

Institut für Rechnergestützte Ingenieursysteme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 45

Metriken in Geschäftsprozessmodellen und BPMN

Özgür Kibarogullari

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Roller
Betreuer/in:	Dipl.-Inf. Felix Baumann

Beginn am: 18. Februar 2013

Beendet am: 14. August 2013

CR-Nummer: D.2.8, G.2.2, D.2.2

Kurzfassung

Die vorliegende Arbeit beschreibt die verschiedenen Metriken in der Geschäftsprozessmodellierung und der BPMN. Die Metriken werden detailliert mit jeweiligen mathematischen Kenntnissen und Beispielen dargestellt. Darüber hinaus wird die Programmierung von Metriken mit der Programmiersprache JAVA implementiert.

Die Metriken sind in drei Hauptkategorien unterteilt, nämlich Fehlermetriken, Komplexitätsmetriken und Qualitätsmetriken. Die Fehlermetriken besagen, wie hoch die Wahrscheinlichkeit ist, dass ein Fehler im Modell vorkommt. Die Komplexitätsmetriken zeigen, wie komplex das Modell ist, so dass das Modell noch einfacher dargestellt werden soll. Die Qualitätsmetriken sind meist die Adaptation von Softwaremetriken in den Prozessmetriken. Anhand Qualitätsmetriken kann herausgefunden werden, wie qualitativ das Modell ist.

Abstract

The working paper describes the different metrics in business process models. The metrics will be described with mathematical basics and examples. At the end, the programming part about the metrics will be implemented using programming language Java.

The metrics are divided in three different main categories including Error metrics, Complexity metrics and Quality metrics. Error metrics show how big the probability is, that the model contains an error. The complexity metrics give us the values, how complex the model is. The quality metrics are adaptations from software metrics to process metrics. The quality metrics give an idea how qualitative the model is.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Aufgabenstellung	10
2	Die Grundlagen der Geschäftsprozessmanagement	13
2.1	Business Process	14
2.2	Business Process Management	16
3	Modellierungssprachen	19
3.1	Petri-Netze	20
3.2	EPK	22
3.3	UML - Aktivitätsdiagramm	25
3.4	BPMN	27
4	Metriken	37
4.1	Fehlermetriken	37
4.2	Komplexitätsmetriken	50
4.3	Qualitätsmetriken	60
4.4	Liste aller Metriken	73
5	Modellkatalog	77
6	Programmierung der Metriken	83
6.1	Aufbau des Programms	83
6.2	Implementierung von Metriken	84
6.3	Programmtest	87
7	Zusammenfassung und Fazit	89
	Literaturverzeichnis	93

Abbildungsverzeichnis

3.1	Diagrammbasierte Methoden [Gadatsch 04]	19
3.2	Elemente in Petri-Netzen [Jochim 01]	20
3.3	Die Jahreszeiten als Petri-Netz Beispiel [Wikimedia 11]	21
3.4	Die Überprüfung eines Antrages als EPK-Model [Schmitzl 10]	22
3.5	EPK Symbole [Wikimedia 06]	23
3.6	Der Prüfungsprozess als EPK Beispiel [Decker 09]	24
3.7	Die Notation des Aktivitätsdiagramms [Dumke 01]	25
3.8	Beispiel für ein UML-Aktivitätsdiagramm [Wikipedia 05]	26
3.9	Versandprozess eines Hardware Teils [EAP 12]	27
3.10	Hauptereignisarten	28
3.11	Die Event Typen in BPMN [BPMO 11]	30
3.12	Aktivitäten Typen in BPMN	31
3.13	Aktivitäten Markierungen und Aufgaben-Typen [BPMO 11]	31
3.14	Gateways in BPMN (erstellt mit Signavio BPMN Editor) ¹	32
3.15	Die Art der Verbindungen in BPMN	33
3.16	Darstellung eines Pools mit 2 Lanes	34
3.17	3 Arten von Artefakten in BPMN	34
3.18	Ein Beispiel mit Artefakten in BPMN [Wikimedia 06]	35
4.1	Beispiel eines Cut-Vertex Punktes	39
4.2	Zwei verschiedene EPK Modelle mit unterschiedlichen Werten der Trennbarkeitsmetrik [Mendling 07]	40
4.3	Zwei EPK Modelle mit unterschiedlichen Werten der Strukturierheitsmetrik [Mendling 07]	42
4.4	Zwei EPK Modelle mit unterschiedlichen Werten der Zyklizitätsmetrik	43
4.5	Zwei EPK Modelle mit gleicher Werten der Parallelitätsmetrik	44
4.6	Zwei EPK Modelle mit unterschiedlicher Dichte [Mendling 07]	46
4.7	Ein EPK Beispiel mit in-depth und out-depth Werten [Mendling 07]	49
4.8	Ein BPMN Modell mit CFC Wert 6	52
4.9	Ein BPMN Modell mit maximaler Verschachtelungstiefe von 6	53
4.10	Ein BPMN Modell zur Fan-in und Fan-out Metrik	55
4.11	Ein BPMN Modell zur Schreibrichtung und Kantenüberschneidungen	57
4.12	Ein BPMN Modell mit RFP Metrik Berechnung	62
4.13	Ein BPMN Modell zur IFCP Metrik	63
4.14	Ein BPMN Modell zur LDA Metrik Berechnung	64
4.15	Ein BPMN Modell mit unterschiedlichen ICP und ECP Metrik Werte	65

4.16	Ein BPMN Modell zur PCpl Metrik	67
4.17	Drei verschiedene Teile eines Modells für die Coupling Metrik [Vanderfeesten 09]	68
4.18	Ein EPK Modellbeispiel für die Coupling Metrik [Vanderfeesten 09]	69
5.1	BPMN Diagramm über Versandprozess eines Gerätes	78
5.2	BPMN Diagramm über die Reisebuchung	80
6.1	Ein Screenshot aus dem Programm mit Metrikwerten	87

Tabellenverzeichnis

4.1	KognitiveWahrnehmung	54
4.2	Qualitätsmerkmale	60
4.3	OO-Software ins Prozessmodellierung	61
4.4	Fehlermetriken	73
4.5	Komplexitätsmetriken	74
4.6	Qualitätsmetriken	75

Verzeichnis der Listings

5.1	XML Code Abschnitt des BPMN Modells 5.1	79
5.2	XML Code Abschnitt des BPMN Modells 5.2	81
6.1	Untersuchung von Verknüpfungen	84
6.2	Untersuchung von Kanten	84
6.3	Untersuchung von Aktivitäten	85
6.4	Untersuchung von Knoten	85
6.5	Untersuchung von XOR-Verknüpfung und Einfluss auf die Metriken	86

Kapitel 1

Einleitung

Ich messe den Erfolg nicht an
meinen Siegen, sondern daran, ob
ich jedes Jahr besser werde.

(Tiger Woods)

Business Process Management ist schon seit langer Zeit ein wichtiger Teil jeden Arbeitsprozesses. Als vor ca. 100 Jahren realisiert wurde, wie das Geschäftsprozessmanagement die Qualität, die Geschwindigkeit und die Ressourcen der Arbeit beeinflussen kann, hat es sehr viel an Bedeutung gewonnen. Seit dem hat sich die Intention zur Verbesserung der Arbeit enorm gesteigert und die Prozessmodellierung hat immer mehr an Bedeutung gewonnen.

Die Prozessmodellierung ist schon seit einiger Zeit ein sehr wichtiges Werkzeug, um die Prozesse weiter zu optimieren und besser darzustellen. Durch gute Prozessmodellierung spart man Beiträge, die in Millionenhöhe liegen. Die großen Firmen stellen mittlerweile Experten für die Prozessmodellierung ein, was natürlich zeigt, wie wichtig dieses Arbeitsgebiet geworden ist. Während kleine Firmen durch intuitive Bearbeitung der Prozesse versuchen voran zu kommen, haben die großen Firmen die Möglichkeit Experten zu gewinnen und nur für die Prozessmodellierung einzusetzen.

Der gleiche Entwicklungsprozess galt auch für die Softwareentwicklung. Zunächst wurde realisiert, wie wichtig digitales Leben sein kann. Anschließend wurde erkannt, wie diese digitale Welt unser Leben erleichtern kann und dann wurde programmiert, programmiert und programmiert. Je wertvoller die Softwarebranche wurde, desto mehr neue Arbeitsgebiete wurden erschaffen. Neue Themenfelder tauchten auf. Was könnte getan werden, um eine konkrete Meinung über die Programmcodes zu bilden? Ist der Programmcode gut, schnell, leserlich, verständlich? In diesem Moment tauchten die Metriken auf. Die Metriken haben die Werte über die gesuchten Eigenschaften geliefert. Anhand Metriken konnte man die Softwarecodes „messen“.

Nach dem die Prozessmodelle immer mehr an Bedeutung gewonnen haben, entwickelte sich die Lage in der Geschäftsprozessmodellierung genauso wie bei der Softwarebranche. Um bessere Kontrolle über die Prozesse zu haben, werden Metriken verlangt und entwickelt. Da dies etwas neues in der Prozessmodellierung ist, gibt es keine große Menge an Metriken,

die zusammengesammelt worden ist. Die Metriken werden durch verschiedene Forscher auf der Welt entwickelt und freigegeben. So hat man im Land A gewisse Metriken und im Land B gewisse Metriken mehr und im Land C wieder gewisse andere Metriken. Hierzu reicht es aus, die Herkünfte der verschiedenen wissenschaftliche Arbeiten zu betrachten wie bei [Mendling 07], [Cardoso 05], [Gruhn 06], [Khelif 09], [Reijers 04]. Da die Metriken ihre Wurzeln in der Softwarebranche haben, wurde oft versucht die Softwaremetriken direkt umzuwandeln, so dass diese in Prozessmodellierung genauso eingesetzt werden können. Während J. Mendling eine sehr detaillierte Arbeit [Mendling 07] über die Fehlermetriken geschrieben hat, hat J. Cardoso, aus Portugal, seine Leistung beigetragen, um Komplexitätsmetriken zu entwickeln [Cardoso 06]. Vor allem an der "Technische Universität Eindhoven" arbeiten die Forscher speziell auf dieses Thema hin und sind dabei sehr erfolgreich. In Deutschland beschäftigen sich die Forscher an der Universität Leipzig ebenfalls sehr viel mit diesem Thema.

Die Metriken unterteilen sich in drei Hauptkategorien; Fehlermetriken, Komplexitätsmetriken und Qualitätsmetriken. Anhand dieser Metriken kann man herausfinden, wie fehleranfällig, komplex oder qualitativ ein Prozessmodell ist. Vor allem die Qualitätsmetriken sind beinahe komplett eine Umwandlung von Softwaremetriken in Prozessmetriken.

Da in Deutschland die BPMN die beliebteste Prozessmodellierungssprache ist, ist es natürlich sehr interessant, inwiefern diese Metriken auf BPMN einsetzbar sind.

1.1 Aufgabenstellung

Das Ziel der Bachelorarbeit ist es, vor allem die verfügbaren Metriken in der Prozessmodellierung herauszufinden und mit Details darstellen. Nebenbei spielt die Erstellung eines Modellkatalogs mit verschiedenen BPMN Modellen bei der Beurteilung von Metriken eine wichtige Rolle. Dadurch können die gefundenen Metriken, anhand des Programms, welches eines der Hauptziele für die Bachelorarbeit ist, getestet werden. Der Modellkatalog ist wichtig, da sowohl im Netz als auch in universitären Möglichkeiten, es schwer oder sogar unmöglich ist, an eine große Menge von Modellen zu kommen, die in vielen Hinsichten sehr hilfreich sein könnte. Anhand von erstellten Metrikenlisten, Modellkatalogen und programmierter Software kann es einfacher sein, in weiteren wissenschaftlichen Arbeiten direkt diese Bachelorarbeit zu nutzen.

Gliederung

Die Arbeit ist folgender Maßen gegliedert:

Kapitel 2 – Die Grundlagen der Geschäftsprozessmanagement: In diesem Kapitel wird die Geschichte von Geschäftsprozessmanagement kurz dargestellt und anschließend werden die atomaren Begriffe von Geschäftsprozessmodellierung detailliert erläutert. Das Manifest für die Geschäftsprozesse wird helfen, um dessen Ziele besser zu verstehen.

Kapitel 3 – Modellierungssprachen: Dieses Kapitel beschäftigt sich mit der Notation der meist genutzten Modellierungssprachen. Diese werden mit Beispielen dargestellt und erläutert. Da eine Vielzahl unterschiedlicher Modellierungssprachen vorhanden ist, wurden nur fünf dieser Sprachen dargestellt.

Kapitel 4 – Metriken: Das Hauptziel der Bachelorarbeit wird in diesem Kapitel erläutert. Die Metriken unterteilen sich in drei Hauptkategorien, welche jeweils mehr als zehn Metriken beinhalten. Fehlermetriken, Komplexitätsmetriken und Qualitätsmetriken werden genau mit Beispielen und Definitionen erklärt.

Kapitel 5 – Modellkatalog: Hier sind einige Beispielmodelle vom Modellkatalog zu sehen. Die XML Versionen der Modelle werden aufgezeigt.

Kapitel 6 – Programmierung der Metriken Hier werden einige Details zu der Programmierung von Metriken dargestellt. Wie wird die XML Datei analysiert ? Wie werden die Elemente aus dem XML Dokument herausgefiltert ? Usw.

Kapitel 2

Die Grundlagen der Geschäftsprozessmanagement

Probleme kann man niemals mit derselben Denkweise lösen, durch die sie entstanden sind.

(Albert Einstein)

Seit hundert Jahren ist es das Ziel der Menschheit ihre Lebensverhältnisse zu verbessern, die Arbeitsschritte und die Prozesse hierfür zu vereinfachen und diese zu kontrollieren. Jean-Paul Satre, französischer Schriftsteller und Philosoph des zwanzigsten Jahrhunderts, beschrieb den diesen Veränderungen zugrunde liegenden Fortschritt als das Werk der Unzufriedenheit. Und so wundert es den Einzelnen nicht, dass sich von den Anfängen im achtzehnten Jahrhundert durch Adam Smith über bedeutende Schlüsselfiguren des zwanzigsten Jahrhunderts – Henry Ford – bis in die heutige Zeit auch das Feld des „Business Process Management“ deutlich veränderte und weiterentwickelt hat. Inzwischen ist der Begriff „Business Process Management“ in unserem Arbeitsleben ein fester Bestandteil geworden und hat vielmehr an Bedeutung gewonnen. Bevor wir auf diese Bedeutung per se eingehen, sollten wir zuvor die Teilaspekte der Begriffe in ihrer Einzelheit betrachten.

Business: Dieser Begriff umfasst alle Organisationsarten, von der Kirche über Vereine und Firmen bis hin zu Tätigkeiten im Haushalt. All diese Organisationsstrukturen beinhalten Aufgaben, die zu erledigen sind. Der Begriff „Business“ umfasst nicht nur kaufmännische Tätigkeiten. Gibt es Tätigkeiten, die zu erledigen sind, so lässt sich die Gesamtheit dieser Aufgaben als „Business“ bezeichnen.

Prozess: Ein Prozess ist eine Entwicklung oder auch eine Vorgang, die eine bestimmte Zeit in Anspruch nimmt und etwas produziert, herausbildet. Unter diesem Begriff versteht sich jede Tätigkeit/ Aufgabe, welche durch Menschen oder Maschinen erledigt wird. *„Prozesse werden von spezifischen Ereignissen ausgelöst (Input) und haben ein oder mehrere Ergebnisse (Output), die entweder zum Abschluss des Prozesses führen oder in einen weiteren Prozess münden.“* [BPM-CBOK 09]

Management: Etymologisch ist dieser Begriff mit hoher Wahrscheinlichkeit von den lateinischen Begriffen manus agere oder mansionem agere (= an der Hand führen / das Haus (für den Eigentümer) bestellen) abzuleiten [Stahle 99] und bedeutet laut Duden sinngemäß das

Führen und Leiten eines Unternehmens. Aufgaben wie Planung, Organisation, Kontrolle und Führung fallen unter diesen Oberbegriff.

2.1 Business Process

Ein „Business Process“ beinhaltet alles, was man zur Verwaltung von Tätigkeiten braucht, um sicher die gewünschten Ergebnisse zu erzielen. Ein Geschäftsprozess kann als ein zielgerichteter zeitlicher und logischer Ablauf von Prozessen betrachtet werden [Gadatsch 04]. Bei solch einem Ablauf werden Eingaben gemacht, auf dessen Grundlagen Arbeitsergebnisse erzeugt werden können. Der Punkt dabei ist, nach P. Harmon von „Business Process Trends“, dass Kunden mit den erzielten Ergebnissen zufrieden sind. Ist dies nicht der Fall, so muss der Geschäftsprozess verändert werden. Betrachten wird folgendes Beispiel. Eine Person kauft ein Produkt von „Amazon.com“ und anstelle der bestellten Ware bekommt sie ein falsches, nicht bestelltes Produkt. Nachdem sie dies bei „Amazon“ gemeldet hat, wird der Fehler beseitigt, sie bekommt ihre gewünschte Ware und zusätzlich ein Geschenk. Genannter Ablauf stimmt sie im Endeffekt zufrieden. Somit kann man sagen, dass es sich im Falle von „Amazon“ um einen guten Geschäftsprozess handelt. R. Burlton, Gründer von „BPTrends“ entwickelte nach zwanzigjähriger Erfahrung folgendes, durch den Autor dieser Arbeit ins Deutsche übersetzte Manifest [Burlton 13] für die Geschäftspresse, welches die genauen Ziele und Absichten der Geschäftsprozesse darstellt:

Die Arbeit: Ein Geschäftsprozess beschreibt die Arbeit einer Organisation:

- ◇ Aus materiellen oder digitalen Eingaben werden die Ausgaben erzeugt.
- ◇ Ein Geschäftsprozess umfasst mehrere Aktivitäten. Diese Aktivitäten können wiederum für sich eine eigene interne Menge von Aktivitäten beinhalten.
- ◇ Ein Geschäftsprozess beschreibt die komplette übernommene Arbeit der Organisation.
- ◇ Die Erstellung des Inhaltes: Ein Geschäftsprozess erstellt Inhalte für den Kunden oder andere Interessenvertreter.
- ◇ Ein Geschäftsprozess soll messbare Inhalte für den Kunden erstellen, so dass die Erwartungen erfüllt werden können und die Kundenzufriedenheit garantiert werden kann.
- ◇ Die erzeugten Inhalte durch einen Geschäftsprozess sollen bezüglich ihrer Qualität kontrolliert werden.
- ◇ Die Inhalte sollen auch vergleichbar sein, so dass man Besseres und Schlechteres herausarbeiten kann.

Die Ressourcen: Ein Geschäftsprozess wird durch verschiedene Ressourcen durchgeführt:

- ◇ Allen Ressourcen sollen genaue Rollen und Verantwortlichkeiten zugewiesen werden.

- ◇ Die Humanressourcen sollen über praktisches Wissen verfügen, um bestmögliche Ergebnisse der Arbeit liefern zu können.

Der Kontext: Ein Geschäftsprozess existiert nur mit einem Kontext in sich:

- ◇ In einem Geschäftsprozess soll klar definiert werden zu welcher Organisation er gehört.
- ◇ Inputs können durch externe Kunden oder andere Geschäftsprozesse innerhalb der Organisation kommen.
- ◇ Ein Geschäftsprozess startet durch das Auslösen der Ereignisse oder eines Ereignisses.
- ◇ Er endet nur wenn die Endstation erreicht ist oder die betroffenen Entscheidungen zu einem frühzeitigen Ende führen.

Die Motivation: Die Ziele des Geschäftsprozesses unterstützen die Ziele des Geschäfts:

- ◇ Ein Geschäftsprozess soll mit den Zielen, Missionen, Visionen und den Leistungsstärken des Geschäfts geführt werden.
- ◇ Die Prinzipien und Werte der Organisation sollen genauso ein Bestandteil des Geschäftsprozesses sein.
- ◇ Für die Entscheidungen innerhalb eines Geschäftsprozesses sollen die Geschäftsstrategien und Geschäftsregeln beachtet werden.
- ◇ Die gewünschte Ausgabe zu erzeugen soll die größte Motivation darstellen.

Der Name: Ein idealer Geschäftsname sollte eindeutig, geschäftsfreundlich und durchgängig sein:

- ◇ Ein Geschäftsprozess sollte nur einen Namen haben.
- ◇ Der Name soll den Prozess widerspiegeln.
- ◇ Er soll das gewünschte Ziel des Geschäftsprozesses eindeutig übermitteln.
- ◇ Der Name soll klar und deutlich von den Kunden und Interessenvertretern verstanden werden.

Die Modelle: Die Geschäftsprozessmodelle ermöglichen vielfältige Perspektiven, Notationen und Diagramme.

- ◇ Ein Modell beinhaltet all die Informationen, welche in dem Geschäftsprozess enthalten sein sollen.
- ◇ Ein Geschäftsprozessmodell stellt eine verständliche Perspektive für den Geschäftsprozess her.
- ◇ Hierbei sollte eine feste Modellierungssprache gewählt werden.

2.2 Business Process Management

„Wer (Akteure) macht was (Aufgaben), wann (zeitliche Abfolge), wie (Qualität), womit (Ressourcen) und zu welchem Zweck (Unternehmensziele)?“ [Sesselmann 10] erklärt sehr gut „Business Process Management“. Jegliche Art von Geschäften kann große und komplexe Aufträge und Aufgaben erhalten, bei welchen die Realisierung des gewünschten Ergebnisses nicht einfach ist und viel Mühe verlangt. Die Arbeitsschritte werden komplexer und unübersichtlicher. In solchen Fällen sollte Schritt für Schritt gearbeitet werden, um erfolgreich voranzukommen. Daher ist in dieser Zeit das Business Process Management eine sehr gute Managementdisziplin, die dabei hilft, diese Schritte zu kontrollieren. Business Process Management – BPM ist eine Managementdisziplin, welche ganz strukturiert, die verschiedenen Aufgaben gestaltet, dokumentiert, misst, überwacht und auch steuert. Die Firmen können ihre Ziele noch einfacher und sparsamer erreichen, da die Prozessschritte vom Anfang bis zum Ende ganz klar sind. Die Prozessschritte werden für jeden ganz klar und einfach dargestellt. Aufgrund der komplexeren Organisation bei großen Firmen erzielt die BPM hier bessere Wirkungen als bei kleineren. Zum Beispiel wäre es für viele Banken eine sehr hilfreiche Strategie BPM einzusetzen. Für kleine Firmen oder Gruppierungen ist es eher nicht empfehlenswert. Man darf Business Process Management nicht mit Business Process Improvement – BPI verwechseln [Sesselmann 10], da die BPI nur ein einmaliger Prozessablauf ist, während die BPM ein ständig sich wiederholender Arbeitsprozess ist. Einige Managementdisziplinen für BPI wären SixSigma, Lean Management, Total Quality Management – TQM, oder die Business Re-Engineering [Fuermann 02]. Die große Firma Motorola hatte die Idee, im Jahre 1987 SixSigma einzusetzen. Bis zum Jahre 2000 hat die Firma auf diese Weise über 14 Milliarden Dollar gespart [Fuermann 02]. Es gibt zahlreiche andere Beispiele für Firmen, die durch bestimmte Managementtechniken vergleichbare Ersparnisse erzielen konnten. BPM hat 5 verschiedene Phasen, welche genauer betrachtet werden sollen. Diese Phasen sind; Modellierung, Analyse, Design, Diagnose und Umsetzung. Um die Vorteile von BPM ersichtlich zu machen müssen die Phasen nacheinander von Anfang bis Ende durchgezogen werden.

Business Process Management hat folgende Ziele [Sesselmann 10]:

- ◇ Die Geschäftsprozesse werden bestimmt und klar dargestellt.
- ◇ Die Prozesse werden geplant, strukturiert und gesteuert.
- ◇ Eine übersichtliche Kostenkalkulation wird durchgeführt.
- ◇ Die Prozesse werden ständig unter Kontrolle gehalten.
- ◇ Die unterschiedlichen Arbeitsschritte werden dokumentiert.
- ◇ Die Aufgaben werden mit genauen Regeln und Inputs definiert. die Rollen bei verschiedenen Prozessen werden klar definiert um eine kontinuierliche Verbesserung von Prozessen zu gewährleisten.

Wie schon erwähnt ist BPM ein dauerhafter und kontinuierlicher Arbeitsprozess, welcher durch einen BPM-Zyklus dargestellt wird. Mithilfe von BPM-Zyklen werden wiederum die BPM Arbeitsprozesse kontrolliert. Es gibt verschiedene BPM-Zyklen. Jeder BPM Zyklus beinhaltet folgende fundamentale Phasen: Analyse, Diagnose, Modellierung, Design, Umsetzung. Es ist wichtig zu wissen, dass BPM nicht unbedingt etwas mit IT zu tun haben muss. Die IT erleichtert vieles, so dass es verschiedene Softwares gibt, um solch verschiedene Prozessabläufe darzustellen. BPM wird bei den folgenden drei Fällen eingesetzt:

- ◇ Falls die existierenden Prozesse durch irgendeine Form verbessert werden sollen.
- ◇ Falls die Arbeitsschritte dokumentiert werden sollen.
- ◇ Falls neue Prozesse eingeführt werden sollen.

Nun kommen wir zu Erklärungen für die verschiedenen Phasen des Geschäftsprozessmanagements:

Modellierung: Die Aufgaben werden in der Modellierungsphase definiert, festgelegt und dementsprechend beschrieben. Somit kann einfach und schnell gesehen werden, was für die bevorstehenden Aufgaben/Prozesse benötigt wird.

