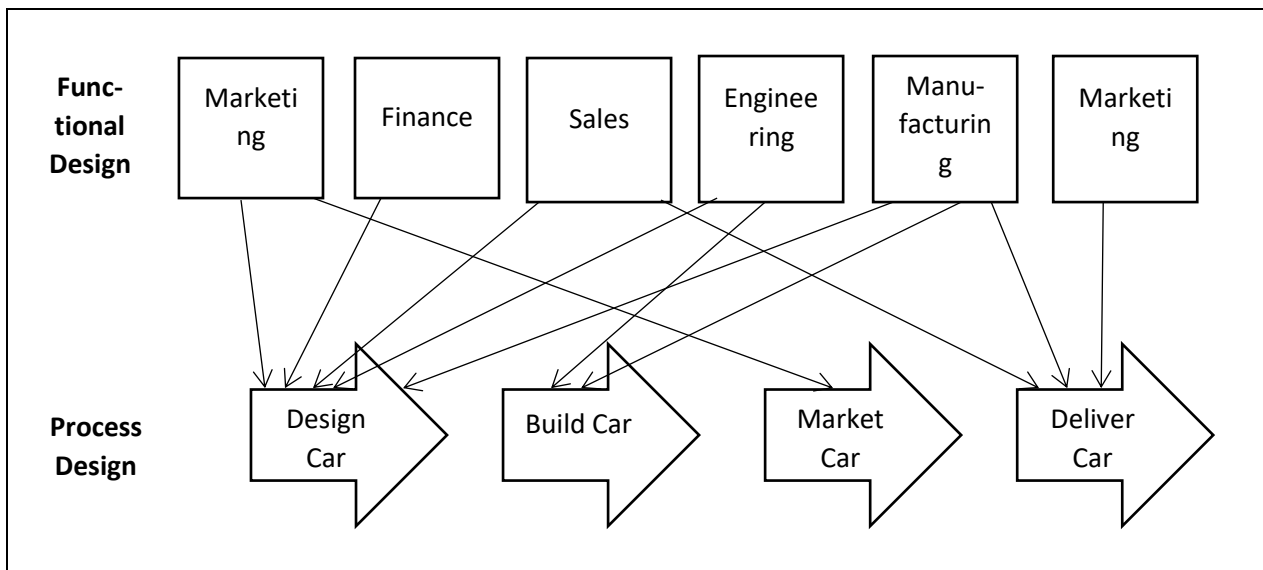


Abbildung 2.5: Beispiel für die Funktions- versus Prozessorientierung in einem Automobilunternehmen (vgl. Schmelzer und Sesselmann [20, S. 54])

Schnittstellen zwischen den einzelnen Abteilungen wie beim funktionsorientierten Ansatz existieren beim prozessorientierten Ansatz nicht. Durch Prozessmodellierung unter dem Aspekt des prozessorientierten Ansatzes versuchen die Unternehmen die stetig wachsenden und komplexen Aufgaben zu bewältigen. Nach DIN EN ISO 9000:2005 ist der Zweck des prozessorientierten Ansatzes, die Wirksamkeit und die Effizienz einer Organisation bei der Erreichung ihrer festgelegten Ziele zu verbessern. [1, 75f] Wie in der Abbildung 2.4 gut zu erkennen ist, verlaufen die Prozesse über die einzelnen Funktionen und Abteilungen hinweg. Auf diese Weise führen sie zu einer höheren Wertschöpfung, da der einzelne Mitarbeiter Einblick auf das Gesamtbild hat.

In der Tabelle 2.2 sind die Merkmale des funktionsorientierten und prozessorientierten Ansatzes in der Unternehmensorganisation, welche teilweise schon gut in der Abbildung 2.4 zu erkennen sind, nochmals zusammengefasst.

Funktionsorientierter Ansatz	Prozessorientierter Ansatz
Vertikale Ausrichtung	Horizontale Ausrichtung
Starke Arbeitsteilung	Arbeitsintegration
Verrichtungsorientierung	Objektorientierung
Tiefe Hierarchie	Flache Hierarchie
Statusdenken	Unternehmerisches Erfolgsdenken
Machtorientierung	Kunde- und Terminorientierung
Abteilungsziele	Ziel: Kundenzufriedenheit, Produktivität
Zentrale Fremdkontrolle	Dezentrale Kontrolle
Kontrollierte Informationen	Freie und offene Informationen
Ratioprojekte	Kontinuierliche Verbesserung
Ersatzprozesse, Redundanz	Konzentration auf Wertschöpfung
Komplexität	Transparenz

Tabelle 2.2: Merkmale der Funktionsorganisation vs. Merkmale der Prozessorganisation (vgl. Schmelzer und Sesselmann [1, S. 73])

Bei der Prozessmodellierung werden wichtige Abläufe, die der Erreichung eines bestimmten meist vom Unternehmen vorgegebenen Ergebnis dienen, im Unternehmen identifiziert. Nicht optimale und überflüssige Prozesse werden erkannt, verbessert und korrigiert bzw. beseitigt. Dadurch werden Ressourcen wieder frei, welche wiederum besser eingesetzt werden können. Eine Folge daraus ist die zeitgleiche Steigerung der Effizienz und Effektivität. Durch die Prozessmodellierung werden die Abläufe für den einzelnen Mitarbeiter transparenter, wodurch jedem klar ist, was zu erledigen ist und wer dafür verantwortlich ist. Das gewährleistet verbesserte, beständige sowie vorhersehbare Ergebnisse. Der Organisation wird zusätzlich ermöglicht Defizite zu erkennen und diesen entsprechend entgegenzuwirken. Weiterhin lässt sich auf verändernden Kundenpräferenzen und auf unerwartet eintretende Situationen schneller reagieren. Dies führt zu einem wachsenden Vertrauen beim Kunden, bei gleichzeitiger Senkung der Kosten und dem effizienten Einsatz der verfügbaren Ressourcen. [17, S. 2f] Auf den ersten Blick bedeutet die Prozessdarstellung zwar einen Mehraufwand, doch zahlt sich dieser auch langfristig gesehen aus. Das Unternehmen, das die Prozessorientierung und Prozessdarstellung nutzt, erhält gegenüber anderen einen Wettbewerbsvorteil, da es auf verändernde Anforderungen schnell reagieren und Anpassungen vornehmen kann.

3 Die Modelliersprachen EPK und BPMN

In diesem Kapitel werden die Modelliersprachen EPK und BPMN genauer betrachtet. Doch zuvor wird die Object Management Group (OMG), welche die BPMN zu ihren Standards zählt, vorgestellt. Dabei wird erläutert wie die OMG einen Standard wie die BPMN erhält und pflegt. Weiterhin wird das ARIS-Konzept vorgestellt und eine Einordnung der Modelliersprachen in diesem vorgenommen. Danach werden die Modellsprachen genauer betrachtet. Zuerst erfolgt jeweils ein kurzer Einblick in ihre Geschichte und ihren Zweck. Außerdem werden die wichtigsten Notationselemente der beiden Modelliersprachen kurz vorgestellt. Anhand des Workflow Pattern Framework und der Bewertungskriterien von Bösing, welche auch kurz erläutert werden, erfolgt eine kritische Auseinandersetzung bezüglich der Eignung, Prozesse darstellen zu können.

3.1 Die Object Management Group (OMG)

Die Object Management Group (OMG) wurde im Jahr 1989 von elf Unternehmen gegründet und ist eine internationale Non-Profit Organisation, welche sich mit Technologiestandards beschäftigt. Zu den Gründungsunternehmen gehören unter anderem die International Business Machines Corporation (IBM), die Apple Inc. und Sun Microsystems. Der Hauptsitz des Konsortiums ist derzeit Needham, Massachusetts, USA. Auf der Website der OMG ist folgendes Mission Statement zu finden:

OMGs Mission ist es, mit Hilfe unserer weltweiten Mitglieder Enterprise Integration Standards zu entwickeln, die einen Wert für die reale Welt schaffen. Die OMG hat sich ebenfalls dem Zusammenbringen von Endnutzern, Regierungsbehörden, Universitäten und Forschungseinrichtungen in unseren Praxisgemeinschaften verschrieben, um Erfahrungen beim Übergang zu neuen Management- und Technologie-Ansätzen, wie dem Cloud-Computing, zu teilen. [21]

Die Mitglieder der OMG bestehen aus hunderten von Unternehmen aus den verschiedensten Bereichen, wie Finanz-, Gesundheitswesen als auch der Automobilindustrie und dem Versicherungswesen. Außerdem ist nach Angabe der OMG fast jede große Organisation der Technologieindustrie vertreten. Viermal im Jahr veranstaltet die OMG eine technische Konferenz, auf der sich die Mitglieder der Arbeitsgruppen treffen und austauschen. Auch OMG-Mitglieder und andere Interessierte können an den Konferenzen teilnehmen und sich über die aktuelle Arbeit informieren. [22, S. 1ff]

Sogenannte Task Forces (Arbeitsgruppen) bestehend aus Herstellern, Anwendern, akademischen Instituten und Regierungsbehörden entwickeln in Kooperation miteinander die Standards bzw. die Spezifikationen der OMG. Ein neu entwickelter Standard wird erst freigegeben, wenn dieser erst einen mehrstufigen Prozess durchlaufen hat. In diesem wird ermittelt, ob sich der Standard implementieren lässt und auch in der Praxis benötigt wird. [21]

Der erste Schritt bei dieser Entscheidungsfindung ist die Request of Information (zu Deutsch: Informationsanfrage; Kurz: RFI) und ist optional. Dabei werden einer Arbeitsgruppe die benötigten Informationen zur Verfügung gestellt, um ein industrielles Problem zu lösen. Der Gruppen werden allgemeine und spezifische Informationen über die Anforderungen der Industrie vermittelt. Weiterhin wird gegebenenfalls Hilfe in Form von technischen Ressourcen zur Verfügung gestellt. Auf diese Weise kann die Arbeitsgruppe ihren Plan zur Lösungsvorstellung validieren und bei Bedarf überarbeiten und anpassen.

Der zweite Schritt des mehrstufigen Prozesses ist die Request for Proposal (zu Deutsch: Angebotsanfrage; Kurz: RFP). In diesem Schritt werden die Bedürfnisse der Industrie identifiziert und die Software-Entwickler eingeladen eine Lösung für das Problem zu entwickeln. Dies stellt zugleich den „Höhepunkt des Erfahrungsaustausches einer technischen OMG-Gruppe (sei es eine Task Force oder eine spezielle Interessengruppe)“ dar. Vor der Anfertigung einer RFP ist es notwendig zu verifizieren, ob und dass es jemanden gibt, der darauf reagiert und sich verpflichtet, eine Lösung zu implementieren. Durch den Letter of Intent (zu Deutsch: Absichtsbekundung; Kurz: LOI) von mindestens einem Mitglied des OMG wird dies gewährleistet. Gleichzeitig verdeutlicht das Unternehmen mit dem LOI seine Absicht eine Lösung für die in der RFP genannten Bedürfnisse zu entwickeln. Wenn ein Unternehmen den LOI eingereicht hat und sich die Absicht zur Problemlösung erkennen lässt, wird das RFP angefertigt. [23, S. 1]

Anschließend hat das Unternehmen bis zu drei Wochen vor der nächsten technischen Konferenz Zeit eine erste Vorlage für die Umsetzung einzureichen. Während der Konferenz wird die erste Vorlage von interessierten OMG Mitgliedern diskutiert und kommentiert. Dem Unternehmen wird danach Zeit gegeben, um eine überarbeitete und verbesserte Version vorzulegen. Diese wird ebenfalls von den OMG Mitglieder gelesen und evaluiert. Wenn die Mehrheit der Mitglieder die überarbeitete Version für würdig hält, erfolgt eine Reihe von Abstimmungen über die Annahmen der OMG-Spezifikation. Die Abstimmung findet während der Konferenz statt, die unmittelbar der Einreichungsfrist der überarbeiteten Version folgt. Wenn keine Ablehnung durch eine höhere Instanz erfolgt, bekommt der Vorschlag den Status „angenommene Spezifikation“, hat jedoch keine Versionsnummer.

Nachdem der Vorschlag den Status „angenommene Spezifikation“ erhalten hat, wird eine finale Arbeitsgruppe gebildet. In dieser werden erste Wartungsarbeiten an der Spezifikation durchgeführt und eine Implementierung erstellt. Die von der Finalen Task Force überarbeitete Version durchläuft nochmals einen Abstimmungsprozess und wird so durch die höheren Instanzen bestätigt. Weiterhin erhält die Version abschließend eine Versionsnummer und den Status „gültige Spezifikation“. In diesem Zeitraum erscheinen die Produkte auf dem Markt, wodurch gesichert wird, dass die Spezifikation auch wirklich genutzt und nicht nur entwickelt, dann aber vergessen wird.

Mit Hilfe des Wartungskreislaufes gewährleistet die OMG, dass die Spezifikationen beziehungsweise Standards aktuell bleiben und auch benutzt werden. Dabei wird eine Arbeitsgruppe gebildet, die die Probleme und Belange bezüglich des Standards sammelt und behandelt. Wenn die Notwendigkeit besteht, wird eine neue Version veröffentlicht. [23] Unter anderem gehören die Modelliersprachen UML und BPMN zu den Standards der OMG.

3.2 ARIS-Konzept

Das ARIS-Konzept wurde 1992 von August-Wilhelm Scheer am Institut für Wirtschaftsinformatik an der Universität des Saarlandes entwickelt. Ziel des Konzeptes war es eine „gemeinsame Sprache für IT und Management“ zu entwickeln. Der Anspruch war es, Fragen der Organisation, der Funktionalität der benötigten Daten und der zu erbringenden Leistung eines Geschäftsprozesses eines Geschäftsprozesses einzeln zu betrachten, um so die die Komplexität zu reduzieren. Andererseits sollte es möglich sein die Zusammenhänge zwischen den einzelnen statischen Sichten in einer dynamischen Sicht, der Prozess- oder Steuerungssicht darzustellen. [24, S. 30] Das ARIS-Konzept besteht, wie in der Motivation schon beschrieben aus fünf unterschiedlichen Sichten, welche in Abbildung 3.1 nochmals dargestellt sind.

Die Organisationssicht beschreibt die Organisationseinheiten und Mitarbeiter eines Unternehmens sowie ihre Beziehungen und Strukturen zueinander. Die Datensicht hingegen dient der Beschreibung von Informationsobjekten und deren Beziehung zueinander. In der Funktionssicht werden die ausführenden Vorgänge und ihre hierarchischen Zusammenhänge eines Unternehmens beschrieben. In der Leistungssicht werden materielle und immaterielle Leistungen beschrieben, die im Verlauf des Wertschöpfungsprozesses erbracht werden. Die Prozess-/Steuerungssicht dient als Verbindungsglied der anderen einzelnen Sichten und beschreibt die Zusammenhänge und Beziehungen zwischen diesen. [24, S. 31f]

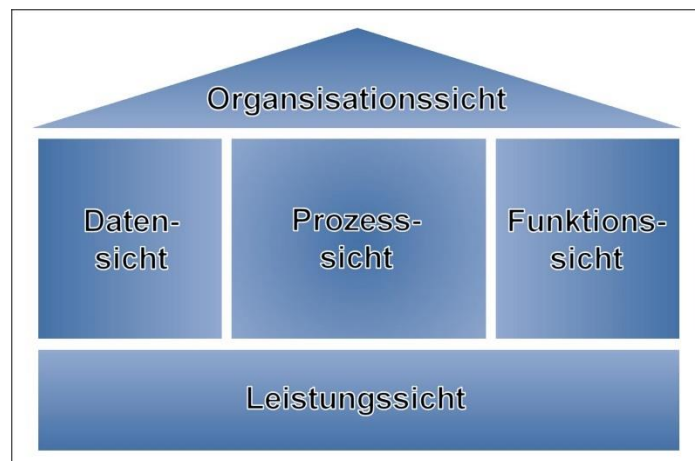


Abbildung 3.1: ARIS-Haus mit den fünf Sichten

Die vorgestellten fünf Sichten des ARIS-Hauses unterteilen sich wiederum in die drei Beschreibungsebenen Fachkonzept, DV-Konzept und Implementierung um die Schrittweise Durchführung von Projekten herauszustellen. Die Abbildung 3.2 stellt diese Unterteilung nochmals grafisch dar. Im Fachkonzept wird der Ist-Zustand und der Soll-Zustand in Modellen formalisiert beschrieben, welches als Ausgangspunkt für eine konsistente Umsetzung in eine informationstechnische Anwendung dient. Werden die Inhalte des Fachkonzepts in die Welt der Datenverarbeitung übertragen so geschieht dies in dem DV-Konzept. Hier wird die organisatorische Beschreibung der Zustände in die Sprache der Informationstechnik umgesetzt. In der Implementierungsebene wird das DV-Konzept konkret durch Hardware- und Software-Komponenten realisiert. [24, S. 33f]

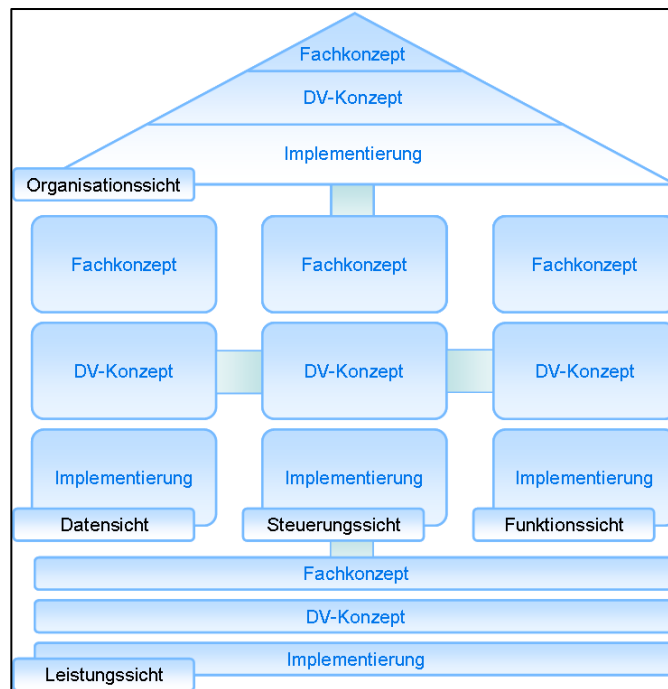


Abbildung 3.2: ARIS-Haus mit Unterteilung in Beschreibungsebenen

Einordnung der EPK und der BPMN in das ARIS Haus

Wie im einleitenden Kapitel schon erwähnt, befasst sich die Arbeit mit der Prozess-/Steuerungssicht. Die Abbildung 3.3 zeigt eine genauere Einordnung der zu behandelnden Modelliersprachen BPMN und der EPK im ARIS-Haus. Anhand der Darstellung lässt sich gut erkennen, dass sowohl die BPMN als auch die EPK in der Fachkonzept-Ebene angewendet werden. Dies zeigt, dass eine Transformation zwischen den beiden Modelliersprachen in Bezug auf das ARIS-Haus sinnvoll ist. Weiterhin verdeutlicht die Abbildung auch, dass die BPMN ebenfalls in der DV-Konzept Ebene verwendet werden kann, wodurch die Möglichkeit besteht, dass beide Modelle mit jeweils unterschiedlichen Schwerpunkten nebeneinander bestehen können. Die EPK hat ihren Schwerpunkt dabei in der Beschreibung der betriebswirtschaftlichen Sicht wohingegen die BPMN ihren in der formalen Beschreibung mit Bezug zur IT aufweist.

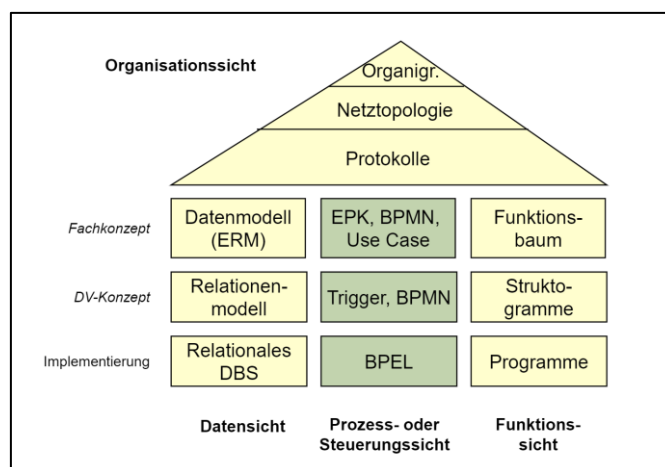


Abbildung 3.3: Einordnung EPK und BPMN im ARIS-Haus (vgl. [25, S. 14])

3.3 Ereignisgesteuerter Prozessketten (EPK)

3.3.1 Ursprung und Zweck

Ihren Ursprung findet die Ereignisgesteuerte Prozesskette (EPK) im Jahr 1992 als sie erstmalig von Keller, Nüttgens und Scheer im Rahmen des ARIS-Konzepts entwickelt und dann vorgestellt wurde. [26, S. 59, 27]. Die EPK ist eine der Hauptkomponenten der Architektur Integrierter Informationssysteme (ARIS) und beschreibt wie schon erwähnt dort die Steuerungs- bzw. Prozesssicht. Sie ist auch ein zentraler Bestandteil des SAP-Konzepts für die Modellierung von Unternehmensstrukturen und -abläufen [26, S. 33] und wird dadurch auch zu großen Teilen im Bereich des industriellen Geschäftsprozessmanagements eingesetzt. [27]

Als Grundlage für die EPK dienten Petri-Netze und wurde nach und nach um neue Symbole und Semantik erweitert, sodass man in der heutigen Literatur sowohl ereignisgesteuerte Prozessketten (EPK) als auch erweiterte ereignisgesteuerte Prozessketten vorfindet. Jedoch werden diese Begriffe heute als synonym zueinander verwendet. [18, S. 43]

Bei der EPK handelt es sich um eine semi-formale Modellierungssprache und dient zur Modellierung von Prozessen. [18, S. 43, 26, S. 59] Wie auch bei Petri-Netzen handelt es sich bei der EPK um eine Modellierungstechnik, welche gerichtete Grafen verwendet. Im Kern beinhaltet sie die Elemente Funktion, Ereignis, Kontrollfluss (gerichtete Kanten) und Konnektoren. Im folgenden Abschnitt werden die Notationen der EPK genauer beleuchtet.

3.3.2 Notationselemente der EPK

In der Tabelle 3.1 sind die wichtigsten Notationselemente mit einer kurzen Beschreibung dargestellt. Eine ausführlichere Auflistung und detailliertere Beschreibung der verwendeten Notationselemente befindet sich im Anhang.

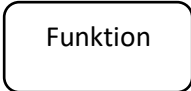
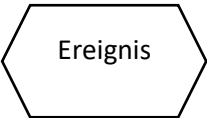
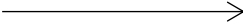
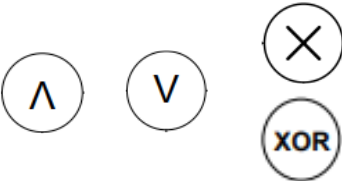


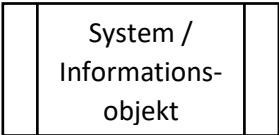

Element	Kurzbeschreibung
	Die Funktion steht für eine Tätigkeit, welche einen Input und Output transformiert. Mit ihr werden die Bearbeitungsschritte des Prozesses abgebildet. Sie werden durch ein auftretendes Ereignis durchgeführt und lösen neue Ereignisse aus.
	Jeder Prozess beginnt und endet mit mindestens einem Ereignis . Es beschreibt einen eingetretenen Systemzustand, dem eine oder mehrere Funktionen folgen.
	Der Kontrollfluss setzt die Funktionen und Ereignisse gerichtet miteinander in Verbindung, wodurch sachlogische und zeitliche Abhängigkeiten abgebildet werden.
	Mit Hilfe der Konnektoren können Prozessverzweigungen modelliert werden. Man unterscheidet dabei 3 Arten von Konnektoren: das logische UND , INKLUSIVE ODER und EXKLUSIVE ODER .
	Eine organisatorische Rolle beschreibt einen Tätigkeitsbereich, der von einer oder von mehreren Personen, die diesem Bereich zugeordnet ist, ausgeführt wird.
	Eine organisatorische Stelle beschreibt einen Tätigkeitsbereich, der von einer konkreten Person ausgeführt wird.
	Ein System modelliert ein Softwaresystem, das für eine bestimmte Funktion oder Anwendungsdomäne erstellt wurde. Das Informationsobjekt liefert die für die Durchführung einer Funktion benötigten Daten.
	Die Prozess-Schnittstelle dient dazu, den Gesamtprozess in Teilprozesse zu untergliedern und verweist auf einen anderen Teilprozess. Die Prozessschnittstelle ist dabei weder Funktion noch Ereignis und kann daher nicht alleine genutzt werden.