Analyse: In dieser Phase wird der gegenwärtige Status der Prozesse genau betrachtet und überprüft, ob das Ziel weiterhin erreicht werden kann. Sowohl Vorteile, als auch die Nachteile der bestehenden Prozessentwicklungen werden kritisch ausgearbeitet. "Dazu werden Informationen aus der strategischen Planung, aus Prozessmodellen, Ergebnisse aus Leistungsmessungen, Daten über Umweltveränderungen, Anforderungen der Prozessbeteiligten und andere Faktoren einbezogen, um ein umfassendes Verständnis des Prozesses im Kontext der Gesamtunternehmung zu erhalten." [BPM-CBOK 09]

Design: In der Designphase wird die Reihenfolge der Prozessschritte ganz detailliert festgehalten. Wann etwas gemacht wird, und bis wann es fertig sein soll, wo es gemacht wird, unter welchen Bedingungen die Prozesse verwirklicht werden sollen, welche Techniken eingesetzt werden usw. [Moch 13]. Das Aussehen des Prozesses wird erarbeitet. Die Rollenverteilungen werden nach dieser Phase sehr übersichtlich zu erkennen sein.

Diagnose: In dieser Phase findet eine ständige Kontrolle von Prozessen statt. Die benötigten Informationen müssen zuerst erhoben werden, so dass sie schnellstmöglich geliefert werden können, um einen schnellen und reibungslosen Prozessverlauf zu gewährleisten. Falls ein Prozessschritt verbessert werden kann, so wird dies gemacht und dies beeinflusst gleichzeitig auch die Designphase.

Umsetzung: Hier werden die Prozessschritte komplett umgesetzt. Nach der Umsetzung ist dies jedoch noch nicht das Ende. Business Process Management ist ein endloser Kreislauf und setzt somit eine ständige Kontrolle und Verbesserung bestehender Prozesse voraus.

Business Process Management ist eine sehr wichtige Strategie, um Arbeitszeit zu sparen, was natürlich wirtschaftliche Vorteile mit sich bringt. Aber vor allem, da die Arbeitsprozesse immer komplexer werden und die Menschen immer mehr den Wunsch nach Verbesserung äußern, ist es absehbar, dass verschiedene Taktiken und Strategien entwickelt werden müssen, um die große Fülle an Informationen bewältigen zu können und darauf aufbauend Kundenzufriedenheit gewährleisten zu können. Der Gedanke, dass man nur sehr viel Durchhaltevermögen zeigen muss um das Ziel letztendlich zu erreichen, rückt immer weiter

in den Hintergrund. Heutzutage ist eher die Planung wichtig um schnellst- und bestmöglich seine Ziele zu erreichen.

Kapitel 3

Modellierungssprachen

Wenn ein Film Erfolg hat, ist er ein Geschäft. Wenn er keinen Erfolg hat, ist er Kunst.

(Carlo Ponti)

Heutzutage gibt es viele Modellierungssprachen, um verschiedene Prozesse zu beschreiben. Die Modellierungssprachen können für ganz allgemeine Sachverhalte im Alltag, wie die Schachzüge oder Lernprozesse für die Prüfungen, genutzt werden. Nun können Modellierungssprachen auch verwendet werden, um die Geschäftsprozesse im Unternehmen zu beschreiben. Hierbei werden die Abläufe des Prozesses modelliert. Damit die Unternehmen wettbewerbsfähig bleiben können, werden diese modellierten Abläufe ständig kontrolliert, um Verbesserungen im Prozess zu erreichen. Es gibt mehrere Modellierungssprachen, um die Geschäftsprozesse graphisch darzustellen.

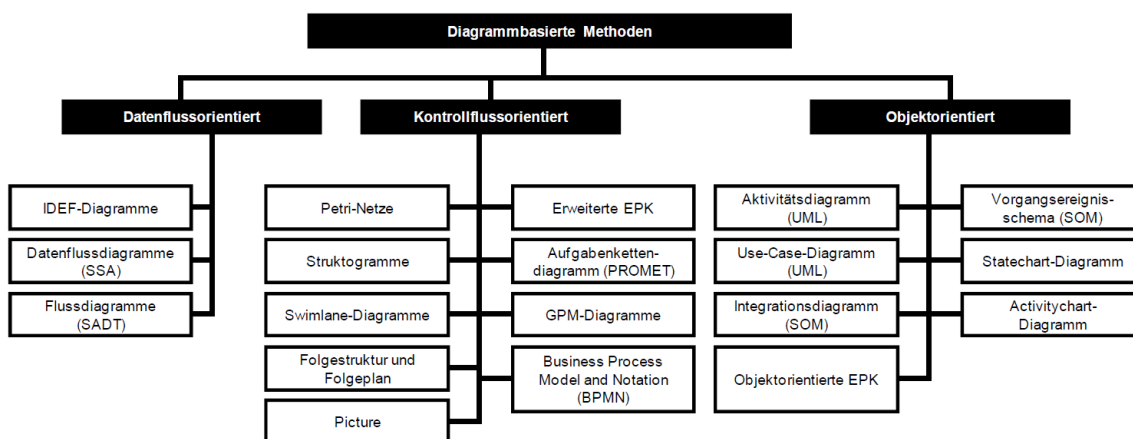


Abbildung 3.1: Diagrammbasierte Methoden [Gadatsch 04]

in Abbildung 3.1 sind die diagrammbasierten Methoden von Modellierungssprachen zu sehen. In diesem Kapitel werden einige Modellierungssprachen vorgestellt, darunter Petri-Netze, Ereignisgesteuerte Prozesskette (EPK), Unified Modeling Language (UML) und Business Process Management Notation (BPMN). Diese vier Modellierungssprachen verwenden die meisten Nutzer. Das ist der Grund, warum genau diese vier Modellierungssprachen hier beschrieben werden. Der Fokus wird jedoch auf BPMN liegen. BPMN ist die meist genutzte Modellierungssprache nach einer Deutschlandweiten Untersuchung von Mitarbeitern der “Camunda - The Business Process Company” [Francko8].

3.1 Petri-Netze

Petri-Netze wurden durch Carl Adam Petri im Jahre 1962 in seiner Dissertation mit dem Titel „Kommunikation mit Automaten“ [Petri 62] an der Technischen Universität Darmstadt eingeführt. Die Petri-Netze gelten als eine Erweiterung von Automaten. Dadurch werden sequentielle und nebenläufige Abläufe von Geschäftsprozessen beschrieben. Ein Vorteil von Petri-Netzen ist, dass trotz der wenig verwendeten Elemente in der graphischen Darstellung, eine große Menge von Informationen beschrieben werden können [Chen 94].

Ein Petri-Netz besteht aus Stellen und Transitionen. Hierzu gibt es zusätzlich noch Marken, welche sich in Stellen befinden und sich von einer Stelle zur anderen Stelle bewegen können. Die Stellen werden als Kreise dargestellt und verfügen über eine Kapazität,

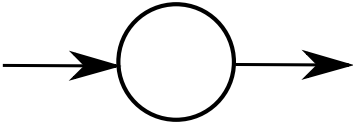
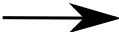
Graphiksymbol	Bezeichnung	Interpretation
	Stelle, Platz, Bedingung	Situation, Zustand, Aktivität (steuerungstech. Interpretation)
	Transition, Ereignis	(Zustands-) Übergang
	Kante, Bogen	Zusammenhang zwischen Situation und Übergang
	Marke, Token	Markierung der aktiven Situation

Abbildung 3.2: Elemente in Petri-Netzen [Jochim 01]

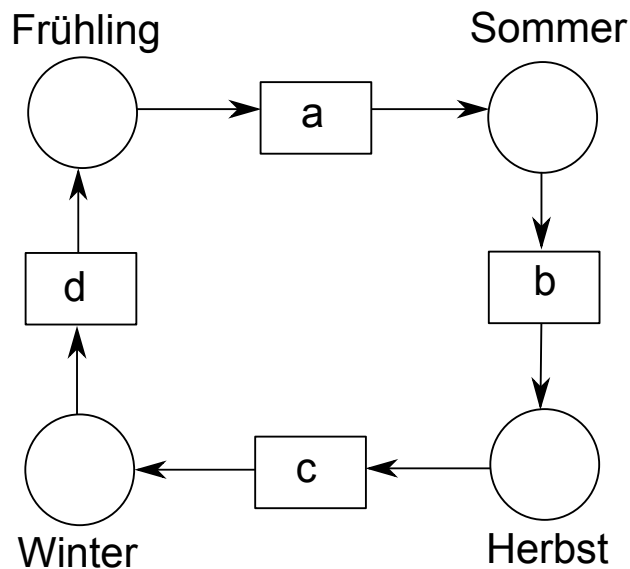


Abbildung 3.3: Die Jahreszeiten als Petri-Netz Beispiel [Wikimedia 11]

so dass die Anzahl der Marken diese Kapazitätsgrenze nicht überschreiten kann. Die Belegungen der Stellen zeigen den aktuellen Zustand des Petri-Netzes. Ist die Kapazität der Stellen nicht angegeben, so beträgt die Kapazität stets den Wert eins. Eine Stelle darf nur mit einer Transition verbunden werden, welche durch gerichtete Kanten zustande kommt.

Die Transitionen veranschaulichen die Aktivitäten, bzw. Ereignisse eines Prozesses, und werden als Rechtecke dargestellt. Die Marken bewegen sich zu nächstmöglichen Stellen, nach dem die jeweiligen Transitionen geschaltet werden. Eine Transition kann erst dann geschaltet werden, wenn sich in der Stelle davor eine Marke befindet. In der Abbildung 3.3 befindet sich die Marke in der Stelle Frühling. Nur die Transition „a“ darf geschaltet werden, worauf die Marke sich von Frühling zu Sommer bewegt. Ab diesem Zeitpunkt kann die Transition „b“ geschaltet werden.

Petri-Netze existieren seit über 50 Jahren und gehören somit zu den ältesten Systemmodellen der Informatik. „Petri-Netze sind nie, wie einige andere Methoden, eine Weile bevorzugt und dann wieder vergessen worden, sondern haben über Jahrzehnte ihren Platz in der Methodenvielfalt behalten.“ [Reisig 10]

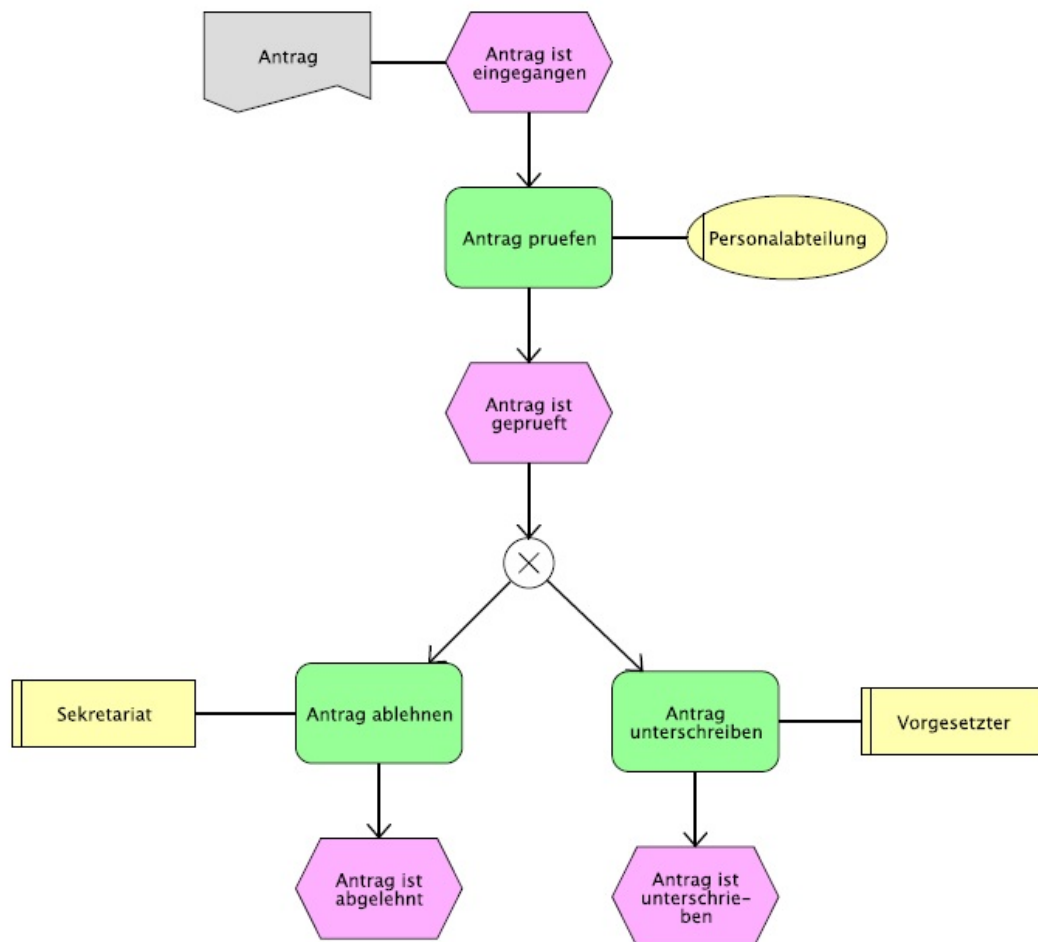


Abbildung 3.4: Die Überprüfung eines Antrages als EPK-Model [Schmitzl 10]

3.2 EPK

Zur Modellierung von Geschäftsprozessen wurde im Jahre 1992 die Notation der Ereignisgesteuerten Prozesskette - EPK entwickelt. Die Entwicklung der Notation fand an der Universität des Saarlandes unter Leitung von August-Wilhelm Scheer in Zusammenarbeit mit SAP AG statt. EPK hat sich wegen der anschaulichen Repräsentation schnell verbreitet und war lange Zeit die am häufigsten genutzte Modellierungssprache [Schmitzl 10]. In der Abbildung 3.4 ist ein einfaches Beispiel mit EPK dargestellt, welches einen Antragprüfungsprozess zeigt. Eine EPK besteht aus Funktionen, Ereignissen und Konnektoren. Die Zustände werden als Ereignisse abgebildet und die zu erledigenden Aufgaben werden als Funktionen abgebildet. Es wird immer mit einem Ereignis angefangen und mit einem Ereignis beendet. Es kommt auch vor, dass ein Modell mehrere Start- und Endereignisse hat.

Es findet immer eine Abwechslung zwischen Funktionen und Ereignissen statt, d.h. die Funktionen werden durch Ereignisse ausgelöst. Um diese Elemente zusammenzuführen oder zu spalten, existieren Konnektoren (UND, ODER, XOR). Darüber hinaus gibt es weitere Elemente, welche in Abbildung 3.5 erklärt werden.

Begriff	Symbol	Definition
Funktion		Beschreibt die Transformation von einem Eingangszustand in einen Zielzustand
Ereignis		Beschreibt das Eintreten eines Zustandes, der eine Funktion auslöst oder das Ergebnis einer Funktion ist.
Prozessschnittstelle		Beschreibt die Verknüpfung zu vor- oder nachgelagerten Geschäftsprozessen.
Verknüpfung		Beschreiben die logische Verknüpfung von Ereignissen und Funktionen.
Org. Einheit		Beschreibt die Gliederungsstruktur eines Unternehmens.
Informationsobjekt		Beschreibt die datentechnische Abbildung der realen Welt.
Belege		Beschreibt die Eingangs- und Ausgangsbelege zu oder von einer Funktion.
Funktionsfluß		Beschreibt, ob von einer Funktion geändert, gelesen oder geschrieben wird.
Zuordnung		Beschreibt die Zuordnung einer Organisationseinheit zu einer Funktion.

Abbildung 3.5: EPK Symbole [Wikimedia 06]

EPK ist zwar ausreichend für eine übersichtliche Modellierung, aber je komplexer die Sachverhalte sind, desto schwieriger ist es, mit EPK zu modellieren. Aus diesem Grund ist EPK für komplexere Geschäftsprozesse nicht ratsam. Für mehr Informationen wird auf die zahlreichen Literaturen wie [Scheer 01] und [Gadatsch 04] verwiesen. Ein detailliertes Beispiel für EPK ist in Abbildung 3.6 zu sehen.

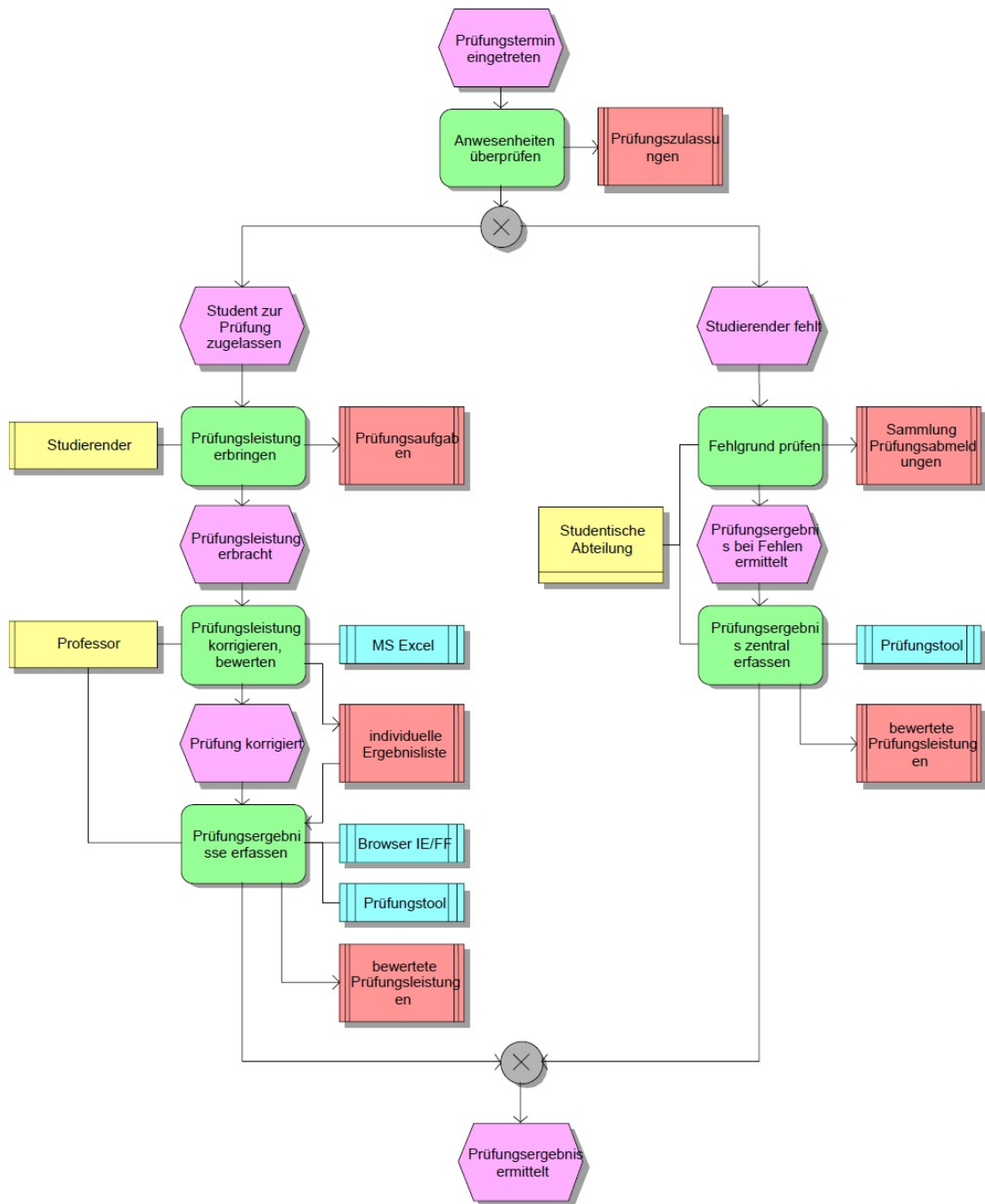


Abbildung 3.6: Der Prüfungsprozess als EPK Beispiel [Decker 09]

3.3 UML - Aktivitätsdiagramm

Aktivitätsdiagramm ist einer der 13 Diagrammtypen von UML. UML – Unified Modeling Language wurde durch OMG – Object Management Group entwickelt. UML ist ein Standard von OMG und beinhaltet 13 verschiedene Diagrammtypen, wie Klassendiagramme, Sequenzdiagramme usw. Eines davon ist das Aktivitätsdiagramm, welche die Abläufe von Prozessen beschreibt. Es wird gezeigt, in welcher Reihenfolge die Aktionen ausgeführt werden. Das Aktivitätsdiagramm gehört zur Kategorie Verhaltensdiagramme innerhalb der UML [Schmitzl 10].

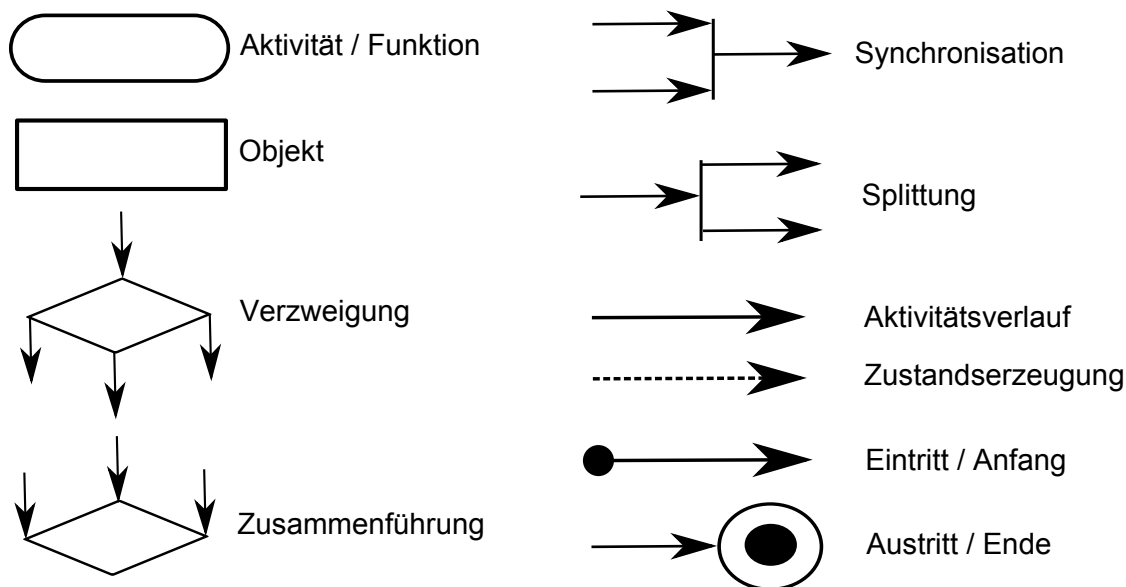


Abbildung 3.7: Die Notation des Aktivitätsdiagramms [Dumke 01]

Ein Aktivitätsdiagramm kann einen oder mehrere Start- und Endpunkte haben. Dazwischen befinden sich die Kanten und Knoten, wodurch die Aktivitäten dargestellt werden. Unten ist eine Liste der Elemente zu sehen, welche in Aktivitätsdiagrammen vorkommen.

Eintritt/Anfang(Startknoten): Ein möglicher Eintrittspunkt in das System. Es können mehrere Startpunkte im Diagramm enthalten sein.

Austritt/Ende(Endknoten): Ein möglicher Austrittspunkt aus dem System. Es können mehrere Endpunkte im Diagramm enthalten sein.

Aktivität/Funktion: Die Aktivitäten stellen die einzelnen Schritte dar und lassen sich nicht mehr unterteilen.

Objekt: Objekte symbolisieren Datenspeicher. Hierdurch werden Daten transportiert.

Verzweigung: Unter Befolgung von Regeln wird entschieden, welcher weitere Weg im

Diagramm genommen werden soll.

Zusammenführung: Dadurch werden die getrennten Wege wieder zu einem Pfad vereint.

Splittung: Der aktuelle Vorgang wird in parallel ablaufende Vorgänge aufgeteilt.

Synchronisation: Parallel verlaufende Vorgänge werden zusammengeführt.

Nachfolgend ist ein Beispiel für einen Aktivitätsdiagramm zu sehen.

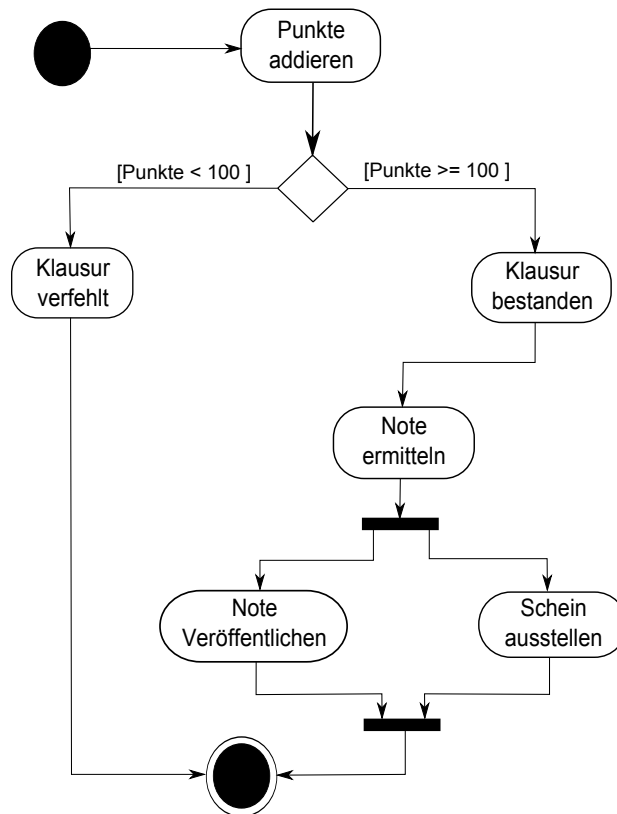


Abbildung 3.8: Beispiel für ein UML-Aktivitätsdiagramm [[Wikipedia 05](#)]

Die Punkte bei einer Klausur werden addiert. Hat man weniger Punkte als 100, so hat man die Klausur verfehlt. Hat man 100 oder mehr Punkte erreicht, so hat man die Klausur bestanden. Während der Notenermittlung wird die Note veröffentlicht und der Schein wird ausgestellt. Erst wenn beide dieser Aufgaben erledigt sind, kommt man aus dem System heraus.

3.4 BPMN

BPMN – Business Process Model and Notation ist eine standardisierte Modellierungssprache für die Geschäftsprozessmodelle. Die BPMN wurde unter der Leitung von S. A. White in der Firma IBM entwickelt und im Jahre 2004 veröffentlicht [White 04]. Das Ziel war, eine Notation zu entwickeln, die sehr verständlich für alle Nutzer sein soll. S. A. White beschreibt das Ziel der BPMN wie folgendes:

“The primary goal of the BPMN effort was to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. BPMN will also be supported with an internal model that will enable the generation of executable BPEL4WS. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation.” (S. White, 2004: Seite 1 [White 04])

Bei der Entwicklung von BPMN wurde versucht, dass die besten Ideen aus verschiedenen Notationen unter einer einzigen Notation zusammengebracht werden. Für die Weiterentwicklung wurde die BPMN durch OMG – Object Management Group übernommen. BPMN 1.0 wurde offiziell im Jahre 2006 und ab 2011 wurde die neueste Version, nämlich BPMN 2.0 veröffentlicht [OMG 10]. BPMN hat sich so verbreitet, dass diese Modellierungssprache sehr beliebt geworden ist. Laut einer Umfrage in Deutschland wird die BPMN immer beliebter und ist in jetziger Zeit schon die häufigste genutzte Notation für die Prozessdokumentationen [Franco8].

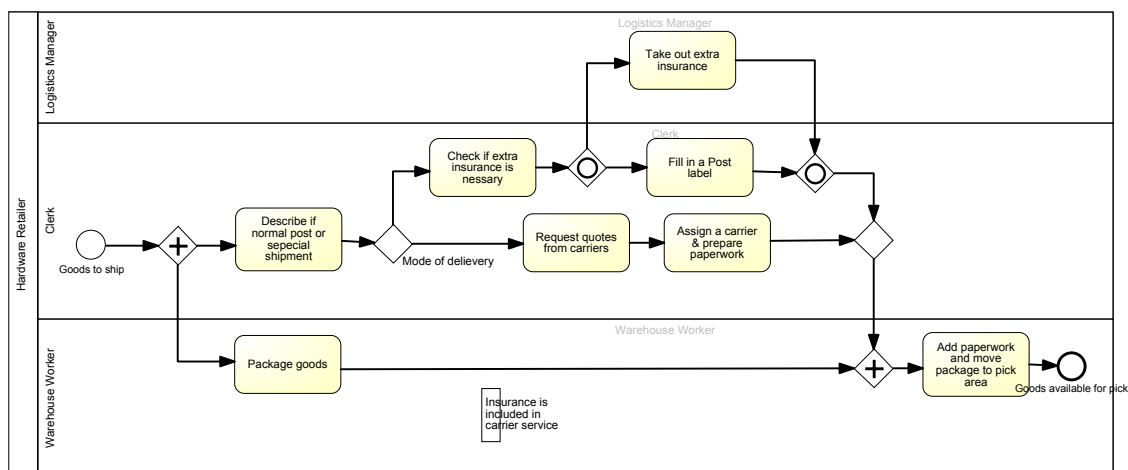


Abbildung 3.9: Versandprozess eines Hardware Teils [EAP 12]

Bei der BPMN 2.0 sind neue Symbole dazugekommen und somit ist diese Modellierungssprache ausdrucksstärker geworden. Es wurde so entwickelt, dass das BPMN Modell ausgeführt und semantisch genau herausgegeben werden kann, wie die Aktivitäten voneinander abhängen. Die BPMN 2.0 ist auf XML (eXtensible Markup Language) basiert [OMG 10]. Dies erleichtert den Transfer und die Bearbeitung von BPMN Modellen. Die Spezifikation der BPMN 2.0 ist ein 538 seitiges Dokument, welches kostenlos von der Homepage von OMG heruntergeladen werden darf [OMG 10].

Ein BPMN Prozessmodell besteht aus Aktivitäten, welche durch Kontrollknoten geregelt werden, in welcher Art und Weise sie miteinander verbunden sind. Die Kontrollknoten und Aktivitäten werden mittels Sequenz- und Nachrichtenflüssen miteinander verbunden.

In der Abbildung 3.9 ist ein in englischer Sprache verfasstes BPMN Diagramm zu sehen. Es geht um einen Versandprozess. Im folgenden Abschnitt werden die jeweiligen Symbole im BPMN detailliert erklärt.

3.4.1 Basiselemente

In diesem Abschnitt werden die grundlegenden Symbole von BPMN mit Hilfe von Beispielen erklärt. Es gibt 4 Hauptkategorien für die grafischen Elemente in BPMN [White 04]. Flow Objects, Connecting Objects, Swimlanes und Artifacts.

3.4.1.1 Flow Objects

Es gibt 3 Arten von Flow Objects, nämlich Events (Ereignisse), Activities (Aktivitäten) und Gateways (Entscheidungspunkte).

- ◇ **Event (Ereignisse):** Die Ereignisse werden durch einen Kreis dargestellt und haben entweder einen Auslöser oder ein Ergebnis. In anderen Worten, Ereignisse markieren die Zeitpunkte im Modell. Es gibt 3 verschiedene Ereignisse, Start-, Zwischen-



Abbildung 3.10: Hauptereignisarten

(Intermediate) und Endereignisse. Ein Startereignis wird als ein Auslöser eines Prozesses bezeichnet. Wer ein Startereignis genutzt hat, muss auch ein Endereignis nutzen [Kocian 11]. Die Zwischenereignisse treten innerhalb des Modells ein und zeigen die Zustandsänderungen im Prozess an. Ein Endereignis beendet das Prozessmodell

und erzeugt somit ein Ergebnis. Ab BPMN 2.0 ist es erlaubt ein Prozessmodell ohne Ereignisse darzustellen [[Kocian 11](#)] .

	Start			Zwischen			Ende	
	Standard	Ereignis-Teilprozess Unterbrechend	Ereignis-Teilprozess Nicht-unterbrechend	Eingetreten	Angeheftet unterbrechend	Angeheftet Nicht-unterbrechend	Ausgelöst	Standard
Blanko: Untypisierte Ereignisse, i. d. R. am Start oder Ende eines Prozesses.								
Nachricht: Empfang und Versand von Nachrichten.								
Timer: Periodische zeitliche Ereignisse, Zeitpunkte oder Zeitspannen.								
Eskalation: Meldung an den nächsthöheren Verantwortlichen.								
Bedingung: Reaktion auf veränderte Bedingungen und Bezug auf Geschäftsregeln.								
Link: Zwei zusammengehörige Link-Ereignisse repräsentieren einen Sequenzfluss.								
Fehler: Auslösen und behandeln von definierten Fehlern.								
Abbruch: Reaktion auf abgebrochene Transaktionen oder Auslösen von Abbrüchen.								
Kompensation: Behandeln oder Auslösen einer Kompensation								
Signal: Signal über mehrere Prozesse. Auf ein Signal kann mehrfach reagiert werden.								
Mehrfach: Eintreten eines von mehreren Ereignissen. Auslösen aller Ereignisse.								
Mehrfach/Parallel: Eintreten aller Ereignisse.								
Terminierung: Löst die sofortige Beendigung des Prozesses aus.								

Abbildung 3.11: Die Event Typen in BPMN [BPMO 11]

Es gibt verschiedene Typen von Ereignissen, welche in der Abbildung 3.11 dargestellt sind. Die Abbildung 3.11 ist ein Ausschnitt vom BPMN Poster [BPMO 11].

- ◇ **Activities (Aktivitäten):** Die Aktivitäten bezeichnen Tätigkeiten, Aufgaben in Prozessmodellen. Sie werden als Vierecke bezeichnet. Es gibt 2 Typen von Aktivitäten, nämlich Tasks (Aufgaben) und Subprocesses (Teilprozesse).

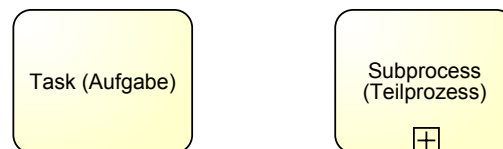
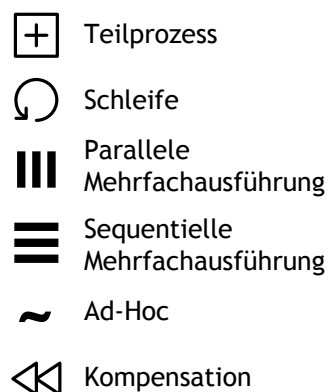


Abbildung 3.12: Aktivitäten Typen in BPMN

Die Aufgaben sind atomare Arbeitseinheiten und die Teilprozesse sind zusammengesetzte Aktivitäten innerhalb eines Prozessmodells. Die Aktivitäten können mit Markierungen, wie „plus“ Zeichen bei dem Teilprozess, und mit Aufgaben-Typen bezeichnet werden.

Markierungen

Sie beschreiben das Ausführungsverhalten von Aktivitäten:



Aufgaben-Typen

Sie beschreiben den Charakter einer Aufgabe:



Abbildung 3.13: Aktivitäten Markierungen und Aufgaben-Typen [BPMO 11]

- ◇ **Gateways (Entscheidungspunkte)** : Mit Entscheidungspunkten werden Verzweigungen (Splits) und Zusammenführungen (Joins) von Sequenzflüssen dargestellt [Mebrahtu 12]. Die Logik des Prozessmodelles wird anhand von Entscheidungspunkte bestimmt. Es gibt verschiedene Typen von Gateways.

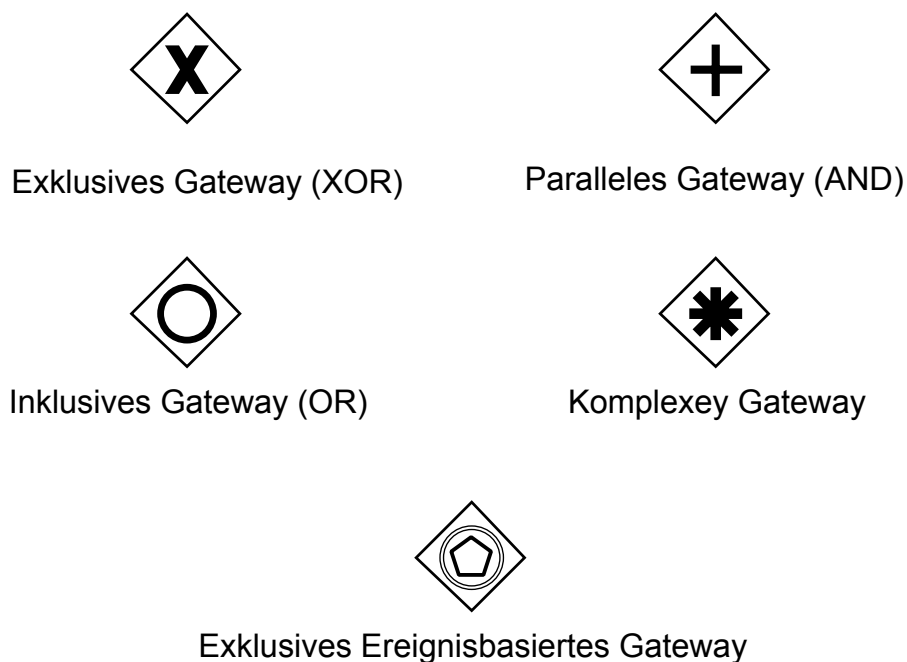


Abbildung 3.14: Gateways in BPMN (erstellt mit Signavio BPMN Editor)¹

¹ **Exklusives Gateway** gilt als XOR-Verknüpfung und genau nur eines der möglichen verbundenen Aktivitäten wird ausgeführt. Bei der Zusammenführung wird auf eines der eingehenden Kanten gewartet. Ein Raute Zeichen ohne „X“ in sich gilt genauso als ein Exklusives Gateway.

Paralleles Gateway ist eine AND-Verknüpfung. Alle ausgehende Kanten werden ausgeführt und dies führt zu einer parallelen Ausführung im Prozessmodell. Bei der Zusammenführung muss auf alle eingehende Kanten gewartet werden, damit es weiterausgeführt werden kann.

Inklusives Gateway ist eine OR-Verknüpfung. Je nach dem, wie die Bedingung ist, kann eine oder mehrere ausgehenden Kanten ausgeführt werden.

Komplexes Gateway ist eine neue logische Verknüpfung. Der Nutzer kann selber definieren, wie das Verzweigungs- oder Vereinigungsverhalten sein soll.

¹<http://academic.signavio.com/>

Exklusives Ereignisbasiertes Gateway beruht auf den Aktivitäten danach. Falls eines der nachfolgenden Ereignisse ausgeführt wird, wird der Prozess gestartet.

3.4.1.2 Connecting Objects

Die Connecting Objects sind dafür zuständig, dass die Flow Object miteinander verbunden werden. Es gibt 3 Arten von Connecting Objects: Sequence Flow, Message Flow und Association [White 04].

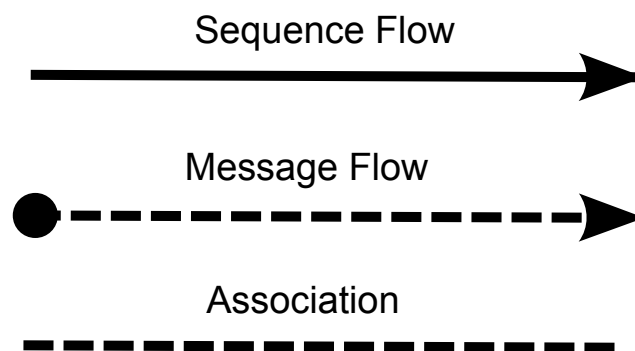


Abbildung 3.15: Die Art der Verbindungen in BPMN

Sequence Flow verbindet die Elemente der Flow Objects miteinander.

Message Flow ist für die Nachrichtenfluss zwischen den Pools im BPMN Diagramm zuständig. Mit Pools sind verschiedene Prozessteilnehmer gemeint.

Association verbindet die Datenobjekte, Artefakte mit den Flow Objects. Somit werden die Inputs und Outputs im Prozessmodell verständlicher dargestellt.

3.4.1.3 Swimlanes

Anhand von Swimlanes kann man die Aktivitäten in einem Prozessmodell besser organisieren und verständlicher darstellen. Dadurch können mehrere Rollen untereinander kommunizieren und die Verantwortlichkeiten lassen sich einfacher abbilden. Hier unterscheidet man zwischen zwei Typen, Pools und Lanes.



Abbildung 3.16: Darstellung eines Pools mit 2 Lanes

Pool: Die Pools werden eingesetzt, falls mehrere Akteure in einem Prozessmodell vorhanden sind. So stellt ein Pool, ein Akteur dar. Der Nachrichtenaustausch findet innerhalb der Pools mit Sequence Flow statt, während der Informationsaustausch zwischen den Pools mit Message Flows stattfindet. Die Pools können als "Black Box" oder als "White Box" dargestellt werden. In der "Black Box" bleiben die Aktivitäten verborgen und sind nicht zu sehen, während im "White Box" die Aktivitäten sichtbar sind [OMG 10].

Lane: Mit Hilfe von Lanes können die Pools unterteilt werden, so dass die Aktivitäten innerhalb eines Pools wieder noch genauer organisiert werden können.

3.4.1.4 Artefacts

Anhand von Artefakten können zusätzliche Informationen und Aspekte im Modell hinzugefügt werden. Hierzu gibt es drei verschiedene Arten von Artefakten. Datenobjekte (Data Object), Textanmerkungen (Text Annotation) und Gruppen (Group).



Abbildung 3.17: 3 Arten von Artefakten in BPMN

Datenobjekte (Data Object): Diese sind Daten, die durch Aktivitäten als Input oder Output hervorgerufen werden können. Nehmen wir an, eine Aktivität braucht ein Input für die

Weiterverarbeitung, so wird dies anhand von Datenobjekten dargestellt. Anschließend gibt diese Aktivität womöglich ein Datenobjekt als Output aus.

Textanmerkungen (Text Annotation): Diese sind nur hilfreiche Anmerkungen für den Leser des Modells. Hierdurch wird die Struktur des Modells nicht verändert.

Gruppen (Group): Die Gruppen sind nur für eine klare Darstellung der Modellelemente gedacht. Mehrere Elemente werden zusammen in einem Viereck gezeigt, um den Zusammenhalt optisch hervorzuheben.

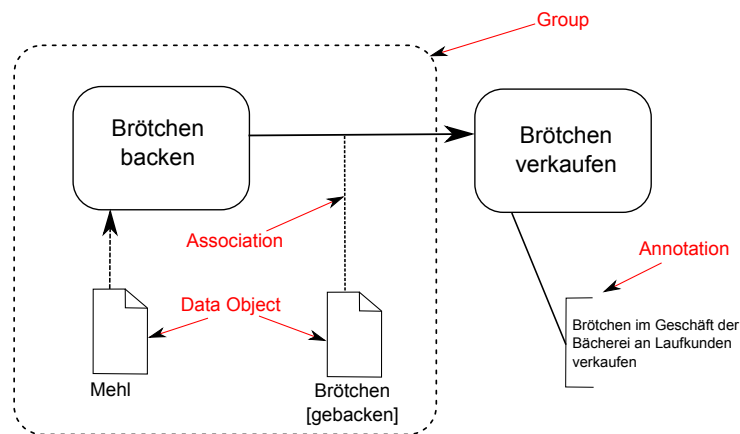


Abbildung 3.18: Ein Beispiel mit Artefakten in BPMN [Wikimedia o6]

Kapitel 4

Metriken

You cannot control what you cannot measure.

(De Marco)

„*You cannot control what you cannot measure*“, sagte DeMarco. In diesem Sinne braucht man die Metriken, um die Modelle zu analysieren und miteinander vergleichen zu können. Messungen zu einem Prozess bedeuten, dass man Zahlen und Symbole zu einer Eigenschaft von Entitäten in die Realität umwandelt, um die Anzahl oder den Grad der Eigenschaften der Entitäten zu zeigen. Die Metriken dienen dazu das Verständnis über die Prozesse zu erhöhen, die Kontrolle über die Prozesse zu haben und die Verbesserungsmöglichkeiten besser zu erkennen. In der Geschäftsprozessmodellierung ist es wichtig, dass man die passenden Messungen macht, denn nur die passenden Messungen führen zu angestrebten Verbesserungen der Prozesse. „*What to measure will be improved.*“ - Hawthorne Effect. Wir wollen Metriken über die Geschäftsprozessmodelle finden und deshalb sind in unserem Fall die Metriken für Netzwerkanalyse bzw. Graphenanalyse sehr wichtig. Die Modelle sollen als Graphennetze dargestellt werden, worüber wir unsere Analyse mittels vordefinierten Metriken machen können.

All die gefundene Metriken können bei alle Modellierungssprachen, welche als Graphennetzwerk dargestellt werden kann, eingesetzt werden.

4.1 Fehlermetriken

Große und komplexe Modelle bereiten den Nutzern oft Schwierigkeiten. Es kommt oft vor, dass man bei großen Modellen einige Fehler findet. Die Anzahl der Fehler hängt von bestimmten Faktoren ab. Das Ziel der Fehlermetriken ist die Fehlerwahrscheinlichkeit bei den Geschäftsprozessmodellen zu messen. Einige Bestimmungsfaktoren für die Geschäftsprozessmodelle sind die in diesem Kapitel erwähnten Metriken. J. Mendling hat in seiner Dissertation mit dem Titel „Detection and Prediction of Errors in EPC Business Process Models“ [[Mendling 07](#)] das Thema Fehlermetriken sehr gründlich behandelt und bietet in diesem Feld eine große Hilfestellung für die Prozessexperten. Neben J. Mendling beschäftigten sich G. Neumann und W.van der Aalst auch mit diesem Thema. Die Zusammenarbeit zwischen den drei Wissenschaftler hat einige Publikationen über die Fehlermetriken mit sich gebracht. Während in der ersten Publikation [[Mendling 06](#)]

von J. Mendling und G. Neumann über die Fehlermetriken, noch bevor die Dissertation von J. Mendling veröffentlicht wurde, die ersten sechs Fehlermetriken bekanntgegeben wurden, wurde das Thema in der Dissertation von J. Mendling sehr ausführlich analysiert. Anschließend haben die genannten Wissenschaftler die Publikation mit den Messungen über die Fehlermetriken [Mendling+ 07] veröffentlicht.

In diesem Kapitel wurden nur EPK Modelle als Beispiel genutzt, da in den Literaturen über die Fehlermetriken ausschließlich solche verwendet wurden.

Für die Fehlermetriken definieren wir das Modell als $\text{Graph}=(N,A)$, wobei N als Knoten mit der Eigenschaft $N = T \cup S \cup J$, und A als Kante mit der Eigenschaft $A \subseteq N \times N$ gilt. Aufgaben/Tasks T , Split-Verknüpfungen S und Join-Verknüpfungen J bestimmen die Zahl der N . $C = S \cup J$ wird als Verknüpfungssymbol genutzt.

4.1.1 Metrik: Größe des Modells (engl. modelsize)

Die Größe des Modells $S(\text{Graph})$ beeinflusst die Verständlichkeit des Modells in sehr großem Maße. So gibt es Größenmetriken, um die Größe des Modells zu messen. In den Modellen ist dies die Anzahl der Knoten im Prozess Modell Graph.

Formel:

$$S(\text{Graph}) = | \text{Knoten} |$$

Korrelation: Eine Steigerung der Größe in Modellen verursacht eine höhere Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling 06] [Mendling+ 07]

Beispiel: In der Abbildung 4.2 beträgt der Wert des Metriks sowohl im linken als auch im rechten Modell 10.

In BPMN entspricht dies der Anzahl an Aktivitäten. Nach J. Mendling ist es unwahrscheinlich, dass das Modell Fehler enthält, falls das Modell sequentiell und ohne Verknüpfungen ist [Mendling 07].

4.1.2 Metrik: Trennbarkeit (engl. separability)

Diese Metrik beruht auf der Trennbarkeit des Graphs, bezüglich des „Cut-Vertex Punkt“ Algorithmus. Cut-Vertex Punkt ist ein Schnittknoten in einem Graph, dessen Entfernung im Graph zu einem multiplen Graphen führt. Das bedeutet, dass der Graph aus mehreren Graphen bestehen kann, nachdem der Schnittknoten entfernt wurde.

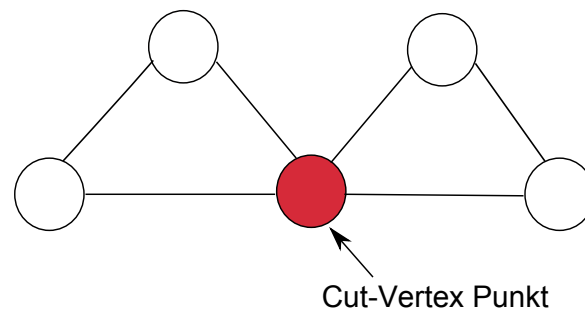


Abbildung 4.1: Beispiel eines Cut-Vertex Punktes

Cut-Vertex kann durch einen Algorithmus gefunden werden, welcher mit Hilfe von Tiefensuche funktioniert [Mendling 07]. Zuerst werden mittels Tiefensuche die zusammenhängende Teilgraphen gezählt, bevor ein Knoten entfernt wird. Nachdem ein Knoten entfernt wurde, findet die Tiefensuche, um die zusammenhängende Teilgraphen zu finden, nochmal statt. Ist die Anzahl der zusammenhängenden Teilgraphen höher, als die Anzahl vor der Entfernung des Knoten, so ist der entfernte Knoten ein Cut-Vertex Punkt [Cormen 09].

Formel:

$$\Pi(\text{Graph}) = \frac{|\{n \in \text{Knoten} \mid n \text{ ist Cut-Vertex}\}|}{|\text{Knoten}| - 2}$$

Korrelation: Eine hohe Steigerung des Trennbarkeitsverhältnisses $\Pi(\text{Graph})$ des Graphs führt zu einer Senkung der Fehlerwahrscheinlichkeit in einem Modell. Je höher die Anzahl der Gelenkpunkte, desto geringer ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling 06] [Mendling+ 07]

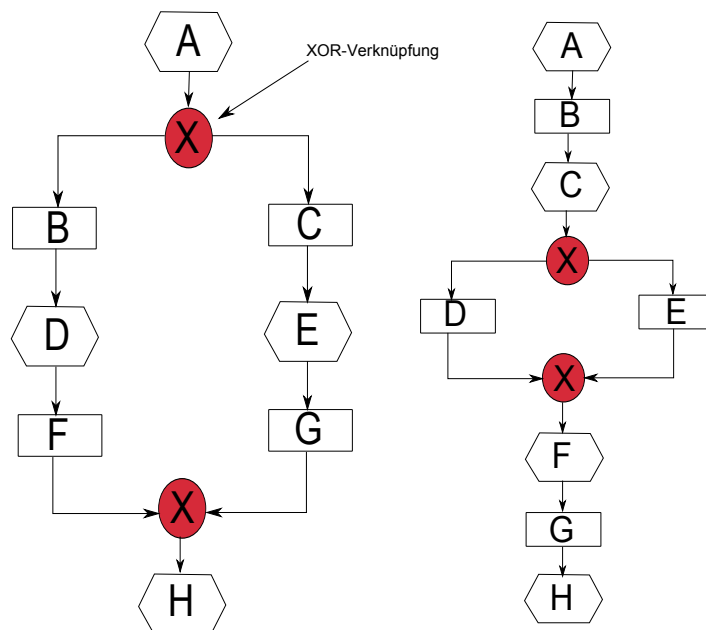


Abbildung 4.2: Zwei verschiedene EPK Modelle mit unterschiedlichen Werten der Trennbarkeitsmetrik [Mendling 07]

Beispiel: In der Abbildung 4.2 sind 2 verschiedene EPK Modelle zu sehen. Im Modell links sind 2 Cut-Vertex Knoten zu sehen, nämlich beide XOR-Verknüpfungen. Im Modell rechts sind 6 Cut-Vertex Knoten zu sehen, nämlich B, C, F, G und die beiden XOR-Verknüpfungen. Berechnet man die Trennbarkeitsmetrik, so stellt sich heraus, welches der beiden Modelle fehleranfälliger ist;

$$\prod(Graph_{links}) = \frac{2}{8} = 0,25$$

$$\prod(Graph_{rechts}) = \frac{6}{8} = 0,75$$

Im Modell rechts ist die Fehlerwahrscheinlichkeit niedriger als im Modell links.