Tabelle 3.1: Übersicht der EPK Notationselemente (vgl. Baumgartner [28, S. 6])

Außerdem sind bei der Modellierung von EPK-Modellen die in den Abbildung 3.4 bis Abbildung 3.11 dargestellten Verknüpfungsregeln der UND, OR und XOR Konnektoren zu beachten.

Und-Regeln:

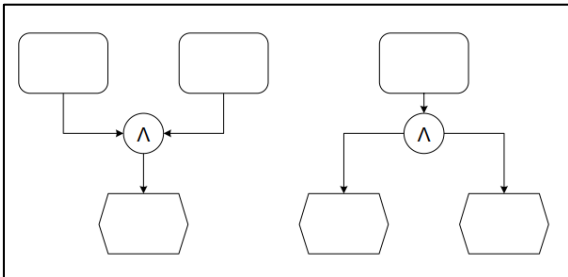


Abbildung 3.4: UND-Regel mit eingehender Funktion und ausgehendem Ereignis

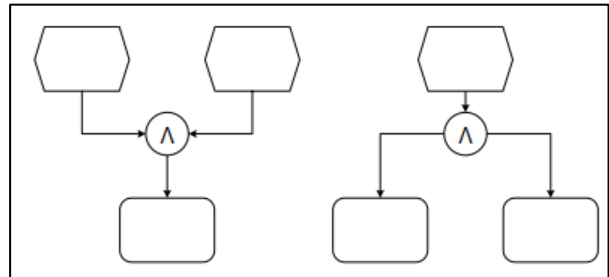


Abbildung 3.5: UND-Regel mit eingehendem Ereignis und ausgehender Funktion

Oder-Regeln:

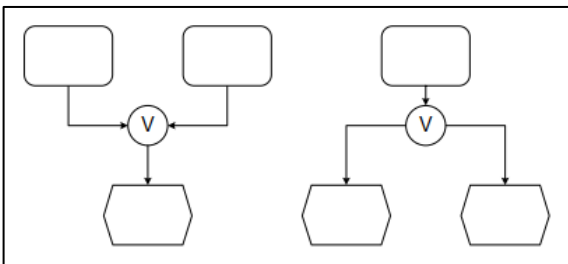


Abbildung 3.6: ODER-Regel mit eingehender Funktion und ausgehendem Ergebnis

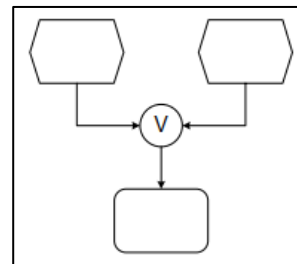


Abbildung 3.7: ODER-Regel mit eingehendem Ereignis und ausgehender Funktion

XOR-Regeln:

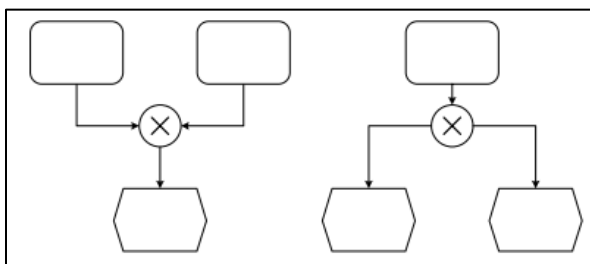


Abbildung 3.8: XOR-Regel mit eingehender Funktion und ausgehendem Ereignis

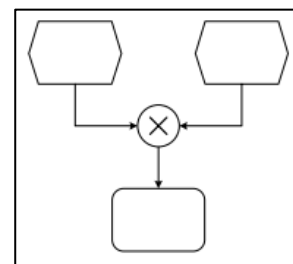


Abbildung 3.9: XOR-Regel mit eingehendem Ereignis und ausgehender Funktion

Verbotene Verknüpfungen:

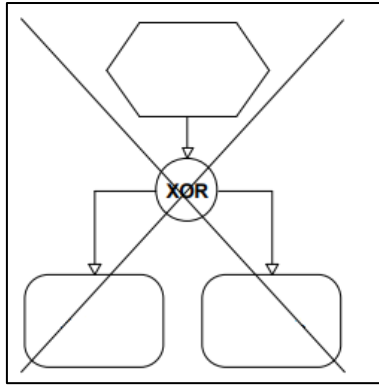


Abbildung 3.10: Verbotene XOR-Verknüpfung mit eingehendem Ereignis und ausgehender Funktion

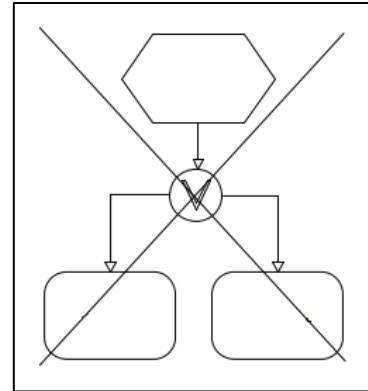


Abbildung 3.11: Verbotene OR-Verknüpfung mit eingehendem Ereignis und ausgehender Funktion

3.4 Business Process Model and Notation (BPMN)

3.4.1 Ursprung und Zweck

Die Business Process Model and Notation (BPMN) wurde vom IBM-Mitarbeiter Stephen A. White entwickelt und im Jahr 2004 von der Business Process Management Initiative (BPMI) in der Version 1.0 veröffentlicht. Im Folgejahr übernahm die OMG die BPMI und schon im Jahr 2006 wurde die BPMN zu einem OMG-Standard. [29, S. 8f] Dank der stetigen Weiterentwicklung der BPMN durch die OMG konnte im Jahr 2008 die Version 1.1 und im Jahr darauf die Version 1.2 veröffentlicht werden. 2011 brachte die OMG dann die BPMN in der Version 2.0 heraus [21], welche derzeit als „State of the Art bei der Modellierung von Geschäftsprozessen und deren Automatisierung“ gilt. [19, V]

„Das primäre Ziel der BPMN Anstrengungen war es, eine Notation zur Verfügung zu stellen, die leicht verständlich für alle Business-Anwender ist; von den Business-Analysten, welche die ersten Konzepte für die Prozesse entwerfen, über die technischen Entwickler, die verantwortlich für die Umsetzung der Technologie sind, die diese Prozesse durchführt, bis hin zu den Geschäftsleuten, die diese Prozesse verwalten und überwachen.“ [30, S. 1]

Die BPMN wurde entwickelt, um die Lücke zwischen den einzelnen Parteien zu schließen und die Geschäftsprozesse für alle verständlich zu modellieren. Durch diesen Ansatz unterscheidet sie sich beispielsweise von der UML und der EPK. Durch ihre grafische Notation sollte dies bewerkstelligt werden. [19, S. 11]

Die BPMN enthält auch weitere Diagrammtypen, wie die Konversation und die Choreographie. [29, S. 142ff, 31, S. 31] Diese Typen von Diagrammen werden in dieser Arbeit jedoch nicht betrachtet. Wenn im weiteren Verlauf der Arbeit von „BPMN“ die Rede ist, bezieht sich dies ausschließlich auf die Prozessdiagramme.

3.4.2 Notationselemente der BPMN

Mit der Verabschiedung der BPMN in Version 2.0 durch die OMG unterteilen sich die Notationselemente in fünf Basiskategorien: Flussobjekt, Daten, Verbindungsobjekt, Teilnehmer und Artefakte. [31, S. 27] Durch die geringe Anzahl an Kategorien kann der Leser die Basistypen der Elemente leicht erkennen und somit die Diagramme schneller verstehen. In der Tabelle 3.2 sind die einzelnen Elemente der fünf Kategorien zugeordnet. [31, S. 27f]

Kategorie	Enthaltene Elemente
Flussobjekt	Ereignisse, Aktivitäten, Gateways
Daten	Datenobjekte, Dateneingang, Datenausgang, Datenspeicher
Verbindungsobjekt	Sequenzflüsse, Nachrichtenflüsse, Assoziationen, Datenassoziationen
Teilnehmer	Pools, Lanes
Artefakte	Gruppe, Anmerkungen

Tabelle 3.2: Kategorien und enthaltene Elemente in der Übersicht

Tabelle 3.3 zeigt die wichtigsten Elemente der BPMN mit einer kurzen Beschreibung. Alle weiteren Elemente der BPMN, die in dieser Arbeit verwendet werden, befinden sich im Anhang. Außerdem stehen weiterführende Informationen zu den einzelnen Elementen in den Spezifikationen zur BPMN von der OMG bereit. [31, S. 29ff]








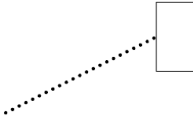
Element	Kurzbeschreibung
 Ereignis	Ein Ereignis ist etwas, das im Verlauf des Prozesses passiert. Die Ereignisse beeinflussen den Ablauf eines Modells und haben in der Regel eine Ursache (Trigger) oder eine Auswirkung (Ergebnis).
 Aktivität	Eine Aktivität ist wie ein Oberbegriff für Arbeit, die eine Firma ausführt. Eine Aktivität kann atomar oder nicht-atomar (zusammengesetzt) sein.
 Gateway	Ein Gateway wird benutzt, um Divergenz und Konvergenz der Sequenzflüsse in einem Prozess zu kontrollieren. Auf diese Weise wird die Verzweigung und Zusammenführung der Pfade bestimmt.
 Sequenzfluss	Der Sequenzfluss wird benutzt, um eine Reihenfolge darzustellen, in der Aktivitäten in einem Prozess ausgeführt werden.
 Pool	Ein Pool ist die graphische Darstellung eines Teilnehmers. Ein Pool kann interne Details in Form eines Prozesses, der ausgeführt wird, enthalten.
 Pool mit Lanes	Eine Lane ist eine Sub-Aufteilung innerhalb eines Pools. Lanes werden verwendet, um die Aktivitäten zu organisieren und zu kategorisieren.
 Datenobjekt	Ein Datenobjekt stellt Informationen darüber zur Verfügung, was eine Aktivität benötigt, um ausgeführt zu werden und / oder was diese erzeugt.
 Anmerkung	Eine Anmerkung ist ein Mechanismus für einen Modellierer, um zusätzliche Textinformationen für den Leser eines Diagramms zur Verfügung zu stellen.

Tabelle 3.3: kurze Übersicht der BPMN Notationselemente (vgl. OMG – Business Process Model and Notation [31, S. 28f])

Die Abbildung 3.12 zeigt einen simplen Beispielprozess mit zwei Notationselementen der BPMN. Hierbei wird das Ereignis in der einfachsten Form (Durstig), die Aktivität (Wasser trinken) und einem durch einen dickeren Rand variierten Ereignis (Durst gelöscht) verwendet.

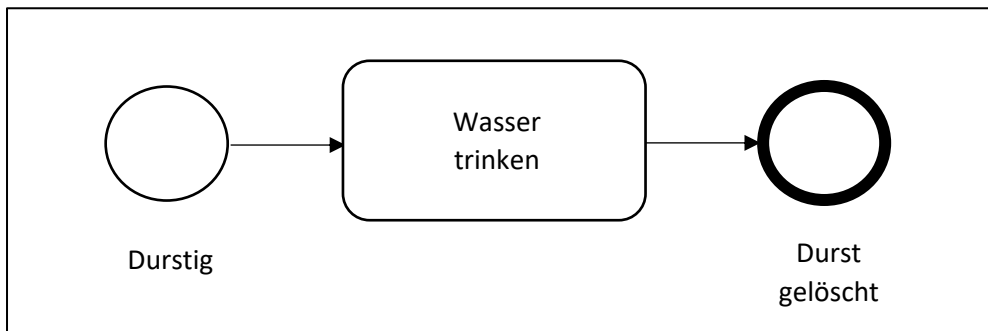


Abbildung 3.12: Simpler Beispielprozess in BPMN

3.5 Bewertungsmethoden für die Eignung zur Prozessdarstellung

Es stellt sich nun die Frage inwieweit die beiden Modelliersprachen zur Prozessdarstellung geeignet sind und welche Bewertungsmethoden man dafür heranzieht. Bösing et. al. und Wohed et al. haben dabei unterschiedliche Ansätze verfolgt, welche sich auch in ihrer Komplexität unterscheiden.

Um die EPK und die BPMN voneinander abzugrenzen hat Bösing et al. eindeutige Kriterien formuliert und die Modelliersprachen einem Vergleich unterzogen. Der Fokus liegt dabei auf eine mögliche Automatisierung, welches gleichzeitig eines der Hauptkriterien ist. Ein weiterer Maßstab für die Bewertung ist die gegebene Formalität der Sprachen, welche wichtig für die Weitergabe von Informationen ist. Da die Prozessmodellierung zunehmend fachübergreifend ist und somit auch in ihrer Komplexität steigt, wurden ebenfalls die Kriterien der Übersichtlichkeit und der einfachen Handhabung als herangezogen. Damit alle Unternehmen auf der gleichen Grundlage bei Prozessmodellierung arbeiten können, ist die Standardisierung das ebenfalls ein Unterscheidungskriterium von Bösing.[32, S. 100]

Wohed et al. haben eine weitaus komplexere Bewertungsmethode genutzt und die Ergebnisse veröffentlicht [33]. Dabei haben sie die Modellsprachen gegen die von Wil van der Aalst und Arthur ter Hofstede entwickelten Workflow Pattern getestet. Die Workflow Pattern und die dahinterstehende Workflow Pattern Initiative wird im folgenden Abschnitt 3.5.1 genauer beleuchtet.

3.5.1 Workflow Pattern Initiative

Die Workflow Pattern Initiative ist eine im Jahr 1999 von der Technischen Universität Eindhoven (unter Führung von Professor Will van der Aalst) und der Technischen Universität Queensland (unter Führung von Professor Arthur ter Hofstede) gemeinsames Projekt. „Das Ziel der Initiative ist es, eine konzeptuelle Grundlage für die Prozesstechnik anzubieten.“ [34] Dabei werden verschiedene Perspektiven überprüft, die von einer Arbeitsablauf- oder einer Geschäftsprozessmodellierungssprache unterstützt werden müssen, wie zum Beispiel der Kontrollfluss.

Der Workflow Pattern Framework (WPF) umfasst zum Zeitpunkt der Veröffentlichung 20 Kontrollflussmuster (Control Flow Patterns), 40 Datenmuster (Data Patterns) und 43 Ressourcenmuster (Resource Patterns). Die Kontrollflussmusterperspektive befasst sich mit den Aspekten, welche mit den Abhängigkeiten des Kontrollflusses zwischen den verschiedenen Aufgaben zusammenhängen. Die Datenperspektive umfasst die Weitergabe von Information und den Umfang der Variablen, wohingegen sich die Ressourcenperspektive mit der Aufteilung der Ressourcen für die entsprechenden Aufgaben und der Delegation befasst.

Die Kontrollflussmuster werden in sechs Gruppen aufgeteilt: *Basismuster* (Basic Control Flow Pattern), *erweiterte Verzweigung und Synchronisation* (Advanced Branching and Synchronization), *Muster für Multi-Instanzen* (Multiple Instance Patterns), *statusbasierte Muster* (State-based Patterns), *Abbruchmuster* (Cancellation Patterns) und *strukturelle Muster* (Structural Patterns). Die Basismuster umfassen die Aspekte der Prozesskontrolle wie die Reihenfolge der einzelnen Aufgaben. Die Muster der erweiterten Synchronisation dienen hingegen der Erweiterung des Prozessflusses durch Verzweigungen und Synchronisationen. Einschränkungen bezüglich der Art und Weise wie Prozesse gestaltet werden können, prüfen die strukturellen Muster. Die statusbasierten Muster reflektieren Situationen, in denen der weitere Prozessablauf vom Status der Prozessinstanz abhängig ist. Die Muster für Multi-Instanzen beschreiben Situationen, in denen mehrere Ausführungsthreads zur gleichen Zeit in einem Prozessmodell aktiv sind und sich diese auf die gleiche Tätigkeit beziehen. Die

Abbruchmuster stellen die Fähigkeit dar, den Abbruch einer einzelnen Aktivität oder eines ganzen Prozesses zu verwirklichen. [35]

In den Tabelle 3.4 und Tabelle 3.5 sind die 20 Kontrollflussmuster mit je einer Kurzbeschreibung aufgeführt. Um Übersetzungsfehler vorzubeugen wurde auf eine Übersetzung der einzelnen Musterbezeichnungen verzichtet.

Bezeichnung	Kurzbeschreibung
Basic Controll Flow	
1. Sequence	Eine Aufgabe im Prozess wird erst aktiviert, wenn die vorhergehende Aufgabe abgeschlossen ist.
2. Parallel Split	Eine Aufspaltung des Kontrollflusses in zwei oder mehrere parallele Kontrollflüsse findet statt.
3. Synchronization	Es findet eine Zusammenführung von zwei oder mehreren Kontrollflüssen zu einem Kontrollfluss statt. Es muss auf alle eingehenden Kontrollflüsse gewartet werden.
4. Exclusive Choice	Aus mehrere Kontrollflussalternative wird genau eine ausgewählt, um den Kontrollfluss fortzusetzen.
5. Simple Merge	Es findet eine Zusammenführung von zwei oder mehreren Kontrollflüssen zu einem Kontrollfluss statt.
Advanced Synchronisation	
6. Multiple Choice	Aus mehreren Alternativen werden maximal alle ausgewählt, um den Kontrollfluss fortzusetzen.
7. Synchronising Merge	Es findet eine Zusammenführung von zwei oder mehreren Kontrollflüssen zu einem Kontrollfluss statt. Es muss auf alle eingehenden Kontrollflüsse gewartet werden. Das Muster bezieht sich auf einen eindeutig identifizierbaren vorausgegangenen Punkt im Prozessablauf.
8. Multiple Merge	Es findet eine Zusammenführung von zwei oder mehreren Kontrollflüssen zu einem Kontrollfluss statt.
9. Discriminator	Von mehreren, konkurrierenden Alternativen wird genau die erste ankommende akzeptiert. Die Restlichen werden verworfen.
Structural Patterns	
10. Arbitrary Cycles	Fähigkeit zur Darstellung einer Schleife, welche über mehrere Punkte betreten und verlassen werden kann.
11. Implicit Termination	Ein Prozess (oder Sub-Prozess) wird beendet, wenn es keine weiteren Ausführungsoptionen gibt.

Tabelle 3.4: Kontrollflussmuster des WPF in einer Kurzübersicht (vgl. WPI [36])

Bezeichnung	Kurzbeschreibung
Multiple Instances Pattern	
12. MI without Synchronisation	Von einer Aufgabe können mehrere, voneinander unabhängige Instanzen erzeugt werden. Es besteht keine Forderung zur abschließenden Synchronisation
13. MI with a priori Design Time Knowledge	Von einer Aufgabe können mehrere, voneinander unabhängige Instanzen erzeugt werden. Die Anzahl der Instanzen ist zum Zeitpunkt der Erstellung bekannt und es ist eine abschließende Synchronisation erforderlich, bevor weitere Aufgaben begonnen werden können.
14. MI with a priori Run Time Knowledge	Von einer Aufgabe können mehrere, voneinander unabhängige Instanzen erzeugt werden. Die Anzahl der Instanzen ist abhängig von Faktoren, die erst während der Laufzeit des Prozesses bekannt werden. Es ist eine abschließende Synchronisation erforderlich, bevor weitere Aufgaben begonnen werden können.
15. MI without a priori Run Time Knowledge	Von einer Aufgabe können mehrere, voneinander unabhängige Instanzen erzeugt werden. Die Anzahl der Instanzen ist abhängig von Faktoren, die erst während der Laufzeit des Prozesses bekannt werden. Es können jederzeit weitere Instanzen erzeugt werden. Es ist eine abschließende Synchronisation erforderlich, bevor weitere Aufgaben begonnen werden können.
State-Based Patterns	
16. Deferred Choice	Aus mehreren Alternativen wird eine abhängig von äußeren Faktoren ausgewählt.
17. Interleaved Parallel Routing	Die Aufgaben können in beliebiger Reihenfolge ausgeführt werden. Es muss jede Aufgabe ausgeführt werden und es kann je nur eine Aufgabe erledigt werden.
18. Milestone	Eine Aufgabe wird erst dann ausgeführt, wenn eine Prozess-Instanz einen bestimmten Status erreicht hat.
Cancellation Patterns	
19. Cancel Activity	Eine Aufgabe wird vor ihrem natürlichen Ende abgebrochen.
20. Cancel Case	Eine Prozessinstanz wird vor ihrem natürlichen Ende abgebrochen. Dies beinhaltet alle laufenden Aufgaben sowie alle nachfolgenden.

Tabelle 3.5: Kontrollflussmuster des WPF in einer Kurzübersicht (Fortsetzung) (vgl. WPI [36])

Eine detaillierte Übersicht über die Daten- und Ressourcenmuster sowie eine genauere Beschreibung befindet sich im Anhang. Diese Muster werden in dieser Arbeit nicht näher betrachtet, weil für die Prozessdarstellung einzig die Kontrollflussmuster von besonderem Interesse sind. Diese können direkt graphisch durch die Modellsprache beschrieben werden. Von besonderem Interesse sind die fünf Basismuster der Kontrollflussmuster, da durch sie die grundlegende Prozessdarstellung sichergestellt wird: Darstellung einer Reihenfolge von Aktivitäten (Sequence), parallele Ausführung von Aktivitäten (parallel Split), Synchronisation (Synchronization), exklusive Auswahl einer Alternative (exclusive Choce) sowie einfache Zusammenführung (Simple Merge). Mit Hilfe der erweiterten Verzweigung und Synchronisation wird der Kontrollfluss besser gelenkt als einzig mit den Basiselementen. Die Mehrfachauswahl ermöglicht es beispielsweise neben der Auswahl eines einzigen oder aller Stränge auch die Auswahl einer Anzahl dazwischen. Der Diskriminator ermöglicht es im Gegenzug, dass von mehreren Alternativen nur die erste fortgeführt wird. Die strukturellen Muster stellen sicher, dass zum Beispiel Schleifen modelliert werden können und dass ein (Sub-)Prozess beendet wird, wenn keine

weitere Ausführung möglich ist. Weniger von Bedeutung für die Prozessdarstellung sind die Multi-Instanzen, weshalb sie im Rahmen dieser Arbeit nicht näher erläutert werden. Die status-basierten Muster greifen Impulse von außerhalb auf. Ein einfaches Beispiel für die verzögerte Entscheidung ist der Anruf eines Kunden, von welchem die Fortführung eines Prozesses abhängig ist. Ein Meilenstein hingegen ist ein besonderer Punkt im Verlauf eines Prozesses: Der nächste Schritt kann erst gemacht werden, wenn der Meilenstein erreicht wurde. Die letzte Gruppe Kontrollflussmuster stellen die Abbruchmuster dar. Diese Muster gewährleisten, dass unter bestimmten Umständen eine Aktivität oder ein Prozess augenblicklich abgebrochen werden kann. [36]

3.5.2 Eignung von EPK zur Prozessdarstellung

Nun stellt sich die Frage, in wie weit sich die EPK, respektive die eEPK, für die Prozessdarstellung eignet. Dieser Frage sind Wohed et al. nachgegangen und haben die EPK gegen die vorgestellten Workflow Pattern getestet. Die Ergebnisse dazu haben sie auf ihrer Website veröffentlicht.