4.1.3 Metrik: Sequenzialität (engl. sequentiality)

Diese Metrik beruht auf den aufeinanderfolgenden Aufgaben in Modellen. Die aufeinanderfolgenden Aufgaben sind die einfachsten Blöcke in Geschäftsprozessmodellen.

Formel:

$$\omega(Graph) = \frac{|\{a \in Kanten \mid a \in T \times T\}|}{|Kanten|}$$

Korrelation: Je höher die Sequenzialitätsmetrik $\omega(\text{Graph})$ ist, desto kleiner ist die Fehlerwahrscheinlichkeit. Demnach verursacht eine Steigerung der Sequenzialität eine Senkung der Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling 06] [Mendling+ 07]

Beispiel: In der Abbildung 4.2 haben wir sowohl im linken als auch im rechten Modell 2 sequentielle Blöcke. Im linken Modell sind diese von B zu F und von C zu G. Im rechten Modell sind diese von A zu C und von F zu H.

$$\omega(\text{Graph}_{\text{links}}) = \frac{4}{10} = 0,4 \qquad \omega(\text{Graph}_{\text{rechts}}) = \frac{4}{10} = 0,4$$

So sind beide Modelle gleichermaßen anfällig, was die Fehlerwahrscheinlichkeit betrifft.

4.1.4 Metrik: Strukturiertheit (engl. structuredness)

Das Ziel dieser Metrik ist es zu überprüfen wie weit ein Geschäftsprozessmodell mit Verschachtelungen (nesting blocks) erstellt werden kann. Der Grad der Strukturiertheit $\phi(\text{Graph})$ kann durch einen Vergleich vom reduzierten Graphen $S_N(\text{Graph}')$ zum ursprünglichen Graphen festgestellt werden.

Formel:

$$\phi(\text{Graph}) = 1 - \frac{S_N(\text{Graph}')}{S_N(\text{Graph})}$$

Korrelation: Eine Steigerung in Strukturiertheit führt zu einer Senkung der Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling 06] [Mendling+ 07]

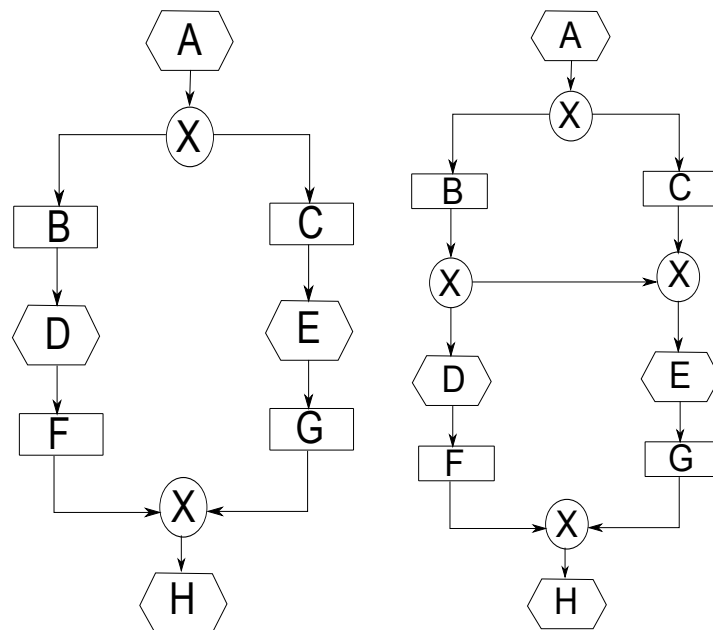


Abbildung 4.3: Zwei EPK Modelle mit unterschiedlichen Werten der Strukturiertheitsmetrik [Mendling 07]

Beispiel: Das linke EPK Modell in der Abbildung 4.3 ist total strukturiert und hat den Wert 1 für die Strukturiertheitsmetrik. Beim rechten EPK Modell gibt es eine zusätzliche Kante und zwei XOR-Verknüpfungen mehr. Dies beeinflusst den Wert der Strukturiertheitsmetrik. Nach den Reduzierungsregeln [Sadiq 99] können die Knoten von B zu G entfernt werden. Die Verknüpfungen dürfen nicht entfernt werden. Dementsprechend werden 6 von 12 Knoten entfernt. So beträgt der Wert 0,5 für die Strukturiertheitsmetrik vom rechten EPK Modell.

4.1.5 Metrik: Zyklizität (engl. cyclicity)

Zyklische Teile eines Modells sind schwieriger zu verstehen als die sequentiellen Teile. Bei dieser Metrik geht es um die Anzahl der Zyklen. Ein Zyklus in einem Graph bedeutet, dass der Start- und Endknoten gleich sind [Diestel 10]. Der Graph startet an dem Startknoten und das Ende des Graphs ist ebenso an diesem.

Formel:

$$CYC_N(Graph) = \frac{| \text{Knoten, die in einem Zyklus sind} |}{| \text{Knoten} |}$$

Korrelation: Eine Steigerung der Zyklusanzahl $CYC_N(Graph)$ führt gleichzeitig zu einer Steigerung der Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling 06] [Mendling+ 07]

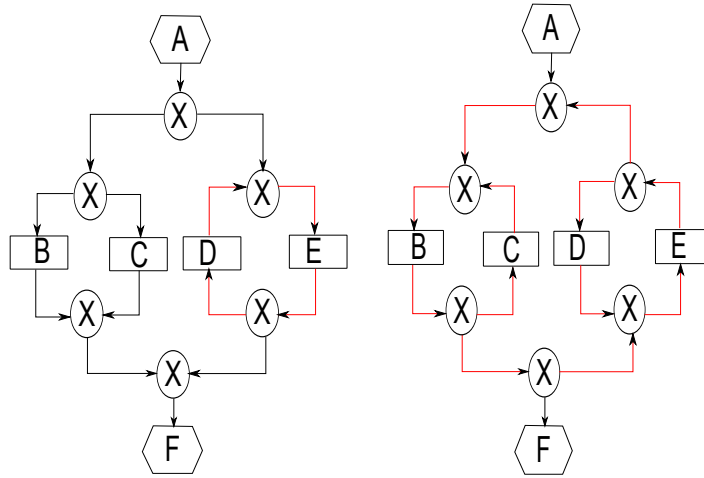


Abbildung 4.4: Zwei EPK Modelle mit unterschiedlichen Werten der Zyklizitätsmetrik

Beispiel: In der Abbildung 4.4 haben wir zwei verschiedene EPK Modelle mit unterschiedlichen Zyklizitätsmetrik-Werten. Im linken Modell gibt es nur einen Zyklus mit 4 Knoten. D, E und die beiden XOR-Verknüpfungen befinden sich in diesem Zyklus. Dagegen haben wir im rechten Modell einen großen Zyklus und 2 weitere Zyklen welche sich in diesem großen Zyklus befinden. Es sind 6 XOR-Verknüpfungen und B,C,D,E vorhanden, die sich im Zyklus befinden.

$$CYC_N(Graph_{links}) = \frac{4}{12} = 0,33$$

$$CYC_N(Graph_{rechts}) = \frac{10}{12} = 0,83$$

Somit ist die Fehlerwahrscheinlichkeit im rechten Modell noch höher als im linken Modell.

4.1.6 Metrik: Parallelität (engl. token split)

Ein nebenläufiges Modell verlangt eine hohe Aufmerksamkeit an synchronisierenden Pfaden. AND und OR Verknüpfungen führen neue "Controlthreads" ein, was neue Token bedeutet. Diese Metrik $TS(Graph)$ zählt die neu eingeführten Token.

Formel:

$$TS(Graph) = \left(\sum_{(c \in C_{OR}) \cup (c \in C_{AND})} d_{out}(c) \right) - 1$$

C ist ein Symbol für die Verknüpfungen und d_{out} steht für die erzeugten Token.

Korrelation: Eine Steigerung der neue Token führt zu einer Steigerung der Fehlerwahrscheinlichkeit. Je mehr Token, desto höher ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling 06] [Mendling+ 07]

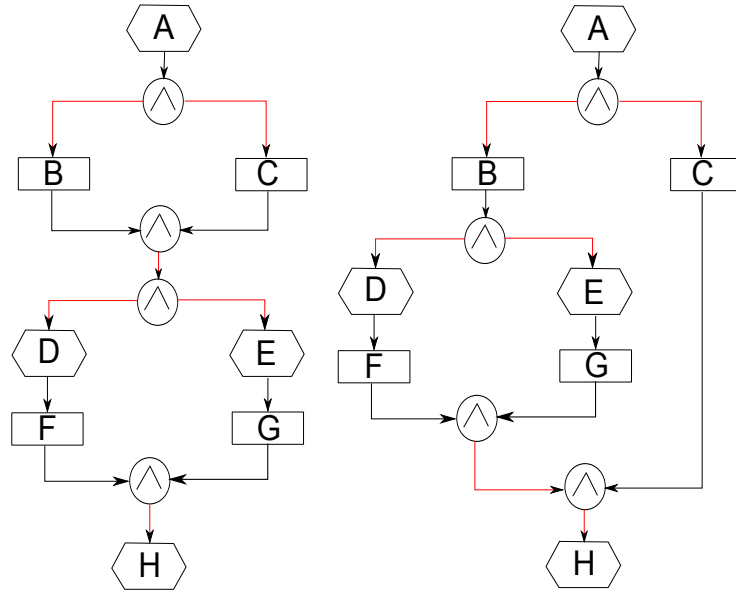


Abbildung 4.5: Zwei EPK Modelle mit gleicher Werten der Parallelitätsmetrik

Beispiel: In beiden Modellen in Abbildung 4.5 sind parallele Ausführungen vorhanden. Obwohl beide unterschiedlich aussehen, haben beide die gleichen Parallelität Werte.

$$TS(Graph_{links}) = \left(\sum_{(c \in C_{OR}) \cup (c \in C_{AND})} d_{out}(c) \right) - 1 = (2 + 1 + 2 + 1) - 1 = 5$$

$$TS(Graph_{rechts}) = \left(\sum_{(c \in C_{OR}) \cup (c \in C_{AND})} d_{out}(c) \right) - 1 = (2 + 2 + 1 + 1) - 1 = 5$$

Die Fehlerwahrscheinlichkeit ist in beiden Modellen dieselbe.

4.1.7 Metrik: Durchmesser (engl. diameter)

Ein Durchmesser bezeichnet die Länge des längsten Pfades vom Startpunkt bis zum Ende. Ein Modell mit einem größeren Durchmesser hat statistisch gesehen eine höhere Fehlerwahrscheinlichkeit wie ein Modell mit einem kleinen Durchmesser. Allerdings gilt diese Wahrscheinlichkeit nicht für die sequentiellen Modelle [Mendling 07]. Definieren wir den Pfad vom Startknoten bis zum Endknoten als $p=(s,e)$ mit Startknoten s und Endknoten e .

Formel:

$$\delta(Graph) = \max\{p(s,e)\}$$

Korrelation: Eine Steigung in $\delta(\text{Graph})$ erhöht die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling+ 07]

Beispiel: Als Beispiel kann man die Modelle in der Abbildung 4.2 betrachten. Die $\delta(\text{Graph})$ im Modell links beträgt 6, während $\delta(\text{Graph})$ im Modell rechts 8 beträgt. So ist die Fehlerwahrscheinlichkeit im rechten Modell noch höher als im linken Modell.

4.1.8 Metrik: Dichte (engl. density)

Hier spielt die Anzahl der Kanten eine große Rolle. Die Dichte $\Delta(\text{Graph})$ bezeichnet das Verhältnis von Anzahl der Kanten zu Knoten in Bezug auf die Verbundenheit innerhalb des Graphes [Turau 10]. Je geringer die Dichte ist, desto schneller werden die Informationen verteilt.

Formel:

$$\Delta(\text{Graph}) = \frac{|\text{Kanten}|}{|\text{Knoten}| * (|\text{Knoten}| - 1)}$$

Korrelation: Je höher die Dichte ist, desto höher ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling+ 07]

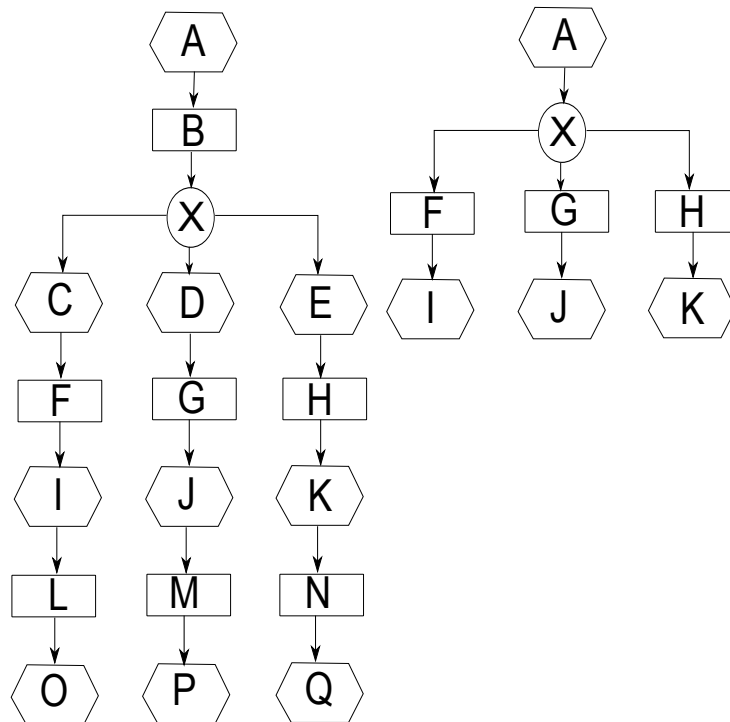


Abbildung 4.6: Zwei EPK Modelle mit unterschiedlicher Dichte [Mendling 07]

Beispiel: Nun wird die Dichte Metrik über die beiden Modelle in Abbildung 4.6 berechnet.

$$\Delta(\text{Graph}_{links}) = \frac{17}{18 * 17} = 0,055 \qquad \Delta(\text{Graph}_{rechts}) = \frac{7}{8 * 7} = 0,125$$

Nach dieser Berechnung ist die Fehlerwahrscheinlichkeit im rechten Modell noch höher als im linken. Auf den ersten Blick, ohne Berechnungen, scheint der Wert des Metriks im linken Modell höher zu sein. Nach der Berechnung des Wertes ist zu erkennen, dass Überraschungen anhand Metriken beseitigt werden können.

4.1.9 Metrik: Verbindungskoeffizient (engl. CNC – coefficient of connectivity)

Der Konnektivitätskoeffizient $\text{CNC}(\text{Graph})$ beschreibt das Verhältnis von Kanten zu Knoten.

Formel:

$$\text{CNC}(\text{Graph}) = \frac{|\text{Kanten}|}{|\text{Knoten}|}$$

Korrelation: Je höher die $CNC(Graph)$ ist, desto größer ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling+ 07]

Beispiel: Betrachten wir die beiden Modelle in Abbildung 4.6.

$$CNC(Graph_{links}) = \frac{17}{18} = 0,945 \qquad CNC(Graph_{rechts}) = \frac{7}{8} = 0,875$$

Somit ist die Fehlerwahrscheinlichkeit im linken Modell höher als im rechten Modell. In einem dichteren Modell ist es wahrscheinlicher einen Fehler zu finden, da es viele Verbindungen zwischen Knoten innerhalb des Modells gibt.

4.1.10 Metrik: Durchschnittsgrad der Verknüpfungen (engl. average degree of connectors)

Diese Metrik gibt die Anzahl der Knoten an, mit der eine Verknüpfung im Durchschnitt verbunden ist.

Formel:

$$\bar{d}_c(Graph) = \frac{1}{|Verknüpfungen|} \sum_{(c \in Verknüpfungen)} d(c)$$

Korrelation: Je höher die $\bar{d}_c(Graph)$ ist, desto höher ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling+ 07]

$$\bar{d}_c(Graph_{links}) = \frac{1}{1} * (3) = 3 \qquad \bar{d}_c(Graph_{rechts}) = \frac{1}{1} * (3) = 3$$

Beispiel: Die beiden Modelle in der Abbildung 4.6 werden betrachtet. Sowohl im linken als auch im rechten Modell gibt es nur eine Verknüpfung und diese hat nur 3 Ausgänge. So haben wir für diese Metrik für beide Modelle die Zahl 3. Demzufolge haben beide Modelle die gleiche Fehlerwahrscheinlichkeit.

4.1.11 Metrik: Maximaler Grad der Verknüpfung (engl. maximum degree of a connector)

Der maximale Grad $\hat{d}_c(Graph)$ an Verbindungen, wird folgendermaßen definiert: Die Zahl der ausgehenden Kanten bei der Verknüpfung mit den meisten Ausgängen.

Formel:

$$\hat{d}_c(Graph) = \max\{d(c) \mid c \in Verknüpfungen\}$$

Korrelation: Je höher die $\hat{d}_c(Graph)$ ist, desto höher ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling+ 07]

$$\hat{d}_c(Graph_{links}) = 3$$

$$\hat{d}_c(Graph_{rechts}) = 3$$

Beispiel: Die Modelle in Abbildung 4.6 werden betrachtet und nach der obigen Formel ist die maximale Zahl an Ausgängen an einer Verknüpfung 3. Auch hier haben beide Modelle die gleiche Fehlerwahrscheinlichkeit von 3.

4.1.12 Metrik: Maximale Tiefe aller Knoten (engl. maximum depth of all nodes)

Die Tiefe bezieht sich auf die maximale Verschachtlung von strukturierten Blöcken in einem Prozess. Nun, um die Tiefe von unstrukturierten Modellen zu finden, gibt es folgenden Algorithmus.

- ◇ Alle Knoten sind mit in-depth Wert "null" initialisiert.
- ◇ Das ganze Prozessmodell wird von Startknoten bis Endknoten durchquert..
- ◇ Bei jedem besuchten Knoten n , wird der neue in-depth Wert $\lambda'_{in}(n)$ aktualisiert, bezogen auf die Werte der Vorgänger $\lambda_{in}(pre)$ und den aktuellen Wert $\lambda_{in}(n)$, gemäß Folgendem:

$$\lambda'_{in}(n) = \begin{cases} \max(\lambda_{in}(n), \lambda_{in}(pre) + 1) & \text{wenn } pre \in S \text{ und } n \notin J \\ \max(\lambda_{in}(n), \lambda_{in}(pre)) & \text{wenn } pre \in S \text{ und } n \in J \\ \max(\lambda_{in}(n), \lambda_{in}(pre)) & \text{wenn } pre \notin S \text{ und } n \notin J \\ \max(\lambda_{in}(n), \lambda_{in}(pre) - 1) & \text{wenn } pre \notin S \text{ und } n \in J \end{cases}$$

- ◇ Die out-depth $\lambda_{out}(n)$ ist gemäß dem Nachfolger-Knoten definiert: Verringerung bei Splits und Erhöhung bei Joints.

Die $\lambda(G)$ wird als Minimum von $\lambda_{in}(n)$ und $\lambda_{out}(n)$ definiert. Somit wird die Tiefe aller Knoten $\Lambda(Graph)$ definiert als Maximum von $\lambda(G)$

Folgende Gleichung gibt die maximale Tiefe des Knotens aus:

Formel:

$$\Lambda(Graph) = \max\{\lambda(n) \mid n \in \text{Knoten}\}$$

Korrelation: Je geringer die $\lambda(G)$, desto niedriger ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling+ 07]

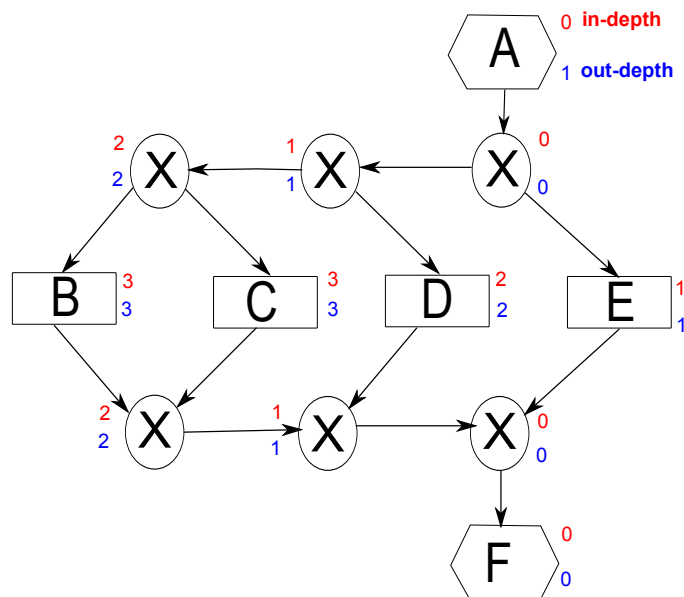


Abbildung 4.7: Ein EPK Beispiel mit in-depth und out-depth Werten [Mendling 07]

Beispiel: In der Abbildung 4.7 ist zu sehen, wie sich die in-depth und die out-depth Werte verändern. Die maximale Tiefe hat den Wert 3.

4.1.13 Metrik: Kontrollflusskomplexität (engl. CFC – control flow complexity)

In der CFC Metrik haben alle Verknüpfungen eine bestimmte Gewichtung. All diese Gewichtungen werden zusammengezählt und abschließend erhalten wir einen CFC Wert über dem Geschäftsprozessmodell. Da die CFC Metrik auch ein Teil der Komplexitätsmetriken ist, wird dies im Kapitel Komplexitätsmetriken genauer erklärt. Mit dieser Formel kann man den CFC Wert berechnen.

Formel:

$$CFC(Graph) = \left(\sum_{(c \in S_{AND})} 1 \right) + \left(\sum_{(c \in S_{XOR})} |c_{XOR Ausgaenge}| \right) + \left(\sum_{(c \in S_{OR})} 2^{|c_{OR Ausgaenge}|} - 1 \right)$$

Korrelation: Je höher die CFC Zahl ist, desto höher ist die Fehlerwahrscheinlichkeit.

Quelle: [Mendling 07] [Mendling+ 07]

Beispiel: siehe Abschnitt 4.2.2

4.2 Komplexitätsmetriken

Damit die Menschen keine Fehler machen oder die Modelle nicht missverstehen, sollen die Modelle einfach zu lesen und leicht zu verstehen sein. Es gibt zahlreiche Modelle, die für den Nutzer schwer zu verstehen sind. Komplexitätsmetriken sagen aus, ob ein Modell einfach oder schwer zu verstehen ist. F. Meyer befasst sich in seiner Diplomarbeit explizit mit dem Thema der Verständlichkeit von graphischen Prozessmodellen [Meyer 06]. Hat man durch diese Metriken herausgefunden, dass die Modelle zu komplex sind, so kann man daran arbeiten die Modelle einfacher darzustellen. Dies hat sogar Auswirkungen auf die Fehlerwahrscheinlichkeit in den Modellen. Je einfacher die Modelle sind, desto geringer ist die Wahrscheinlichkeit, dass man Fehler macht. Die Komplexitätsmetriken können in manche Modellen trotzdem wirkungslos bleiben, da gewisse Komplexität in manche Fällen unvermeidbar ist. So können manchmal die Werte der Komplexitätsmeriken ein Widerspruch zeigen. Einige der Komplexitätsmetriken wurden aus dem Softwarebereich in Prozessbereich angepasst. Hierzu bieten die Forschungsarbeiten von J. Cardoso [Cardoso 06] und V. Gruhn [Gruhn 09] ausführliche Informationen, wie die Metriken aus dem Softwarebereich in Prozessmodellierung angepasst werden. J. Shao betrachtet die Komplexität aus einem unterschiedlicheren Perspektive und entwickelt die kognitiven Wahrnehmungsmetriken [Shao 03]. V. Gruhn gibt in seine Arbeiten [Gruhn+ 06] und [Gruhn 06] einen Überblick über die vorhandene Komplexitätsmetriken.

4.2.1 Metrik: Anzahl der Aktivitäten (engl. number of activities)

Dies ist eine einfache Metrik, bei der es um die Anzahl der Aktivitäten $\alpha(\text{Graph})$ geht. T ist die Menge der Aktivitäten innerhalb eines Modells.

Formel:

$$\alpha(\text{Graph}) = | T |$$

Korrelation: Je mehr Aktivitäten in einem Modell enthalten sind, desto komplexer können diese Modelle sein.

Quelle: [Cardoso 05]

Man sollte darauf achten, dass auch ein Modell mit 50-60 Aktivitäten einfach und leicht zu verstehen ist. Aus diesem Grund ist es nicht sehr empfehlenswert allein diese Metrik zu nutzen.

Beispiel: In der Abbildung 4.8 beträgt der Wert 8, da insgesamt 8 Aktivitäten vorhanden sind.

4.2.2 Metrik: Kontrollflusskomplexität (engl. CFC – control flow complexity)

Diese Metrik basiert auf der Analyse von XOR, OR und AND Splits [Cardoso 05]. Jeder Split in einem Modell erhöht die Anzahl möglicher Entscheidungen folgendermaßen:

AND-Split: Als Ergebnis der AND-Split Ausführung wird nur ein Zustand betrachtet, da nur ein ausgehender Pfad ausgeführt wird, falls die nötigen Eingangspfade eingetroffen sind. Aus diesem Grund wird die CFC Anzahl bei jedem AND-Split um 1 erhöht.

XOR-Split: Die Logik der XOR-Verknüpfung ist, dass auch bei mehreren Eingängen, am Ende nur ein ausgehender Pfad ausgeführt wird. Bei der XOR-Verknüpfung wird einer der möglichen Zustände ausgewählt. Es gibt die Möglichkeit, dass jeder Zustand ausgeführt werden kann. So muss man alle Zustände mitbetrachten. Die CFC Zahl wird bei jedem XOR-Split mit n Ausgängen um die Anzahl n erhöht [Gruhn 06].

OR-Split: Bei OR-Verknüpfung gibt es $2^n - 1$ verschiedene Möglichkeiten, um aus den Ausgängen die gewünschten Pfade auszuwählen. So werden mindestens 1, höchstens n Ausgänge ausgeführt [Gruhn 06]. Das bedeutet, dass jeder OR-Split mit n Ausgängen die CFC Zahl um $2^n - 1$ erhöht.

Formel:

$$CFC(Graph) = \left(\sum_{(c \in S_{AND})} 1 \right) + \left(\sum_{(c \in S_{XOR})} |c_{XOR} Ausgaenge| \right) + \left(\sum_{(c \in S_{OR})} 2^{|c_{OR} Ausgaenge|} - 1 \right)$$

Korrelation: Je höher die CFC Zahl ist, desto höher ist die Komplexität.

Quelle: [Gruhn 06] [Gruhn+ 06] [Cardoso 05]

Beispiel: In der Abbildung 4.8 ist ein einfaches BPMN Diagramm zu sehen. Es gibt insgesamt 4 Verknüpfungen, aber da in der CFC Metrik nur die Split-Verknüpfungen betrachtet werden, haben wir 3 Verknüpfungen, welche wir in Betracht ziehen müssen. Ein XOR-Split, OR-Split und AND-Split Verknüpfungen mit jeweils 2 Ausgängen. Der XOR-Split ergibt ein Wert von 2, die AND-Verknüpfung ergibt ein Wert von 1 und die OR-Verknüpfung ergibt ein Wert von 3.

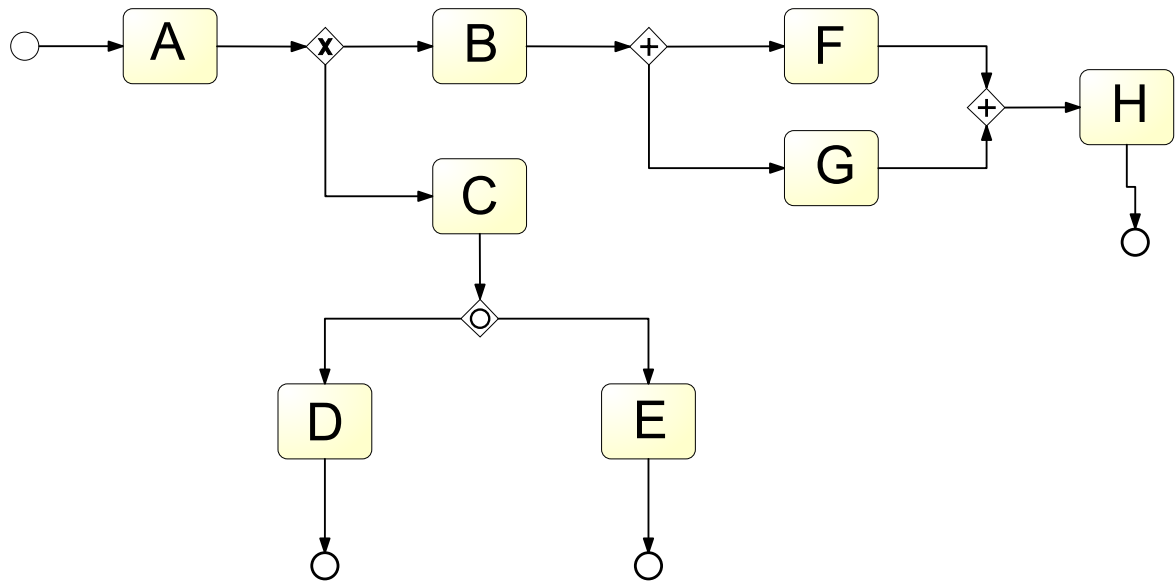


Abbildung 4.8: Ein BPMN Modell mit CFC Wert 6

So haben wir ein CFC Wert von 6.

$$CFC(Graph) = 1 + 2 + 3 = 6$$

4.2.3 Metrik: Zyklomatische Komplexität (engl. cyclomatic number)

Die zyklomatische Zahl, oder auch McCabe Metrik genannt, misst die Anzahl der möglichen binären Entscheidungen im Diagramm [McCabe 89]. Nach dem die Anzahl der binären Entscheidungen errechnet wird, wird die gefundene Zahl um 1 erhöht und somit ergibt sich die zyklomatische Zahl $v(Graph)$. Die binäre Entscheidung wird als $\beta(C)$ mit der Verknüpfung C definiert.

Formel:

$$v(Graph) = |\beta(C)| + 1$$

Korrelation: Je geringer die zyklomatische Zahl ist, desto geringer ist die Komplexität.

Quelle: [Gruhn 06] [Gruhn+ 06] [Cardoso 06] [McCabe 89]

Beispiel: Betrachten wir als Beispiel weiterhin die Abbildung 4.8. Es gibt 2 Verknüpfungen, bei denen eine binäre Entscheidung stattfindet. Die XOR-Verknüpfung mit den Ausgängen zu B und C. Die OR-Verknüpfung mit den Ausgängen D und E. So hat man 2 binäre Entscheidungen. Diese Zahl erhöht man um 1. Der zyklomatische Wert ist nun der Wert 3.

4.2.4 Metrik: Maximale und durchschnittliche Verschachtelungstiefe (engl. maximum and mean nesting depth)

Mit dieser Metrik kann man die Komplexität mit Hilfe von der Verschachtelungstiefe herausfinden. Die Verschachtelungstiefe (nesting depth) $\eta(\text{Graph})$ von einer Aktivität ist die Anzahl der Entscheidungen $\beta(C)$ im Graph, welche für die Ausführung von bestimmten Aktivitäten notwendig sind [Harrison 81].

Formel:

$$\eta(\text{Graph}) = |\beta(C)|$$

Korrelation: Eine große Verschachtelungstiefe erzeugt eine größere Komplexität.

Quelle: [Harrison 81] [Gruhn+ 06] [Gruhn 06]

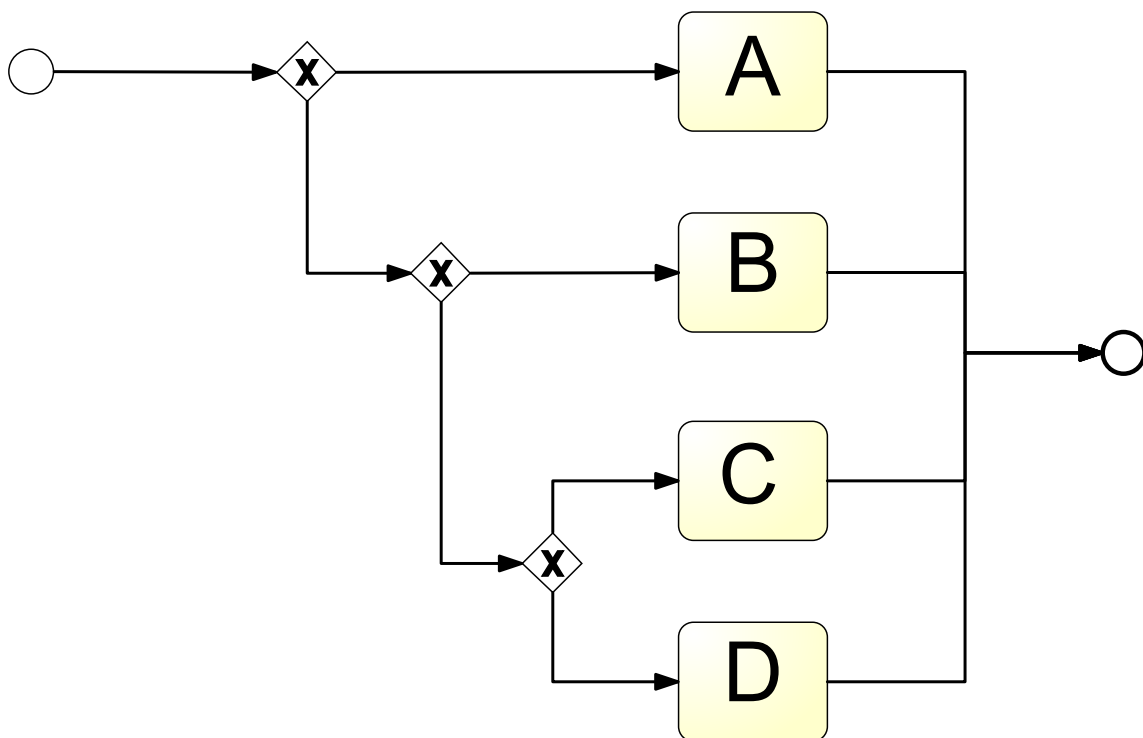


Abbildung 4.9: Ein BPMN Modell mit maximaler Verschachtelungstiefe von 6

Beispiel: Um die Aktivitäten C oder D in der Abbildung 4.9 auszuführen, müssen 3 Entscheidungen an XOR-Verknüpfungen getroffen werden, welcher Pfad ausgeführt werden soll. Genau die Anzahl dieser Entscheidungen ergibt den Wert dieser Metrik. In diesem Diagramm beträgt dieser Wert 3, um die Aktivitäten C oder D auszuführen.

4.2.5 Metrik: Kognitive Wahrnehmung (engl. cognitive weight)

Die kognitive Wahrnehmung der Menschen beschreibt, wie die Informationen wahrgenommen und weiterverarbeitet werden. Kognitive Wahrnehmungsfähigkeiten der Menschen beeinflussen natürlich die Komplexität der Modelle. Der Mensch nimmt jeden Konstruktor mit einer anderen kognitiven Gewichtung wahr [Shao 03]. In der folgenden Tabelle 4.1 ist zu erkennen, welche Constructoren mit welcher Gewichtung wahrgenommen werden.

Kontrollstruktur	Gewichtung
XOR Verknüpfung mit genau 2 Ausgängen	2
XOR Verknüpfung mit mehr als 2 Ausgängen	3
AND-Split	4
OR-Split	7
Subtask	2
Parallele Ausführungen	6

Tabelle 4.1: Kognitive Werte über die Wahrnehmung [Gruhn 09]

Formel:

Der Wert dieser Metrik $\kappa(\text{Graph})$ errechnet sich aus der Summe der Werte in der Tabelle 4.1, welche sich im Diagramm befinden.

Korrelation: Je höher $\kappa(\text{Graph})$ ist, desto höher ist die Komplexität.

Quelle: [Gruhn 09] [Shao 03] [Cardoso 06] [Gruhn 09]

Beispiel: In der Abbildung 4.8 beträgt der Wert für $\kappa(\text{Graph})$ 19. Es gibt eine XOR-Verknüpfung mit zwei Ausgängen, eine OR-Verknüpfung, eine AND-Verknüpfung und eine parallele Ausführung (F und G). Die Summe der Gewichtungen der genannten Kontrollstrukturen ergibt 19.

4.2.6 Metrik: Prozessaufwurf (engl. fan-in / fan-out)

Ein Geschäftsprozessmodell in Untermodelle zu unterteilen kann zu wiederverwendbaren, einfacheren oder kleineren Modellen führen. Explizit hierfür wurden neue Metriken entwickelt, wie Fan-In und Fan-Out Metriken.

Fan-In Metrik (fan_{in}) bezeichnet die Anzahl der Aktivitäten, die einen Unterprozess aufrufen.

Fan-Out Metrik (fan_{out}) bezeichnet die Anzahl der Unterprozesse, die durch andere Aktivitäten aufgerufen werden.

Formel:

$$\rho(Graph) = fan_{in} + fan_{out}$$

Korrelation: Hat ein Modell einen hohen Prozessaufwurf $\rho(Graph)$ Wert, so ist dieses Modell schwer zu verstehen.

Quelle: [Gruhn 06]

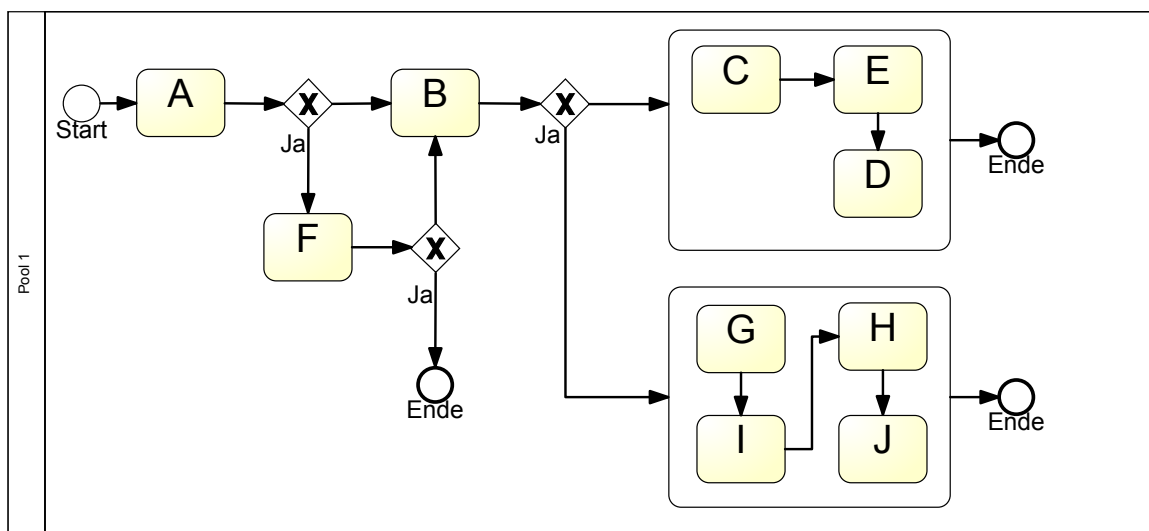


Abbildung 4.10: Ein BPMN Modell zur Fan-in und Fan-out Metrik

Beispiel: In der Abbildung 4.10 ist es zu sehen, dass die Fan-In Metrik den Wert 1 hat, da nur die Aktivität B einer der Unterprozesse aufrufen kann. Die Fan-Out Metrik hat den Wert 2, da es zwei Unterprozesse gibt, die aufgerufen werden. Somit beträgt der Wert für die $\rho(Graph)$ 3.

4.2.7 Metrik: Umfangsmetriken (engl. structural metrics)

Wie bei der ersten Metrik über die Anzahl der Aktivitäten, kann man hier, all die anderen Bestandteile von Modellen zählen.

Formel:

trivial

Mit der Größe eines Modells steigt auch die Komplexität. So kann man mit den Umfangsmetriken, welche unten aufgelistet sind, alle Bestandteile zählen [Meyer 06]. Hierzu gehören folgende Werte zur Größe des Modells.:

- | | |
|----------------------|---------------------|
| ◇ Zwischenereignisse | ◇ Split Operationen |
| ◇ Funktionen | ◇ XOR-Split |
| ◇ Startereignisse | ◇ OR-Split |
| ◇ Endereignisse | ◇ AND-Split |
| ◇ Kanten | ◇ XOR-Join |
| ◇ Verküpfungen | ◇ OR-Join |
| ◇ Join Operationen | ◇ AND-Join |

Korrelation: Eine hohe Anzahl an oben genannten Bestandteile des Modells erhöht gleichzeitig die Komplexität des Modells.

Quelle: [Meyer 06]

Beispiel: In der Abbildung 4.10 ist die Anzahl der verschiedenen Bestandteile folgendermaßen:

- | | |
|--------------------------|-------------------------|
| ◇ Zwischenereignisse = 0 | ◇ Split Operationen = 3 |
| ◇ Funktionen = 0 | ◇ XOR-Split = 3 |
| ◇ Startereignisse = 1 | ◇ OR-Split = 0 |
| ◇ Endereignisse = 3 | ◇ AND-Split = 0 |
| ◇ Kanten = 17 | ◇ XOR-Join = 0 |
| ◇ Verküpfungen = 3 | ◇ OR-Join = 0 |
| ◇ Join Operationen = 0 | ◇ AND-Join = 0 |

4.2.8 Metrik: Kantenüberschneidungen (engl. edge intersection)

Dies ist eine graphische Metrik. Bei Kantenüberschneidungen Metrik $\Psi(\text{Graph})$ geht darum, wie oft sich zwei Kanten im Modell überschneiden, was den Lesefluss behindern würde.

Formel:

$$\Psi(\text{Graph}) = | \text{Kantenberschneidungen} |$$

Korrelation: Je mehr Überschneidungen von Kanten existieren, desto komplexer ist das Modell.

Quelle: [Meyer o6]

Beispiel: In der Abbildung 4.11 ist das Überschneiden zweier Kanten zu sehen. Somit gibt es nur eine Überschneidung und der Wert dieser Metrik beträgt 1.

4.2.9 Metrik: Schreibrichtung (engl. writing direction)

Ein Geschäftsprozessmodell, das keine genaue und klare Leserichtung besitzt, ist schwer zu verstehen [Meyer o6]. Üblicherweise werden die Modelle von oben nach unten oder von links nach rechts dargestellt. Diese Metrik zählt, wie oft von der Hauptleserichtung abgewichen wird. Bei der Schreibrichtung Metrik $\Xi(\text{Graph})$ werden die Kanten κ , die von dieser Darstellung abweichen, gezählt.

Formel:

$$\Xi(\text{Graph}) = |\kappa|$$

Korrelation: Je höher die Anzahl der abgewichenen Kanten sind, desto höher ist die Komplexität für den Nutzer.

Quelle: [Meyer o6]

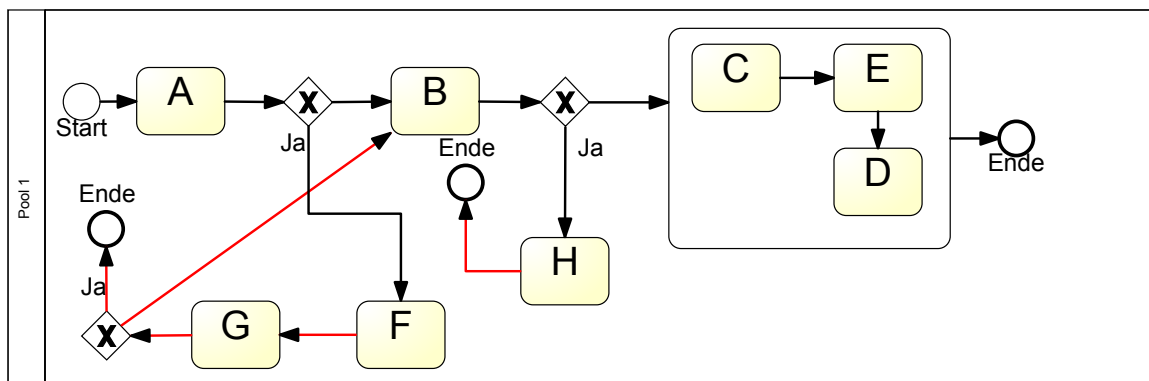


Abbildung 4.11: Ein BPMN Modell zur Schreibrichtung und Kantenüberschneidungen

Beispiel: In der Abbildung 4.11 gibt es fünf Kanten, welche von der Schreibrichtung abweichen und mit rot bezeichnet sind. Somit beträgt der Wert dieser Metrik 5.

4.2.10 Metrik: Halstead Metrik (engl. halstead metric)

Halstead Metrik ist im Softwarebereich die bekannteste Metrik für die Softwarekomplexität. Diese wurde umgewandelt, so dass die Metrik für die Geschäftsprozessmodelle eingesetzt werden kann [Cardoso 06]. Halstead Metrik basiert direkt auf der Anzahl von Aktivitäten, Verknüpfungen und Daten im Modell.

n_1 = Die Anzahl der verschiedenen Aktivitäten, Splits, Joins und Kontrollflusselementen

n_2 = Die Anzahl der verschiedene Daten-variablen

N_1 = Die Anzahl aller Ereignissen

N_2 = Die Anzahl aller Daten-variablen

Nun sieht die Halstead Metrik folgendermaßen aus:

Formel:

Prozesslänge $N = n_1 * \log_2(n_1) + n_2 * \log_2(n_2)$

Prozessvolumen $V = (N_1 + N_2) * \log_2(n_1 + n_2)$

Prozessschwierigkeit $D = (n_1/2) * (N_2/n_2)$

Korrelation: Je höher die Werte der N, V und D sind, desto höher ist die Komplexität.

Quelle: [Cardoso 06] [Thammarak 10] [Makni 10]

Einer der vielen Vorteile von Halstead Metrik ist, dass keine tiefgründige Analyse stattfinden muss und dadurch ist es einfacher und schneller, eine Meinung über die Komplexität zu bilden.

Beispiel: Hier wird empfohlen Beispiele aus dem Softwarebereich anzuschauen.

4.2.11 Metrik: Schnittstelle (engl. IC – interface)

Hier spielt die fan-in und fan-out Metrik eine indirekte Rolle, da diese Metrik aus dem Softwarebereich hier in Inputs und Outputs umgewandelt ist. Der Grund ist, dass die Unterprozesse erst dann aufgerufen werden können, wenn die Inputs schon vorhanden sind. Dementsprechend bekommt man eine Menge von Outputs, wenn die aufgerufenen Unterprozesse ausgeführt worden sind.

Formel:

$$IC(Graph) = Laenge * (| Inputs | * | Outputs |)^2$$

Wenn die Aktivitäten in der „Black Box“ sind, so ist die Länge = 1.

Sind die Aktivitäten in der „White Box“, so nutzt man die zyklomatische Metrik als Länge. Die Definition bezüglich „Black Box“ und „White Box“ sind im Abschnitt 3.4.1.3 zu finden.

Korrelation: Je höher der Wert IC(Graph) desto höher ist die Komplexität.

Quelle: [Cardoso 06]

Beispiel: -

4.2.12 Metrik: Netzwerkkomplexitätskoeffizient (engl. CNC – coefficient of network complexity)

Dies ist eine einfache graphische Metrik, welche nur durch Zählen von graphischen Objekten eine Meinung über die Komplexität zeigen kann. Durch folgende Formel wird der Wert dieser Metrik ausgerechnet.

Formel:

$$CNC(Graph) = \frac{|Kanten|}{|Aktivitäten, Splits, Joins|}$$

Korrelation: Eine hohe CNC(Graph) Zahl deutet auf eine hohe Komplexität hin.

Quelle: [Cardoso 06]

Beispiel: Die Abbildung 4.11 wird betrachtet:

$$CNC(Graph) = \frac{15}{8 + 3} = 1,36$$

4.3 Qualitätsmetriken

Betrachten wir die Qualität der Geschäftsprozesse im Allgemeinen, so haben wir 4 Kategorien: Funktion, Input/Output, Non-Human Ressources, Human Ressources. Unten sind diese vier verschiedenen Kategorien mit verschiedenen Dimensionen der Qualität zu sehen [Heravizadeh 08]. Anhand der Messungen der unten genannten Eigenschaften lässt sich eine allgemeine Aussage über die Qualität treffen. Es kommt oft vor, dass man die Qualität anhand nur einer der folgenden Punkte bestimmt, was natürlich zu falschen Aussagen führen kann.

Funktion	Input/Output	Non-Human Resource	Human Resource
Genauigkeit	Genauigkeit	Eignung	Fachwissen
Sicherheit	Sachlichkeit	Genauigkeit	Qualifizierung
Zuverlässigkeit	Glaubwürdigkeit	Sicherheit	Zertifizierung
Verständlichkeit	Ansehen	Zuverlässigkeit	Erfahrung
Erlernbarkeit	Erreichbarkeit	Zeiteffizienz	Zeiteinteilung
Zeiteffizienz	Sicherheit	Ressourcennutzung	Rhetorik
Effektivität	Relevanz	Effektivität	
Produktivität	Wertzuwachs	Gefahrlosigkeit	
Gefahrlosigkeit	Pünktlichkeit	Nutzerzufriedenheit	
Nutzerzufriedenheit	Vollständigkeit	Robustheit	
Robustheit	Datenmenge	Verfügbarkeit	

Tabelle 4.2: Qualitätsmerkmale [Heravizadeh 08]

Kommen wir ganz konkret zu der Definition von Qualität in Geschäftsprozessmodellen. In diesem Bereich wird versucht, die Qualitätsmerkmale von Softwarebereichen an BPMN anzupassen, da die Qualitätsmetriken erst für die Softwarequalität eingeführt wurden. Es gibt viele Ähnlichkeiten. Einige davon sind [Khelif 10]:

Module/Klasse = Aktivität
Methode/Funktion = Operation
Variable/Konstante = Datenelement
Interface = Subtask

Die Qualitätsmetrik in Geschäftsprozessmodellen kann in drei Kategorien unterteilt werden. Coupling (Kopplung), Cohesion (Zusammenhang) und Komplexität.

4.3.1 Coupling (Kopplung) Metriken

Kopplungsmetriken beziehen sich auf die Beziehung der Aktivitäten in einem Modell [Khelif 10]. Wie stark sind die Aktivitäten in einem Modell miteinander verbunden. Der Begriff „verbunden“ bedeutet, dass eine Aktivität mit den anderen eine oder mehrere Informationen teilt.

Für ein gegebenes Geschäftsprozessmodell versteht man unter der Kopplungsmetrik die Zahl der Verbindungen zwischen den Aktivitäten. Zusätzlich wird festgestellt, wie kompliziert diese Verbindungen sind und welche Art von Verbindungen diese sind. Hieraus wird reflektiert, wie kritisch und wichtig eine Aktivität ist. Demnach bedeutet dies, dass ein hoher Wert bei der Berechnung einer Kopplungsmetrik, eine hohe Anzahl an Aktivitäten bedeuten kann. In anderen Worten: Es gibt eine hohe Informationsabhängigkeit zwischen Aktivitäten, wenn der Wert der Kopplungsmetrik hoch ist.