Bewertung nach Workflow Pattern Framework

	EPK
Basic Control Flow	
1. Sequence	+
2. Parallel Split	+
3. Synchronisation	+
4. Exclusive Choice	+
5. Simple Merge	+
Advance Synchronisation	
6. Multiple Choice	+
7. Synchronising Merge	+
8. Multiple Merge	-
9. Discriminator	-
Structural Patterns	
10. Arbitrary Cycles	+
11. Implicit Termination	+
Multiple Instance Patterns	
12. MI without Synchronisation	-
13. MI with a priori Design Time Knowledge	-
14. MI with a priori Run Time Knowledge	-
15. MI without a priori Run Time Knowledge	-
State-Based Patterns	
16. Deferred Choice	-
17. Interleaved Parallel Routing	-
18. Milestone	-
Cancellation Patterns	
19. Cancel Activity	-
20. Cancel Case	-

Tabelle 3.6: Unterstützung der Kontrollflussmuster des Workflow Pattern Frameworks durch die EPK (vgl. WPI, Wohed et al [36])

Die Tabelle 3.6 zeigt die einzelnen Kontrollflussmuster und deren Unterstützung durch die EPK. Ein „+“ in der Tabelle bedeutet die direkte Unterstützung des Musters und ein „-“ eine fehlende Unterstützung. Aus der Tabelle ist gut zu erkennen, dass die ereignisgesteuerten Prozessketten nur die Pattern des Basic Control Flows und die Structural Patterns vollständig unterstützen, wohingegen die Multiple Instance Patterns, State-Based Patterns und die Cancellation Patterns nicht unterstützt werden. Außerdem erhalten nur die Hälfte der Advance Synchronising Patterns in Form von Multiple Choice und Synchronising Merge volle Unterstützung. Multiple Merge und Discriminator erhalten dagegen keine Unterstützung.

Auch wenn die Datenfluss- und Ressourcenmuster nicht näher betrachtet werden, weil sie für die Prozessdarstellung keine tragende Rolle spielen, ist zu erwähnen, dass die EPK keine dieser Muster nach Wohed et. al unterstützt. Trotz der Defizite bei den Daten-, Ressourcenmuster und auch bei den Kontrollflussmustern ist die EPK stark eingeschränkt für die Prozessdarstellung geeignet, da sie die wichtigsten Kontrollflussmuster unterstützt.

Bewertung nach Bösing

Methoden	EPK
Kriterien	
Automatisierung möglich	-
Formalität gegeben	-
Übersichtlichkeit	-
Einfache Handhabung	+
Methode als Standard	-

Tabelle 3.7: Bewertungsübersicht der EPK nach Bösing (vgl. Bösing [32, S. 101])

Die Tabelle 3.7 zeigt die Eignung der EPK nach der Bewertungsmethode von Bösing. Auch hier bedeutet ein „+“ in der Tabelle die Einhaltung des Kriteriums und ein „-“ die Nichterfüllung eines Kriteriums. Wie schon die Eignungsprüfung durch das Workflow Pattern Framework zeigt auch die Bewertung nach Bösing, dass die EPK nur bedingt für die Prozessmodellierung geeignet ist. Lediglich die „Einfache Handhabung“ wird durch die EPK unterstützt. Die restlichen Kriterien werden nicht erfüllt.

Fasst man nun die Ergebnisse der beiden Bewertungsmethoden zusammen, so wird deutlich, dass die EPK nur eingeschränkt für die Prozessdarstellung geeignet ist. Außerdem wird deutlich, dass die EPK eine weniger formelle Modelliersprache ist, was wiederum zu ihrer einfachen Handhabung und einer leichteren Verständlichkeit führt.

3.5.3 Eignung von BPMN zur Prozessdarstellung

Auch bei der BPMN stellt sich die Frage, in wie weit die Sprache für die Prozessdarstellung geeignet ist. Da die Modelliersprache für die Geschäftsprozessdarstellung entwickelt wurde, kann man annehmen, dass sie sich für die Prozessdarstellung eignet. Dieser Frage wurde ebenfalls durch Wohed et. al ausführlich nachgegangen. Wie bei der EPK haben sie die BPMN in der Version 1.0 gegen die vorgestellten Workflow Pattern getestet. Die Ergebnisse wurden sowohl auf der Website als auch in mehreren Publikationen veröffentlicht. [33]

Bewertung nach Workflow Pattern Framework

	BPMN
Basic Control Flow	
1. Sequence	+
2. Parallel Split	+
3. Synchronisation	+
4. Exclusive Choice	+
5. Simple Merge	+
Advance Synchronisation	
6. Multiple Choice	+
7. Synchronising Merge	+ / -
8. Multiple Merge	+
9. Discriminator	+
Structural Patterns	
10. Arbitrary Cycles	+
11. Implicit Termination	+
Multiple Instance Patterns	
12. MI without Synchronisation	+
13. MI with a priori Design Time Knowledge	+
14. MI with a priori Run Time Knowledge	+
15. MI without a priori Run Time Knowledge	-
State-Based Patterns	
16. Deferred Choice	+
17. Interleaved Parallel Routing	+ / -
18. Milestone	-
Cancellation Patterns	
19. Cancel Activity	+
20. Cancel Case	+

Tabelle 3.8: Unterstützung der Kontrollflussmuster des Workflow Pattern Framework durch die BPMN (vgl. WPI [36], Wohed et al [33, S. 13])

In der Tabelle 3.8 sind die einzelnen Kontrollflussmuster sowie deren Unterstützung durch die BPMN abgebildet. Ein „+“ in der Tabelle bedeutet die direkte Unterstützung des Musters, ein „+/-“ eine teilweise Unterstützung und ein „-“ eine fehlende Unterstützung. Aus der Tabelle ist deutlich zu lesen, dass die BPMN fast alle Kontrollflussmuster direkt oder wenigstens teilweise unterstützt. Keine volle Unterstützung erhalten die Multi-Instanzen ohne priori Wissen zur Laufzeit (MI without a priori Run

Time Knowledge) und der Meilenstein (Milestone). Wohed et al. bietet für beide Muster Hilfskonstruktionen an. [33, S. 11-12]

Trotz der Defizite in den Daten- und Ressourcenmustern, ist die BPMN 1.0 für die Prozessdarstellung geeignet, da fast alle Kontrollflussmuster mindestens indirekt dargestellt werden können. Alle für die Darstellung der Kontrollflussmuster benötigten Elemente sind auch in der BPMN 2.0 vorhanden, weshalb auch diese selbst für die Prozessdarstellung geeignet ist. [31, 27ff]

Bewertung nach Bösing

Methoden	BPMN
Kriterien	
Automatisierung möglich	+
Formalität gegeben	+
Übersichtlichkeit	+
Einfache Handhabung	+
Methode als Standard	+

Tabelle 3.9: Bewertung der BPMN nach Bösing (vgl. Bösing [32, S. 101])

Die Tabelle 3.9 zeigt, welche Kriterien von der BPMN erfüllt oder nicht erfüllt werden. Auch hier bedeutet ein „+“ in der Tabelle die Einhaltung des Kriteriums und ein „-“ die Nichterfüllung eines Kriteriums. In der Tabelle ist gut zu erkennen, dass die BPMN als Prozessmodelliersprache alle Kriterien von Bösing erfüllt und somit auch sehr gut für die Prozessdarstellung geeignet ist.

Fasst man nun die Ergebnisse von Wohed et al und Bösing zusammen, stellt sich heraus, dass die BPMN sehr gut für Prozessmodellierung geeignet ist. Dies ist auch nicht verwunderlich, da sie eigens für die Geschäftsprozessdarstellung entwickelt wurde. Weiterhin lässt sich feststellen, dass BPMN eine formelle Modelliersprache ist und dadurch, wie in Kapitel 1 erwähnt, ihren Anwendungsbereich in Anwendungssystemen finden kann.

3.6 Fazit

Die EPK und die BPMN wurden für den gleichen Zweck, der Geschäftsprozessmodellierung, entwickelt. Jedoch hat sich mit den genutzten Bewertungsmethoden gezeigt, dass beide Modelliersprachen unterschiedlich gut dafür geeignet sind. In der Tabelle 3.10 sind die unterstützten, teilweise unterstützten und nicht unterstützten Muster der beiden Modelliersprachen gegenübergestellt. Hierbei wird nochmal deutlich, dass die BPMN weitaus mehr Kontrollflussmuster unterstützt bzw. teilweise unterstützt als die EPK. Zu den Daten- und Ressourcenmustern, die hier nur eine untergeordnete Rolle spielen, ist anzumerken, dass die EPK keine der Muster unterstützt wohingegen die BPMN wenige Muster unterstützt.

	EPK	BPMN
Basic Control Flow		
21. Sequence	+	+
22. Parallel Split	+	+
23. Synchronisation	+	+
24. Exclusive Choice	+	+
25. Simple Merge	+	+
Advance Synchronisation		
26. Multiple Choice	+	+
27. Synchronising Merge	+	+ / -
28. Multiple Merge	-	+
29. Discriminator	-	+
Structural Patterns		
30. Arbitrary Cycles	+	+
31. Implicit Termination	+	+
Multiple Instance Patterns		
32. MI without Synchronisation	-	+
33. MI with a priori Design Time Knowledge	-	+
34. MI with a priori Run Time Knowledge	-	+
35. MI without a priori Run Time Knowledge	-	-
State-Based Patterns		
36. Deferred Choice	-	+
37. Interleaved Parallel Routing	-	+ / -
38. Milestone	-	-
Cancellation Patterns		
39. Cancel Activity	-	+
40. Cancel Case	-	+

Tabelle 3.10: Unterstützung der Kontrollflussmuster des Workflow Pattern Framework im Vergleich [vgl. Wohed et al. [33, S. 13], [36)]

Die Tabelle 3.11 zeigt die einzelnen Bewertungskriterien von Bösing und ihre Erfüllung durch die EPK sowie der BPMN. Wie in den vorherigen Tabellen auch bedeutet ein „+“ die Erfüllung des Kriteriums und ein „-“ die Nichterfüllung des Kriteriums. In der Tabelle zeigt sich, dass die EPK lediglich die Eigenschaft der „Einfachen Handhabung“ erfüllt, wohingegen die BPMN alle Kriterien von Bösing unterstützt.

Methoden	EPK	BPMN
Kriterien		
Automatisierung möglich	-	+
Formalität gegeben	-	+
Übersichtlichkeit	-	+
Einfache Handhabung	+	+
Methode als Standard	-	+

Tabelle 3.11: Gegenüberstellung von EPK und BPMN in einer Bewertungsmatrix (vgl. Bösing [32, S. 101])

Vergleicht man nun die beiden vorgestellten Modelliersprachen mit Hilfe der Bewertungsmatrix von Bösing und der Bewertungsmethode von Wohed et al., so wird deutlich gezeigt, dass die EPK im Vergleich zur BPMN eine nur zum Teil geeignete Sprache für Prozessmodellierung ist. So bietet die BPMN weitaus mehr Möglichkeiten das Prozessmodell zu strukturieren und auch mehr Informationen in das Prozessmodell zu integrieren. Wie schon im ersten Abschnitt des Kapitels erwähnt, wird ebenfalls ersichtlich, dass beide Modelliersprachen auch unterschiedlich Anwendungskontexte bedienen können. So ist die EPK für weniger formale Darstellungen mit betriebswirtschaftlicher Nähe geeignet, wohingegen die BPMN für formale Beschreibungen mit stärkerer Nähe zur Informatik genutzt wird. Jedoch lässt sich die BPMN auch zur Beschreibung von betriebswirtschaftlichen Prozessen nutzen. Auf Basis der gewonnenen Informationen wird im folgenden Kapitel eine Überführung der EPK in die BPMN erarbeitet.

4 Transformation von EPK in BPMN

4.1 Erste Überlegungen

Der logischste und auch vermeintlich einfachste Transformationsweg zwischen den Modelliersprachen EPK und BPMN wäre eine direkte Transformation oder auch 1-zu-1-Transformation. Dabei würde man ein Element der EPK genau auf ein Element der BPMN abbilden. Als Grundlage für eine sauber direkte Transformation ist eine präzise Definition der Semantik der Quell- und Zielsprache nötig.

Als Alternative zu der direkten Transformation von EPK zu BPMN kann auch eine zweistufige Transformation vorgenommen werden. Dabei wird zunächst die EPK in eine dritte Sprache abgebildet und das entstandene Zwischenergebnis wiederum nach BPMN übersetzt. Allerdings muss bei der Wahl der dritten Sprache ein besonderes Augenmerk daraufgelegt werden, ob diese auch eine Obermenge der Quell- und Zielsprache darstellt. Da sowohl die BPMN als auch die EPK XML-basiert beschrieben werden können, würde sich diese als dritte Sprache anbieten.

Diese Arbeit befasst sich jedoch nur mit der direkten Transformation von EPK zu BPMN, da die Überführung über eine dritte Sprache, den Umfang der Arbeit übersteigen würde.

4.2 Direkte Transformation von EPK zu BPMN

Dieser Abschnitt untersucht die direkte Transformation von EPK zu BPMN. Das wichtigste Kriterium bei der Transformation ist der Erhalt der Informationen, d.h. geht bei der Überführung von EPK zu BPMN Information verloren oder bleibt diese vollständig vorhanden? Wenn Informationen verloren gehen, können als Folge bestimmte Fragen im Zielmodell nicht gleichwertig beantwortet werden, wie dies im Quellmodell der Fall war. Um diesen Umstand aus dem Wege zu gehen, müssen im Vorfeld zwei Fragen für jedes Notationselement beantwortet werden: [37, S. 99]

1. Existiert eine BPMN-Notation, die die EPK-Semantik abbildet?
Falls nicht, welche Notation kommt dem möglichst nahe?
2. Werden keine zwei EPK-Notationen auf die BPMN-Notation abgebildet?
Falls nein: Handelt es sich dabei um einen Informationsverlust?

#1 Funktion

Funktionen in der EPK repräsentieren Bearbeitungsschritte innerhalb des Prozesses und können in der BPMN als Aktivität direkt abgebildet werden (Abbildung 4.1).

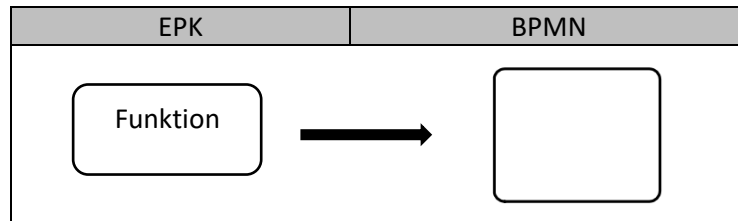


Abbildung 4.1: #1a EPK Funktion zu BPMN Aktivität

#2 Ereignis

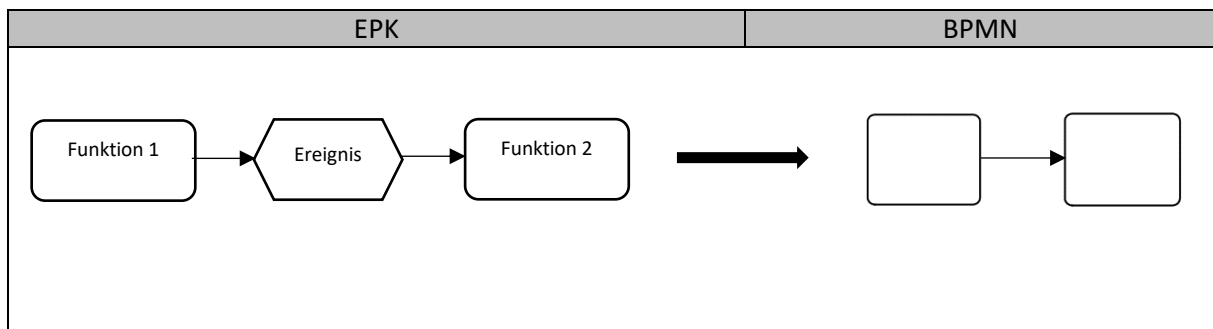


Abbildung 4.2: #2a EPK-Ereignis zu BPMN

In der EPK repräsentiert das Ereignis zumeist einen Zustand, wobei es in den meisten Fällen keine nennenswerte Mehrinformation liefert. Da die BPMN den Fokus mehr auf die Aktivitäten legt und eine Darstellung von Prozesszuständen im Normalfall nicht vorgesehen ist, lassen sich die meisten EPK-Ereignisse bei der direkten Transformation zur BPMN ignorieren (siehe Abbildung 4.2). Soll jedoch ein Prozesszustand zwingend angelegt werden, lässt sich dieses in der BPMN mit einem Datenobjekt und einer entsprechenden Zustandsbeschreibung wie in Abbildung 4.3 realisieren.

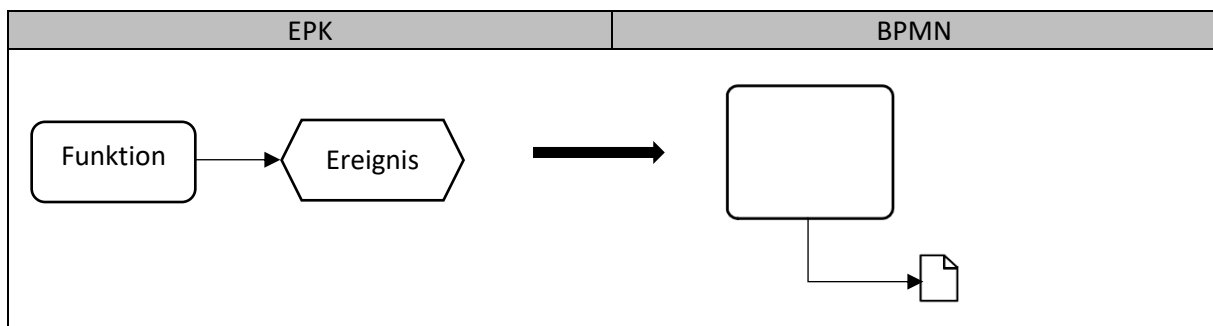


Abbildung 4.3: #2b EPK-Ereignis zu BPMN mittels Datenobjekt

Allerdings existiert auch ein Spezialfall, bei dem das Ereignis in der Transformation beachtet werden muss. Mittels der XOR- und ODER-Verknüpfungen mit mehreren ausgehenden Kanten werden in der EPK Entscheidungsfindungen modelliert. Die nachfolgenden Ereignisse bilden dabei die Bedingungen für die Verzweigung. Wie in Abbildung 4.4 dargestellt, werden die Verzweigungsbedingungen in der BPMN dann über Kantenbeschriftungen realisiert. [37, S. 101]

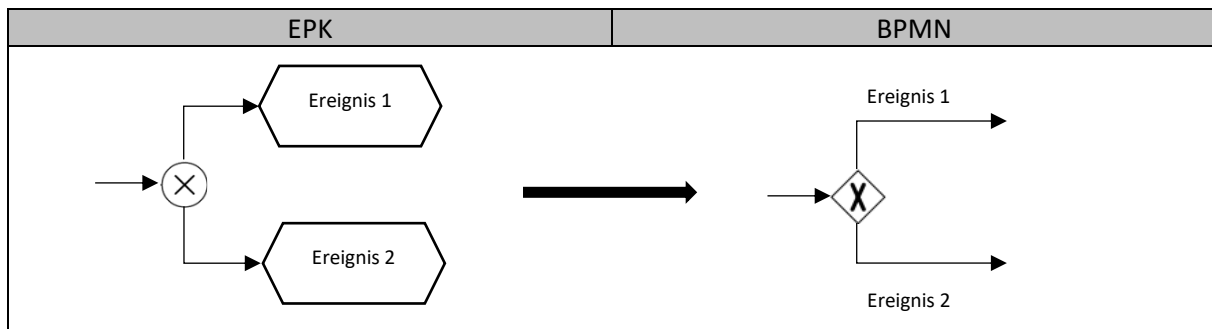


Abbildung 4.4: #2c EPK-Ereignis zu BPMN mit XOR-Verknüpfung

Es gibt auch Modellierungen, bei der ein Ereignis als Startereignis dient. Dabei sind die EPK-Ereignisse entweder Trigger oder Zustände. Der Typ kann lediglich durch die Beschriftung des Ereignisses oder durch den Prozesskontext ermittelt werden. Verfügt die EPK nur über ein Startereignis lässt sich dieses im BPMN-Modell ebenfalls als Startereignis darstellen (Abbildung 4.5). Da auch die BPMN mit einem Endereignis abgeschlossen wird, können die EPK-Endereignisse einfach in BPMN-Endereignisse übersetzt werden (Abbildung 4.6). Verfügt die EPK allerdings über mehrere Startereignisse, ist eine gültige Transformation nur in bestimmten Fällen möglich. Den Idealfall stellt dabei die Abbildung 4.7 dar. Dabei handelt es sich um alternative Startereignisse, die später mittels eines XOR-Konnektors zusammengeführt werden. [37, S. 101]

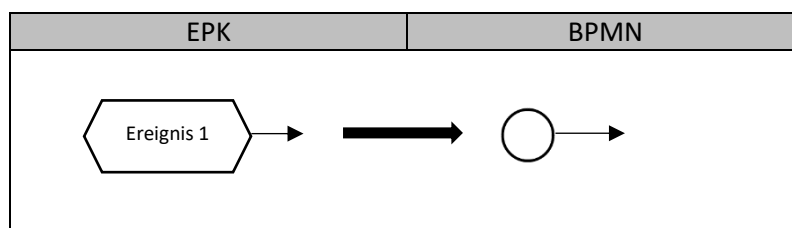


Abbildung 4.5: #2d EPK-Ereignis zu BPMN als Startereignis

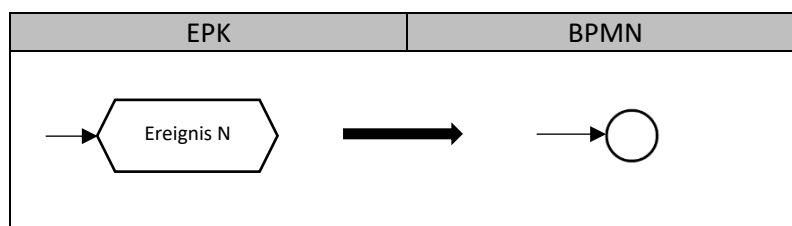


Abbildung 4.6: #2e EPK-Ereignis zu BPMN als Endereignis

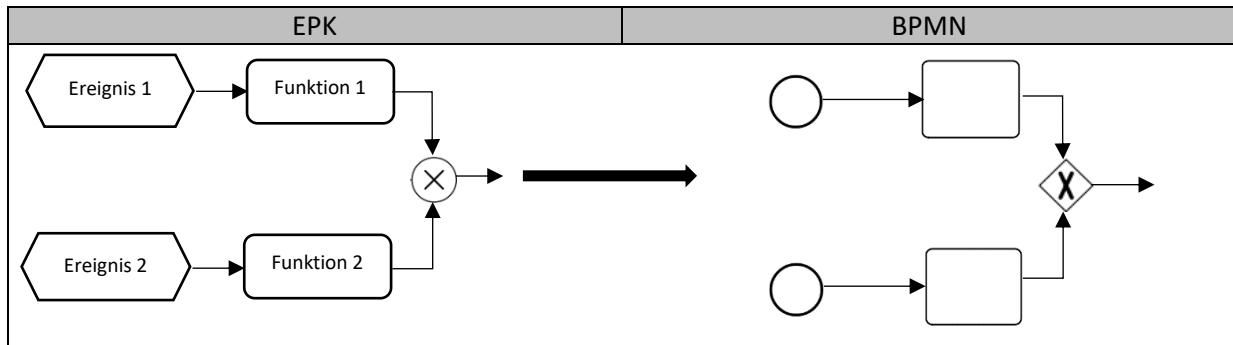


Abbildung 4.7: #2f EPK-Ereignis zu BPMN mit alternativen Startereignissen

#3 Konnektoren

In der EPK werden durch die Verwendung von Konnektoren Prozessverzweigungen modelliert. Gleiches gilt für die Gateways bei der BPMN, wodurch sich die einzelnen Typen der Konnektoren direkt abbilden lässt. Die Abbildungen Abbildung 4.8, Abbildung 4.9 und Abbildung 4.10 zeigen die genauere Überführung der EPK Konnektoren in BPMN. Der XOR-Konnektor wird direkt als XOR-Gateway abgebildet (Abbildung 4.8). Gleiches gilt für den UND-Konnektor (Abbildung 4.9) und den OR-Konnektor (Abbildung 4.10). Jedoch ist bei der direkten Transformation zu beachten, dass die Verzweigungsbedingungen in EPKs erfüllt werden, während bei der BPMN dann Kantenbeschriftungen vorgesehen sind. [37, S. 101]

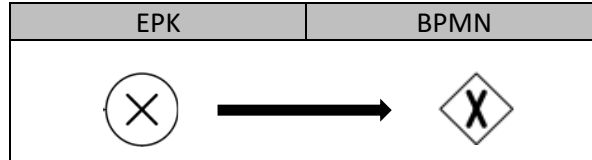


Abbildung 4.8: #3a EPK XOR-Konnektor zu BPMN

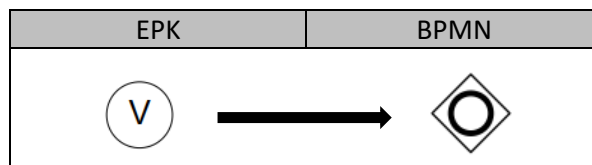


Abbildung 4.9: #3b EPK OR-Konnektor zu BPMN

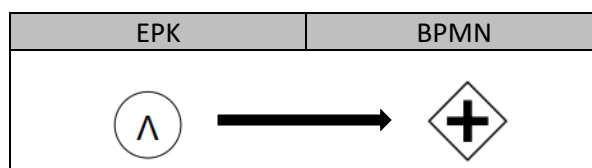


Abbildung 4.10: #3c EPK AND-Konnektor zu BPMN

#4 Prozessschnittstelle

Die Prozessschnittstelle dient in der EPK zur Herstellung von Beziehungen zwischen einzelnen Modellen. Diese werden in zwei verschiedenen Varianten verwendet. Entweder verweist die Prozessschnittstelle auf ein anderes Modell als Ganzes oder auf ein Startereignis in einem anderen Modell. Dabei muss der zweite Prozess erst beendet sein, damit die Funktion nach der Prozessschnittstelle angestoßen werden kann. Als BPMN würde dies, wie in Abbildung 4.11 gezeigt, als Unterprozess abgezeichnet.