Kurz gefasst: je stärker der Wert der Kopplungsmetrik zwischen Prozessen ist, desto schwieriger ist es die Prozesse zu verstehen, zu ändern und zu korrigieren.

Für die Umsetzung zwischen objektorientierter Software und BPMN gibt es folgende Vergleiche.

OO-Software	BPMN Notation
Class/Package	Subprocess, Process
Method	Task
Method invocation	Reception of a Sequence flow or a message flow by a task
Variable/Constant	Data Object
Commentline	Annotation
Interface of class	The set of tasks in a process, which send or receive a message flow
Local data in a class (Attributes)	Data Object generated by a process and not used by an activity from outside the process

Tabelle 4.3: Adaptation von OO-Software Konstrukten ins Prozessmodellierung [Khelif 10]

4.3.1.1 Metrik: Nachrichten-/ Sequenzflussmetrik (engl. MSPC – message or sequence flow passing coupling)

Dies ist eine umgewandelte Version von einer “Message Passing Coupling” (MPC) Metrik aus dem Software Bereich. In Geschäftsprozessmodellen heißt dies „Message or sequence flow passing coupling“ [Khelif 10].

Formel:

MSPC ist die Anzahl der Nachrichten oder Sequenzflüsse, welche direkt von einem Prozess-task zu einem anderen Task gesendet werden.

Korrelation: Je höher der Wert dieser Metrik ist, desto schwieriger ist es, das Modell zu verstehen.

Quelle: [Khelif 10]

Beispiel: trivial.

4.3.1.2 Metrik: Prozessrückmeldung
(engl. RFP – response for process)

Eine direkte Beschreibung der Metrik ist durch folgende Formel zu erkennen. Die Anzahl der aufgerufenen Tasks und Tasks, die im Prozess sind ergeben den Wert dieser Metrik.

Formel:

$$RFP = | RS | \quad RS = \{T_j\} \cup \{R_j\}$$

T_j = die Menge aller Tasks im Prozess

R_j = die Menge aller Tasks, die aufgerufen worden sind.

Korrelation: Je größer die RFP ist, desto komplexer ist das Modell.

Quelle: [Khelif 09] [Khelif 10]

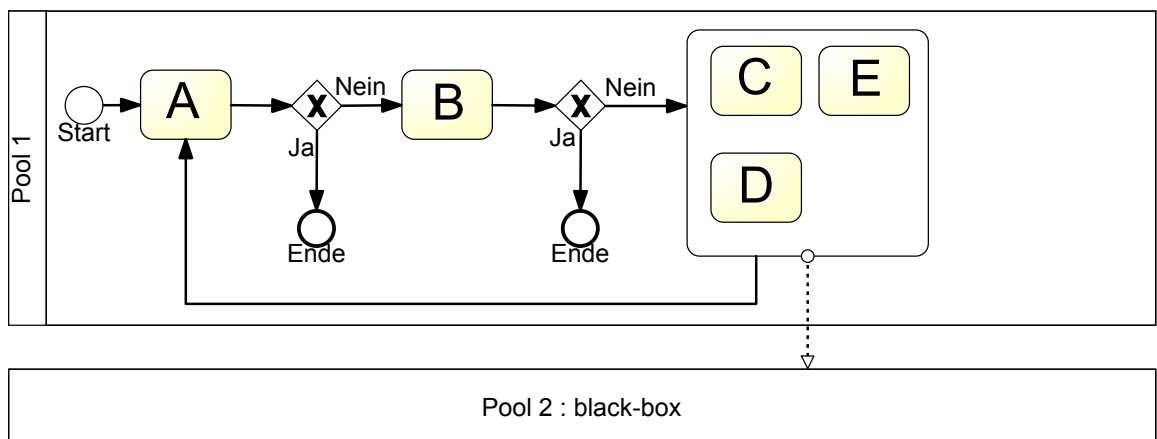


Abbildung 4.12: Ein BPMN Modell mit RFP Metrik Berechnung

Beispiel: In der Abbildung 4.12 haben wir den RFP Wert 5 für den aufgeklappten Unterprozess. Die T_j beträgt 3, da sich 3 Aktivitäten (C,D,E) im Prozess befinden. Die R_j beträgt 2, da zuerst die Aktivität A aufgerufen wird und Pool 2 wird auch aufgerufen.

$$RFP = | RS | = 5$$

$$RS = \{T_j\} \cup \{R_j\} = \{C, D, E\} \cup \{A, Pool2\} = \{C, D, E, A, Pool2\}$$

4.3.1.3 Metrik: Informationsflussdichte (engl. IFCP - information flow coupling process)

Diese Metrik ist eine Umwandlung der „ICP – Information flow based coupling“ Metrik aus dem Softwarebereich [Khelif 10].

Formel: Es geht um die Anzahl von Tasks in einem Prozess P, welche Sequenzflüsse oder Nachrichtenflüsse empfangen und nebenbei Datenobjekte als Input haben. Die Anzahl der genutzten Daten beeinflussen den Metrikwert sehr.

Korrelation: Je mehr Daten genutzt wurden, desto stärker ist die Kopplung zwischen den Tasks.

Quelle: [Khelif 10]

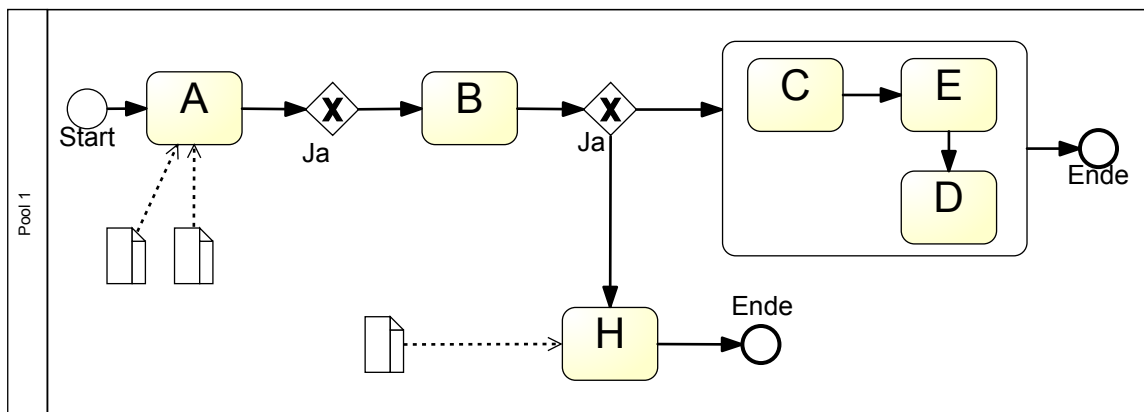


Abbildung 4.13: Ein BPMN Modell zur IFCP Metrik

Beispiel: In der Abbildung 4.13 sind zwei Aktivitäten dargestellt, welche jeweils Datenobjekte als Input bekommen. Diese zwei Aktivitäten werden im Prozess aufgerufen. So beträgt der Wert dieser Metrik 2.

4.3.1.4 Metrik: Datenlokalität (engl. LDA – locality of data activity)

Diese Metrik ist eine Umwandlung der „LD – Locality of Data“ Metrik aus dem Softwarebereich. Während in Softwaremetriken die Daten untersucht werden, wird in Prozessmodellen die Datenobjekte untersucht. [Khelif 10].

Formel:

$$LDA(Graph) = \frac{\sum_{i=1}^n |L_i|}{\sum_{i=1}^n |DT_i|}$$

Eine Aktivität (Subprozess/Task) mit n Tasks kann mathematisch wie oben ausgedrückt werden.

$DT_i (1 \leq i \leq n)$ ist die Menge an Daten, verbunden zu Task T_i innerhalb einer Aktivität.

$L_i (1 \leq i \leq n)$ ist die Menge an Daten, die durch die Aktivität/Task T_i produziert wurden.

Korrelation: Je höher die LDA ist, desto aussagekräftiger ist der Prozess. Anders geschrieben: je höher der LDA Wert ist, desto einfacher ist es, den Prozess zu testen.

Quelle: [Khelif 10] [Khelif 09]

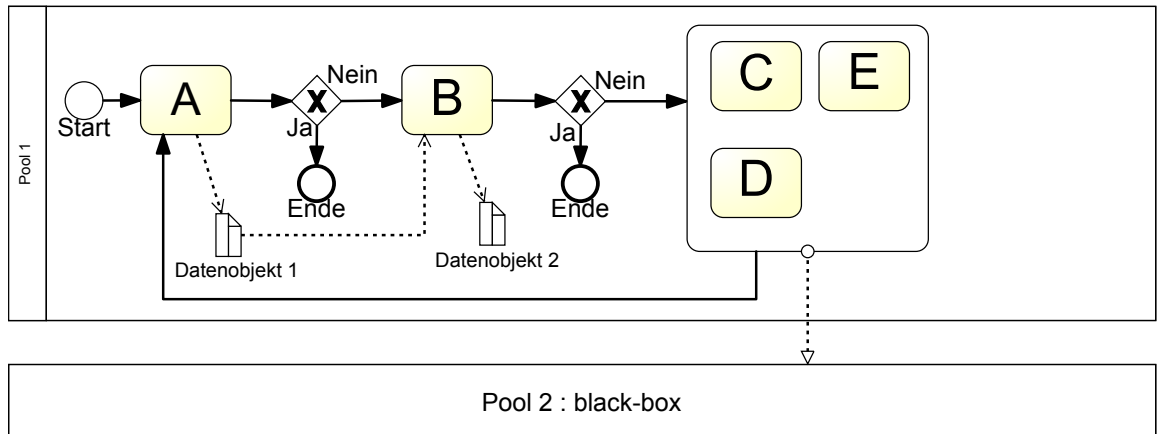


Abbildung 4.14: Ein BPMN Modell zur LDA Metrik Berechnung

Beispiel: In der Abbildung 4.14 berechnen wir die LDA Metrik von der Aktivität B. 2 Datenobjekte sind mit der Aktivität B verbunden. So ist der Wert von DT_i gleich 2. Datenobjekt 2 wird durch die Aktivität B erzeugt. So beträgt der Wert für die L_i gleich 1.

$$LDA(Graph) = \frac{\sum_{i=1}^n |L_i|}{\sum_{i=1}^n |DT_i|} = \frac{1}{2} = 0,5$$

Der LDA Wert für die Aktivität B in der Abbildung 4.14 ist somit 0,5.

4.3.1.5 Metrik: Eingehende-/Ausgehendeflussnachrichten (engl. ICP & ECP – imported/exported coupling of a process)

Formel: ICP zählt für jeden Prozess die Anzahl der Nachrichten / Sequenzflüsse, welche bei Tasks innerhalb des Prozesses oder direkt vom Prozess gesendet wurden.

ECP zählt für jeden Prozess die Anzahl der Nachrichten / Sequenzflüsse, welche bei Tasks vom Prozess oder direkt vom Prozess empfangen wurden.

Korrelation: Ein hoher ECP Wert erhöht die Wahrscheinlichkeit für die Entstehung von Problemen.

Quelle: [Khelif 10]

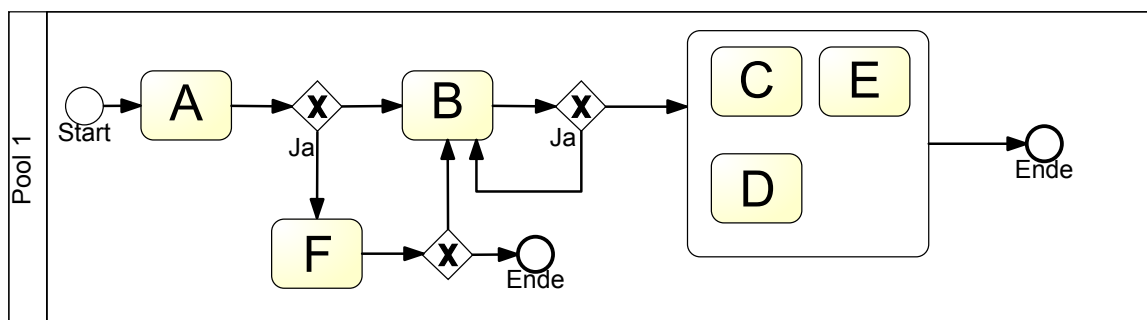


Abbildung 4.15: Ein BPMN Modell mit unterschiedlichen ICP und ECP Metrik Werte

Beispiel: In der Abbildung 4.15 berechnen wir die ICP und ECP Metrikwerte von der Aktivität B. Die Aktivität B hat den ECP Wert 3, weil sie 3 Sequenznachrichten bekommt, alle drei von Gateways. Der ICP Wert beträgt 1, da nur eine Sequenznachricht ausgesendet wird.

Ein hoher ICP Wert hängt von mehreren externen Diensten ab, welche von anderen Prozessen angeboten werden. Dies kann die Verzögerungen, die Kosten und die Fehlerwahrscheinlichkeit erhöhen.

4.3.1.6 Metrik: Aufgabenkopplung (engl. TCpl – task coupling)

Dies ist eine Umwandlung von „OpCpl – Operation Coupling“ [Khelif 10] aus dem Softwarebereich. Diese Metrik misst die Kopplung zwischen Aufgaben/Tasks von einem Prozess P und andere Prozessaufgaben.

Formel:

Die Summe der Anzahl von Aufgaben in P, welche direkte Sequenz- oder Nachrichtenflüsse an andere Prozesse sendet. Überdies wird die Anzahl der Aufgaben in P, welche Sequenzflüsse / Nachrichtenflüsse empfangen, ebenso mitgezählt.

Korrelation: Je höher der Wert, desto schwieriger ist es das Modell zu verstehen.

Quelle: [Khelif 10]

Beispiel: siehe [Khelif 10]

4.3.1.7 Metrik: Prozesskopplung (engl. PCpl – process coupling)

PCpl misst die Kopplung zwischen einem Aktivität und anderen Aktivitäten und Pools.

Formel:

Es ist die Summe aller Nachrichten- oder Sequenzflüsse, gesendet von einem Aktivität P zu anderen Aktivitäten [Khelif 10]. Es ist gleichzeitig auch die Zahl der Nachrichten- / Sequenzflüsse, welche direkt von anderen Aktivitäten zu dem Aktivität P gesendet sind.

Korrelation: Je höher der Wert desto stärker ist die Kopplung.

Quelle: [Khelif 10]

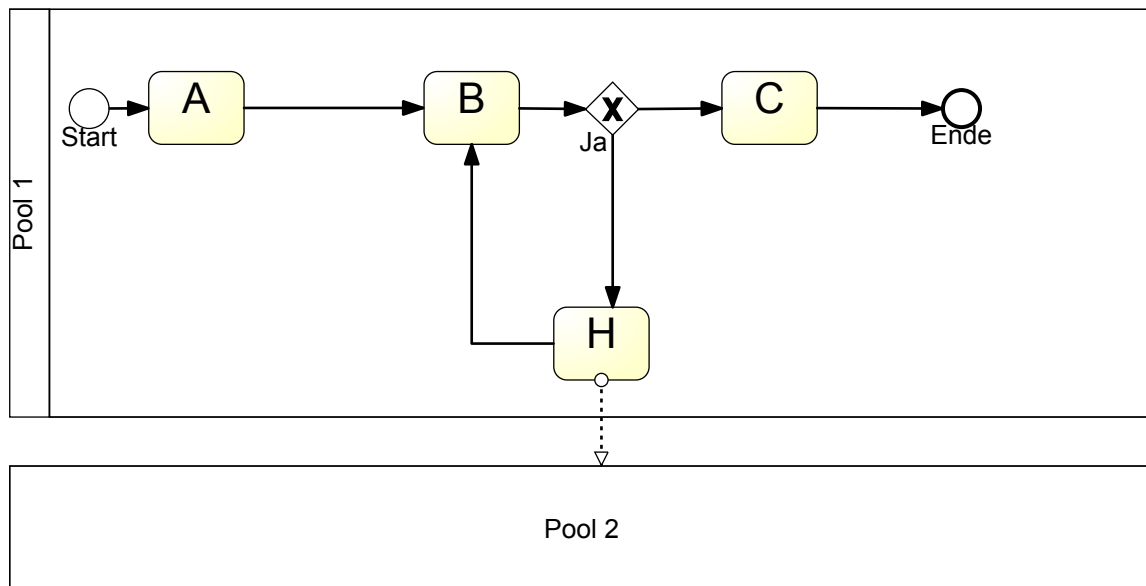


Abbildung 4.16: Ein BPMN Modell zur PCpl Metrik

Beispiel: In der Abbildung 4.16 sind drei Aktivitäten (A, B und H) dargestellt, welche jeweils eine Verbindung direkt zu einem anderen Aktivität oder Pool haben. Die Aktivität A hat den Wert 1, da sie eine Sequenzfluss direkt an B schickt. Die Aktivität B hat den Wert 2, da sie direkt von A und H eine Sequenzfluss bekommt. Die Aktivität H hat ebenso den Wert 2, da sie eine Sequenzfluss zu B sendet und einen Nachrichtenfluss an dem "Black Box" Pool. Der Wert für die PCpl Metrik ergibt sich aus der Summe dieser Werte. PCpl Wert ist hier $1 + 2 + 2 = 5$.

4.3.1.8 Metrik: Daten-/Kontrollflusskopplung (engl. DFC – data and control flow coupling)

Dies ist eine umgewandelte Version von einer "CBD - Coupling between object classes" Metrik aus dem Softwarebereich.

Formel:

DFC zählt die Sequenzflüsse und die Datenobjekte. Es wird bei dieser Metrik kein Unterschied zwischen Import und Export Kopplung gemacht.

Korrelation: Je höher der Wert dieser Metrik ist, desto stärker ist die Kopplung. Gleichzeitig erhöht sich die Komplexität.

Quelle: [Vanderfeesten 09]

Beispiel: trivial.

4.3.1.9 Metrik: Kopplungsmetrik (engl. CP – coupling)

CP zählt alle Aktivitäten-paare in einem Geschäftsprozessmodell, welche miteinander verbunden sind [Vanderfeesten 09].

Formel:

$$CP(Graph) = \sum_{t_1, t_2 \in T} \frac{connected(t_1, t_2)}{|T| * (|T| - 1)}$$

t_1 und t_2 sind einzelne Aktivitäten aus der Menge der Aktivitäten T. Jede Verknüpfung zwischen zwei Aktivitäten hat eine andere Gewichtung. Die Gewichtung hängt von der Wahrscheinlichkeit für die Ausführung ab. Die Gewichtung sieht folgendermaßen aus:

AND Verknüpfung ist die stärkste Verknüpfung, weil jede AND Verknüpfung 100 Prozent zu einem Fall führen wird.

XOR Verknüpfung ist die schwächste Verknüpfung, weil aus allen Fällen nur einer gewählt wird. Die Wahrscheinlichkeit dieser Verbindung ist $\frac{1}{m*n}$. m steht für die Anzahl der ankommenden (incoming) Verzweigungen und n steht für die abgehenden (outgoing) Verzweigungen.

OR Verknüpfung hat eine Gewichtung zwischen AND und XOR Verknüpfungen, weil bei den OR Verknüpfungen fest vorgegeben ist, welche Verzweigung gewählt wird. Die Gewichtung der Kante hängt davon ab, ob diese Kante ausgewählt wurde oder nicht. Es gibt insgesamt $(2^m - 1) * (2^n - 1)$ Kombinationen, welche Kante gewählt werden soll. Die Gewichtung der OR-Verknüpfung kann anhand folgender Formel berechnet werden.

$$CP(OR - Verknüpfung) = \frac{1}{(2^m - 1) * (2^n - 1)} + \frac{(2^m - 1) * (2^n - 1) - 1}{(2^m - 1) * (2^n - 1)} - \frac{1}{m * n}$$

Korrelation: Je höher der Wert ist, desto stärker ist die Kopplung.

Quelle: [Vanderfeesten 09]

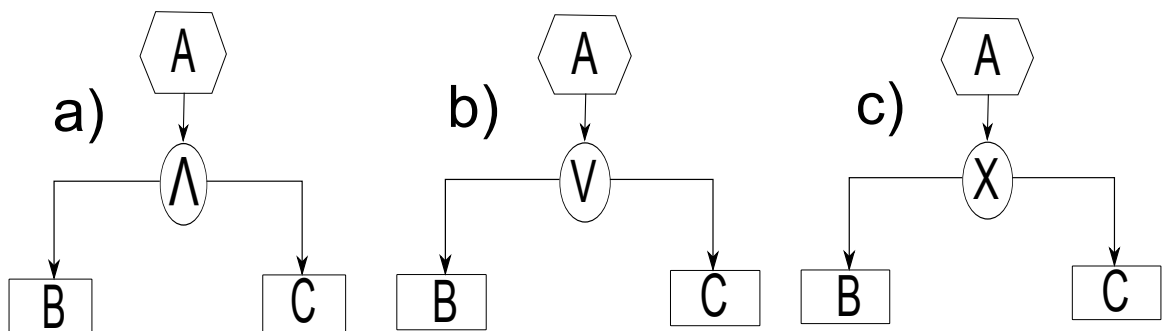


Abbildung 4.17: Drei verschiedene Teile eines Modells für die Coupling Metrik [Vanderfeesten 09]

Beispiel: In der Abbildung 4.17 sind drei Beispiele zu sehen. Bei dem Beispiel "a)" mit AND-Verknüpfung beträgt der CP Wert $\frac{1}{3}$. Der Wert zwischen A und B ist 1 und der Wert zwischen A und C ist genauso 1. Setzt man die Werte in die Formel des CP Werts, so bekommt man das Ergebnis $\frac{1}{3}$.

Bei dem Beispiel "c)" mit XOR-Verknüpfung beträgt der CP Wert $\frac{1}{6}$ und der CP Wert bei dem Beispiel "b)" mit der AND-Verknüpfung beträgt 0,222.

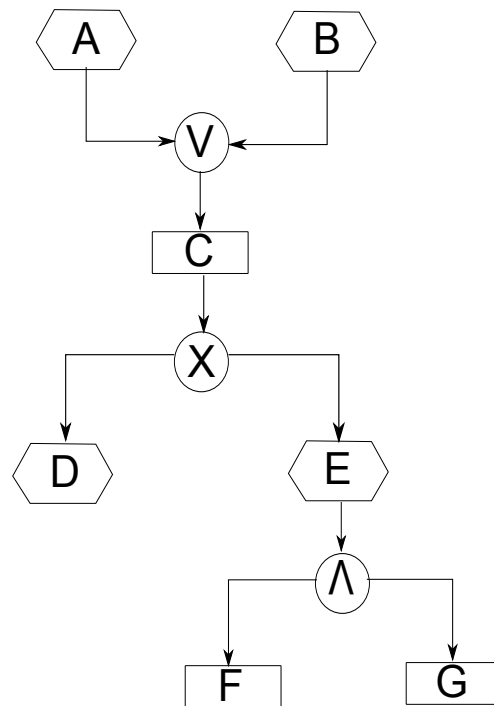


Abbildung 4.18: Ein EPK Modellbeispiel für die Coupling Metrik [Vanderfeesten 09]

In der Abbildung 4.18 sehen wir ein komplettes EPK Modell. Die Coupling Metrik der Abbildung 4.18 beträgt folgendes:

$$CP(Graph) = \frac{\frac{2}{3} + \frac{2}{3} + \frac{1}{2} + \frac{1}{2} + 1 + 1}{7 * 6} = 0,103$$

4.3.2 Cohesion (Zusammenhang) Metriken

Die Cohesion Metrik misst den Zusammenhang von Teilen innerhalb der Modelle. Der Wert der Metrik ist immer zwischen 0 und 1. Die Cohesion einer Aktivität ist das Produkt von Relation- und Information-Cohesion. Die Relations-Cohesion besagt inwieweit die verschiedenen Operationen innerhalb einer Aktivität miteinander in Verbindung stehen. Sie legt für jede Operation eine Aktivität fest, wieviele andere Operationen mit den Teilen von Inputs und Outputs überlappt werden.

Ein großer Vorteil von der Cohesion Metrik ist, dass kritische Daten herausgefunden werden können.

Nun gilt „strong cohesion, loose coupling“.

Die folgenden Metriken für die Cohesion stammen aus den Workflow Modellen.

4.3.2.1 Metrik: Zusammenhang der Aktivitätsbeziehungen (engl. Activity Relation Cohesion)

Diese Metrik erlaubt eine Beurteilung der Alternativen in einer Informationen-Element-Struktur. Erster Bestandteil davon ist die Relations-Cohesion [Reijers 04]. Diese besagt, wieviele verschiedene Operationen innerhalb einer Aktivität in Beziehung stehen. Um die relative Metrik zwischen 0 und 1 zu bekommen macht man folgendes: Den Durchschnitt an Überschneidungen geteilt durch maximale Anzahl an Überschneidungen, sprich die Anzahl der Operationen minus 1. In der Publikation von H. A. Reijers and I.T.P. Vanderfeesten [Reijers 04] sind ausführliche Erklärungen mit Beispielen vorhanden.

Formel: siehe [Reijers 04]

Korrelation: Der Wert dieser Metrik wird bei den nächsten Metriken genutzt, um am Ende ein Wert über die Zusammenhang innerhalb des Modells zu bekommen.

Quelle: siehe [Reijers 04]

Beispiel siehe [Reijers 04]

4.3.2.2 Metrik: Zusammenhang der Aktivitätsdaten (engl. Activity Information Cohesion)

Diese Metrik fokussiert sich auf alle Informationselemente, welche als Input oder Output bei jeder Operation innerhalb einer Aktivität genutzt werden [Reijers 04]. Sie besagt, wieviele Informationselemente genutzt sind, die mehr als einmal, in Proportion zu allen genutzten Informationselemente, eingesetzt sind. Alle verschiedenen Informationselemente werden gezählt, welche bei dem Schnittpunkt zweier Operationen erscheinen. Für nähere Informationen wird auf die Literatur [Reijers 04] verwiesen.

Formel: siehe [Reijers 04]

Korrelation: Der Wert dieser Metrik wird bei den nächsten Metriken genutzt, um am Ende ein Wert über die Zusammenhang innerhalb des Modells zu bekommen.

Quelle: siehe [Reijers 04]

Beispiel siehe [Reijers 04]

4.3.2.3 Metrik: Zusammenhang der Aktivitäten (engl. Activity Cohesion)

Die Gesamt-Cohesion einer Aktivität ist gegeben als Produkt von Relations- und Informations-Cohesion, welche in den vorherigen Metriken definiert wurden. Dies reflektiert, dass die Operationen zueinander in Verbindung stehen sollten, die Informationen jedoch geteilt werden sollten. Durch die zwei obigen Gleichungen, Zusammenhang der Aktivitätsbeziehungen $\lambda(t)$ und der Aktivitätsdaten $\mu(t)$ lässt sich dieser Metrik $c(t)$ bestimmen.

Formel:

$$c(t) = \lambda(t) * \mu(t)$$

Für ausführliche Erklärungen und Beispielen wird auf die Literatur [Reijers 04] verwiesen.

Korrelation: Der Wert dieser Metrik wird bei der nächsten Metrik genutzt, um am Ende ein Wert über die Zusammenhang innerhalb des Modells zu bekommen.

Quelle: siehe [Reijers 04]

Beispiel siehe [Reijers 04]

4.3.2.4 Metrik: Prozesszusammenhangsmetrik (engl. Process Cohesion)

Die Prozess-Cohesion kann als durchschnittliche Aktivitäts-Cohesion gelten. Mit den oben berechneten Werten kann man den endgültigen Zusammenhangsmetrik c der ganzen Prozess berechnen.

Formel:

$$c = \frac{\sum_{t \in T} c(t)}{|T|}$$

T ist die Menge aller gültigen Aktivitäten.

Um über diese Berechnung ein Entschluss ziehen zu können, wird noch eine Berechnung mit Prozesszusammenhangsmetrik c und Kopplungsmetrik 4.3.1.9 gemacht. Das Verhältnis

zwischen beiden Berechnungen gibt den Wert für die Qualität des Modells.

$$\rho = \frac{CP(Graph)}{c}$$

Korrelation: Je niedriger der Wert des Verhältnisses zwischen Prozesszusammenhangsmetrik und Kopplungsmetrik ist, desto hochqualitativ, besser ist das Modell.

Quelle: [[Reijers 04](#)]

Beispiel: Ein sehr ausführliches Beispiel zu den obengenannten Metriken ist in [[Reijers 04](#)] zu sehen.

4.4 Liste aller Metriken

Hier ist eine Liste aller Metriken zu sehen mit einer Literaturangabe zu jeweilige Metriken.

Nr.	Fehlermetrik	Quelle
4.1.1	Größe des Modells (Modelsize)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.2	Trennbarkeit (Separability)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.3	Sequenzialität (Sequentiality)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.4	Strukturiertheit (Structuredness)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.5	Zyklizität (Cyclicity)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.6	Parallelität (Token Split)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.7	Durchmesser (Diameter)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.8	Dichte (Density)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.9	Verbindungskoeffizient (CNC-Coefficient of Connectivity)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.10	Durchschnittsgrad der Verknüpfungen (Average degree of connectors)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.11	Maximaler Grad der Verknüpfungen (Maximum degree of a connector)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.12	Maximale Tiefe alle Knoten (Max depth of all nodes)	[Mendling 07] [Mendling 06] [Mendling+ 07]
4.1.13	Kontrollflusskomplexität (CFC–Control Flow Complexity)	[Mendling 07] [Mendling 06] [Mendling+ 07]

Tabelle 4.4: Fehlermetriken

Nr.	Komplexitätsmetrik	Quelle
4.2.1	Anzahl der Aktivitäten (Number of Activities)	[Gruhn 06]
4.2.2	Kontrollflusskomplexität (CFC - Control Flow Complexity)	[Gruhn 06] [Gruhn+ 06] [Cardoso 05]
4.2.3	Zyklomatische Komplexität (Cyclomatic Number)	[Gruhn 06] [Gruhn+ 06] [Cardoso 06] [McCabe 89]
4.2.4	Max. und durchschnittliche Verschachtelungstiefe (Maximum and mean nesting depth)	[Harrison 81] [Gruhn+ 06] [Gruhn 06]
4.2.5	Kognitive Wahrnehmung (Cognitive Weight)	[Gruhn 09] [Shao 03] [Cardoso 06] [Gruhn 09]
4.2.6	Prozessaufruf (Fan-in / Fan-out)	[Gruhn 06]
4.2.7	Umfangsmetriken (Structural Metrics)	[Meyer 06]
4.2.8	Kantenüberschneidungen (Edge Intersection)	[Meyer 06]
4.2.9	Schreibrichtung (Writing Direction)	[Meyer 06]
4.2.10	Halstead Metrik (Halstead Metric)	[Cardoso 06] [Thammarak 10] [Makni 10]
4.2.11	Schnittstelle ((IC – Interface)	[Cardoso 06]
4.2.12	Netzwerkkomplexitätskoeffizient (CNC – Coefficient of network complexity)	[Cardoso 06]

Tabelle 4.5: Komplexitätsmetriken

Nr.	Qualitätsmetrik	Quelle
4.3.1.1	Nachrichten-/ Sequenzflussmetrik (MSPC – Message or Sequence flow passing coupling)	[Khelif 10]
4.3.1.2	Prozessrückmeldung (RFP – Response for Process)	[Khelif 09] [Khelif 10]
4.3.1.3	Informationsflussdichte (IFCP - Information Flow Coupling Process)	[Khelif 10]
4.3.1.4	Datenlokalität (LDA – Locality of data activity)	[Khelif 10] [Khelif 09]
4.3.1.5	Eingehende-/ Ausgehendeflüsse (ICP/ECP – Imported/Exported Coupling of a Process)	[Khelif 10]
4.3.1.6	Aufgabenkopplung (TCpl – Task Coupling)	[Khelif 10]
4.3.1.7	Prozesskopplung (PCpl – Process Coupling)	[Khelif 10]
4.3.1.8	Daten-/ Kontrollflusskopplung (DFC – Data and Control Flow Coupling)	[Khelif 10]
4.3.1.9	Kopplungsmetrik (CP – Coupling)	[Vanderfeesten 09]
4.3.2.1	Zusammenhang der Aktivitätsbeziehungen (Activity Relation Cohesion)	[Reijers 04]
4.3.2.2	Zusammenhang der Aktivitätsdaten (Activity Information Cohesion)	[Reijers 04]
4.3.2.3	Zusammenhang der Aktivitäten (Activity Cohesion)	[Reijers 04]
4.3.2.4	Prozesszusammenhangsmetrik (Process Cohesion)	[Reijers 04]

Tabelle 4.6: Qualitätsmetriken

Kapitel 5

Modellkatalog

Der einzige Mensch, der sich
vernünftig benimmt, ist mein
Schneider. Er nimmt jedesmal neu
Maß, wenn er mich trifft, während
alle anderen immer die alten
Maßstäbe anlegen in der Meinung,
sie passten auch heute noch.

(George Bernard Shaw)

Das Ziel des Modellkatalogs ist es, eine Menge von BPMN Modellen mit Diagrammen oder XML Dateien zu sammeln. Es ist schwer an große Mengen von Modellen zu kommen.

So wurden mehrere BPMN Modelle mit oder ohne XML Dateien unter der Bachelorarbeit zusammengesammelt. Diese Modelle helfen dabei, dass die gefundenen Metriken getestet werden können. Die BPMN Modelle mit XML Codes helfen dabei, diese in dem programmierten Software einzuladen und das Programm zu testen, ob die programmierten Metriken richtig implementiert sind.

Es ist sehr hilfreich zu sehen, wie bestimmte Prozesse modelliert sind. Die Möglichkeit nach der Analyse der Modelle hilft dabei herauszufinden, wo die Lücken und Stärken der Modelle liegen.

Die Modelle in dem Modellkatalog stammen aus dem Netz, Büchern und wissenschaftlichen Arbeiten. Die Diagramme sind teilweise eingescannte Versionen aus den Büchern.

Unten sind zwei Beispiele aus dem Modellkatalog zu sehen. Zu den Diagrammen werden noch Abschnitte aus dem XML Code gezeigt. In der Abbildung 5.1 geht es um den Versand eines Geräts. In der Abbildung 5.2 geht es um das Buchen von Urlaub. Unter den beiden BPMN Diagrammen sind ihre jeweiligen XML Code Abschnitte zu sehen.

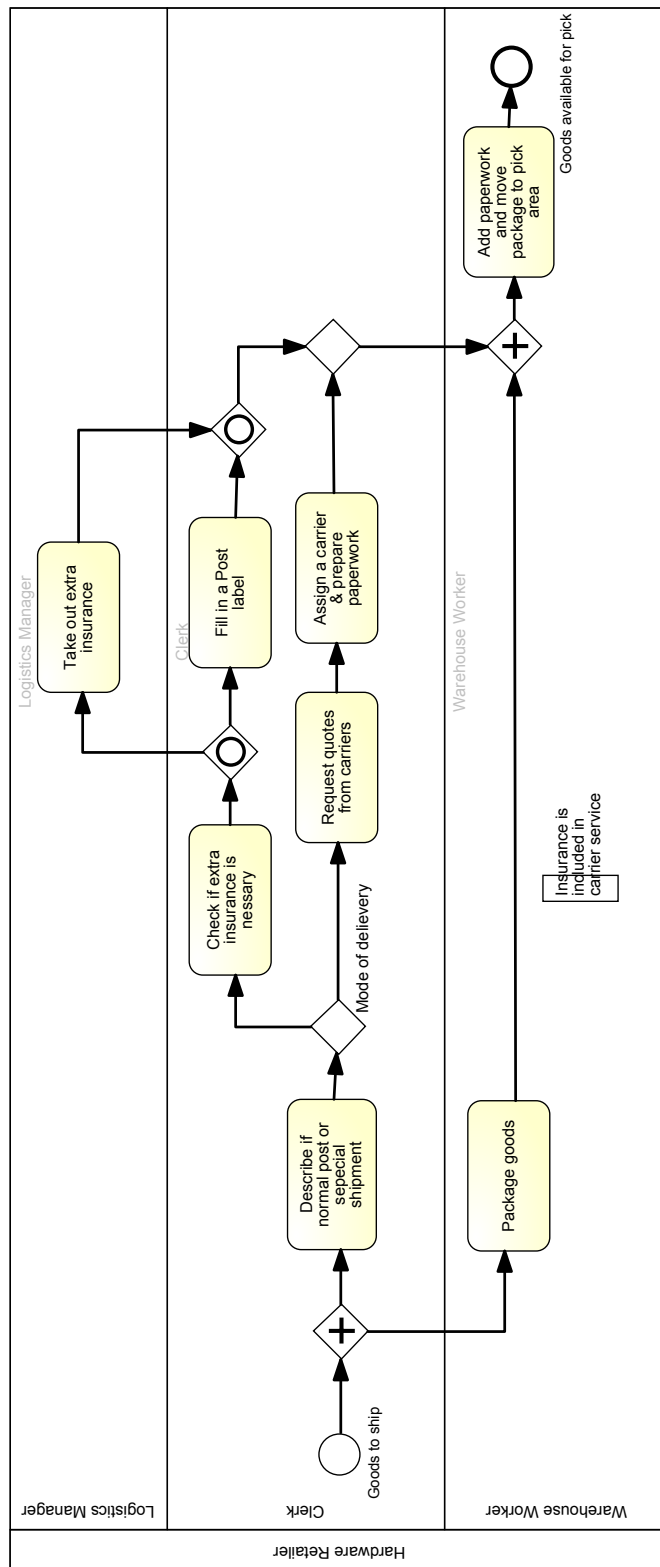
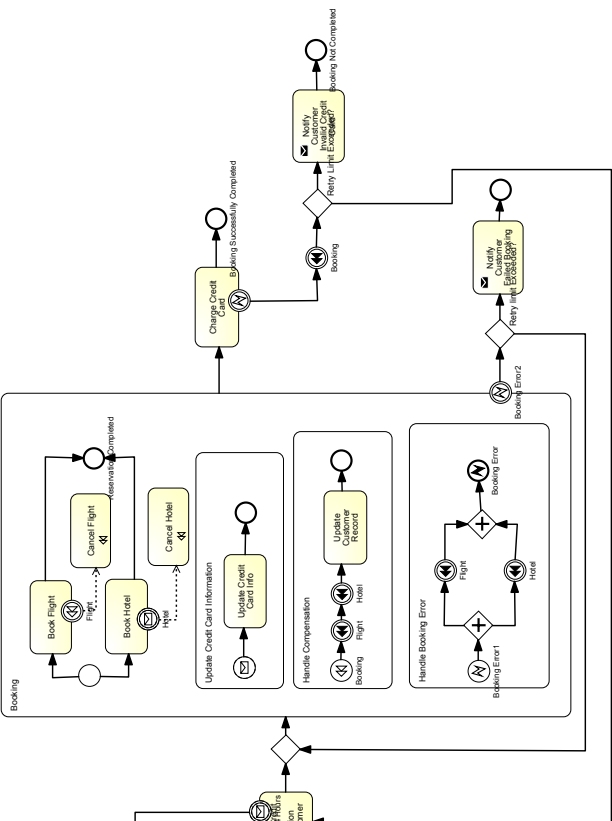


Abbildung 5.1: BPMN Diagramm über Versandprozess eines Gerätes

Listing 5.1 XML Code Abschnitt des BPMN Modells 5.1

```
1 <bpmn:process id="EAID_DP000000_4861_4fa7_958B_7B76F6DD02FF">
2   <bpmn:textAnnotation id="EAID_1569F293_2AE1_4005_A400_713B9F5C252D"
3     textFormat="text/plain">
4     <bpmn:text>Insurance is included in carrier service</bpmn:text>
5   </bpmn:textAnnotation>
6 </bpmn:process>
7 <bpmn:collaboration id="EAID_DC000000_4861_4fa7_958B_7B76F6DD02FF">
8   <bpmn:participant id="EAID_PR000000_4861_4fa7_958B_7B76F6DD02FF"
9     processRef="EAID_DP000000_4861_4fa7_958B_7B76F6DD02FF"/>
10  <bpmn:participant id="EAID_PR000000_530C_47bc_9B02_7021A040806C" name="Hardware
11    Retailer" processRef="EAID_BD779988_530C_47bc_9B02_7021A040806C"/>
12 </bpmn:collaboration>
13 <bpmn:process id="EAID_BD779988_530C_47bc_9B02_7021A040806C" name="Hardware Retailer">
14   <bpmn:laneSet id="EAID_LS000000_530C_47bc_9B02_7021A040806C">
15     <bpmn:lane id="EAID_95B7B659_EB77_49d9_9540_EB23FCDoF9B3" name="Clerk">
16       <bpmn:flowNodeRef>EAID_824F8C72_914A_4cc6_8546_651F8EAB8CC7</bpmn:flowNodeRef>
17       <bpmn:flowNodeRef>EAID_633D0647_3D5A_4151_BF19_C9FDBF482D56</bpmn:flowNodeRef>
18       <bpmn:flowNodeRef>EAID_F6AEB148_F84B_4fb1_A352_9253FE323E05</bpmn:flowNodeRef>
19       <bpmn:flowNodeRef>EAID_8603B768_F928_440b_89D3_094B4E343886</bpmn:flowNodeRef>
20     </bpmn:lane>
21     <bpmn:lane id="EAID_DC6DFBAA_4D56_406e_BAD7_1442D7D9DB8E" name="Logistics
22       Manager">
23       <bpmn:flowNodeRef>EAID_0740946F_4484_491c_B48E_912C339E3966</bpmn:flowNodeRef>
24     </bpmn:lane>
25     <bpmn:lane id="EAID_EDC5919A_2577_4011_AD9D_7B3D3Co87E8E" name="Warehouse
26       Worker">
27       <bpmn:flowNodeRef>EAID_6C0B4862_8710_44f7_8C23_52BB8926056A</bpmn:flowNodeRef>
28       <bpmn:flowNodeRef>EAID_6E1CBB69_F1D1_4bf4_AF15_C6B95D9D3F2A</bpmn:flowNodeRef>
29     </bpmn:lane>
30   </bpmn:laneSet>
31   <bpmn:parallelGateway id="EAID_6CoB4862_8710_44f7_8C23_52BB8926056A"
32     gatewayDirection="Unspecified">
33     <bpmn:incoming>EAID_E695EA42_DE02_4186_B7DE_2513E231EDD9</bpmn:incoming>
34     <bpmn:incoming>EAID_BCA03077_6FF1_4b92_8A93_256A65CE614C</bpmn:incoming>
35     <bpmn:outgoing>EAID_CEB14088_05A6_429c_8919_9272E4F009DF</bpmn:outgoing>
36   </bpmn:parallelGateway>
37   <bpmn:endEvent id="EAID_6E1CBB69_F1D1_4bf4_AF15_C6B95D9D3F2A" name="Goods
38     available for pick">
39     <bpmn:incoming>EAID_5FDC2D48_4A47_4019_AD51_B1C04A9E759E</bpmn:incoming>
40   </bpmn:endEvent>
41   <bpmn:task id="EAID_7889A18C_F418_41cf_B23F_37A96404FB04" name="Package goods"
42     startQuantity="1" completionQuantity="1" isForCompensation="false">
43     <bpmn:incoming>EAID_AF39E203_065C_49d6_AA68_74DBDA0E30A1</bpmn:incoming>
44     <bpmn:outgoing>EAID_E695EA42_DE02_4186_B7DE_2513E231EDD9</bpmn:outgoing>
45   </bpmn:task>
46   <bpmn:sequenceFlow id="EAID_E695EA42_DE02_4186_B7DE_2513E231EDD9"
47     sourceRef="EAID_7889A18C_F418_41cf_B23F_37A96404FB04"
48     targetRef="EAID_6CoB4862_8710_44f7_8C23_52BB8926056A"/>
49 </bpmn:process>
50 </bpmn:definitions>
```



Listing 5.2 XML Code Abschnitt des BPMN Modells 5.2

```
1
2 <bpmn:process id="EAID_DP000000_0518_43d7_8075_C818DC8BD838">
3   <bpmn:exclusiveGateway id="EAID_F508DD70_6701_4a3e_8E69_4D09504F4DA5" name="Retry
4     limit Exceeded?" gatewayDirection="Unspecified">
5     <bpmn:incoming>EAID_51D2BE56_634F_48ae_A705_E1584AE64266</bpmn:incoming>
6     <bpmn:outgoing>EAID_A633C195_7AD2_4625_92D9_92D588A159EC</bpmn:outgoing>
7     <bpmn:outgoing>EAID_4B42C6B5_6EF3_4396_9C6D_B3AF4FB62730</bpmn:outgoing>
8   </bpmn:exclusiveGateway>
9   <bpmn:exclusiveGateway id="EAID_E8F46A82_7645_4cco_846B_07C7A860F3CD" name="Retry
10     Limit Exceeded?" gatewayDirection="Unspecified">
11     <bpmn:incoming>EAID_A941270C_4153_40e0_80B6_59A3698CA314</bpmn:incoming>
12     <bpmn:outgoing>EAID_8B0DBFA2_BB6D_493f_B2AA_3DDF4EC68F18</bpmn:outgoing>
13     <bpmn:outgoing>EAID_697A3A7F_15DD_44ca_A338_ED714D0F662D</bpmn:outgoing>
14   </bpmn:exclusiveGateway>
15   <bpmn:exclusiveGateway id="EAID_5BCA345F_C1AC_469d_AEBD_0050C3E2BAAA"
16     gatewayDirection="Unspecified">
17     <bpmn:incoming>EAID_4B42C6B5_6EF3_4396_9C6D_B3AF4FB62730</bpmn:incoming>
18     <bpmn:incoming>EAID_3B9EFBE4_ABF4_4077_AAD5_60059D9C273A</bpmn:incoming>
19     <bpmn:outgoing>EAID_296914D3_38F2_43e3_80C6_4021343FA36E</bpmn:outgoing>
20   </bpmn:exclusiveGateway>
21   <bpmn:sequenceFlow id="EAID_4B42C6B5_6EF3_4396_9C6D_B3AF4FB62730"
22     sourceRef="EAID_F508DD70_6701_4a3e_8E69_4D09504F4DA5"
23     targetRef="EAID_5BCA345F_C1AC_469d_AEBD_0050C3E2BAAA"/>
24   <bpmn:exclusiveGateway id="EAID_979A3915_E2C6_4bc4_A325_5DE123AD9B7F"
25     gatewayDirection="Unspecified">
26     <bpmn:incoming>EAID_F25E737E_9661_4515_8FF2_A6C803712715</bpmn:incoming>
27     <bpmn:outgoing>EAID_A1039838_4B6A_41aa_96D7_4676C893315B</bpmn:outgoing>
28     <bpmn:outgoing>EAID_359D8181_F8CE_4042_A074_029E1D11E5E5</bpmn:outgoing>
29     <bpmn:outgoing>EAID_83C60FCA_332A_4530_A84F_C56A29B86FD3</bpmn:outgoing>
30   </bpmn:exclusiveGateway>
31   <bpmn:endEvent id="EAID_2DE57B7D_CBC5_4ace_8C57_F478D589D568" name="Request
32     Cancelled">
33     <bpmn:incoming>EAID_2777CD98_3F86_454c_BE12_07AC73B1C8CD</bpmn:incoming>
34   </bpmn:endEvent>
35   <bpmn:intermediateCatchEvent id="EAID_2D3818E9_9914_4fad_87D9_8735144EF097"
36     name="Customer make Selection" parallelMultiple="false">
37     <bpmn:incoming>EAID_83C60FCA_332A_4530_A84F_C56A29B86FD3</bpmn:incoming>
38     <bpmn:outgoing>EAID_E09947FF_E483_4cb4_BC69_ED4DF5F1937E</bpmn:outgoing>
39     <bpmn:messageEventDefinition/>
40   </bpmn:intermediateCatchEvent>
41   <bpmn:sequenceFlow id="EAID_83C60FCA_332A_4530_A84F_C56A29B86FD3"
42     sourceRef="EAID_979A3915_E2C6_4bc4_A325_5DE123AD9B7F"
43     targetRef="EAID_2D3818E9_9914_4fad_87D9_8735144EF097"/>
44   <bpmn:intermediateCatchEvent id="EAID_71AD8259_0A4A_4fd3_9417_0DEBCDA01A07"
45     name="Cancel Request" parallelMultiple="false">
46     <bpmn:incoming>EAID_359D8181_F8CE_4042_A074_029E1D11E5E5</bpmn:incoming>
47     <bpmn:outgoing>EAID_D22010B1_6CC6_458f_8A52_A22AF692467C</bpmn:outgoing>
48     <bpmn:messageEventDefinition/>
49   </bpmn:intermediateCatchEvent>
50 </bpmn:process>
```

Kapitel 6

Programmierung der Metriken

Measuring programming progress
by lines of code is like measuring
aircraft building progress by weight.

(Bill Gates)

Ein Teil der Bachelorarbeit ist die Implementierung der gefundenen Metriken. Es wurden in der Programmiersprache Java einige der Metriken implementiert, sodass man automatisch ein paar Werte der implementierten Metriken bekommen kann, nachdem man die XML Datei als Input im Programm angegeben hat.

Folgende Metriken werden implementiert:

- ◇ CFC Wert (siehe Abschnitt 4.2.2)
- ◇ Kognitive Wahrnehmung (siehe Abschnitt 4.2.5)
- ◇ Anzahl Aktivitäten (siehe Abschnitt 4.2.1)
- ◇ Anzahl Knoten (siehe Abschnitt 4.2.7)
- ◇ Anzahl Kanten (siehe Abschnitt 4.2.7)
- ◇ Dichte (siehe Abschnitt 4.1.8)
- ◇ CNC (Coefficient of Connectivity) (siehe Abschnitt 4.1.9)

6.1 Aufbau des Programms

Es wird in dem Programm zuerst die ganze XML Datei untersucht. In einer XML Datei von BPMN Diagrammen sind zwei Hauptteile vorhanden. "bpmn:process" und "bpmndi:BPMNDiagram". Während bei "bpmn:process" die Informationen über die Semantik und Teile der Diagramme stehen, stehen im Bereich von "bpmndi:BPMNDiagram" Informationen über die Position, Größe und den Abstand von Diagrammteilen. In diesem Sinne braucht man den Bereich "bpmndi:BPMNDiagram" nicht betrachten, da diese Informationen irrelevant sind. So werden in dem Programm nur die Bereiche von "bpmn:process" betrachtet.

Die Schnittstelle "DOM (Document Object Model)" wurde genutzt um die XML Datei einzulesen und zu analysieren. DOM repräsentiert die XML Datei als eine Baumstruktur und mit Hilfe dieser Repräsentation kann man genau auf die gewünschten Knoten zugreifen und die Informationen herausholen.

6.2 Implementierung von Metriken

Die Implementierung findet anhand der Untersuchung von Knoten, Kanten und Verknüpfungen statt.

6.2.1 Untersuchung von Verknüpfungen

Listing 6.1 Untersuchung von Verknüpfungen

```
1 if (node.getNodeName().endsWith("Gateway")) {  
2     anzahlVerknuepfungen++;  
3 }
```

In der XML Datei von BPMN Modellen ist leicht zu erkennen, was eine Verknüpfung ist. Die Verknüpfungen sind mit einem Gateway bestückt. Zum Beispiel "bpmn:exclusiveGateway". So betrachtet man alle Knotennamen mit einem "Gateway" am Ende ihres Namens und zählt sie zusammen. Die Summe ergibt die Anzahl der Verknüpfungen. Dies wird im Programm wie oben in 6.1 in Java realisiert.