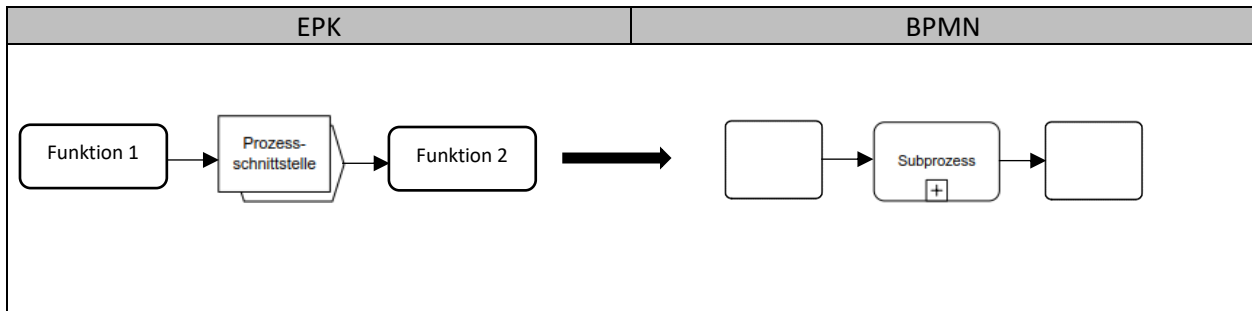


Abbildung 4.11: #4a EPK-Prozessschnittstelle zu BPMN als Unterprozess

Prozessschnittstellen können auch einfache Zerschneidungen von Prozessmodellen darstellen. Dabei wird auf eine Schnittstelle in einem anderen Modell verwiesen. In der BPMN kann diese „Verlinkung“ von Modellen als Link-Ereignis dargestellt werden (Abbildung 4.12 und Abbildung 4.13).

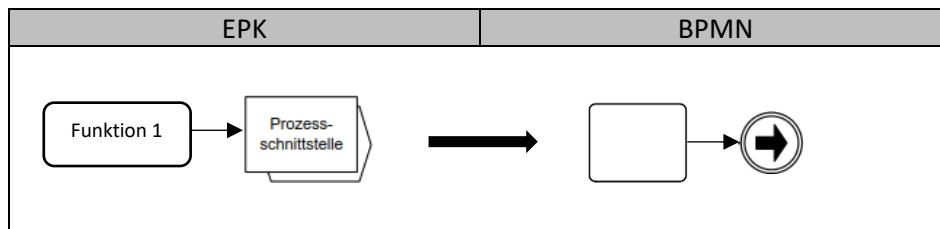


Abbildung 4.12: #4b EPK-Prozessschnittstelle zu BPMN als Austrittsereignis

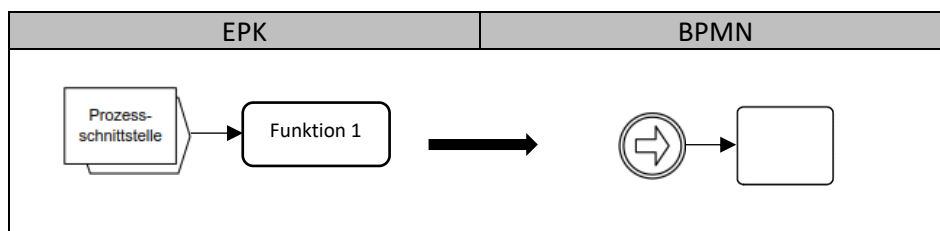


Abbildung 4.13: #4c EPK-Prozessschnittstelle zu BPMN als Eintrittspunkt

Ob die Prozessschnittstelle einen Unterprozess oder eine Verlinkung darstellt, lässt sich mit Hilfe von ein paar Indizien erkennen. Es handelt sich um einen Unterprozess, wenn für die Prozessschnittstelle ein Vorgänger- und ein Nachfolgeelement existieren. Dient die Prozessschnittstelle als Eintrittspunkt in einen Prozess oder wird als Austrittspunkt verwendet, so handelt es sich wahrscheinlich um eine Verlinkung.

#5 Organisationseinheit / Rolle

In der EPK dient die Stelle und die Rolle als Tätigkeitsbereich, welcher mindestens genau einer Person zugeordnet ist. In der BPMN entspricht das der Konstruktion von Pools, Lanes und Sub-Lanes. Jedoch können diese in der BPMN in eine Beziehung gesetzt werden, wohingegen die Stelle und die Rolle in der EPK nicht weiter in eine Beziehung gesetzt werden. Dies geschieht dann zumeist in einem Organigramm außerhalb des eigentlichen Prozessmodells. Wie in Abbildung 4.14 dargestellt, können Stelle und Rolle in der BPMN als nebeneinander angeordnete Lanes des gleichen Pools abgebildet werden. Jedoch steht keine Bezeichnung für diesen Pool zur Verfügung, wodurch generische Namen als Lösungsvariante gewählt werden.

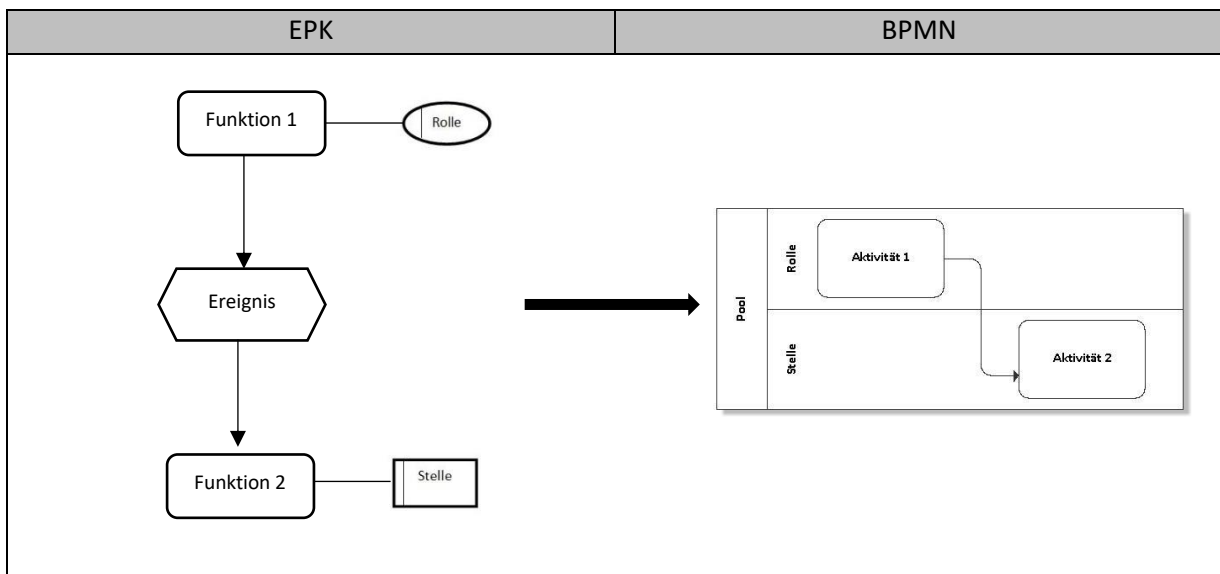


Abbildung 4.14: #5a EPK-Organisationseinheit/Rolle zu BPMN

Werden allerdings mehrere Stellen und/oder Rollen mit einer Funktion verknüpft so lässt sich dies nicht sauber in der BPMN abbilden. Jedoch kann man dafür verschiedene Alternativen nutzen um den Sachverhalt möglichst ohne Informationsverlust abzubilden wie die Grafiken Abbildung 4.15 bis Abbildung 4.17 zeigen. Die Abbildung 4.15 zeigt, dass man separate Lanes schaffen kann, die die Zusammenarbeit mehrerer Stellen oder Rollen modellieren. Die Abbildung 4.16 zeigt eine weitere Darstellungsmöglichkeit. Hierbei wird die gemeinschaftliche Aktivität dupliziert und mit gleicher Bezeichnung in den verschiedenen zugehörigen Lanes erzeugt. Die letzte Möglichkeit einer Transformation von Stellen/Rollen stellt die Abbildung 4.17 dar. Die Aktivität wird nur der verantwortlichen oder führenden Lane zugeordnet und über Textannotationen wird die Beteiligung der weiteren Stellen/Rollen erwähnt. [37, S. 103]

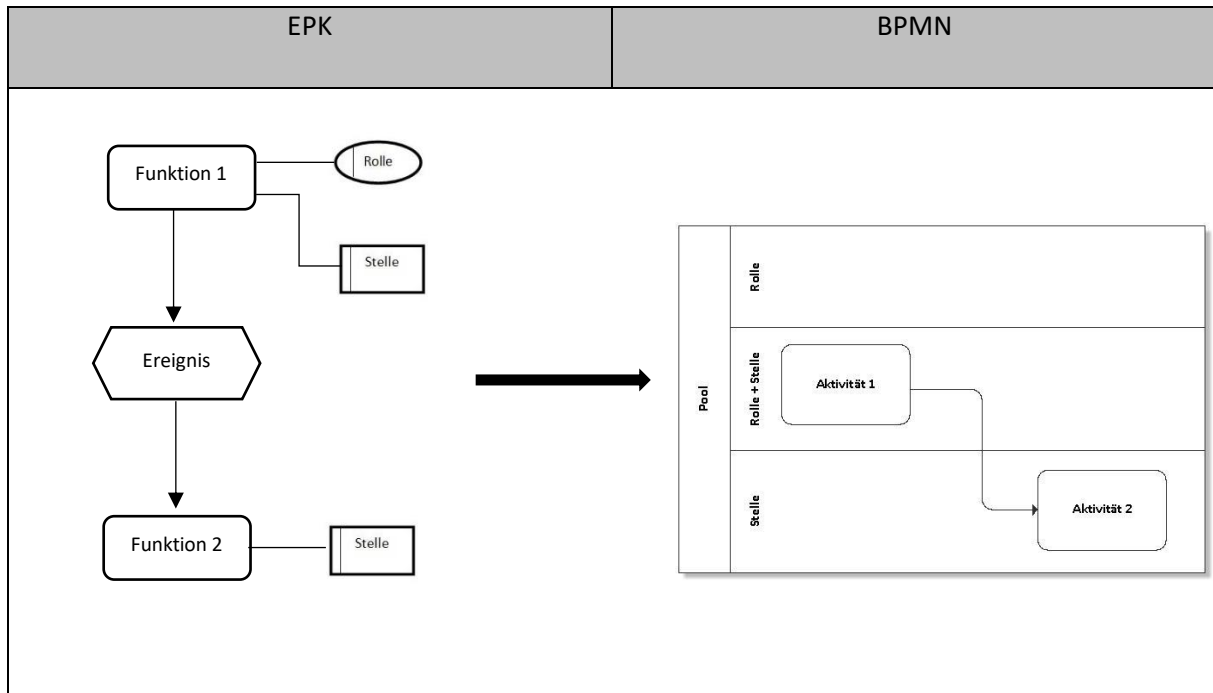


Abbildung 4.15: #5b EPK-Organisationseinheit/Rolle zu BPMN

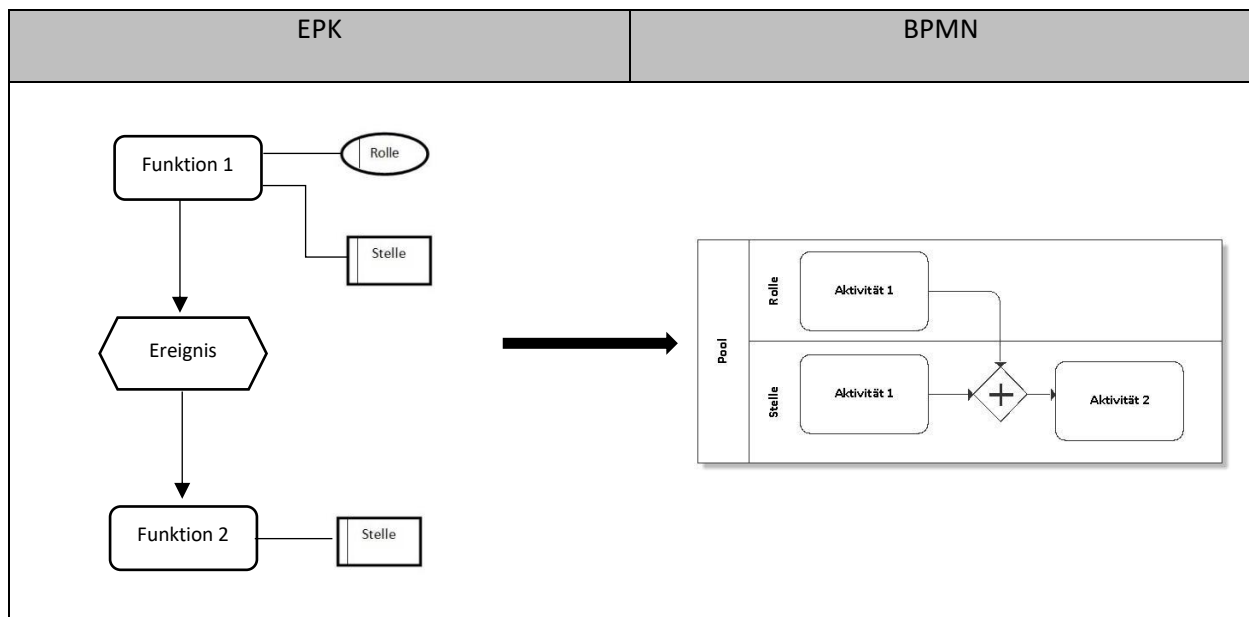


Abbildung 4.16: #5c EPK-Organisationseinheit/Rolle zu BPMN

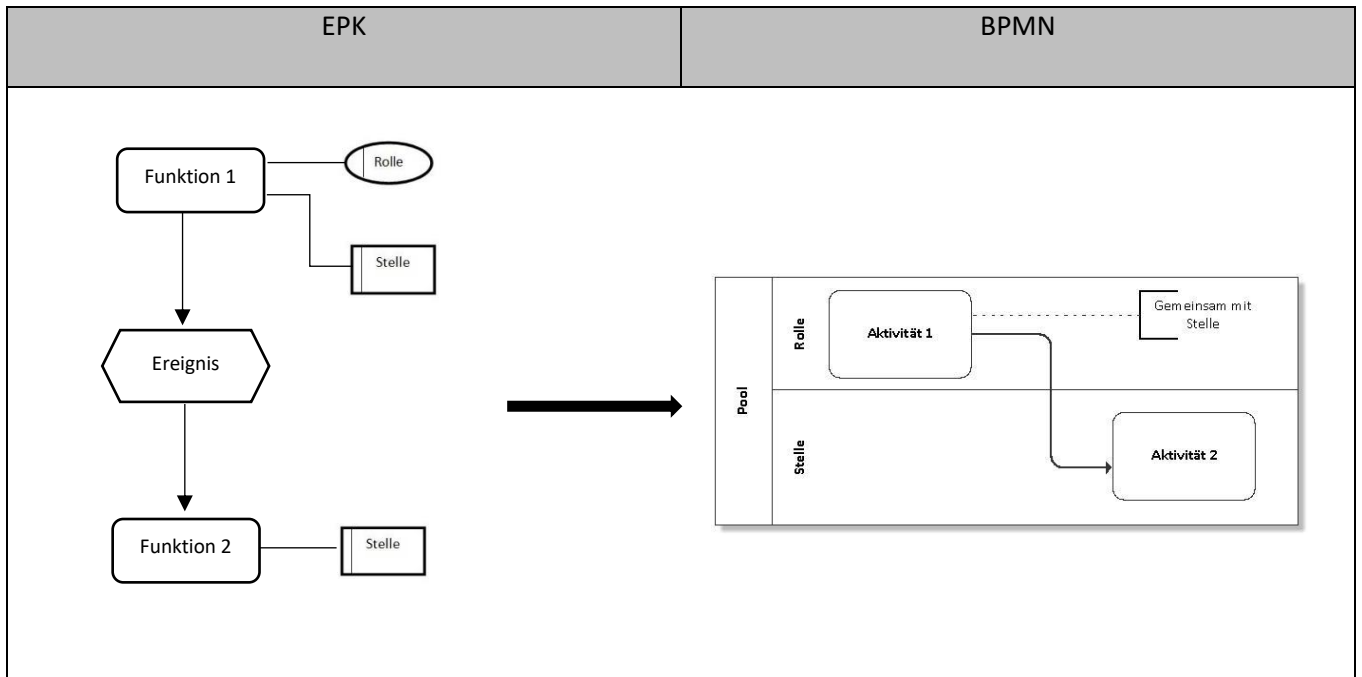


Abbildung 4.17: #5d EPK-Organisationseinheit/Rolle zu BPMN

#6 Datenobjekt

Wie die Abbildung 4.18 zeigt, lassen sich Datenobjekte der EPK direkt mit einer entsprechend gerichteten Assoziation als BPMN-Datenobjekt abbilden.

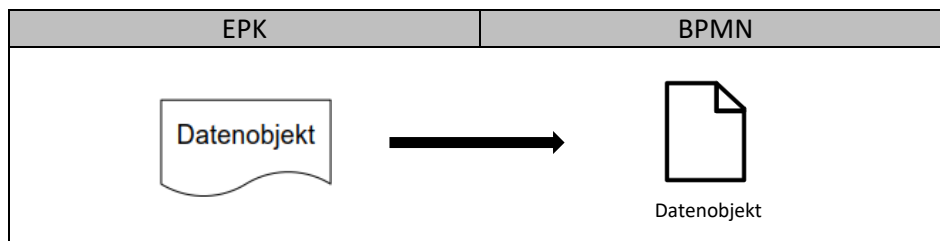


Abbildung 4.18: #6 EPK-Datenobjekt zu BPMN

#7 Systeme

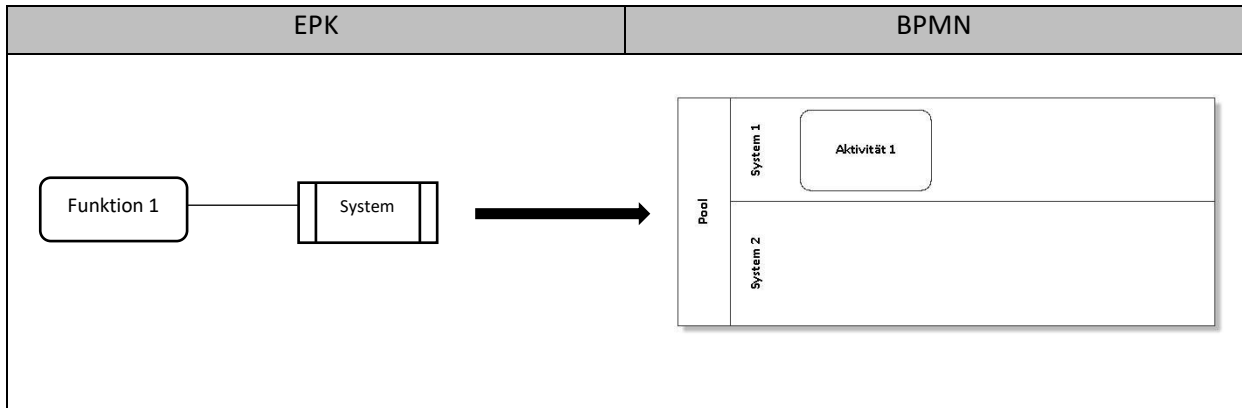


Abbildung 4.19: #7a EPK-System zu BPMN

Systeme lassen sich, ähnlich wie die Stellen und Rollen, in Pools und Lanes abbilden. Voraussetzung dafür ist, dass das Modell eine reine Systemansicht bietet und somit keine Stelle oder Rolle im Modell vorkommt (Abbildung 4.19). Auch hier stellt sich wieder die Frage, wie man die Zuordnung von mehreren Systemen zu einer Funktion abbildet. Außerdem muss man den Fall beachten, dass sowohl Stellen oder Rollen als auch Systeme im Modell existieren. Als Alternative für das Problem bietet sich die in Abbildung 4.20 gezeigte Darstellung an. Hier wird das System als zugeklappter Pool dargestellt, welches mit dem Prozess Nachrichten austauscht. Eine weitere Option zeigt die Abbildung 4.21, bei der mit Hilfe von Textannotationen die Systemzuordnung abgebildet wird. Mit Hilfe von weiteren Modellen, die sich direkt mit der Systemzuordnung beschäftigen, kann die Systemsicht aus der EPK in BPMN überführt werden. Dabei wird dann auf die Darstellung von Systemen im eigentlichen Prozessmodell verzichtet, wie in Abbildung 4.22 gezeigt wird. [37, S. 106]

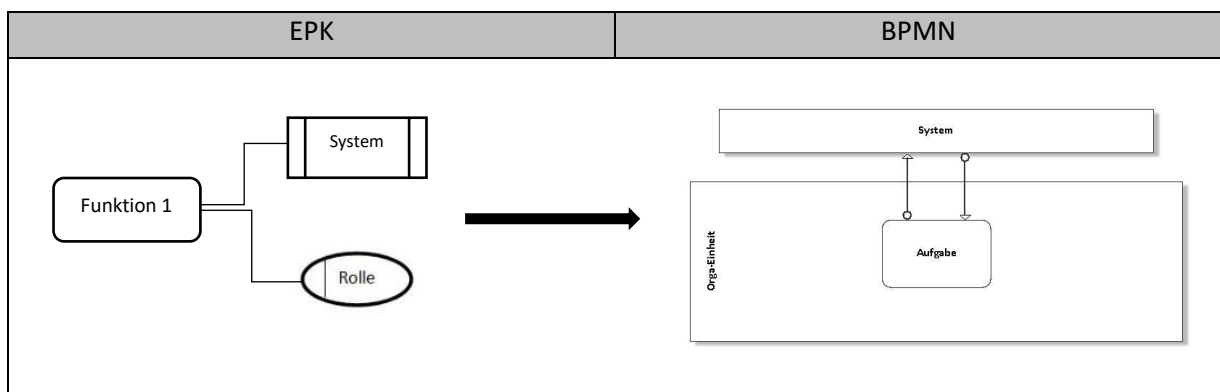


Abbildung 4.20: #7b EPK-System zu BPMN

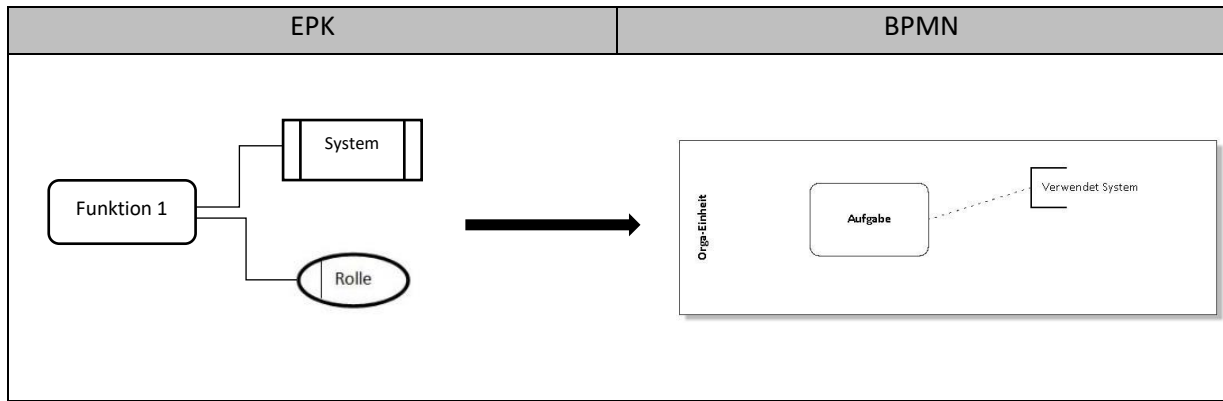


Abbildung 4.21: #7c EPK-System zu BPMN

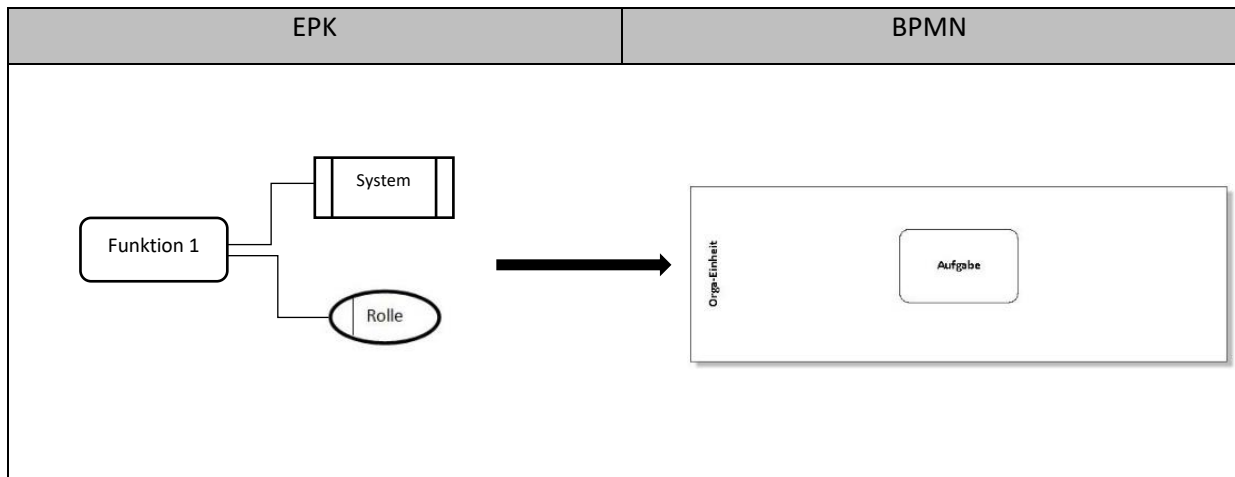


Abbildung 4.22: #7d EPK-System zu BPMN

5 Verifizierung und Validierung

5.1 Begriffserklärung Verifizierung und Validierung

Um eine Verifizierung und Validierung durchführen zu können, muss erst die Frage geklärt werden, wie diese definiert werden und was sie beinhalten. Die Verifizierung ist „das Verfahren zur Beurteilung der Produkte der Software-Entwicklungsphase, um Zusicherung zu leisten, dass sie die Anforderungen, die für sie in der vorherigen Phase definiert wurden, erfüllen“. Bei der Verifizierung stellt sich dementsprechend die Frage: „*Wird das System richtig gebaut?*“. Dahingegen ist die Validierung „das Verfahren zum Testen eines Computerprogramms und der Auswertung der Ergebnisse, um die Einhaltung spezifischer Anforderungen zu gewährleisten“. [38, S. 214] Bei der Validierung stellt sich die Frage: „*Wird das richtige System gebaut?*“. Diese Definitionen gehen auf das Institute of Electrical and Electronics Engineers (IEEE) zurück.

In dieser Arbeit wurden Transformationsregeln aufgezeigt, mit denen eine Transformation von EPK in BPMN ermöglicht werden soll. Um nun die Eignung der aufgestellten Regeln zu prüfen, wird ein Verifizierungsprozess durchgeführt. Dabei wird geprüft, ob die Anwendung der Regeln bei der Modelltransformation zum gewünschten Ergebnis führt. Im Validierungsprozess wird dann überprüft, ob das erhaltene Ergebnis den Ansprüchen der Transformation zwischen den Modellsprachen EPK und BPMN genügt. Das heißt einerseits, ob die Transformation vollständig erfolgt, bzw. alle verwendeten Notationselemente umgewandelt werden und andererseits muss das neu entstandene Diagramm den gleichen Sachverhalt wie das ursprüngliche Diagramm widerspiegeln. Des Weiteren müssen die Grundsätze der ordnungsgemäßen Modellierung (GoM) nach der Transformation noch erfüllt sein.

5.2 Beispieltransformation von EPK zur BPMN

Die Abbildung 5.1 zeigt den Vorgang einer Auftragsbearbeitung in einem produzierenden und verkaufenden Unternehmen. Sobald die Anfrage des Kunden eingegangen ist, werden die Kundendaten im Verkauf abgeglichen. Nach der Überprüfung wird der Auftrag entweder abgelehnt oder angenommen.

Wurde der Auftrag angenommen, so wird die Verfügbarkeit des geordneten Produktes mittels des ERP-Systems im Verkauf überprüft. Nach dem Überprüfungsvorgang gibt es zwei Möglichkeiten mit der Order fortzufahren. Entweder ist das bestellte Produkt vorhanden, dann kann es direkt ausgeliefert werden oder der Artikel muss erst noch hergestellt werden. Die Produktion des Artikels findet in einem extra Sub-Prozess in der Produktion statt. Sobald das Produkt fertig gestellt wurde, wird dieses ebenfalls versandt. Der letzte Schritt im Prozess ist dann das Versenden der Rechnung an den Kunden.

Das Diagramm ist nach den Grundsätzen ordnungsgemäßer Modellierung (GoM) erstellt worden, das heißt, es wurde syntaktisch korrekt nach den Regeln der EPK und den in dieser Arbeit definierten Beschränkungen modelliert. Der dargestellte Sachverhalt wird als korrekt betrachtet, da es sich lediglich um ein veranschaulichendes Beispiel handelt. Es wurden nur Elemente verwendet, die für die Beispieltransformation relevant sind, das heißt, sie stellen in dieser Arbeit definierte Bausteine oder Elemente mit direktem Äquivalent in der Ziel-Modellsprache dar. Dadurch ist auch der Grundsatz der Wirtschaftlichkeit erfüllt. Das Diagramm folgt einer klaren Strukturierung, ist dadurch einfach zu lesen und erfüllt somit den Grundsatz der Klarheit. Des Weiteren wurde das Diagramm systematisch aufgebaut. Die eindeutigen Bezeichnungen der Elemente würden eine weitere Verwendung in anderen Sichten ermöglichen. Ob der Grundsatz der Vergleichbarkeit erfüllt ist, wird sich direkt nach der Transformation zeigen.

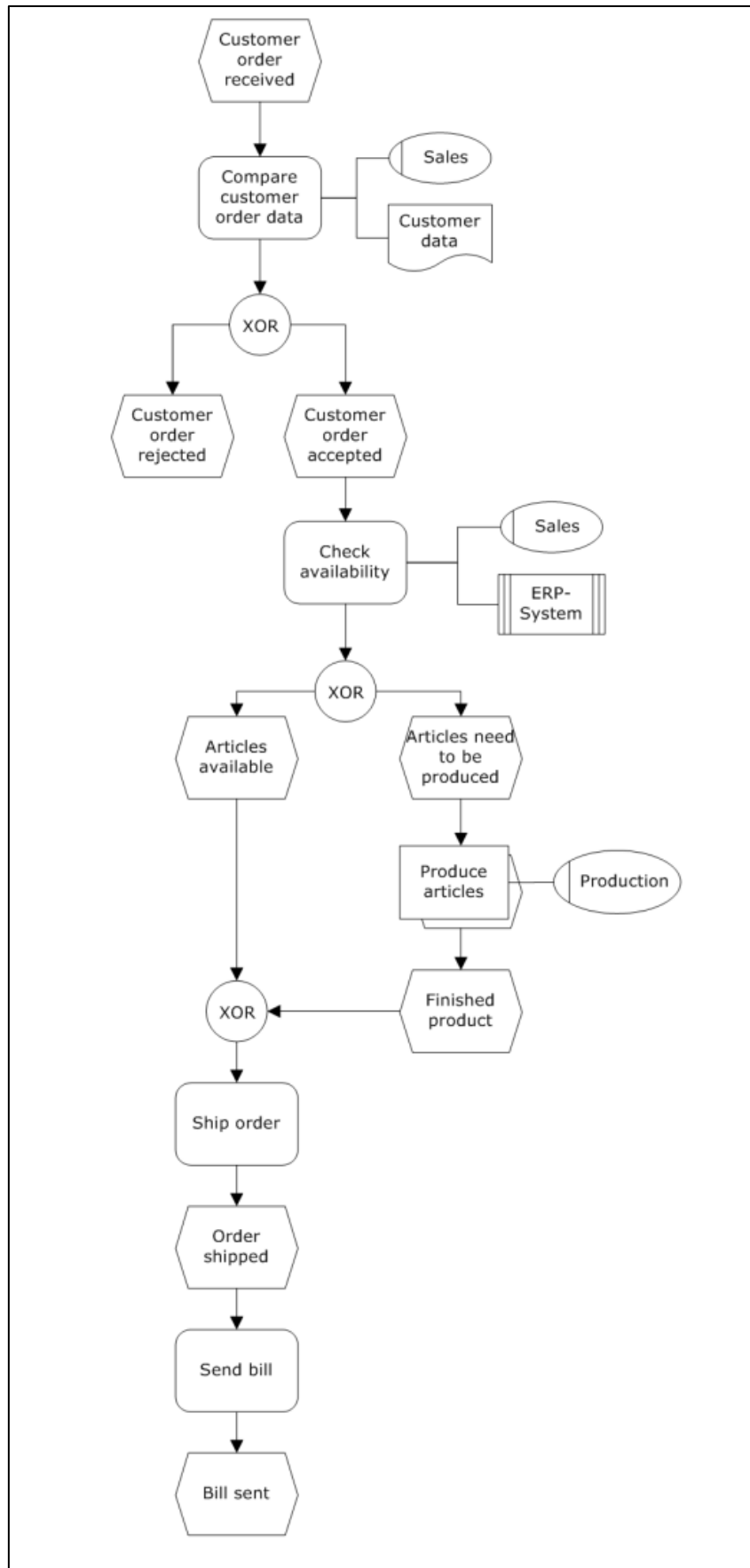


Abbildung 5.1: EPK Beispielprozess Transformation

In der Abbildung 5.2 sind alle identifizierten Regeln graphisch hervorgehoben worden. Die Nummerierung der Regeln finden sich in Kapitel 4.2 in den Abbildungsbezeichnungen wieder. Die abgehandelten Notationselemente wurden im Hintergrund dargestellt. Auch die Verbindungslinien sind aufgrund der besseren Lesbarkeit in den Hintergrund gestellt worden. Sie wurden jedoch noch nicht transformiert. Die Abbildung 5.3 auf der nachfolgenden Seite zeigt das durch die Transformation entstandene BPMN-Modell.

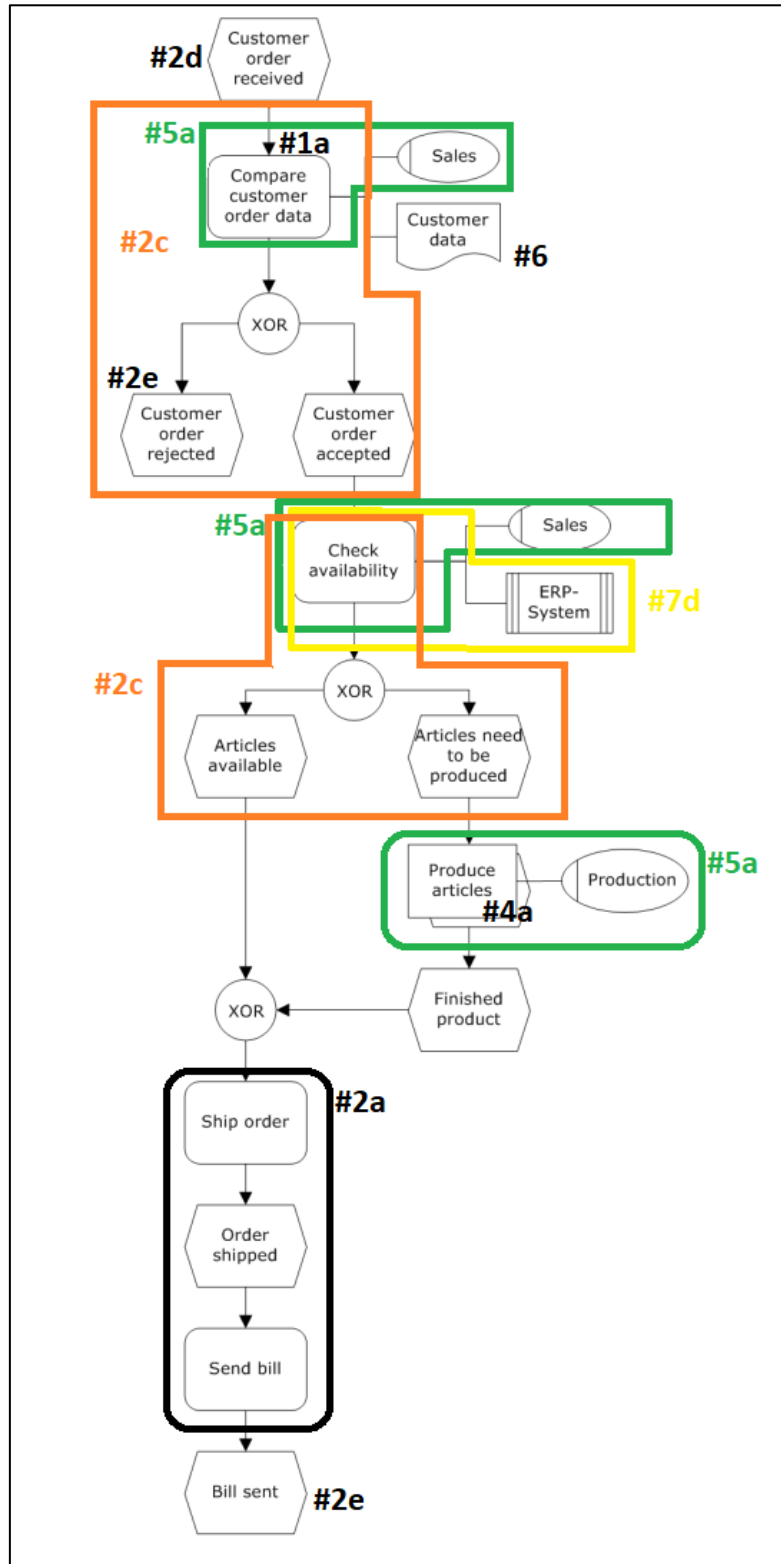


Abbildung 5.2: Diagramm in EPK - Regelidentifizierung

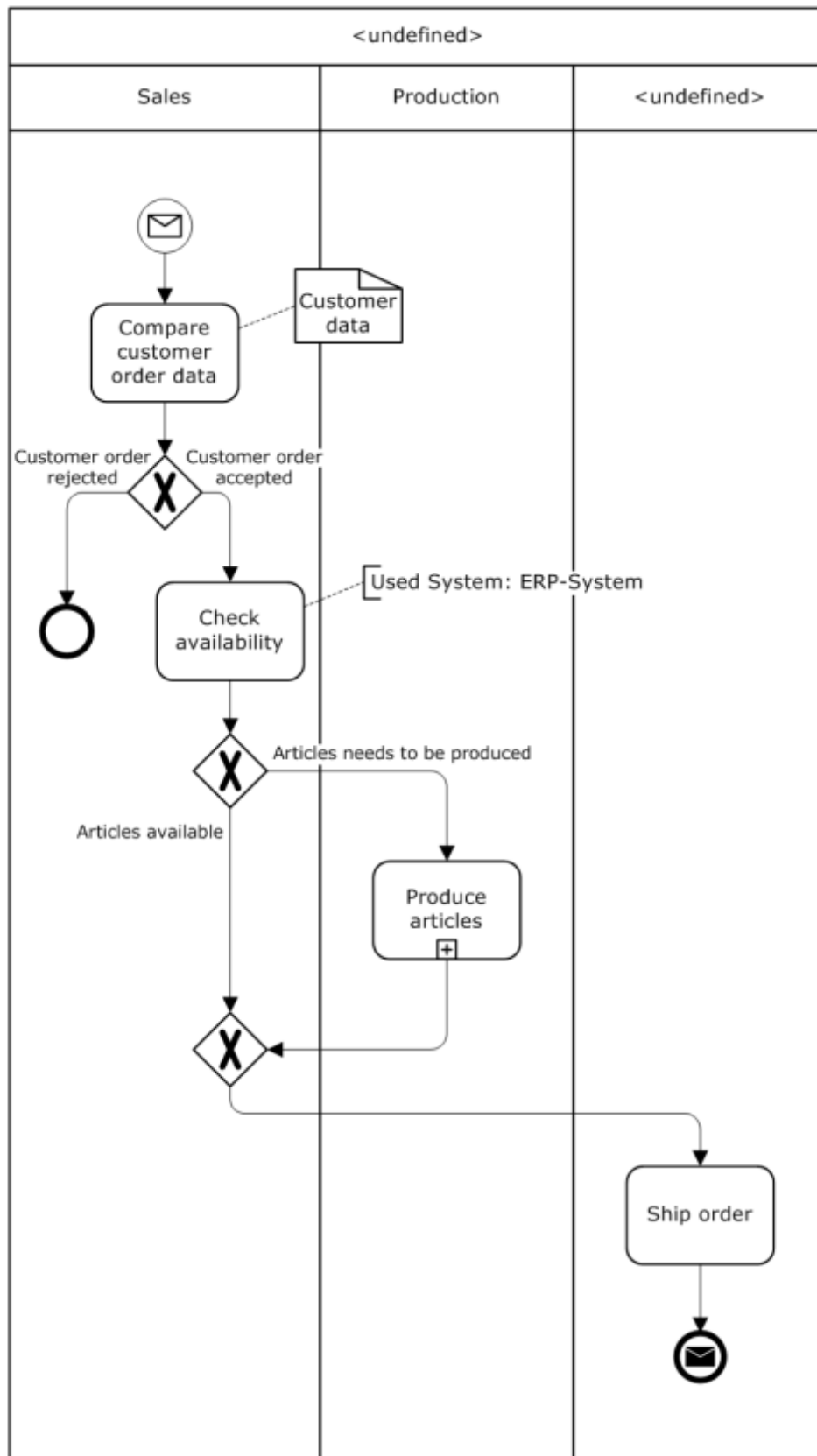


Abbildung 5.3: Transformierter EPK Prozess in BPMN

Verifizierung

Durch die Verwendung der beschriebenen Regeln konnten alle im Diagramm verwendeten Notationselemente transformiert werden. Weiterhin wurden in diesem Diagramm nur Elemente genutzt, die den in dieser Arbeit zuvor beschriebenen Anforderungen genügen. Bei der Verwendung von nicht abgedeckten Elementen wäre es zu Fehlern in Form von Lücken im transformierten Diagramm gekommen. Die Ausführungsreihenfolge der einzelnen Schritte hat zu keinerlei Fehlern geführt und wird daher als richtig angesehen.

Validierung

Das durch die Transformation erhaltene BPMN modelliert den gleichen Sachverhalt wie das originale Diagramm in der EPK-Notation. Außerdem wurde das gesamte Diagramm lückenlos transformiert. Die Verwendung von nicht abgedeckten Notationselementen würde an dieser Stelle zu Fehlern führen, wie bereits angesprochen wurde. Die M2M-Transformation über den beschriebenen Transformationsweg von EPK zu BPMN ist erfolgreich.

Die GoM sind auch nach der Transformation noch erfüllt. Das Diagramm ist sowohl syntaktisch als auch semantisch korrekt. Es hat den gleichen Detaillierungsgrad des Originaldiagramms, wodurch es sich weder beim Grundsatz der Wirtschaftlichkeit noch bei dem der Relevanz verschlechtert. Weiterhin besitzt das Diagramm den gleichen Grad an Klarheit wie das ursprüngliche Diagramm, da die Strukturen direkt übernommen wurden. Weil das erhaltene Diagramm den gleichen Sachverhalt wie das originale Diagramm wiedergibt, ist der Grundsatz der Vergleichbarkeit vollständig erfüllt. Zum Grundsatz des systematischen Aufbaus kann wenig gesagt werden, da hier nur die Prozesssicht betrachtet wird. Allerdings sind die Namen der einzelnen Elemente weiterhin eindeutig, sodass sie in anderen Sichten ebenfalls eindeutig identifizierbar sind.

6 Zusammenfassung und Ausblick

In den vorangegangenen Kapiteln wurde eine Methode zur Transformation von Diagrammen in EPK zu BPMN Aktivitätsdiagrammen für die Prozessdarstellung beschrieben. Dazu wurden zunächst in Kapitel 2 wichtige Begriffe für dieser Arbeit erläutert und ein kurzer Einblick in die Geschichte und die Bedeutung von Prozessen für Unternehmen gegeben. Im anschließenden Kapitel 3 wurden die Object Management Group (OMG), welche die Verantwortung für die BPMN trägt vorgestellt. Außerdem wurde das ARIS-Konzept, woraus die EPK entspringt, näher betrachtet. Die beiden Standard wurden ebenfalls vorgestellt.

Es wurde weiterhin geprüft, inwieweit sich die beiden Modellsprachen für die Darstellung von Prozessen eignen. Dazu wurde unter anderem das Workflow Pattern Framework verwendet. Dieses enthält drei Arten von Mustern: Kontrollflussmuster, Datenmuster und Ressourcenmuster. Es wurde festgestellt, dass beide Modellsprachen erhebliche Defizite bei den Datenmustern und Ressourcenmustern aufweisen. Zugleich wurde aber auch festgestellt, dass für die Prozessdarstellung die Kontrollflussmuster von besonderer Bedeutung sind und diese sowohl bei der BPMN zu großen Teilen abgedeckt werden. Die EPK wird durch die Kontrollflussmuster schlechter aber noch ausreichend abgedeckt. Aus diesem Grund wurde geschlussfolgert, dass sich beide Modellsprachen für die Prozessdarstellung eignen. Im Anschluss wurden ihre jeweiligen Notationselemente vorgestellt und Einschränkungen bezüglich ihrer Verwendung definiert.

In Kapitel 4 wurde dann der Transformationsweg beschrieben. Dort fand die Überlegung statt, eine vollständige 1-zu-1 Transformation zwischen den beiden Modellersprachen aufzuzeigen. Es wurden für die 1-zu-1 Transformation von EPK zu BPMN Regeln beschrieben und auch genauer erläutert wann welche Regel angewendet wird.

In Kapitel 5 wurde graphisch gezeigt, wie aus einem Diagramm in der EPK Notation mit Hilfe des erarbeiteten Regelwerks ein BPMN Diagramm entsteht. Diese M2M-Transformation geschah vollständig, das heißt, dass alle verwendeten Elemente transformiert wurden. Außerdem erzielte die Transformation ein zufriedenstellendes Ergebnis, was bedeutet, dass das erhaltene Diagramm den ursprünglichen Sachverhalt wiedergab.

Die Forschungsfrage „Wie können Transformationswege in Hinblick auf die Prozessdarstellung zwischen den Modellsprachen EPK und BPMN aussehen?“ wurde bearbeitet und es ist ein möglicher Transformationsweg zwischen den Diagrammen der EPK und BPMN aufgezeigt worden. Dieser Transformationsweg erfolgt über ein Regelwerk. Jedoch lässt sich so die Transformation nur semi automatisch bewerkstelligen, was einen erheblichen Aufwand darstellt. Außerdem können durch diese Transformationmethode weiterhin Fehler entstehen, man sich vor der Transformation für eine Regelwerk entscheiden muss. Dies kann zu Fehlern führen, wenn mit dem gleichen Regelwerk sämtliche Modelle eines Unternehmens transformiert werden sollen. Weiterhin muss man auch erörtern inwieweit das Regelwerk zur Transformation von EPK zu BPMN auch für die Transformation von BPMN zu EPK geeignet ist. Daher wäre eine formale Methode wünschenswert zur Transformation der Modellersprachen wünschenswert.

Als Lösung zu der direkten Transformation von EPK zu BPMN kann eine zweistufige Transformation vorgenommen werden, welche formaler und somit weniger fehlerbehaftet sein könnte. Dabei wird zunächst die EPK in eine dritte Sprache abgebildet und das entstandene Zwischenergebnis wiederum nach BPMN übersetzt. Allerdings muss bei der Wahl der dritten Sprache ein besonderes Augenmerk daraufgelegt sein, ob diese auch eine Obermenge der Quell- und Zielsprache darstellt. Da sowohl die BPMN als auch die EPK XML-basiert beschrieben werden können, würde sich XML als dritte Sprache anbieten. Durch die Verwendung der XML-Sprache als Zwischenschritt könnte man den Transformationsvorgang mit Hilfe von geeigneten Programmen automatisieren.

7 Tabellenverzeichnis

Tabelle 2.1: Überblick der Grundsätze ordnungsgemäßer Modellierung.....	12
Tabelle 2.2: Merkmale der Funktionsorganisation vs. Merkmale der Prozessorganisation	15
Tabelle 3.1: Übersicht der EPK Notationselemente (vgl. Baumgartner [28, S. 6]).....	22
Tabelle 3.2: Kategorien und enthaltene Elemente in der Übersicht.....	26
Tabelle 3.3: kurze Übersicht der BPMN Notationselemente (vgl. OMG – Business Process Model and Notation [31, S. 28f])	27
Tabelle 3.4: Kontrollflussmuster des WPF in einer Kurzübersicht (vgl. WPI [36])	30
Tabelle 3.5: Kontrollflussmuster des WPF in einer Kurzübersicht (Fortsetzung) (vgl. WPI [36]).....	31
Tabelle 3.6: Unterstützung der Kontrollflussmuster	32
Tabelle 3.7: Bewertungsübersicht der EPK nach Bösing (vgl. Bösing [32, S. 101])	33
Tabelle 3.8: Unterstützung der Kontrollflussmuster	34
Tabelle 3.9: Bewertung der BPMN nach Bösing (vgl. Bösing [32, S. 101])	35
Tabelle 3.10: Unterstützung der Kontrollflussmuster	36
Tabelle 3.11: Gegenüberstellung von EPK und BPMN	37
Tabelle I.1: BPMN-Startereignisse (vgl. OMG-BPMN [31]).....	66
Tabelle I.2: BPMN-Zwischen- und Endereignisse (vgl. OMG-BPMN [31]).....	67
Tabelle I.3: Datenmuster des Workflow Pattern (vgl. WPI, Wohed et al [36])	78
Tabelle I.4: Ressourcenmuster des Workflow Pattern (Teil 1) (vgl. WPI, Wohed et al [36])	83
Tabelle I.5: Ressourcenmuster des Workflow Pattern (Teil 2) (vgl. WPI, Wohed et al [36])	84

8 Abbildungsverzeichnis

Abbildung 1.1: ARIS-Haus	4
Abbildung 1.2: historische Entwicklung wichtiger Standards in der Prozessmodellierung	4
Abbildung 1.3: Popularität von Prozessnotationen	6
Abbildung 2.1: Grundsätzlicher Prozess.....	9
Abbildung 2.2: Mögliche Prozessreihenfolge.....	10
Abbildung 2.3: Grundsätze ordnungsgemäßer Modellierung.....	11
Abbildung 2.4: Funktions- versus Prozessorientierung (vgl. Schmelzer und Sesselmann [20, S. 54]) ..	14
Abbildung 2.5: Beispiel für die Funktions- versus Prozessorientierung in einem Automobilunternehmen	15
Abbildung 3.1: ARIS-Haus mit den fünf Sichten	19
Abbildung 3.2: ARIS-Haus mit Unterteilung in Beschreibungsebenen.....	20
Abbildung 3.3: Einordnung EPK und BPMN im ARIS-Haus (vgl. [25, S. 14])	20
Abbildung 3.4: UND-Regel mit eingehender Funktion und ausgehendem Ereignis	23
Abbildung 3.5: UND-Regel mit eingehendem Ereignis und ausgehender Funktion	23
Abbildung 3.6: ODER-Regel mit eingehender Funktion und ausgehendem Ergebnis.....	23
Abbildung 3.7: ODER-Regel mit eingehendem Ereignis und ausgehender Funktion.....	23
Abbildung 3.8: XOR-Regel mit eingehender Funktion und ausgehendem Ereignis	23
Abbildung 3.9: XOR-Regel mit eingehendem Ereignis und ausgehender Funktion	23
Abbildung 3.10: Verbotene XOR-Verknüpfung mit eingehendem Ereignis und ausgehender Funktion	24
Abbildung 3.11: Verbotene OR-Verknüpfung mit eingehendem Ereignis und ausgehender Funktion	24
Abbildung 3.12: Simpler Beispielprozess in BPMN	28
Abbildung 4.1: #1a EPK Funktion zu BPMN Aktivität	39
Abbildung 4.2: #2a EPK-Ereignis zu BPMN	39
Abbildung 4.3: #2b EPK-Ereignis zu BPMN mittels Datenobjekt.....	39
Abbildung 4.4: #2c EPK-Ereignis zu BPMN mit XOR-Verknüpfung	40
Abbildung 4.5: #2d EPK-Ereignis zu BPMN als Startereignis	40
Abbildung 4.6: #2e EPK-Ereignis zu BPMN als Endereignis	40
Abbildung 4.7: #2f EPK-Ereignis zu BPMN mit alternativen Startereignissen.....	41
Abbildung 4.8: #3a EPK XOR-Konnektor zu BPMN	41
Abbildung 4.9: #3b EPK OR-Konnektor zu BPMN	41
Abbildung 4.10: #3c EPK AND-Konnektor zu BPMN.....	41
Abbildung 4.11: #4a EPK-Prozessschnittstelle zu BPMN als Unterprozess	42
Abbildung 4.12: #4b EPK-Prozessschnittstelle zu BPMN als Austrittsereignis.....	42
Abbildung 4.13: #4c EPK-Prozessschnittstelle zu BPMN als Eintrittspunkt.....	42
Abbildung 4.14: #5a EPK-Organisationseinheit/Rolle zu BPMN	43
Abbildung 4.15: #5b EPK-Organisationseinheit/Rolle zu BPMN	44
Abbildung 4.16: #5c EPK-Organisationseinheit/Rolle zu BPMN.....	44
Abbildung 4.17: #5d EPK-Organisationseinheit/Rolle zu BPMN	45
Abbildung 4.18: #6 EPK-Datenobjekt zu BPMN.....	45
Abbildung 4.19: #7a EPK-System zu BPMN	46
Abbildung 4.20: #7b EPK-System zu BPMN.....	46
Abbildung 4.21: #7c EPK-System zu BPMN	47
Abbildung 4.22: #7d EPK-System zu BPMN.....	47
Abbildung 5.1: EPK Beispielprozess Transformation.....	50
Abbildung 5.2: Diagramm in EPK - Regelidentifizierung	51
Abbildung 5.3: Transformierter EPK Prozess in BPMN.....	52

Abbildung I.1: BPMN Ereignis.....	61
Abbildung I.2: BPMN Aktivität.....	61
Abbildung I.3: BPMN Gateway	62
Abbildung I.4: BPMN Sequenzfluss	62
Abbildung I.5: BPMN Nachrichtenfluss	62
Abbildung I.6: BPMN Assoziation	63
Abbildung I.7: BPMN Pool	63
Abbildung I.8: BPMN Pool mit Lane	64
Abbildung I.9: BPMN Datenobjekt	64
Abbildung I.10: BPMN Nachricht.....	64
Abbildung I.11: BPMN Gruppe	65
Abbildung I.12: BPMN Anmerkung.....	65
Abbildung I.13: Aufgabe	68
Abbildung I.14: zusammengeklappter Sub-Prozess	68
Abbildung I.15: expandierter Sub-Prozess	68
Abbildung I.16: parallele Multiinstanzen #1	69
Abbildung I.17: sequentielle Multiinstanzen #2.....	69
Abbildung I.18: Schleife.....	69
Abbildung I.19: Schleife als Sub-Prozess	70
Abbildung I.20: Sub-Prozess Kompensation.....	70
Abbildung I.21: Kompensation	70
Abbildung I.22: bedingter Sequenzfluss.....	71
Abbildung I.23: Default Sequenzfluss.....	71
Abbildung I.24: XOR-Gateway / exklusives Gateway	71
Abbildung I.25: ODER-Gateway / inklusives Gateway.....	71
Abbildung I.26: UND-Gateway / paralleles Gateway	72
Abbildung I.27: Komplexes Gateway.....	72
Abbildung I.28: ereignis-basiertes Gateway.....	72
Abbildung I.29: paralleles ereignis-basiertes Gateway (Instanziierung)	72
Abbildung I.30: exklusives ereignis-basiertes Gateway (Instanziierung)	73
Abbildung I.31: Ad-hoc-Sub-Prozess	73
Abbildung I.32: Transaktions-Sub-Prozess	73
Abbildung I.33: Ereignis-basierter-Sub-Prozess	74
Abbildung I.34: Datensammlung	74
Abbildung I.35: Dateninput/-eingabe.....	74
Abbildung I.36: Datenoutput/-ergebnis	74
Abbildung I.37: Datenspeicher	74
Abbildung I.38: Funktion	75
Abbildung I.39: Ereignis.....	75
Abbildung I.40: Dokument	75
Abbildung I.41: System / Informationsobjekt	75
Abbildung I.42: Kontrollfluss	76
Abbildung I.43: UND-Konnektor.....	76
Abbildung I.44: ODER-Konnektor	76
Abbildung I.45: XOR-Konnektor	76
Abbildung I.46: Rolle	76
Abbildung I.47: Stelle.....	76
Abbildung I.48: Prozessschnittstelle.....	77

9 Literaturverzeichnis

Literatur

- [1] H. Schmelzer, W. Sesselmann, H. J. Schmelzer und W. Sesselmann, *Geschäftsprozessmanagement in der Praxis: Kunden zufrieden stellen - Produktivität steigern // Geschäftsprozessmanagement in der Praxis: Kunden zufriedenstellen, Produktivität steigern, Wert erhöhen ; [das Standardwerk, 7. Aufl. München: Hanser Wirtschaft; Hanser, 2010.*
- [2] H. Seidlmeier, *Prozessmodellierung mit ARIS: Eine beispielorientierte Einführung für Studium und Praxis, 3. Aufl. Wiesbaden: Vieweg + Teubner, 2010.*
- [3] W. Bandara, A. Alibabaei und M. Aghdasi, „Means of achieving business process management success factors“ in *Proceedings of the 4th Mediterranean Conference on Information Systems, 2009.*
- [4] R. K. L. Ko, S. S. G. Lee und E. Wah Lee, „Business process management (BPM) standards: a survey“, *Business Process Management Journal*, Jg. 15, Nr. 5, S. 744–791, 2009.
- [5] D. Miers, „The keys to BPM project success“, *Retrieved July*, Jg. 20, S. 2011, 2006.
- [6] H. Seidlmeier, Hg., *Prozessmodellierung mit ARIS // Prozessmodellierung mit ARIS®: Eine beispielorientierte Einführung für Studium und Praxis in ARIS 9, 4. Aufl. Wiesbaden: Springer; Springer Vieweg, 2015.*
- [7] O. Omar, „Eine BPMN-nach-BPEL-Transformation“.
- [8] J. Freund und B. Rücker, *Praxishandbuch BPMN 2.0, 4. Aufl. München: Carl Hanser Verlag GmbH Co KG; Hanser, 2014.*
- [9] F.A. Brockhaus GmbH, Hg., *Brockhaus Universal Lexikon Band 15. BROCKHAUS, 2003.*
- [10] J. Becker, M. Rosemann und R. Schütte, „Grundsätze ordnungsmäßiger modellierung“, *Wirtschaftsinformatik*, Jg. 37, Nr. 5, S. 435–445, 1995.
- [11] Anu Maria, „Introduction to Modeling and Simulation“ in *Proceedings of the 1997 Winter Simulation Conference, 1997.*
- [12] Richard F. Paige, Jonathan S. Ostroff und Phillip J. Brooke, „Principles for Modeling Language Design“ in 2000.
- [13] F. A. Brockhaus GmbH, Hg., *Brockhaus Universal Lexikon Band 23. BROCKHAUS, 2003.*
- [14] F. A. Brockhaus GmbH, Hg., *Brockhaus Universal Lexikon Band 18. BROCKHAUS, 2003.*
- [15] „Wirtschaftslexikon Gabler“, <http://wirtschaftslexikon.gabler.de/Definition/prozess.html>.
- [16] Normenausschuss Qualitätsmanagement, Statistik und Zertifizierungsgrundlagen im DIN Unterausschuss 5, im DIN Deutsches Institut für Normung e. V., „ISO 9000 Einführungs- und Unterstützungspaket: Leitfaden zum Konzept und zur Anwendung des prozessorientierten Ansatzes für Managementsysteme“ in 2011.
- [17] DIN-ISO, *Qualitätsmanagementsysteme-Grundlagen und Begriffe (ISO 9000: 2005)*. September: September.
- [18] J. Becker, C. Mathas, A. Winkelmann und O. Günther, *Geschäftsprozessmanagement*. Berlin, Heidelberg: Springer-Verlag; Springer, 2009.

- [19] J. Göpfert und H. Lindenbach, *Geschäftsprozessmodellierung mit BPMN 2.0: Business Process Model and Notation // Geschäftsprozessmodellierung mit BPMN 2.0: Business Process Model and Notation*. Walter de Gruyter GmbH & Co KG, 2013.
- [20] H. J. Schmelzer und W. Sesselmann, *Geschäftsprozessmanagement in der Praxis: Kunden zufrieden stellen, Produktivität steigern, Wert erhöhen*. 8. Auflage ISBN: 3446434608. *Change Management // Geschäftsprozessmanagement in der Praxis: Kunden zufriedenstellen, Produktivität steigern, Wert erhöhen : [das Standardwerk*, 8. Aufl. München: München: Carl Hanser Verlag GmbH & Co. KG (siehe S. 107); Hanser, 2013.
- [21] OMG-Website, *OMG - We set the Standard - About OMG*. [Online] Verfügbar unter: <http://www.omg.org/gettingstarted/gettingstartedindex.htm>.
- [22] OMG, *OMG - We Set the Standard*. [Online] Verfügbar unter: http://www.omg.org/memberservices/OMG_Backgrounder.pdf.
- [23] OMG, „Snapshot: The omg technology adoption process“ most, 2016.
- [24] A. C. Schwickert, „Geschäftsprozessmodellierung mit ARIS“.
- [25] Prof. Dr. Martin Leischner, „Prozessmodellierung“, Nov. 21 2014.
- [26] J. L. Staud, *Geschäftsprozessanalyse: ereignisgesteuerte Prozessketten und objektorientierte Geschäftsprozessmodellierung für betriebswirtschaftliche Standardsoftware // Geschäftsprozessanalyse: Ereignisgesteuerte Prozessketten und objektorientierte Geschäftsprozessmodellierung für Betriebswirtschaftliche Standardsoftware (German Edition)*. Dordrecht: Springer Science & Business Media; Springer, 2006.
- [27] G. Keller, A.-W. Scheer und M. Nüttgens, *Semantische Prozeßmodellierung auf der Grundlage" Ereignisgesteuerter Prozeßketten (EPK)*. Inst. für Wirtschaftsinformatik, 1992.
- [28] Baumgartner, Ebert, Schleider, „Regeln zur Modellierung von EPK's“.
- [29] J. Freund und B. Rücker, *Praxishandbuch BPMN 2.0*. Carl Hanser Verlag GmbH Co KG, 2012.
- [30] S. A. White, „Introduction to BPMN. 2004“, *IBM Corporation*, 2004.
- [31] OMG, *Business Process Model and Notation*. OMG, 2011.
- [32] K. D. Bösing und J. Geldmacher, „Transformation einer EPK in eine formalisierte Methode“, *Wissenschaftliche Beiträge der Technischen Hochschule Wildau, Wildau*, S. 95–101, 2014.
- [33] P. Wohed, van der Aalst, Wil MP, M. Dumas, A. H. M. ter Hofstede und N. Russell, „Pattern-based Analysis of BPMN“, 2005.
- [34] *Workflow Pattern Initiative*. [Online] Verfügbar unter: <http://www.workflowpatterns.com/>.
- [35] Petia Wohed, Wil M.P. van der Aalst, Marlon Dumas, Arthur H.M. ter Hofstede und Nick Russell, „Pattern-based Analysis of UML Activity Diagrams“ in 2006.
- [36] WPI, *Workflow Process Initiative - Workflow Patterns*. [Online] Verfügbar unter: www.workflowpatterns.com.
- [37] G. Decker, W. Tscheschner und J. Puchan, „Migration von EPK zu BPMN“ in *Proceedings: EPK 2009 Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten 8. Workshop der Gesellschaft für Informatik e. V.(GI) und Treffen ihres Arbeitskreises Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)*, 2009, S. 91–117.

- [38] W. L. Oberkampf und T. G. Trucano, „Verification and validation in computational fluid dynamics“, *Progress in Aerospace Sciences*, Jg. 38, Nr. 3, S. 209–272, 2002.

I Anhang

Notationselemente der BPMN

Basiselemente

Ereignis

Ein Ereignis ist etwas, das im Verlauf eines Prozesses passiert. Die Ereignisse haben Einfluss auf den Ablauf des Modells und in der Regel eine Ursache (Trigger) oder eine Auswirkung (Ergebnis). Aus diesem Grund wird zwischen „throwing“ und „catching“ Ereignissen unterschieden. Ereignisse werden als Kreise dargestellt, in denen Marker zur Unterscheidung zwischen Triggern und Ergebnissen gesetzt werden. Mit Start-, Zwischen- und Endereignis existieren drei Arten von Ereignissen. Start- und Endereignis müssen nicht zwingend modelliert werden, jedoch muss bei einem existierendem Startereignis auch ein Endereignis im Modell dargestellt werden. Um das Lesen der Diagramme zu vereinfachen, werden in dieser Arbeit Start- und Endereignis modelliert.

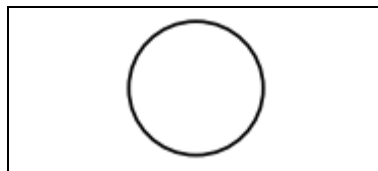


Abbildung I.1: BPMN Ereignis

Aktivität

Eine Aktivität ist ein Oberbegriff für Arbeit, die jemand oder etwas ausführt. Die Aktivität kann dabei atomar oder nicht-atomar sein. In der BPMN wird die Aktivität als abgerundetes Rechteck dargestellt. Die Aktivität kann als Prozess, Sub-Prozess und Aufgabe in einem Prozessmodell vorkommen.

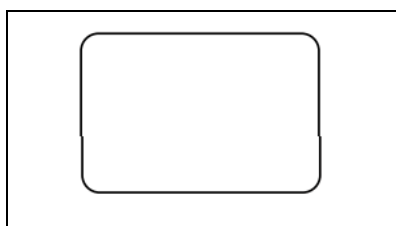


Abbildung I.2: BPMN Aktivität

Gateway

Das Gateway wird zur Kontrolle der Divergenz und der Konvergenz der Sequenzflüsse in einem Prozess benutzt, wodurch die Verzweigung und Zusammenführung der Pfade bestimmt werden. Mit Hilfe von internen Marken wird bestimmt, auf welche Art die Verzweigung oder Zusammenführung geschieht.

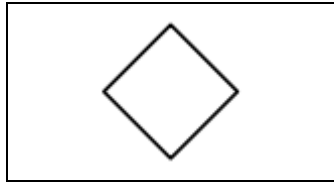


Abbildung I.3: BPMN Gateway

Sequenzfluss

Durch den Sequenzfluss wird die Reihenfolge dargestellt, in der Aktivitäten in einem Prozess ausgeführt werden. Er wird als durchgezogene Linie mit einer Pfeilspitze dargestellt. Das Diagramm kann sowohl von links nach rechts als auch von oben nach unten gezeichnet werden. Zur besseren Lesbarkeit ist eine andere Richtung nicht empfehlenswert.

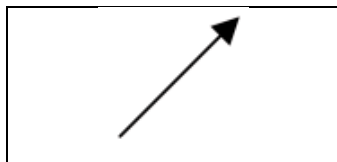


Abbildung I.4: BPMN Sequenzfluss

Nachrichtenfluss

Der Nachrichtenfluss wird verwendet, um den Fluss von Nachrichten zwischen einem sendenden und einem empfangenden Teilnehmer darzustellen. Die Teilnehmer werden in der BPMN als zwei separate Pools im Modell dargestellt. Eine gestrichelte Linie mit einem nicht ausgefüllten Kreis am Anfang und einer nicht ausgefüllten Pfeilspitze dient der Darstellung des Nachrichtenflusses.

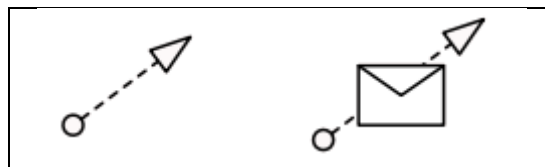


Abbildung I.5: BPMN Nachrichtenfluss

Assoziation

Die Assoziation wird verwendet, um Informationen und Artefakte mit graphischen Elementen zu verbinden. Text-Anmerkungen und andere Artefakte können den graphischen Elementen zugeordnet werden. Eine Pfeilspitze an der Assoziation zeigt eine Richtung des Flusses an, wenn diese angemessen ist. Ansonsten wird nur eine gepunktete Linie verwendet.

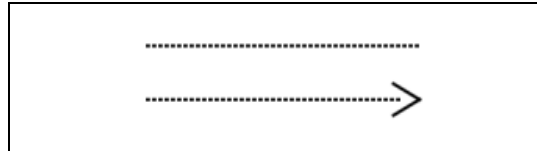


Abbildung I.6: BPMN Assoziation

Pool

Mit dem Pool werden Teilnehmer im Prozess dargestellt. Ein Pool kann interne Details in Form eines ausführenden Prozesses enthalten oder er besitzt keine internen Details. Zur Darstellung des Pools wird ein Rechteck mit Bezeichnung benutzt. Dabei kann der Pool vertikal oder horizontal gezeichnet werden. Bei der horizontalen Verwendung ist die Bezeichnung auf der linken Seite, bei der vertikalen oben.

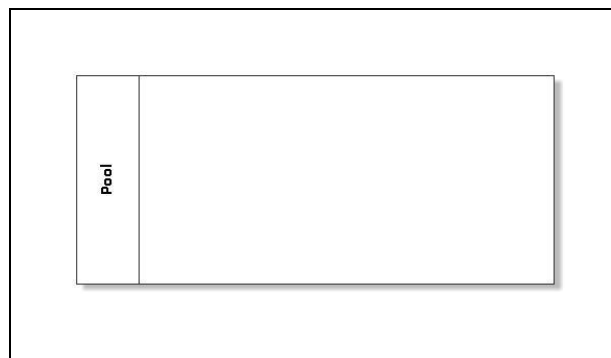


Abbildung I.7: BPMN Pool

Lane

Eine Lane ist eine Unterteilung innerhalb des Pools. Sie erstreckt sich über die gesamte Länge des Prozesses, entweder vertikal oder horizontal. Lanes werden zur Organisation und Kategorisierung von Aktivitäten verwendet. Es besteht auch die Möglichkeit mehrere Lanes in einer Lane, also eine Verschachtelung, darzustellen.

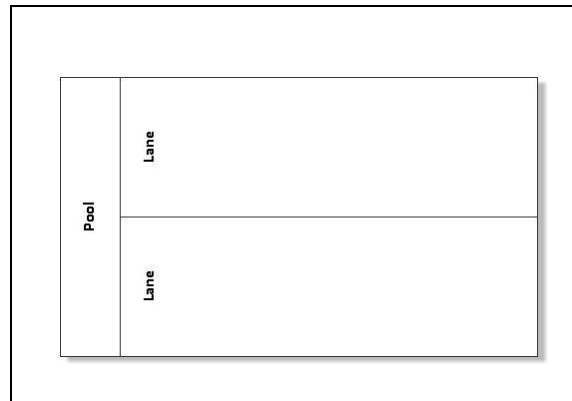


Abbildung I.8: BPMN Pool mit Lane

Datenobjekt

Ein Datenobjekt stellt Informationen zur Verfügung, welche Bedingungen erfüllt sein müssen, damit eine Aktivität ausgeführt wird und/oder was diese erzeugt. Ein Datenobjekt kann sowohl ein einzelnes als auch eine Sammlung von Objekten sein. Datenein- und -ausgabe stellen die gleiche Information für Prozesse zur Verfügung.

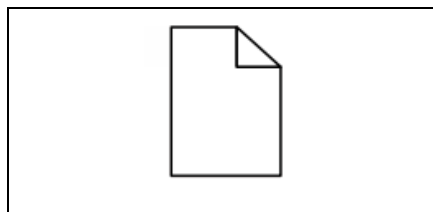


Abbildung I.9: BPMN Datenobjekt

Nachricht

Eine Nachricht wird benutzt, um den Inhalt einer Kommunikation zwischen zwei Teilnehmern darzustellen.



Abbildung I.10: BPMN Nachricht

Gruppe

Eine Gruppe ist eine Bündelung von grafischen Elementen, die innerhalb der gleichen Kategorie sind. Diese Art der Gruppierungen hat keinen Einfluss auf den Sequenzfluss innerhalb der Gruppe. Der Kategorienname erscheint im Diagramm als Gruppen-Label. Kategorien können zur Dokumentation oder Analyse verwendet werden. Gruppen sind eine Möglichkeit, mit der Kategorien von Objekten visuell im Diagramm dargestellt werden können. Sie werden als abgerundetes Rechteck mit Punkt-Strich-Linie dargestellt.



Abbildung I.11: BPMN Gruppe

Anmerkung

Eine Anmerkung dient dazu, zusätzliche Textinformationen für den Leser eines Diagramms zur Verfügung zu stellen.

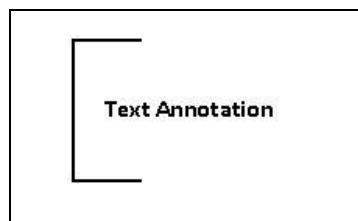


Abbildung I.12: BPMN Anmerkung

Erweiterte Elemente der BPMN

In der BPMN 2.0 Spezifikation existieren noch weitere Elemente, welche die Basiselemente verfeinern. Elemente, die keine Variationen haben, werden im Weiteren nicht dargestellt. [31]

Ereignis

Für das Ereignis existieren drei möglich Typen: Start-, Zwischen- und Endereignis. Bei den Start- und Zwischenereignissen wird außerdem noch zwischen unterbrechenden und nicht-unterbrechenden Ereignis differenziert. Weiterhin wird auch zwischen angehefteten und nicht-angehefteten Zwischenereignissen unterschieden. Die einzelnen Darstellungsformen und die dazugehörige Definition werden in den Tabelle I.1 und Tabelle I.2 dargestellt. Eine detaillierte Beschreibung der einzelnen Ereignisse kann in der Spezifikation der BPMN 2.0 nachgelesen werden.
























	Standard	Ereignis-Teilprozess unterbrechend	Ereignis-Teilprozess nicht-unterbrechend
Blanko			
Nachricht			
Timer			
Fehler			
Eskalation			
Bedingung			
Link	-	-	-
Abbruch	-	-	-
Komepnsation			
Signal			
Mehrfach			
Mehrfach/Parallel			
Terminierung	-	-	-
	-	-	-

Tabelle I.1: BPMN-Startereignisse (vgl. OMG-BPMN [31])









































	Zwischenreignis			Endereignis	
	Eingetreten	Angeheftet unterbrechend	Angeheftet nicht-unterbrechend	Ausgelöst	Standard
Blanko					
Nachricht					
Timer					
Fehler					
Eskalation					
Bedingung					
Link					
Abbruch					
Kompensation					
Signal					
Mehrfach					
Mehrfach/Parallel					
Terminierung					

Tabelle I.2: BPMN-Zwischen- und Endereignisse (vgl. OMG-BPMN [31])

Aktivität

Aktivitäten sind in der BPMN Prozesse, Sub-Prozesse und Aufgaben, wobei diese die kleinste Einheit in einem Prozess bilden. Sie stellen eine atomare Aktivität dar, wodurch eine weitere Verfeinerung nicht möglich ist. Eine Aufgabe wird durch einen Namen im Element gekennzeichnet (Abbildung I.13).

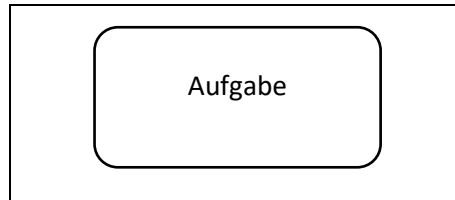


Abbildung I.13: Aufgabe

Die Aufgabe kann mit einem Symbol versehen werden, um die Aufgabenart genauer zu definieren. Die BPMN Spezifikation unterscheidet hier zwischen sieben Arten von Aufgaben: Service Aufgabe, Skript-Aufgabe, Benutzer-Aufgabe, Manuelle Aufgabe, Empfangs-Aufgabe, Sende-Aufgabe und Geschäftsregel-Aufgabe. Der Sub-Prozess ist eine zusammengesetzte Aktivität und ist somit auch nicht-atomarer Art. Der Sub-Prozess kann Sub-Aktivitäten beinhalten, welche wiederum Aufgaben oder Sub-Prozesse sein können. Die Abbildung I.14 zeigt einen zusammengeklappten Sub-Prozess, wohingegen in der Abbildung I.15 ein expandierter Sub-Prozess zu sehen ist. Wichtig ist, dass ein Sequenzfluss die Grenze eines Sub-Prozesses nicht überschreiten kann. Folglich befindet sich innerhalb eines Sub-Prozesses immer ein abgeschlossener (Teil-) Prozess.

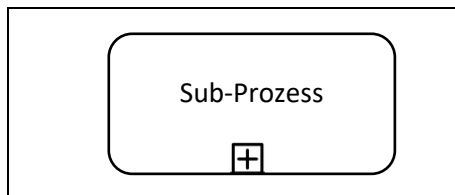


Abbildung I.14: zusammengeklappter Sub-Prozess

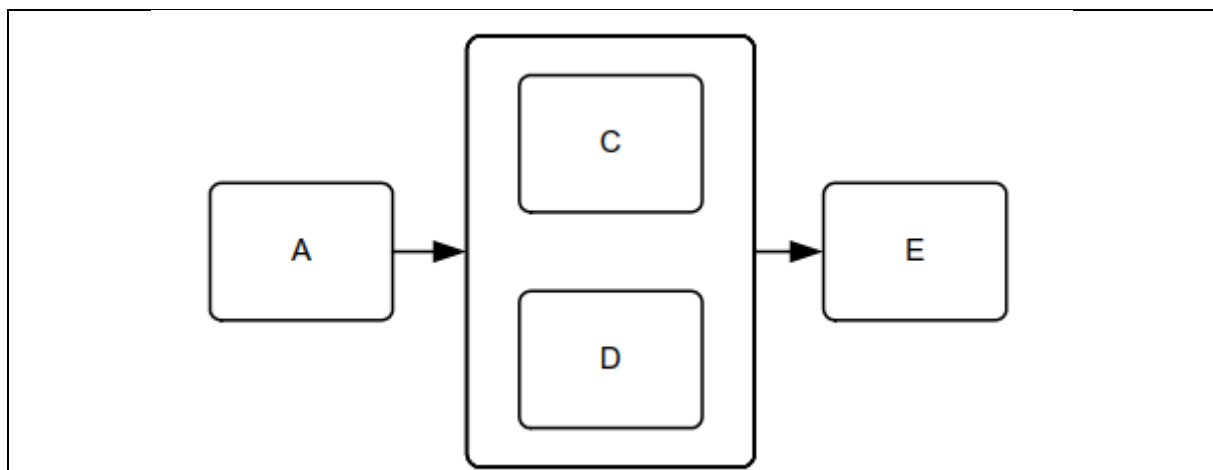


Abbildung I.15: expandierter Sub-Prozess

Multi-Instanzen

Aktivitäten können als Multi-Instanz auftreten. Dabei wird zwischen paralleler und sequentieller Ausführung unterschieden. Die Abbildung I.16 verbildlicht eine parallele Multi-Instanz, wohingegen die Abbildung I.17 eine sequentielle Multi-Instanz darstellt.

Die Multi-Instanzen dienen der Effizienz, indem mehrere Instanzen der Aufgabe gebildet werden. So kann eine parallele Aktivität beispielsweise von mehreren Mitarbeitern bearbeitet werden, bei einer sequentiellen Aktivität wird die Aufgabe für jedes Objekt wiederholt.

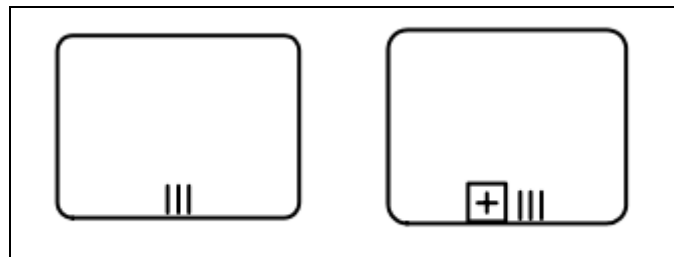


Abbildung I.16: parallele Multiinstanzen #1

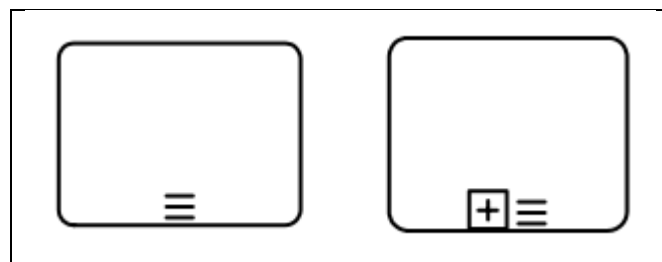


Abbildung I.17: sequentielle Multiinstanzen #2

Schleife

Sowohl die Aktivität als auch der Sub-Prozess können als Schleife auftreten. Die Abbildung I.18 und Abbildung I.19 zeigen die einzelnen Darstellungsformen.

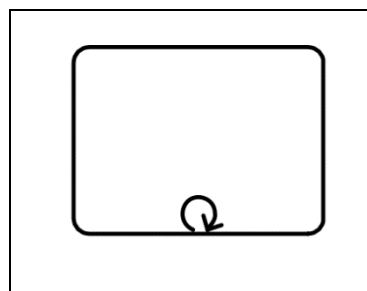


Abbildung I.18: Schleife

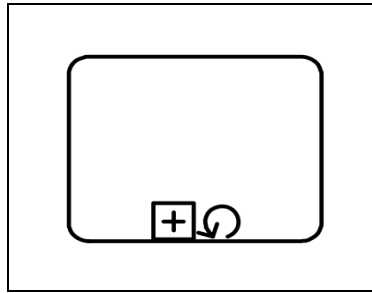


Abbildung I.19: Schleife als Sub-Prozess

Kompensation

Eine weitere spezielle Markierung ist die Kompensation. Die Kompensation hat ein angeheftetes Zwischenereignis, das sogenannte fangende Kompensations-Zwischenereignis. Zwischen diesem Ereignis und der Kompensationsaktivität besteht eine gerichtete Assoziation, kein Sequenzfluss. Dadurch wird der Zusammenhang verdeutlicht. Wird nun irgendwo im Prozess ein werfendes Kompensationsereignis erreicht, so werden die entsprechenden Kompensationsaktivitäten ausgeführt. Die Kompensation wird benutzt, wenn ein Fehler aufgetreten ist und zuvor ausgeführte Aufgaben oder deren Ergebnisse rückgängig gemacht werden müssen.

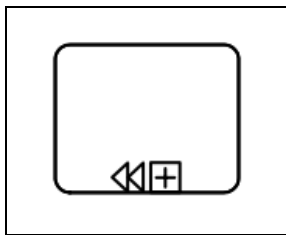


Abbildung I.20: Sub-Prozess Kompensation

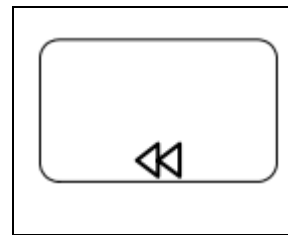


Abbildung I.21: Kompensation

Sequenzfluss

Die Spezifikationen der BPMN enthalten auch verfeinerte Darstellungsformen des Sequenzflusses, indem er als „Default“ Fluss (Abbildung I.23) oder als bedingter Sequenzfluss (Abbildung I.22) ausgewiesen wird.

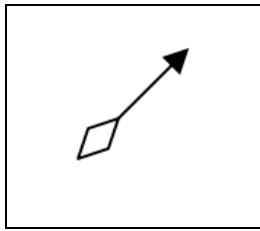


Abbildung 1.22:
bedingter Sequenzfluss

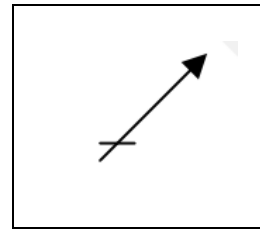


Abbildung 1.23: Default
Sequenzfluss

Gateway

Um den Sequenzfluss zu lenken, werden in der BPMN mehrere Gateways unterschieden: exklusives Gateway, inklusives Gateway, paralleles Gateway, komplexes Gateway sowie ereignis-basiertes Gateway. Die meisten dieser Gateways werden sowohl zur Verzweigung als auch zur Zusammenführung des Flusses genutzt, sodass sie zwei Verhaltensmuster aufweisen. Sie können diese Verhaltensmuster auch gleichzeitig aufweisen, also zusammenführen und verzweigen in einem Schritt. Dieses Verhalten kann allerdings auch durch zwei hintereinander gesetzte Gateways modelliert werden und wird daher auf diese Art und Weise modelliert. Ebenfalls wird in dieser Arbeit darauf geachtet, dass nach Möglichkeit ein verzweigendes Gateway immer seinen korrespondierenden Partner hat, das heißt, dass es zu jedem verzweigenden auch ein zusammenführendes Gateway der gleichen Art gibt. Manchmal soll jedoch ein Verhalten modelliert werden, das so nicht umsetzbar ist. In diesem Fall wird davon abgewichen.

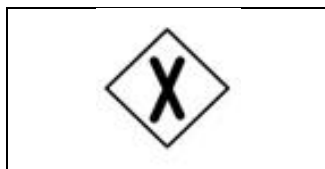


Abbildung 1.24: XOR-Gateway
/ exklusives Gateway

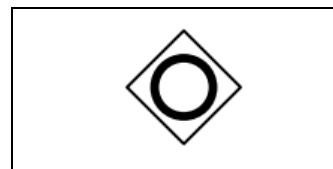


Abbildung 1.25: ODER-
Gateway / inklusives Gateway

Beim exklusiven Gateway wird der Sequenzfluss abhängig von den Bedingungen an genau eine ausgehende Kante weitergeleitet. Die Bedingungen werden nacheinander geprüft. Somit wird eine „XOR“-Beziehung modelliert. Wenn dieses Gateway benutzt wird, um den Fluss zusammenzuführen, dann wird jedes Mal, wenn eine Kante eine Marke mitbringt, der Sequenzfluss fortgesetzt. Es gibt zwei Möglichkeiten, um das exklusive Gateway darzustellen: entweder ohne internen Marker, also als „Blanko-Gateway“, oder durch das „X“ in der Raute. Um Verwirrung vorzubeugen, wird in dieser Arbeit im Weiteren nur die Raute mit dem „X“ verwendet.

Das inklusive Gateway wird durch einen Kreis in der Raute gekennzeichnet. Eingesetzt bei der Verzweigung wird je nach Bedingung mindestens eine, gegebenenfalls aber auch alle ausgehenden Kanten ausgewählt. Hier wird demnach eine „OR“-Beziehung modelliert. Bei der Zusammenführung wird auf so viele Kanten gewartet, wie bei der Verzweigung aktiviert wurden, also mindestens auf eine, aber gegebenenfalls auch alle eingehenden Kanten.

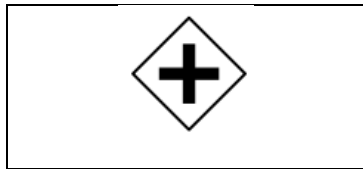


Abbildung I.26: UND-Gateway /
paralleles Gateway

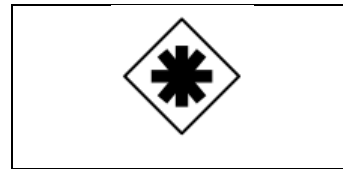


Abbildung I.27: Komplexes
Gateway

Das parallele Gateway modelliert die „AND“-Beziehung. Bei der Verzweigung werden alle ausgehenden Kanten aktiviert, was einer Parallelisierung des Ablaufes entspricht. Das gleiche Verhalten kann erzeugt werden, indem aus einer Aktivität einfach zwei Sequenzflüsse herauskommen. Um Verwirrung vorzubeugen, wird in dieser Arbeit im Weiteren für dieses Verhalten immer das Gateway benutzt. Bei der Zusammenführung mit diesem Gateway wird auf alle eingehenden Kanten gewartet, ehe der Sequenzfluss fortgesetzt wird. Dieses Gateway hat als internen Marker ein Plus.

Das komplexe Gateway wird benutzt, um ein Verzweigungs- und Zusammenführungsverhalten zu modellieren, das nicht von den anderen Gateways erfasst wird. Das Gateway wird durch einen Asterisk gekennzeichnet.

Es gibt noch drei weitere Gateways in der BPMN: das ereignis-basierte Gateway, das exklusive ereignis-basierte Gateway sowie das parallele ereignis-basierte Gateway. Das erstgenannte Gateway erscheint im normalen Sequenzfluss, wenn eine Entscheidung von eintretenden Ereignissen bestimmt wird. Ein solches Ereignis kann beispielsweise der Erhalt einer Nachricht eines anderen Teilnehmers sein. Es müssen mindestens zwei ausgehende Kanten an dieser Verzweigung anliegen, aber es wird nur eine ausgewählt.



Abbildung I.28: ereignis-basiertes Gateway

Das ereignis-basierte Gateway gibt es auch als „Startereignis“. Dann soll es anzeigen, dass mehrere Ereignisse einen Prozess auslösen können. Sobald eines der Ereignisse eintritt, wird der normale Sequenzfluss aufgenommen. Da dieser Umstand allerdings auch mit mehreren Startereignissen modelliert werden kann, wird auf seine Verwendung im Folgenden verzichtet.

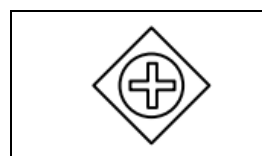


Abbildung I.29: paralleles ereignis-basiertes Gateway (Instanziierung)

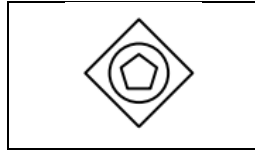


Abbildung I.30: exklusives ereignis-basiertes Gateway (Instanziierung)

Ad-hoc-Sub-Prozess

Für die Sub-Prozesse existieren mehrere spezielle Darstellungsformen. Eine davon ist der Ad-hoc-Sub-Prozess. In Abbildung I.30 ist dieser genauer dargestellt. Der Ad-hoc-Sub-Prozess ist nicht genauer definiert, da in diesem kein Angaben zur Reihenfolge oder zur Anzahl der Wiederholungen einzelner Aktivitäten hinterlegt werden. Auf seine Verwendung wird daher, wenn möglich, verzichtet.

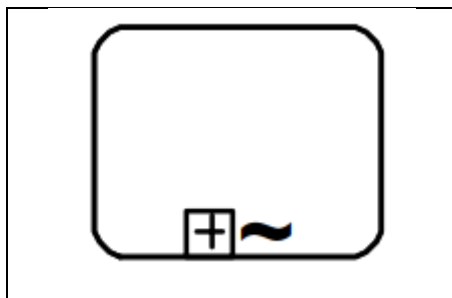


Abbildung I.31: Ad-hoc-Sub-Prozess

Transaktions-Sub-Prozess

Der Transaktions-Sub-Prozess zeichnet sich durch einen doppelten Rand und einem fangendem Abbruch-Zwischenereignis aus. Innerhalb des Sub-Prozesses gibt es ein Abbruch-Endereignis. Der Sub-Prozess versucht das „alles oder nichts“ – Prinzip zu realisieren. Wird im Sub-Prozess das Abbruch-Endereignis erreicht, so werden alle Aktivitäten abgebrochen und alle beinhalteten kompensatorischen Aktivitäten ausgeführt. Danach geht der Sequenzfluss beim fangenden Abbruch-Zwischenereignis weiter. Die Abbruch-Ereignisse treten nur im Zusammenhang mit dem Transaktions-Sub-Prozess auf.

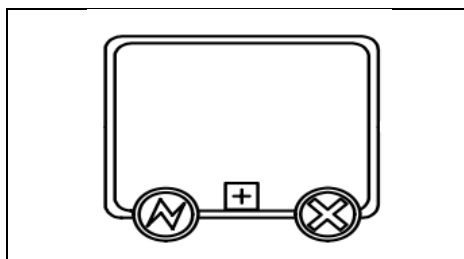


Abbildung I.32: Transaktions-Sub-Prozess

Ereignis-basierter-Sub-Prozess

Der Ereignis-basierte-Sub-Prozess ist ein weiterer Spezialfall für Sub-Prozesse und wird mit einem gestrichelten Rahmen und durch ein fangendes unterbrechendes oder ein nicht-unterbrechendes Starterereignis in der linken oberen Ecke verbildlicht.

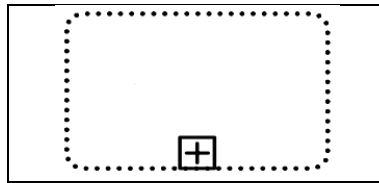


Abbildung I.33: Ereignis-basierter-Sub-Prozess

Datenobjekt

Bei den Basiselementen der BPMN wurde alles Wichtige zu den Datenobjekten schon genau erläutert. Jedoch kann ihre Darstellungsform verfeinert werden. Die unter Abbildungen zeigen die möglichen Darstellungen. In Abbildung I.34 wird eine Sammlung verbildlicht, in Abbildung I.35 und I.36 die Eingabe bzw. das Ergebnis von Daten und in Abbildung I.37 ein Datenlager dargestellt.

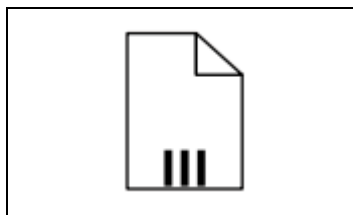


Abbildung I.34: Datensammlung

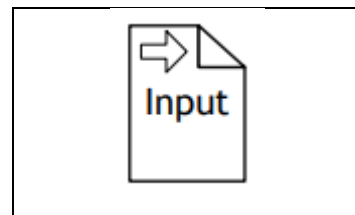


Abbildung I.35: Dateninput/-
eingabe

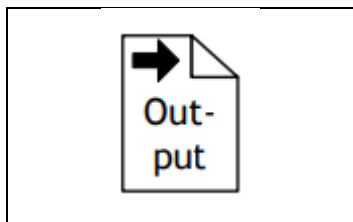


Abbildung I.36: Datenoutput/-
ergebnis

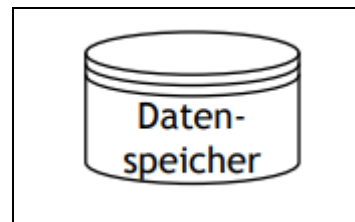


Abbildung I.37: Datenspeicher

Notationselemente der EPK

Funktion

Die Funktion Abbildung I.38 steht für eine Tätigkeit, welche einen Input und Output transformiert. Mit ihr werden die Bearbeitungsschritte des Prozesses abgebildet. Sie werden durch ein auftretendes Ereignis durchgeführt und lösen neue Ereignisse aus.

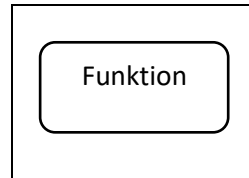


Abbildung I.38: Funktion

Ereignis

Jeder Prozess beginnt und endet mit mindestens einem **Ereignis**. Es beschreibt einen eingetretenen Systemzustand, dem eine oder mehrere Funktionen folgen.

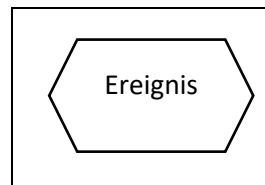


Abbildung I.39: Ereignis

Dokument

Schriftliche Dokumente, die durch das Unternehmen „wandern“ bzw. in den Betrieb gelangen oder nach Außen gesendet werden. Dabei dient die Darstellungsform der Abgrenzung gegen Elemente eines Informationssystems.

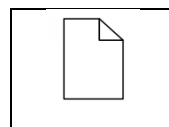


Abbildung I.40: Dokument

System / Informationsobjekt

Ein System modelliert ein Softwaresystem, das für eine bestimmte Funktion oder Anwendungsdomäne erstellt wurde. Das Informationsobjekt liefert die für die Durchführung einer Funktion benötigten Daten.

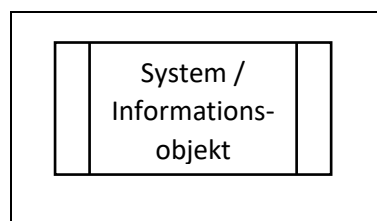


Abbildung I.41: System / Informationsobjekt

Kontrollfluss

Der Kontrollfluss setzt die Funktionen und Ereignisse gerichtet miteinander in Verbindung, wodurch sachlogische und zeitliche Abhängigkeiten abgebildet werden.

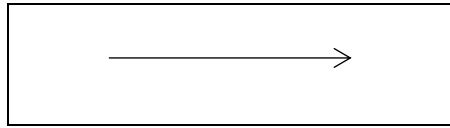


Abbildung I.42: Kontrollfluss

Informationsfluss

Der Informationsfluss beschreibt den Datenfluss zwischen dem Informationsobjekt (Bsp. Dokumente)

Zuordnung

Die Zuordnung zeigt den Zusammenhang zwischen Rolle oder Stelle und Funktion.

Konnektoren

Mit Hilfe der Konnektoren können Prozessverzweigungen modelliert werden. Man unterscheidet dabei 3 Arten von Konnektoren: das logische UND, INKLUSIVE ODER und EXKLUSIVE ODER.



Abbildung I.43: UND-Konnektor

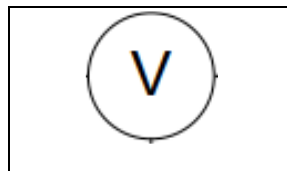


Abbildung I.44: ODER-Konnektor

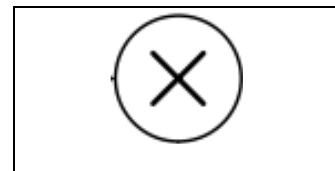


Abbildung I.45: XOR-Konnektor

Rolle

Eine organisatorische Rolle beschreibt einen Tätigkeitsbereich, der von einer von mehreren Personen, die diesem Bereich zugeordnet ist, ausgeführt wird.

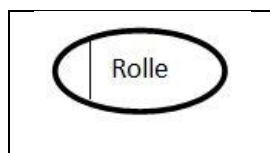


Abbildung I.46: Rolle

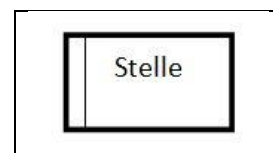


Abbildung I.47: Stelle

Stelle

Eine organisatorische Stelle beschreibt einen Tätigkeitsbereich, der von einer konkreten Person ausgeführt wird.

Prozessschnittstelle

Die Prozess-Schnittstelle dient dazu, den Gesamtprozess in Teilprozesse zu untergliedern und verweist auf einen anderen Teilprozess. Die Prozessschnittstelle ist dabei weder Funktion noch Ereignis und kann daher nicht allein genutzt werden.



Abbildung I.48: Prozessschnittstelle

Workflow Pattern

Datenmuster (Data Patterns)

	BPMN	EPK
Data Visibility		
21. Task Data	+	-
22. Block Data	+	-
23. Scope Data	-	-
24. Multiple Instance Data	+/-	-
25. Case Data	+	-
26. Folder Data	-	-
27. Workflow Data	-	-
28. Environment Data	-	-
Data Interaction (Internal)		
29. Task to Task	+	-
30. Block Task to Sub-Workflow Decomposition	+	-
31. Sub-Workflow Decomposition to Block Task	+	-
32. to Multiple Instance Task	-	-
33. from Multiple Instance Task	-	-
34. Case to Case	-	-
Data Interaction (External)		
35. Task to Environment – Push-Orientated	+	-
36. Environment to Task – Pull-Orientated	+	-
37. Environment to Task – Push-Orientated	+	-
38. Task to Environment – Pull-Orientated	+	-
39. Case to Environment – Push-Orientated	-	-
40. Environment to Case – Pull-Orientated	-	-
41. Environment to Case – Push-Orientated	-	-
42. Case to Environment – Pull-Orientated	-	-
43. Workflow to Environment – Push-Orientated	-	-
44. Environment to Workflow – Pull-Orientated	-	-
45. Environment to Workflow – Push-Orientated	-	-
46. Workflow to Environment – Pull-Orientated	-	-
Data Transfer		
47. By Value – Incoming	+	-
48. By Value – Outgoing	+	-
49. Copy In / Copy Out	+/-	-
50. By Reference – Unlocked	-	-
51. By Reference – Locked	+	-
52. Data Transformation – Input	+/-	-
53. Data Transformation – Output	+/-	-
Date-based Routing		
54. Task Precondition – Data Existence	+	-
55. Task Precondition – Data Value	-	-
56. Task Postcondition – Data Existence	+	-
57. Task Postcondition – Data Value	-	-
58. Event-based Task Trigger	+	-
59. Data-based Task Trigger	+	-
60. Data-based Routing	+	-

Tabelle 1.3: Datenmuster des Workflow Pattern (vgl. WPI, Wohed et al [36])

Ein „+“ in den Tabellen bedeutet direkte Unterstützung des Musters, ein „+/-“ eine teilweise Unterstützung und ein „-“ ein Fehlen der Unterstützung. [36]

- 1. Task Data:** Data Elements can be defined by tasks which are accessible only within the context of individual execution instances of that task.
- 2. Block Data:** Block tasks (i.e. task which can be described in terms of a corresponding subprocess) are able to define data elements which are accessible by each of the components of the corresponding subprocess.
- 3. Scope Data:** Data elements can be defined which are accessible by a subset of the task in a case.
- 4. Multiple Instance Data:** Tasks which are able to execute multiple times within a single case can define data elements which are specific to an individual execution instance.
- 5. Case Data:** Data elements are supported which are specific to process instances or case. They can be accessed by all components of the process during the execution of the case.
- 6. Folder Data:** Data elements can be defined which are accessible by multiple cases on a selective basis. They are accessible to all components of the cases to which they are bound.
- 7. Workflow Data:** Data elements are supported which are accessible to all components in each and every case of the process and are within the context of the process itself.
- 8. Environment Data:** Data elements which exist in the external operating environment are able to be accessed by components of processes during execution.
- 9. Task to Task:** The ability to communicate data elements between one task instance and another within the same case. The communication of data elements between two tasks is specified in a form that is independent of the task definitions themselves.
- 10. Block Task to Sub-Workflow Decomposition:** The ability to pass data elements from a block task instance to the corresponding subprocess that defines its implementation. Any data elements that are available to a block task are able to be passed to (or be accessed) in the associated subprocess although only a specifically nominated subset of those data elements are actually passed to the subprocess.
- 11. Sub-Workflow Decomposition to Block Task:** The ability to pass data elements from the underlying subprocess back to the corresponding block task. Only nominated data elements defined as part of the subprocess are made available to the (parent) block task.
- 12. to Multiple Instance Task:** The ability to pass data elements from a preceding task instance to a subsequent task which is able to support multiple execution instances. This may involve passing the data elements to all instances of the multiple instance task or distributing them on a selective basis. The data passing occurs when the multiple instance task is enabled.

13. **from Multiple Instance Task:** The ability to pass data elements from a task which supports multiple execution instances to a subsequent task. The data passing occurs at the conclusion of the multiple instance task. It involves aggregating data elements from all instances of the task and passing them to a subsequent task.
14. **Case to Case:** The passing of data elements from one case of a process during its execution to another case that is executing concurrently.
15. **Task to Environment – Push:** The ability of a task to initiate the passing of data elements to a resource or service in the operating environment.
16. **Environment to Task – Pull:** The ability of a task to request data elements from resources or services in the operational environment.
17. **Environment to Task – Push:** The ability for a task to receive and utilise data elements passed to it from services and resources in the operating environment on an unscheduled basis.
18. **Task to Environment – Pull:** The ability of a task to receive and respond to requests for data elements from services and resources in the operational environment.
19. **Case to Environment – Push:** The ability of a case to initiate the passing of data elements to a resource or service in the operational environment.
20. **Environment to Case – Pull:** The ability of a case to request data from services or resources in the operational environment.
21. **Environment to Case – Push:** The ability of a case to accept data elements passed to it from services or resources in the operating environment.
22. **Case to Environment – Pull:** The ability of a case to respond to requests for data elements from a service or resource in the operating environment.
23. **Workflow to Environment – Push:** The ability of a process environment to pass data elements to resources or services in the operational environment.
24. **Environment to Workflow – Pull:** The ability of a process environment to request global data elements from external applications.
25. **Environment to Workflow – Push:** The ability of services or resources in the operating environment to pass global data to a process.
26. **Workflow to Environment – Pull:** The ability of the process environment to handle requests for global data from external applications.
27. **Data Transfer by Value – Incoming:** The ability of a process component to receive incoming data elements by value avoiding the need to have shared names or common address space with the component(s) from which it receives them.

- 28. Data Transfer by Value – Outgoing:** The ability of a process component to pass data elements to subsequent components as values avoiding the need to have shared names or common address space with the component(s) to which it is passing them.
- 29. Data Transfer – Copy In/Copy Out:** The ability of a process component to copy the values of a set of data elements from an external source (either within or outside the process environment) into its address space at the commencement of execution and to copy their final values back at completion.
- 30. Data Transfer by Reference – Unlocked:** The ability to communicate data elements between process components by utilizing a reference to the location of the data element in some mutually accessible location. No concurrency restrictions apply to the shared data element.
- 31. Data Transfer by Reference – with Lock:** The ability to communicate data elements between process components by passing a reference to the location of the data element in some mutually accessible location. Concurrency restrictions are implied with the receiving component receiving the privilege of read-only or dedicated access to the data element. The required lock is declaratively specified as part of the data passing request.
- 32. Data Transformation – Input:** The ability to apply a transformation function to a data element prior to it being passed to a process component. The transformation function has access to the same data elements as the receiving process component.
- 33. Data Transformation – Output:** The ability to apply a transformation function to a data element immediately prior to it being passed out of a process component. The transformation function has access to the same data elements as the process component that initiates it.
- 34. Task Precondition – Data Existence:** Data-based preconditions can be specified for tasks based on the presence of data elements at the time of execution. The preconditions can utilize any data elements available to the task with which they are associated. A task can only proceed if the associated precondition evaluates positively.
- 35. Task Precondition – Data Value:** Data-based preconditions can be specified for tasks based on the value of specific parameters at the time of execution. The preconditions can utilize any data elements available to the task with which they are associated. A task can only proceed if the associated precondition evaluates positively.
- 36. Task Postcondition – Data Existence:** Data-based postconditions can be specified for tasks based on the existence of specific parameters at the time of task completion. The postconditions can utilize any data elements available to the task with which they are associated. A task can only proceed if the associated postcondition evaluates positively.
- 37. Task Postcondition – Data Value:** Data-based postconditions can be specified for tasks based on the value of specific parameters at the time of execution. The postconditions can utilize any data elements available to the task with which they are associated. A task can only proceed if the associated postcondition evaluates positively.
- 38. Event-based Task Trigger:** The ability for an external event to initiate a task and to pass data elements to it.

- 39. Data-based Task Trigger:** Data-based task triggers provide the ability to trigger a specific task when an expression based on data elements in the process instance evaluates to true. Any data element accessible within a process instance can be used as part of a data-based trigger expression.
- 40. Data-based Routing:** Data-based routing provides the ability to alter the control-flow within a case based on the evaluation of data-based expressions. A data-based routing expression is associated with each outgoing arc of an OR-split or XOR-split. It can be composed of any data-values, expressions and functions available in the process environment providing it can be evaluated at the time the split construct with which it is associated completes. Depending on whether the construct is an XOR-split or OR-split, a mechanism is available to select one or several outgoing arcs to which the thread of control should be passed based on the evaluation of the expressions associated with the arcs.

Ressourcenmuster (Resource Patterns)

	BPMN	EPK
Creation Patterns		
1. Direct Allocation	+	-
2. Role-based Allocation	+	-
3. Deferred Allocation	-	-
4. Authorization	-	-
5. Separation on Duties	-	-
6. Case Handling	-	-
7. Retain Familiar	-	-
8. Capability-based Allocation	-	-
9. History-based Allocation	-	-
10. Organizational Allocation	-	-
11. Automatic Execution	+	-
Push Patterns		
12. Distribution by Offer-Single Resource	-	-
13. Distribution by Offer-Multiple Resources	-	-
14. Distribution by Allocation-Single Resource	+	-
15. Random Allocation	-	-
16. Round Robin Allocation	-	-
17. Shortest Queue	-	-
18. Early Distribution	-	-
19. Distribution on Enablement	+	-
20. Late Distribution	-	-
Pull Patterns		
21. Resource-Initiated Allocation	-	-
22. Resource-Initiated Execution – Allocated Work Item	-	-
23. Resource-Initiated Execution – Offered Work Item	-	-
24. System-Determined Work Queue Content	-	-
25. Resource-Determined Work Queue Content	-	-
26. Selection Autonomy	-	-
Detour Patterns		
27. Delegation	-	-
28. Escalation	-	-
29. Deallocation	-	-
30. Stateful Reallocation	-	-
31. Stateless Reallocation	-	-
32. Suspension / Resumption	-	-
33. Skip	-	-
34. Redo	-	-
35. Pre-Do	-	-
Auto-Start Patterns		
36. Commencement on Creation	+	-
37. Commencement on Allocation	-	-
38. Piled Execution	-	-
39. Chained Execution	+	-

Table 1.4: Ressourcenmuster des Workflow Pattern (Teil 1) (vgl. WPI, Wohed et al [36])

	BPMN	EPK
Visibility Patterns		
Configurable Unallocated Work Item Visibility	-	-
Configurable Allocated Work Item Visibility	-	-
Multiple Ressource Patterns		
Simultaneous Execution	+	-
Additional Resources	-	-

Tabelle I.5: Ressourcenmuster des Workflow Pattern (Teil 2) (vgl WPI, Wohed et al [36])

Ein „+“ in den Tabellen bedeutet direkte Unterstützung des Musters, ein „+/-“ eine teilweise Unterstützung und ein „-“ ein Fehlen der Unterstützung.

- 1. Direct Distribution:** The ability to specify at design time the identity of the resource(s) to which instances of this task will be distributed at runtime.
- 2. Role-Based Distribution:** The ability to specify at design-time one or more roles to which instances of this task will be distributed at runtime. Roles serve as a means of grouping resources with similar characteristics. Where an instance of a task is distributed in this way, it is distributed to all resources that are members of the role(s) associated with the task.
- 3. Deferred Distribution:** The ability to specify at design-time that the identification of the resource(s) to which instances of this task will be distributed will be deferred until runtime.
- 4. Authorization:** The ability to specify the range of privileges that a resource possesses in regard to the execution of a process. In the main, these privileges define the range of actions that a resource can initiate when undertaking work items associated with tasks in a process.
- 5. Seperation of Duties:** The ability to specify that two tasks must be executed by different resources in a given case.
- 6. Case Handling:** The ability to allocate the work items within a given case to the same resource at the time that the case is commenced.
- 7. Retain Familiar:** Where several resources are available to undertake a work item, the ability to allocate a work item within a given case to the same resource that undertook a preceding work item.
- 8. Capability Based Distribution:** The ability to distribute work items to resources based on specific capabilities that they possess. Capabilities (and their associated values) are recorded for individual resources as part of the organizational model.
- 9. History Based Distribution:** The ability to distribute work items to resources on the basis of their previous execution history.
- 10. Organisational Distribution:** The ability to distribute work items to resources based their position within the organisation and their relationship with other resources.

- 11. Automatic Execution:** The ability for an instance of a task to execute without needing to utilise the services of a resource.
- 12. Distribution by Offer – Single Resource:** The ability to distribute a work item to a selected individual resource on a non-binding basis.
- 13. Distribution by Offer – Multiple Resource:** The ability to distribute a work item to a group of selected resources on a non-binding basis.
- 14. Distribution by Allocation – Single Resource:** The ability to distribute a work item to a specific resource for execution on a binding basis.
- 15. Random Allocation:** The ability to allocate work items to a selected resource chosen from a group of eligible resources on a random basis.
- 16. Round Robin Allocation:** The ability to allocate a work item to a selected resource chosen from a group of eligible resources on a cyclic basis.
- 17. Shortest Queue:** The ability to allocate a work item to a selected resource chosen from a group of eligible resources on the basis of having the shortest work queue.
- 18. Early Distribution:** The ability to advertise and potentially distribute a work items to resources ahead of the moment at which it is actually enabled.
- 19. Distribution on Enablement:** The ability to advertise and distribute a work items to resources at the moment that the task to which it corresponds is enabled for execution.
- 20. Late Distribution:** The ability to advertise and distribute work items to resources after the task to which the work item corresponds has been enabled for execution.
- 21. Resource-Initiated Allocation:** The ability for a resource to commit to undertake a work item without needing to commence working on it immediately.
- 22. Resource-Initiated Execution – Allocated Work Item:** The ability for a resource to commence work on a work item that is allocated to it.
- 23. Resource-Initiated Execution – Offered Work Item:** The ability for a resource to select a work item offered to it and commence work on it immediately.
- 24. System Determined Work Queue Content:** The ability of the system to order the content and sequence in which work items are presented to a resource for execution.
- 25. Resource-Determined Work Queue Content:** The ability for resources to specify the format and content of work items listed in the work queue for execution.
- 26. Selection Autonomy:** The ability for resources to select a work item for execution based on its characteristics and their own preferences.

27. **Delegation:** The ability for a resource to allocate an unstarted work item previously allocated to it (but not yet commenced) to another resource.
28. **Escalation:** The ability of a system to distribute a work item to a resource or group of resources other than those it has previously been distributed to in an attempt to expedite the completion of the work item.
29. **Deallocation:** The ability of a resource (or group of resources) to relinquish a work item which is allocated to it (but not yet commenced) and make it available for distribution to another resource or group of resources.
30. **Stateful Reallocation:** The ability of a resource to allocate a work item that they are currently executing to another resource without loss of state data.
31. **Stateless Reallocation:** The ability for a resource to reallocate a work item that it is currently executing to another resource without retention of state.
32. **Suspension/Resumption:** The ability for a resource to suspend and resume execution of a work item.
33. **Skip:** The ability for a resource to skip a work item allocated to it and mark the work item as complete.
34. **Redo:** The ability for a resource to redo a work item that has previously been completed in a case. Any subsequent work items (i.e. work items that correspond to subsequent tasks in the process) must also be repeated.
35. **Pre-Do:** The ability for a resource to execute a work item ahead of the time that it has been offered or allocated to resources working on a given case. Only work items that do not depend on data elements from preceding work items can be "pre-done".
36. **Commencement on Creation:** The ability for a resource to commence execution on a work item as soon as it is created.
37. **Commencement on Allocation:** The ability to commence execution on a work item as soon as it is allocated to a resource.
38. **Piled Execution:** The ability to initiated the next instance of a task (perhaps in a different case) once the previous one has completed with all associated work items being allocated to the same resource. The transition to *Piled Execution* mode is at the instigation of an individual resource. Only one resource can be in *Piled Execution* mode for a given task at any time.
39. **Chained Execution:** The ability to automatically start the next work item in a case once the previous one has completed. The transition to *Chained Execution* mode is at the instigation of the resource.
40. **Configurable Unallocated Work Item Visibility:** The ability to configure the visibility of unallocated work items by process participants.

- 41. Configurable Allocated Work Item Visibility:** The ability to configure the visibility of allocated work items by process participants.
- 42. Simultaneous Execution:** The ability for a resource to execute more than one work item simultaneously.
- 43. Additional Resources:** The ability for a given resource to request additional resources to assist in the execution of a work item that it is currently undertaking.