6.2.2 Untersuchung von Kanten

Listing 6.2 Untersuchung von Kanten

```
1 if ((node.getNodeName() == "bpmn:outgoing")  
2     || (node.getNodeName() == "bpmn:messageFlow")  
3     || (node.getNodeName() == "bpmn:textAnnotation")) {  
4     kanten++;  
5 }
```

Die Untersuchung auf die Anzahl der Kanten beruht in unserem Programm auf die ausgehenden (outgoing) Kanten, Nachrichtenflüsse und Assoziationen. Da eine Kante gleichzeitig als "incoming" und "outgoing" gilt, muss man nur die eingehenden (incoming) oder ausgehenden Kanten berücksichtigen, so dass es nicht zu einem falschen Ergebnis kommt. Die Kanten in der XML Datei sind entweder als "bpmn:outgoing" oder als "bpmn:incoming" gekennzeichnet. Im Quellcode in 6.2 ist zu erkennen, dass die Summe der Kanten anhand der Anzahl der ausgehenden Kanten, Nachrichtenflüsse und Assoziationen berechnet wird.

6.2.3 Untersuchung von Aktivitäten

Listing 6.3 Untersuchung von Aktivitäten

```
1 if ((node.getNodeName() == "bpmn:task")
2     || (node.getNodeName() = "bpmn:subProcess")) {
3     anzahlAktivitaeten++;
4 }
```

Die Anzahl der Aktivitäten spielt eine Rolle bei der Berechnung der meisten Metriken. So ist es wichtig, dies in unserem Programm berücksichtigt zu haben. Die Aktivität erkennt man an dem Knotennamen "bpmn:task". So kann man alle Knoten mit diesem Namen zusammenzählen und die Summe der Aktivitäten erhalten. Im Quellcode 6.3 ist dies direkt zu erkennen.

6.2.4 Untersuchung von Knoten

Listing 6.4 Untersuchung von Knoten

```
1 if (node.getNodeName().endsWith("Gateway")
2     || (node.getNodeName() == "bpmn:task")
3     || (node.getNodeName().endsWith("Task"))
4     || (node.getNodeName() = "bpmn:subProcess")
5     || (node.getNodeName() == "bpmn:startEvent")
6     || (node.getNodeName() == "bpmn:endEvent")) {
7     knoten++;
8 }
```

Für die Berechnung mancher Metrikwerte muss man das BPMN Diagramm zuerst als einen Graph repräsentieren und anschließend die Knoten betrachten. In der BPMN gelten die Verknüpfungen, Start-, und Endevents genauso als ein Knoten wie die Aktivitäten. So müssen diese genau in Betracht gezogen werden. Im Quellcode 6.4 ist zu sehen, dass die Anzahl der Knoten aus diesen Bestandteilen besteht.

6.2.5 Untersuchung von XOR-Verknüpfungen

Listing 6.5 Untersuchung von XOR-Verknüpfung und Einfluss auf die Metriken

```
1 if (node.getNodeName() == "bpmn:exclusiveGateway") {
2     int Anzahl_XOR_Ausgaenge = 0;
3     AnzahlXOR++;
4     for (int a = 0; a < node.getChildNodes().getLength(); a++) {
5         if ((node.getChildNodes().item(a).getNodeName() == "bpmn:outgoing")) {
6             Anzahl_XOR_Ausgaenge++;
7         }
8     }
9 }
10 System.out.println("Anzahl Ausgaenge: " + Anzahl_XOR_Ausgaenge);
11 CFC = CFC + Anzahl_XOR_Ausgaenge;
12 if (Anzahl_XOR_Ausgaenge == 2) {
13     kognitiveWahrnehmung = kognitiveWahrnehmung + 2;
14 }
15 else if (Anzahl_XOR_Ausgaenge > 2) {
16     kognitiveWahrnehmung = kognitiveWahrnehmung + 3;
17 }
18 }
```

Im Quellcode 6.5 wird zuerst die Anzahl der XOR-Verknüpfung und anschließend die Zahl der Ausgänge ermittelt. Da die XML Datei als eine Baumstruktur repräsentiert wird, haben die Verknüpfungen auch noch Kinder mit Informationen von outgoing oder incoming Kanten. So müssen die Kinder der Knoten betrachtet werden, um Informationen über die Summe der ausgehenden Kanten von XOR-Verknüpfung zu bekommen. Dies ist oben in der for-Schleife zu sehen. Die Kinderknoten werden einzeln betrachtet und die ausgehenden Kanten werden zur Anzahl der ausgehenden Kanten hinzu addiert.

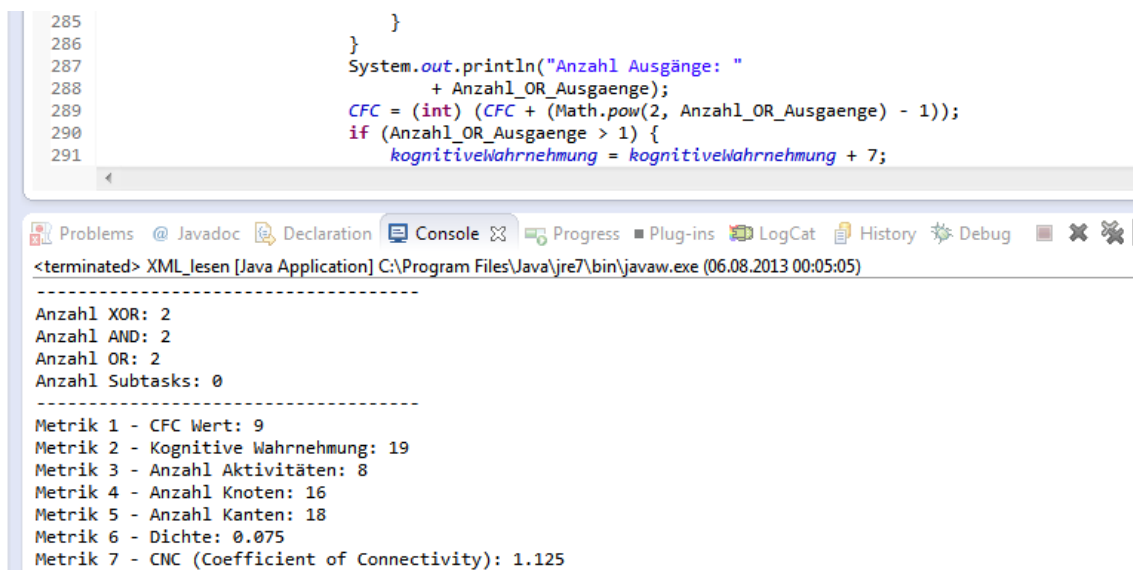
Da die Anzahl der ausgehenden Kanten eine Auswirkung auf die "kognitive Wahrnehmung" Metrik hat, wird es mit einer if-Verzweigung im Code realisiert. Ist die Anzahl der ausgehenden Kanten gleich zwei, so wird eine andere Berechnung gemacht. Ist die Anzahl der ausgehenden Kanten an XOR-Verknüpfung mehr als zwei, so wird auch hier eine andere Berechnung gemacht.

6.3 Programmtest

Hier werden die Testergebnisse von dem Programm beschrieben.

Die BPMN Modell in der Abbildung 5.1 wird anhand implementierten Metriken (Einführung Kapitel 6) getestet.

In dem Modell geht es um den Versand eines Gerätes. Die programmierte Software soll die Werte von sieben implementierten Metriken liefern. Unten ist ein Screenshot aus dem Programm, nach dem das Programm ausgeführt ist. Die Werte der Metriken entsprechen genau die Werte von Hand berechneten Metrikenwerte für die BPMN Modell 5.1.



```

285     }
286   }
287   System.out.println("Anzahl Ausgänge: "
288                       + Anzahl_OR_Ausgaenge);
289   CFC = (int) (CFC + (Math.pow(2, Anzahl_OR_Ausgaenge) - 1));
290   if (Anzahl_OR_Ausgaenge > 1) {
291       kognitivelWahrnehmung = kognitivelWahrnehmung + 7;
  
```

Problems @ Javadoc Declaration Console Progress Plug-ins LogCat History Debug

<terminated> XML_lesen [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (06.08.2013 00:05:05)

```

-----
Anzahl XOR: 2
Anzahl AND: 2
Anzahl OR: 2
Anzahl Subtasks: 0
-----
Metrik 1 - CFC Wert: 9
Metrik 2 - Kognitive Wahrnehmung: 19
Metrik 3 - Anzahl Aktivitäten: 8
Metrik 4 - Anzahl Knoten: 16
Metrik 5 - Anzahl Kanten: 18
Metrik 6 - Dichte: 0.075
Metrik 7 - CNC (Coefficient of Connectivity): 1.125
  
```

Abbildung 6.1: Ein Screenshot aus dem Programm mit Metrikwerten

Insgesamt wurden acht Modelle mit diesem Programm getestet. Die getesteten Modelle sind im Modellkatalog vorhanden. Die durchschnittliche Anzahl an Aktivitäten beträgt bei diesen Modellen 13, während die durchschnittliche Größe 22 ist. Die Größe beruht auf der Anzahl der Knoten. Die Wahrscheinlichkeit, dass die Menschen die größeren Modelle komplexer empfinden, ist hoch. Der Wert der Kognitiven Wahrnehmungsmetrik ist bei Modellen mit größerem Anzahl an Knoten viel höher, als bei kleineren Modellen.

Kapitel 7

Zusammenfassung und Fazit

Seien wir realistisch, versuchen wir
das Unmögliche.

(Che Guevara)

Im Rahmen dieser Arbeit wurde untersucht, welche Metriken für Prozessmodelle, mit besonderem Fokus auf BPMN vorhanden sind und wie diese anhand von BPMN Modellen getestet werden können. Die gefundene Metriken wurden analysiert sowie mit Beispielen aufgezeigt.

Zunächst wurden in Kapitel zwei die Grundlagen der Geschäftsprozessmodellierung erläutert. Danach folgt in Kapitel 3 die Beschreibung der Notation der meist genutzten Modellierungssprachen mit Beispielen. Der Fokus wurde dabei auf BPMN gelegt, da die BPMN laut einer Umfrage die meist genutzte Modellierungssprache in Deutschland ist und eine sehr gute Zukunft hat [Francko8].

Den Hauptteil dieser Arbeit bildet das Herausfinden sowie die Analyse von Metriken in Geschäftsprozessmodellen. Die Metriken wurden in drei Hauptkategorien unterteilt: Fehlermetriken, Komplexitätsmetriken und Qualitätsmetriken.

Die Fehlermetriken stammen hauptsächlich aus den Arbeiten von J. Mendling. Anhand Fehlermetriken wird herausgefunden, wie fehleranfällig ein Modell ist beziehungsweise sein kann. Hierzu gibt es mehrere strukturelle Metriken, welche die Bestandteile des Modells zählen. Darüber hinaus gibt es Metriken wie CFC Metrik (Kontrollflusskomplexität), die mittels Verknüpfungen im Modell ein Resultat über die Fehleranfälligkeit liefern. Die Metriken wurden mit Korrelationen beschrieben und mit Beispielen ausführlich erklärt.

Die Komplexitätsmetriken besagen, wie komplex ein Modell sein kann. Dies hilft dabei herauszufinden, ob das Modell einfacher dargestellt werden muss oder nicht. Einer der interessanten Metriken ist die Kognitive Wahrnehmungsmetrik, die besagt, wie die verschiedenen Teile des Modells von Menschen wahrgenommen werden. Und anhand dieser Wahrnehmung kann herausgestellt werden, ob das Modell einfacher dargestellt werden soll oder nicht.

Qualitätsmetriken beruhen sich sehr auf die Kopplung und Zusammenhang der verschiedenen Teile innerhalb des Modells. Hierzu spielt die Analyse der Kanten im Modell eine wichtige Rolle, da anhand der Kanten herausgefunden wird, wie stark der Zusammenhang oder die Kopplung innerhalb des Modells ist.

Ein anderer wichtiger Teil dieser Arbeit ist das Testen von gefundenen Metriken und die Erstellung eines Modellkatalogs mit verschiedenen BPMN Modellen und deren XML Codes. Mit Hilfe von gefundenen BPMN Modellen kann man die analysierten Metriken auf die Vollständigkeit prüfen. Hierzu wurden einige der Metriken in Programmiersprache JAVA implementiert. In Kapitel sechs werden die wichtigen Teile der Code beschrieben und detailliert erklärt. Es wurde beschrieben, wie die Verknüpfungen, Knoten und Kanten anhand des Programms untersucht werden. Anschließend wurden die Modelle mit dieser Software getestet. Schließlich folgt dann noch die Beschreibung der Ergebnisse.

Fazit

Die Metriken in Geschäftsprozessmodellen sind noch nicht ausgereift. Während im Softwarebereich viele und detaillierte Metriken gibt, ist dies in Geschäftsprozessmodellen nicht der Fall. Es existieren circa 50 Metriken für Prozessmodellen, aber die Metriken beruhen sich viel mehr auf die EPK Modelle und auch auf Modelle, welche leicht als Netzwerkgraphen dargestellt werden. Vor allem gibt es für die BPMN Modelle nicht viele aussagekräftige Metriken; außer Umfangsmetriken, welche die verschiedenen Bestandteile des Modells messen. Während in einem EPK Modell grundlegende Verknüpfungen existieren, gibt es in BPMN Modellen mehrere verschiedene Arten von Ereignissen. Anhand vorhandenen Metriken kann man zunächst noch keine Meinung über die Ereignisse bilden. Fragen wie "Werden die BPMN Modelle komplexer, wenn mehrere verschiedene Ereignisse vorhanden sind? Gibt es Ereignisse, welche das Modell verständlicher machen? Ist die Anzahl der Pools und Lanes ein aussagekräftiges Ergebnis, was für die Prozessexperten sehr behilflich sein kann?" sind nicht beantwortet worden und werden in Zukunft als ein Forschungsthema zu behandeln sein.

In dieser Arbeit wurden viele Metriken vorgestellt, welche für den Nutzer sehr behilflich sein können, um die Modelle verständlicher und einfacher darzustellen. Da die BPMN sich sehr verbreitet hat und weiterhin verbreitet, ist es wichtig Metriken in Bezug auf die BPMN Modellen zu entwickeln.

Danksagung

Danken möchte ich mich in erster Linie bei meinem Betreuer Felix Baumann für seine ausgiebige Unterstützung. Durch stetig kritisches Hinterfragen und konstruktive Kritik verhalf er mir zu einer durchdachten These und Fragestellung. Vielen Dank für die Zeit und Mühen, die Sie, Herr Baumann, in meine Arbeit investiert haben.

Auch möchte ich mich bei Kristoffer Kantschar und Bariya Sindhu bedanken, die viel Zeit in die Korrektur meiner Arbeit investiert haben. Zahlreiche Kommata, Satzstellungen und Rechtschreibfehler flogen dank ihrer Hilfe hinaus oder auch hinein. Ein besonderer Dank gilt Jonathan Schätz, der immer bereit war, mich zu motivieren und zu unterstützen.

Ganz besonders möchte ich mich bei meinen Eltern bedanken, die mich mit Geduld, Motivation und Vertrauen unterstützt haben. Meine Mutter Güllüzar, mein Vater Niyazi und meine Schwester Elif haben mich nicht nur während der Schulzeit, sondern auch während des Studiums so herzlich unterstützt. Dankeschön !

Literaturverzeichnis

- [BPM-CBOK 09] European Association of Business Process Management. *BPM Common Body of Knowledge – BPM-CBOK*. Dr. Götz Schmidt Verlag. Wettenberg, 2009. - ISBN 3921313805.
- [BPMO 11] Berliner BPM-Offensive. *BPMN 2.0 Poster*. URL http://www.bpmb.de/images/BPMN2_0_Poster_DE.pdf. 2011. Abrufdatum: 09.08.2013
- [Burlton 13] R. Burlton. BPTrends. *Business Process Manifesto!* URL <http://www.bptrends.com/bpmmmanifesto.cfm> - Abrufdatum: 09.08.2013.
- [Cardoso 05] J. Cardoso. *Control-flow Complexity Measurement of Processes and Weyuker's Properties*. In Proc. 6th International Enformatika Conference, Transactions on Enformatika, Systems Sciences and Engineering, Budapest. Publikaton, 2005.
- [Cardoso 06] J. Cardoso, J. Mendling, G. Neumann, and H.A. Reijers. *A Discourse on Complexity of Process Models - (Survey Paper)*. In Proc. Proceeding BPM'06 Proceedings of the 2006 international conference on Business Process Management Workshop, Wien. Publikation, 2006
- [Chen 94] R. Chen, A.-W. Scheer. *Modellierung von Prozessketten mittels Petri-Netz Theorie*. Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 107, Universität des Saarlandes, Saarbrücken, 1994.
- [Cormen 09] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C.Stein. *Introduction to Algorithms*. MIT Verlag, 3.Auflage. New York, 2009. - ISBN 0262533057
- [Decker 09] G. Decker, Willi Tscheschner, Jörg Puchan. *Migration von EPK zu BPMN*. GI-Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK), Berlin, 2009.
- [Diestel 10] R. Diestel. *Graphentheorie*. Springer Verlag, 4.Auflage. Hamburg, 2010. - ISBN 3642149111
- [Dumke 01] R. Dumke. URL <http://se.cs.uni-magdeburg.de/tutorial/Aktivittdtsdiagramm.htm>. Universität Magdeburg. Aktualisierungsdatum: 2001. Abrufdatum: 09.08.2013
- [EAP 12] (Sparx Systems) Enterprise Architect Program Version 10. *Shipment Process of a Hardware*. 2011.

- [Francko8] A. Franck, T.Henninger. Camunda services GmbH. *BPMN 2008*. Berlin, 2008.
- [Fuermann 02] T. Fuermann, C. Dammasch. *Prozessmanagement, Anleitung zur Steigerung der Wertschöpfung*. Carl Hanser Verlag, 2. Auflage, 2002. - ISBN 3446219161
- [Gadatsch 04] A. Gadatsch. *Grundkurs Geschäftsprozess-Management*. Vieweg+Teubner Verlag, 5. Auflage. Wiesbaden, 2004. - ISBN 3528257598
- [Gruhn 06] V. Gruhn, R. Laue. *Complexity Metrics for Business Process Models*. In 9th International Conference on Business Information Systems (BIS 2006), Klagenfurt, Austria. Publikation, 2006.
- [Gruhn+ 06] V. Gruhn, R. Laue, F. Meyer. *Berechnung von Komplexitätsmetriken für ereignisgesteuerte Prozessketten*. In Proc. 5. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK), Wien. Publikation, 2006.
- [Gruhn 09] V. Gruhn, R. Laue. *Adopting the Cognitive Complexity Measure for Business Process Models*. Leipzig. Publikation 2009.
- [Harrison 81] W. A. Harrison und K. I. Magel. *A complexity measure based on nesting level*. New York. Publikation, 1981.
- [Heravizadeh 08] M. Heravizadeh, J. Mendling, and M. Rosemann. *Dimensions of Business Processes Quality (QoBP)*. Queensland University of Technology, Brisbane. Publikation, 2008.
- [Jochim 01] K. Jochim. *Semantikbeschreibung mit Abläufen von Petri-Netzen*. Karlsruher Institut für Technologie - KIT, Fachbereich Informatik, Seminararbeit.2006
- [Khelif 09] W. Khelif, L. Makni, N. Zaaboub, H. Ben-Abdallah. *Quality metrics for business process modeling* Sfax University, Tunesien. Publikation, 2010.
- [Khelif 10] W. Khelif, N. Zaaboub, H. Ben-Abdallah. *Coupling metrics for business process modeling*. Sfax University, Tunesien, 2010.
- [Kocian 11] C. Kocian. *Geschäftsprozessmodellierung mit BPMN 2.0 – Business Process Model and Notation im Methodenvergleich*. Hochschule für angewandte Wissenschaften Neu-Ulm, Studienarbeit 2011.
- [Makni 10] L. Makni, W. Khelif, N. Z. Haddar, H. Ben-Abdallah. *A Tool for Evaluating the Quality of Business Process Models*. Sfax University, Tunesien. Publikation, 2010.
- [McCabe 89] T. J. McCabe, C.W. Butler: *Design complexity measurement and testing*. In: Communications of the ACM, (1989), New York. Publikation, 1989.

- [Mebrahtu 12] S. Mebrahtu. *Metrics for Updatable Process Views on Large Process Models*. Universität Ulm, Fachbereich Informatik, Bachelorarbeit. 2012.
- [Mendling 06] J. Mendling und G. Neumann. *Error Metrics for Business Process Models*. Wirtschaftsuniversität Wien. Publikation, 2006
- [Mendling 07] J. Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. Wirtschaftsuniversität Wien. Dissertation., 2007.
- [Mendling+ 07] Jan Mendling, G. Neumann, W.v.d.Aalst. *On the correlation between process model metrics and errors*. In Proc. Tutorials, posters, panels and industrial contributions at the 26th International Conference on Conceptual Modeling - ER 2007 Auckland, New Zealand. CRPIT, 83. Publikation, 2007.
- [Meyer 06] F. Meyer. *Untersuchung von Komplexitätsmetriken graphischer Geschäftsprozessmodelle*. Universität Leipzig, Fachbereich Informatik. Diplomarbeit, 2006.
- [Moch 13] S. Moch. Know How & Tools für Unternehmenserfolg, Business Process Management. URL <http://www.manager-wiki.com/index.php/unternehmensanalyse/84-business-process-management-bpm>. - Abrufdatum: 09.08.2013
- [OMG 10] OMG – Object Management Group. *Business Process Model and Notation (BPMN 2.0)*. URL <http://www.omg.org/spec/BPMN/2.0/Beta2/PDF/>. 2010. Abrufdatum: 09.08.2013
- [Petri 62] C. A. Petri. *Kommunikation mit Automaten*. Darmstadt, Technische Universität, Dissertation., 1962.
- [Reijers 04] H. A. Reijers and I.T.P. Vanderfeesten *Cohesion and Coupling Metrics for Workflow Process Design*. International Conference on Business Process Management BPM 2004, volume 3080 of Lecture Notes in Computer Science, Berlin. Publikation, 2004.
- [Reisig 10] W. Reisig. *Petrinetze – Modellierungstechnik, Analysemethoden, Fallstudien*. Vieweg+Teubner Verlag. 2010. - ISBN 3834812900
- [Sadiq 99] W. Sadiq and M. E. Orlowska. *Applying Graph Reduction Techniques for Identifying Structural Conflicts in Process Models*. The University of Queensland, Australien. Publikation, 1999.
- [Scheer 01] A.-W. Scheer. *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. 4. Auflage. Berlin, 2001. - ISBN 3540416013
- [Schmitzl 10] J. Schmitzl. *Durchgängige Modellierung von prozessorientierten Anwendungen mit BPMN 2.0*. Universität Ulm, Fachbereich Informatik, Masterarbeit. 2010

- [Sesselmann 10] W. Sesselmann, H.J. Schmerzel. *Geschäftsprozessmanagement in der Praxis: Kunden zufrieden stellen – Produktivität steigern – Wert erhöhen*. Hanser Wirtschaft Verlag, 7. Auflage, 2010. - ISBN 3446421858
- [Shao 03] J. Shao, Y. Wang. *A new measure of software complexity based on cognitive weights*. In Proc.: Canadian Journal of Electrical and Computer Engineering. Publikation, 2003.
- [Staehle 99] W. H. Staehle, P. Conrad & J. Sydow . *Management - Eine verhaltenswissenschaftliche Perspektive*. Vahlen Verlag, 8. Auflage. München, 1999. - ISBN 3800623447
- [Thammarak 10] K. Thammarak. *Survey Complexity Metrics for Reusable Business Process*. Bangkok Suvarnabhumi College, Bangkok. Publikation, 2010.
- [Turau 10] V. Turau. *Algorithmische Graphentheorie*. Oldenbourg Wissenschaftsverlag, 3. Auflage. Oldenburg, 2009. - ISBN 348659057X
- [Vanderfeesten 09] I. Vanderfeesten, J. Cardoso, H. A. Reijers *A weighted coupling metric for business process models*. CAiSE Forum, volume 247 of CEUR Workshop Proceedings. 2007.
- [White 04] S. A. White. *Introduction to BPMN*. IBM Corporation, URL http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf , 2004. Abrufdatum: 09.08.2013.
- [Wikimedia 11] Wikimedia. URL http://commons.wikimedia.org/wiki/File:Seasons_1.svg. Aktualisierungsdatum: 2011. Abrufdatum: 09.08.2013
- [Wikimedia 06] Wikimedia. URL <http://www.iwiki.de/wiki/index.php/Bild:Epksymbole.JPG>. Aktualisierungsdatum: 2006. Abrufdatum: 09.08.2013
- [Wikipedia 05] Wikipedia. URL http://de.wikipedia.org/wiki/Datei:Activity_diagram_2.png. Aktualisierungsdatum: 2005. Abrufdatum 09.08.2013.
- [Wikimedia 06] Wikimedia. URL <http://commons.wikimedia.org/wiki/File:Bpmn-8.png>. Aktualisierungsdatum: 2006. Abrufdatum: 09.08.2013

Akronyme und Abkürzungen

A	Kanten
BPI	Business Process Improvement
BPM	Business Process Management
BPMN	Business Process Model and Notation
C	Verknüpfungen
CFC	Control Flow Complexity
CNC	Coefficient of Connectivity
CP	Coupling
DFC	Data and Control Flow Coupling
DOM	Document Object Model
ECP	Exported Coupling of a Process
EPK	Ereignisgesteuerte Prozesskette
IBM	International Business Machines Corporation
ICP	Imported Coupling of a Process
IFCP	Information Flow Coupling Process
IT	Informationstechnik
J	Join-Verknüpfung
LDA	Locality of Data Activity
MSPC	Message or Sequence Flow Passing Coupling
N	Knoten
OMG	Object Management Group
PCpl	Process Coupling
RFP	Response For Process
S	Split-Verknüpfung
T	Task / Aufgabe
TCpl	Task Coupling
TQM	Total Quality Management
UML	Unified Modeling Language
XML	Extensible Markup Language
XOR	Exclusive OR

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift