

Vertikale Konsistenz von Geschäftsprozessmodellen

Konzept und prototypische Implementierung
eines Werkzeugs auf Basis von BPMN 2.0

Bachelorarbeit

vorgelegt von André Zensen

Angefertigt im Studiengang Bachelor of Science (B.S.) in
Wirtschaftsinformatik an der Fachhochschule Bielefeld,
Fachbereich Wirtschaft und Gesundheit

Sommersemester 2016

Erstprüfer und Betreuer:

Prof. Dr. Küster

Zweitprüfer:

Prof. Dr. Förster

Para mi Fiorella

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis	III
Tabellenverzeichnis.....	IV
Abkürzungsverzeichnis.....	V
Kurzfassung	1
1. Einführung	2
1.1 Motivation und Zielsetzung	2
1.2 Aufbau der Arbeit	3
2. Grundlagen und Begriffsdefinitionen	4
2.1 Geschäftsprozesse und BPMN 2.0.....	5
2.1.1 Geschäftsprozesse und ihre Bedeutung für Unternehmen	5
2.1.2 BPMN 2.0 und ausführbare Prozesse	6
2.2 Lebenszyklus des Geschäftsprozessmanagements.....	12
2.2.1 Beteiligte Akteure und verschiedene Sichten auf einen Prozess	15
2.2.2 Unterschiede zwischen Sichten und typische Veränderungen.....	16
2.3 Vertikale Konsistenz.....	19
2.3.1 Definition und Abgrenzung zu horizontaler Konsistenz.....	19
2.3.2 Verwandte Arbeiten	23
2.4 BPMN-Modellierungswerkzeug bpmn-js	29
2.4.1 Architektur	29
2.4.2 Verwendungsmöglichkeiten	30
3. Konzept.....	31
3.1 Architektur	31
3.1.1 Zusammenspiel der Komponenten.....	32
3.1.2 User Interfaces	33
3.1.3 Klassenmodell des Werkzeugs	34
3.2 Ausgewählte Use Cases	36
3.3 Anforderungen	37
3.4 Datenmodelle und Formate.....	38
3.4.1 Korrespondenzen	38
3.4.2 Assoziation.....	38
4. Prototypische Implementierung	39
4.1 Verwendete Technologien und Umsetzung der Architektur.....	39
4.1.1 Aufbau des Frontend.....	39
4.1.2 Aufbau des Backend	40
4.2 Arbeitsvorgehen mit dem Werkzeug	40
4.2.1 Ein Projekt einrichten und einen Prozess anlegen (Anforderung R1)	40

4.2.2	Klonen eines Modells und Initialisieren der Korrespondenzen (Anforderung R2)	41
4.2.3	Laden und Speichern von Modellen (Anforderung R1)	42
4.2.4	Anzeigen, Erstellen und Manipulieren von Korrespondenzen (Anforderung R3, R4)	43
4.3	Einfache vertikale Konsistenzprüfung	46
4.3.1	Visuelle Prüfung durch bestehende Korrespondenzen (Anforderung R5, R6)	46
4.3.2	Prüfung durch existierende Elemente	46
5.	Validierung des Prototyps durch Fallbeispiele	48
5.1	Vorgehensweise	48
5.2	Fallbeispiel Urlaubsantrag	48
5.2.1	Ausgangsbasis und Veränderungen	48
5.2.2	Arbeitsschritte mit dem Werkzeug	51
5.2.3	Validierung	51
5.3	Fallbeispiel ATM Machine	52
5.3.1	Ausgangsbasis und Veränderungen	52
5.3.2	Arbeitsschritte mit dem Werkzeug	59
5.3.3	Validierung	59
6.	Fazit und Ausblick	61
	Literaturverzeichnis.....	VI
	Anhang	VIII

Abbildungsverzeichnis

Abbildung 1: Kernelemente der BPMN 2.0 am Beispiel	10
Abbildung 2: Lebenszyklus des Geschäftsprozessmanagements	14
Abbildung 3: Rollen im Lebenszyklus des GPM.....	16
Abbildung 4: Beispiel einer Verfeinerung vom fachlichen zum technischen Modell	17
Abbildung 5: Menüs zur Modellierung in bpmn-js.....	30
Abbildung 6: Top-Level Architektur des Tools	32
Abbildung 7: Mockup des User Interface des Tools	33
Abbildung 8: Klassenmodell des Tools	34
Abbildung 9: Use Cases	36
Abbildung 10: Korrespondenz des Typs 1-n in JSON-Schreibweise	38
Abbildung 11: Eingabemaske für ein neues Projekt	40
Abbildung 12: Einen neuen Prozess zu einem Projekt hinzufügen.....	41
Abbildung 13: Klonen eines Modells.....	42
Abbildung 14: Eine farblich grün dargestellte 1-1 Korrespondenz.....	43
Abbildung 15: Informationen über eine Korrespondenz mit Fokussierungs-Funktion.....	44
Abbildung 16: Erweiterte Selektion einer bestehenden 1-1 Korrespondenz.....	44
Abbildung 17: Neu erstellte 1-n Korrespondenz.....	45
Abbildung 18: Highlighting der Elemente des Ausgangsmodells	46
Abbildung 19: Konsistenzprüfung auf Basis abgedeckter Elemente des Ausgangsmodells.....	46
Abbildung 20: Fachliches Ausgangsmodell des Prozesses "Urlaubsantrag"	49
Abbildung 21: Verfeinertes, technisches Modell des Urlaubsantragsprozesses	50
Abbildung 22: ATM Modell auf Business-Ebene.....	53
Abbildung 23: Technisches Modell des ATM Prozesses.....	54
Abbildung 24: Anzeige der Korrespondenzen im fachlichen Modell nach Verfeinerung des geklonten Modells.....	55
Abbildung 25: Anzeige der Elemente und ihrer Korrespondenzen im technischen Modell nach der Verfeinerung.....	56
Abbildung 26: Visueller Direktvergleich nach Verfeinerung und ohne Anpassung der Korrespondenzen.....	57
Abbildung 27. Visueller Direktvergleich nach möglicher Anpassung der Korrespondenzen	58

Tabellenverzeichnis

Tabelle 1: Abstraktionsebenen eines Prozessmodells differenziert nach Ziel, Gestaltungsebene und Detaillierungsgrad	18
Tabelle 2: Korrespondenztypen und ihre Beschreibung	22
Tabelle 3: Neue Kategorisierung von Veränderungen	28
Tabelle 4: Funktionale Anforderungen	37

Abkürzungsverzeichnis

(e)EPK	(erweiterte) Ereignisgesteuerte Prozess Ketten
AJAX	Asynchronous JavaScript and XML
ACID	Atomicity, Consistency, Isolation und Durability
BNB	Banco do Nordeste do Brasil S.A.
BPEL	Business Process Executin Language
BPMN	Business Process Modeling Notation
BPR	Business Process Reengineering
CSS	Cascade Style Sheet
XML	Extended Markup Language
GPM	Geschäftsprozessmanagement
GPO	Geschäftsprozessoptimierung
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
IT	Informationstechnologie
JSON	JavaScript Simple Object Notation
OMG	Object Modeling Group
PST	Process Structure Tree
SVG	Scalable Vector Graphic
SPM	Shared Process Model
SESE	Singe Entry Single Exit
SysML	Systems Modeling Language
UML	Unified Modeling Language
XAMPP	Apache, MariaDB, Perl und PHP

Kurzfassung

Geschäftsprozessmodelle verändern sich häufig und werden auf verschiedenen Ebenen betrachtet und bearbeitet. Verschiedene an der Modellierung beteiligte Akteure führen ihrer Sicht entsprechend im Prozess der Verfeinerung eines Geschäftsprozessmodells verschiedene Änderungen durch, die zu Inkonsistenzen führen können. Fachliche Modelle unterscheiden sich in Umfang und Darstellung der zu vollziehenden Schritte des Modells. Diese Arbeit beleuchtet die Hintergründe, die zu Inkonsistenzen zwischen Modellen verschiedener Abstraktionsebenen führen. Sie stellt ein prototypisch implementiertes, webbasiertes Werkzeug auf Basis von BPMN 2.0 vor, welches einen Beitrag zum Konsistenzmanagement leisten soll. Mit Hilfe einer grafischen Gegenüberstellung und basierend auf manuell erstellten Korrespondenzen zwischen den Elementen zweier Modelle auf verschiedenen Abstraktionsebenen können die vorgenommenen Veränderungen und entstandene Inkonsistenzen zwischen Modellen strukturiert nachvollzogen und beseitigt werden, um finanzielle Folgeschäden aufkommender Inkonsistenzen zu beseitigen.

1. Einführung

Dieses Kapitel führt in das Thema ein und erläutert dabei die Motivation und Zielsetzung der Arbeit. Es wird außerdem eine grobe Übersicht über den Aufbau der Arbeit gegeben.

1.1 Motivation und Zielsetzung

Geschäftsprozesse und eine Prozessorientierung sind in modern organisierten Unternehmen ein wichtiger Bestandteil zur Leistungserstellung und Erfüllung von Kundenwünschen. An der Modellierung von Geschäftsprozessen sind hierbei verschiedene fachlich und technisch orientierte Akteure beteiligt, die unterschiedliche Sichten auf einen Prozess einnehmen und dementsprechend bei der Modellierung andere Aspekte in den Vordergrund stellen.

Geschäftsprozessmodelle werden dabei auf mehreren Ebenen iterativ bearbeitet. Hierbei werden sie mehrfach Anpassungen und Änderungen unterzogen, um beispielsweise Durchlaufzeiten zu minimieren, oder auf neue Umweltbedingungen reagieren zu können. Sie werden in der Regel von einer fachlichen Sicht in eine technische Sicht überführt, um den Geschäftsprozess rechnergestützt für eine Simulation und etwaige Optimierungen zu verwenden, oder im Rahmen eines Workflowsystems rechnergestützt auszuführen. Die erstellten Modelle dienen somit nicht nur zur Dokumentation der im Unternehmen gelebten Prozesse.

Durch die Verfeinerung der Modelle, wie bei der Überführung eines Modells der fachlichen Ebene zur technischen Spezifikation und IT Ebene, kommt es durch typische Veränderungen zu Inkonsistenzen zwischen den Modellen der verschiedenen Abstraktionsebenen. Diese Inkonsistenzen können zu finanziellen Schäden für Unternehmen führen, wenn sie unentdeckt bleiben. Inkonsistenzen müssen daher möglichst vor der Verwendung der Modelle im operativen Geschäft identifiziert, dokumentiert und behoben werden.

Die Arbeit verfolgt zwei Ziele. Sie soll zunächst einen Kontext für vertikale Konsistenz schaffen. Sie soll dabei die Gründe aufzeigen, die zu vertikaler Inkonsistenz führen, welche Auswirkungen diese Inkonsistenzen haben können und wie ihnen entgegnet werden können. Das zweite Ziel ist es, ein unterstützendes Werkzeug zu entwickeln. Am Markt verfügbare Programme bieten keine oder nur rudimentäre Funktionen zum vertikalen Konsistenzmanagement von Geschäftsprozessmodellen auf Basis von BPMN 2.0 an. Diese Arbeit stellt vor dem Hintergrund des Geschäftsprozessmanagements und bisheriger Arbeiten und Ansätze zur vertikalen Konsistenz von Geschäftsprozessmodellen hierzu ein Konzept und die Implementierung eines webbasierten Werkzeuges vor.

Das Werkzeug, welches als Grundlage auf der Modellierungssprache BPMN 2.0 aufbaut, soll dabei helfen, Änderungen und Inkonsistenzen zu verstehen, aufzudecken und schnell beseitigen zu können. Hierzu werden Korrespondenzen zwischen den Modellelementen auf verschiedenen Abstraktionsebenen erstellt und verwaltet, die eine einfache vertikale Konsistenzprüfung ermöglichen. Eine visuelle Gegenüberstellung soll außerdem als Diskussionsgrundlage für die verschiedenen Prozessbeteiligten und Modellierer dienen.

1.2 Aufbau der Arbeit

Die Arbeit besteht im Kern aus zwei Teilen. Der erste Teil schafft durch Kapitel 2 die Grundlagen und den Kontext für das Thema vertikale Konsistenz. Es wird auf die Bedeutung von Geschäftsprozessen für Unternehmen eingegangen, die Modellierungssprache BPMN 2.0 vorgestellt und im Rahmen des Lebenszyklus des Geschäftsprozessmanagements eine Grundlage für das Thema geschaffen. Vertikale Konsistenz wird definiert und das für das Konzept verwendete Verständnis und Korrespondenzen als Basis einer vertikalen Konsistenzprüfung erläutert. Verwandte Arbeiten und in ihnen vorgestellte Ansätze und Methoden zu vertikaler Konsistenz werden vorgestellt. Außerdem wird auf das im Prototyp verwendete Modellierungswerkzeug bpmn-js eingegangen.

Der zweite Teil behandelt das Konzept, die prototypische Implementierung und Validierung des entwickelten Werkzeugs. In Kapitel 3 wird die konzeptionelle Architektur des entwickelten Werkzeugs vorgestellt und durch ausgewählte Use Cases, erfasste Anforderungen und verwendete Datenmodelle erweitert.

Kapitel 4 widmet sich der prototypischen Implementierung und Umsetzung des Konzepts. Die verwendeten Technologien und die Umsetzung des Konzepts stehen im Vordergrund. Das Arbeitsvorgehen mit dem Werkzeug wird gezeigt und veranschaulicht die im Konzept vorgestellte einfache vertikale Konsistenzprüfung.

In Kapitel 5 wird der Prototyp mit Hilfe von zwei unterschiedlich umfangreichen und komplexen Beispielen validiert. Hierzu werden fachliche Modelle zu technischen Modellen verfeinert. Nach der Verfeinerung wird der Einsatz und Beitrag des Werkzeugs zur vertikalen Konsistenz aufgezeigt.

Kapitel 6 schließt als letztes Kapitel mit einem Fazit über das prototypische Werkzeug und gibt einen Ausblick durch weiterführende Gedanken zum vorgestellten Konzept.

2. Grundlagen und Begriffsdefinitionen

In diesem Kapitel werden die Grundlagen geschaffen, vor dessen Hintergrund das Konzept und die prototypische Implementierung des vorgestellten Werkzeuges entwickelt werden.

Zunächst wird in Kapitel 2.1 geklärt, was Geschäftsprozesse sind, welche Ausprägungen auftreten können und welche Bedeutung sie für Unternehmen haben.

Übergehend von den Geschäftsprozessen werden die Modellierungssprache "Business Process Modeling Notation" (BPMN) und Kernelemente der Sprache vorgestellt, mit derer Geschäftsprozesse auf fachlicher und technischer Ebene modelliert werden können. Es wird kurz die Herkunfts- und Entstehungsgeschichte beleuchtet, die Anwendungsmöglichkeiten im Rahmen von Workflows und ausführbarer Prozessmodelle betrachtet.

Kapitel 2.2 behandelt typische Veränderungen von Prozessmodellen im Rahmen des Geschäftsprozessmanagements und stellt ein häufig verwendetes Lebenszyklusmodell für Prozesse vor. Es werden beteiligte Akteure des Geschäftsprozessmanagements (GPM) und ihre verschiedenen Sichten auf Prozesse und Prozessmodelle, sowie die daraus resultierenden Veränderungen während Modellverfeinerungen näher betrachtet.

In Kapitel 2.3 werden die Veränderungen, die während der Evolution von Prozessmodellen vorgenommen werden, unter dem Aspekt daraus resultierender Inkonsistenzen zwischen verschiedenen Sichten auf einen Prozess und verschiedener Abstraktionsebenen eines Prozesses betrachtet.

Es wird die vertikale Konsistenz von der horizontalen Konsistenz abgegrenzt, die Bedeutung vertikaler Konsistenz aufgezeigt und Korrespondenzen als Ausgangsbasis einer einfachen Prüfung vertikaler Konsistenz vorgestellt. Daneben werden Möglichkeiten zur Berechnung von Korrespondenzen beschrieben.

Weiterführend werden verwandte Arbeiten vorgestellt, die sich mit dem Thema der vertikalen Konsistenz und insbesondere der Berechnung von Korrespondenzen befassen. Diese Arbeiten dienen in Teilen als Basis für das vorgestellte Konzept und prototypisch implementierte Werkzeug.

Abschließend wird in Kapitel 2.4 das im Prototyp verwendete auf JavaScript basierende BPMN 2.0 Modellierungswerkzeug "bpmn-js" aus dem Camunda-Umfeld vorgestellt. Es werden die Architektur und Verwendungsmöglichkeiten des webbasierten Open-Source Werkzeugs beschrieben.

2.1 Geschäftsprozesse und BPMN 2.0

Folgend sollen die Fragen geklärt werden, was Geschäftsprozesse sind, welche Bedeutung sie für Unternehmen haben und woher die Modellierungssprache BPMN 2.0 stammt, was ihre Kernelemente sind und welche Rolle sie im Kontext von ausführbaren Prozessen spielt.

2.1.1 Geschäftsprozesse und ihre Bedeutung für Unternehmen

Geschäftsprozesse sind modellhafte Beschreibungen von Unternehmensfunktionen, die zeitlich und inhaltlich voneinander abhängig sind (vgl. [1], S. 238 ff.). Sie können als funktions- und organisationsübergreifende Kette oder Reihenfolge von Aktivitäten verstanden werden, welche Geschäftsregeln folgen und von Mitarbeitern gelebt werden: "The supply chain process at any large company is complex and could easily be subdivided into subprocesses that contain hundreds of activities and thousands of business rules and are implemented by employees located throughout the world." (s. [2], S. 185 f.). Ein Prozess kann Subprozesse enthalten, die beispielsweise wiederkehrende Aufgaben abbilden.

Die Aktivitäten eines Prozesses sind inputabhängig von der vorgelagerten Aktivität und können durch Ereignisse verknüpft sein. Jede Aktivität und ihre Erfüllung "lebt von den als Input empfangenen Informationen und produziert neue Informationen (Output), die wiederum für die nachgelagerten Aufgabenträger als Eingangsinformationen dienen" (s. [3], S. 84 f.).

Ereignisse können von außen auftreten, oder durch den Prozess bzw. durch eine Aktivität in diesem ausgelöst werden. Ein Ereignis kann zum Beispiel der Auftragseingang eines Kunden sein und Aktivitäten die Aufgaben, die zu bearbeiten sind, um den Auftrag für den Kunden befriedigend zu erfüllen. Es kann auch ein auftretender Fehler sein, der dazu führt, dass eine bestimmte Aktivität ausgeführt wird.

Der Fokus eines Geschäftsprozesses liegt auf der Wertschöpfung für einen unternehmensinternen oder einen externen Kunden (vgl. [4], S. 57 ff.). Ein Prozess konsumiert und produziert in diesem Zusammenhang Leistungen, wobei der Prozessoutput in der Regel einen Wert für den Kunden darstellt.

Most of today's service processes are more likely to take information and modify it to generate new data, recommendations, or a printed document. Some prefer to think of this by speaking of creating value. (s. [2], S. 185)

Geschäftsprozesse stellen eine "spezielle Form der Ablauforganisation" dar, konkretisieren hierdurch eine Geschäftsstrategie und können als "Bindeglied zwischen Unternehmensstrategie und der Systementwicklung bzw. den unterstützenden Informationssystemen" betrachtet werden (vgl. [5], S. 35 ff.).

Ein Geschäftsprozess kann aus verschiedenen Sichten als Modell betrachtet und beschrieben werden, was dazu führt, dass der Detaillierungsgrad einer Sicht sehr unterschiedlich gegenüber einer anderen Sicht eines weiteren Prozessbeteiligten oder Stakeholders ausfallen kann. Eine fachliche Sicht ist in der Regel weniger detailreich, als ein konkretisierter und für das operative Geschäft ausgestalteter Prozess, der durch IT unterstützt ausführbar ist und um entsprechende Elemente angereichert wird.

Ausprägungen von Geschäftsprozessen

Geschäftsprozesse können nach Abstraktionsebene und Aufgabe differenziert werden (vgl. [5], S. 38 f.). Steuerungsprozesse dienen der übergreifenden und integrierten Koordination aller Prozesse und sind somit Bindeglied zwischen wertschöpfenden und unterstützenden Prozessen.

Kerngeschäftsprozesse zeichnen sich durch einen hohen Wertschöpfungsanteil aus und sind in der Regel für die Wettbewerbsfähigkeit eines Unternehmens relevant. Sie bilden die Leistungserstellung vom Kundenwunsch bis zur Erfüllung dieses Wunsches ab.

Unterstützungsprozesse weisen einen geringen Wertschöpfungsanteil auf und sind selten wettbewerbskritisch. Unterstützende Aktivitäten umfassen beispielsweise administrative Aufgaben, Buchhaltung, oder auch Berichtswesen (vgl. [1], S. 242 ff.; [4], S. 65 ff.).

Bedeutung von Prozessen im Unternehmen

In der Vergangenheit war die Organisation von Unternehmen stark von Taylorismus geprägt. Aufgaben wurden in möglichst kleine Aufgaben zerteilt, um eine Spezialisierung und Erhöhung der Leistungsfähigkeit des Einzelnen zu bewirken (vgl. [1], S. 240 ff.).

Diese Spezialisierung führt zu Abteilungsgrenzen und typischen Abteilungen wie Einkauf, Produktion, Vertrieb, oder auch IT. In einer funktional-orientierten Organisation stehen lange, hierarchische Abstimmungswege einer zunehmend globalisierten und dynamischen Umwelt und damit erforderter Flexibilität entgegen. Übergreifende Abläufe werden durch Abteilungsgrenzen behindert oder unterbrochen. Um diesen Problemen entgegenzuwirken, fand eine Entwicklung hin zu einer prozessorientierten Organisation statt, die von IT unterstützt und automatisiert wird (vgl. [3], S. 84; [6] S. 1 ff.).

Modernen Enterprise-Resource-Planning-Systemen liegt beispielsweise eine Prozessphilosophie zu Grunde, um die Integration und die abgestimmte Zusammenarbeit verschiedener Abteilungen oder Unternehmen zu verbessern und zu gewährleisten.

2.1.2 BPMN 2.0 und ausführbare Prozesse

Geschäftsprozesse müssen handhabbar gemacht werden, beispielsweise durch graphische Darstellungen, um als Diskussionsgrundlage zu dienen, oder formalisierte Modelle, die für Simulationen der Prozesse durch Softwareprogramme verwendet werden können. Insbesondere ist ein Formalismus nötig, um ausführbare Prozesse modellieren und in einem Workflow-System ausführen zu können. Die Business Process Modeling Notation (BPMN) (vgl. [7]; [8], S. 21 ff.) vereint hierzu grafische und fachliche, sowie formalisierte und technische Aspekte in der aktuellen Version 2.0 und ist mehr, als lediglich eine Notation und Palette grafischer Elemente. Sie beinhaltet tiefergehende "Vorgaben zur Bedeutung und Verwendung der Zeichen und Symbole" (s. [9], S. 504 ff.), die sich in Syntax und Semantik der Modellierungssprache ausdrücken.

Herkunft und Entwicklungsziel

BPMN wurde zunächst 2002 von IBM zur übersichtlichen grafischen Darstellung von Geschäftsprozessen entwickelt: "Es handelt sich dabei um einen umfassenden Ansatz zur Beschreibung, Analyse, Durchführung, Verwaltung und Verbesserung von Geschäftsprozessen in Großunternehmen" (s. [3], S. 109 ff.; [5], S. 85 ff.).

Ein Ziel der Sprache ist die Definition einer Notation, die für alle an automatisierten Geschäftsprozessen beteiligten Akteure verständlich ist. Vor dem Aufkommen von BPMN gab es bereits verschiedene Modellierungsmöglichkeiten, die ihren Ursprung oft in der Informatik hatten, wie Petri-Netze, Datenflussdiagramme oder Ablaufdiagramme. Diese haben sich als nur bedingt geeignet zur Abbildung betrieblicher Prozesse herausgestellt.

Gemeinsam ist den meisten etablierten GPM-Notationen und –Systemen das Token-basierte und von Petri-Netzen abgeleitete Grundkonzept nebenläufiger, verteilter und paralleler Abläufe. BPMN teilt diesen Ursprung (s. [10], S. 191 ff.), ist aber im Gegensatz zu Petri-Netzen nicht Datenflussorientiert (vgl. [5], S. 64 ff.).

Als besser geeignete Möglichkeiten der Modellierung "haben sich Vorgangskettendiagramme, UML bzw. SysML Aktivitätsdiagramme, die Business Process Modeling Notation und ereignisgesteuerte Prozessketten durchgesetzt" (vgl. [3], S. 105 f.). Auf eine Vorstellung von UML basierten Notationen wird in dieser Arbeit verzichtet, dafür aber im Kontext zu BPMN auf ereignisgesteuerte Prozessketten eingegangen, da diese eine in der Industrie weit verbreitete Alternative zu BPMN darstellt und viele Ähnlichkeiten zu ihr aufweist.

Ereignisgesteuerte-Prozess-Ketten (EPK) wurden 1992 durch eine Arbeitsgruppe um August-Wilhelm Scheer (Software AG) zusammen mit dem Unternehmen SAP entwickelt. Sie ist der BPMN ähnlich ([5], S. 64 ff.; [10], S. 196): Sie ist wie BPMN eine diagrammbasierte Methode, kontrollfluss- und Token-basiert und besteht ähnlich wie BPMN aus Funktionen, Ereignissen und Kontrollflüssen. Funktionen sind hierbei aktive Komponenten und stellen Aufgaben dar. Sie entsprechen den BPMN Activities. Ereignisse sind Zustandsbeschreibung eines Informationssystems und Auslöser und Ergebnis von Funktionen. Ein Prozess wird durch mindestens ein Ereignis begonnen und endet in mindestens einem Ereignis. Kontrollflüsse zwischen Funktionen und Ereignissen sind durch logische Operatoren strukturiert. So kann ein Kontrollfluss durch Entscheidungen geteilt und zusammengeführt werden. Erweiterungen der EPK (dann als eEPK, "erweiterte EPK" bezeichnet) dienen der Abbildung von Organisationseinheiten, In- und Outputdaten und Verweisen auf andere Prozesse. In BPMN werden Organisationseinheiten und Prozessbeteiligte durch Pools und Lanes dargestellt. Durch Annotation von Funktionen mit für Controlling relevanten Informationen wie Bearbeitungszeit und Kostensätzen wird eine rechnergestützte Simulation möglich ([3], S. 105 f.).

Während in Deutschland vor allem in der Industrie die EPK Verwendung findet, hat sich die BPMN bisher besonders in der Schweiz bei Unternehmen und öffentlicher Verwaltung als Wahl zur Modellie-

rung von Geschäftsprozessen unter Berücksichtigung fachlicher und technischer Sichten durchgesetzt (s. [5], S 85 ff.).

Die Standardisierung von BPMN und Weiterentwicklung hin zur Version 2.0 im Jahre 2010 mit einer erweiterten Elementpalette erfolgte unter der Schirmherrschaft der Object Modeling Group (OMG) seit dem Jahre 2005. Die Standardisierung erlaubt und erleichtert den Austausch zwischen unterschiedlichen Modellierungswerkzeugen. Die Version 2.0 berücksichtigt dabei explizit die Ausführbarkeit der Modelle durch Transformation fachlicher in technische Prozessmodelle durch entsprechende Elemente der erweiterten Palette ([9], S. 504 ff.). Die Notation basiert hierbei auf einem Format in der Extended Markup Language (XML), welches zweigeteilt zu betrachten ist. Einerseits enthält ein BPMN im XML-Format die grafischen Definitionen des Prozesses und der enthaltenen Elemente, andererseits die Definitionen zur Ausführung und technische Details.

Die Ausführung von Prozessen konnte bisher beispielsweise durch die "Business Process Execution Language" (BPEL) erreicht werden und kann als XML-Markup definiert werden (vgl. [8], S. 231 ff.). Hierbei werden Funktionen in Prozessen durch aufgerufene Webservices ausgeführt. BPEL dient der ausschließlichen Kommunikation mit Webservices und sieht keine grafische Notation wie die BPMN vor ([3], S. 110 f.). Hier zeigt sich der Vorzug einer Unterstützung der Modellierung von fachlicher und technischer Sicht, wie sie in BPMN 2.0 zu finden ist, besonders in Hinsicht auf die Abbildung von grafischen Elementen ausführbarer Elemente, oder auch Roundtrip-Engineering.

Kernelemente der BPMN

BPMN ist EPK ähnlich¹, verzichtet aber auf Pflichtevents nach Aktivitäten und kennt verschiedene Ereignistypen. Die Kernelemente², aus denen sich die BPMN aufbaut, bestehen aus Objekten, die vier Gruppen zugeordnet werden können: *Flow Objects*, *Connecting Objects*, *Swimlanes* und *Artifacts* (vgl. [3], S. 109 ff.; [9], S. 504 ff.). Die von der OMG herausgegebene Spezifikation sieht außerdem noch eine Kategorie *Data* vor, welche Daten In- und Outputs umfasst (s. [7], S. 27 ff.). Abbildung 1 zeigt beispielhafte, beschriftete Elemente der verschiedenen Gruppen.

Flow Objects umfassen Activities, Gateways und Events. Activities sind (bis auf Subprozesse) atomare Funktionen bzw. Arbeiten, die entweder von einem Menschen oder Computer ausgeführt werden. Sie sind grafisch durch Rechtecke mit abgerundeten Ecken dargestellt und können weiter typisiert werden, beispielsweise in *Script* oder *Service Tasks*, um Ausführung durch IT hervorzuheben (vgl. [7], S. 156 ff., S. 385 f., S. 426 ff.). Gateways dienen der Strukturierung der Ablaufreihenfolge durch Bedingungen. Sie sind durch diamantförmige Symbole dargestellt und umfassen das ex- und inklusive Oder, sowie parallele Verzweigungen. Events sind Ereignisse, die eintreten können, oder ausgelöst werden. Eine eingehende oder ausgehende Nachricht stellt beispielsweise ein Ereignis dar. Start- und Endpunkte eines Prozesses sind Ereignisse. Sie können außerdem auch an Activities angehängt werden, um beispielsweise Sequenzpfade zur Fehlerbehandlung darzustellen.

¹ Eine Darstellung wesentlicher Elemente und ein Methodenvergleich finden sich in [5], S. 88 und S. 100.

² Alle BPMN 2.0 Elemente sind in [7] ausführlich dargestellt. Eine Übersicht bieten [8], S. 21 ff. und [2], S. 465 ff.

Connecting Objects sind die Kanten im Modell, die alle Elemente miteinander verbinden und umfassen Sequence Flows, sowie Message Flows. Sequence Flows verbinden *Flow Objects* innerhalb eines Pools. Message Flows sind temporärer Natur und verbinden typischerweise pool-übergreifend *Flow Objects*. Sie dienen zur Darstellung von Kommunikation zwischen Zuständigkeitsbereichen und Prozessbeteiligten, wie beispielsweise zwei Unternehmen, oder die Kommunikation zwischen Kunde und Unternehmen. Neben Sequence und Message Flows gibt es noch Associations, die Metainformationen über *Flow Objects* beinhalten können.

Swimlanes stellen Zuständigkeitsbereiche dar und umfassen Pools und Lanes. Lanes bilden beispielsweise Funktionsbereiche ab und gruppieren diese und können einen Pool strukturieren. Ein Pool ist ein in sich abgeschlossener Prozess. Eine Kommunikation zwischen in sich abgeschlossenen Prozessen, oder Pools, erfolgt nur über Message Flows.

Artifacts umfassen Datenobjekte und Annotationen. Datenobjekte beschreiben In- und Output von Activities, wie zum Beispiel benötigte oder erzeugte Dokumente, während Annotationen ähnlich der Associations Metainformationen und Kommentare sind.

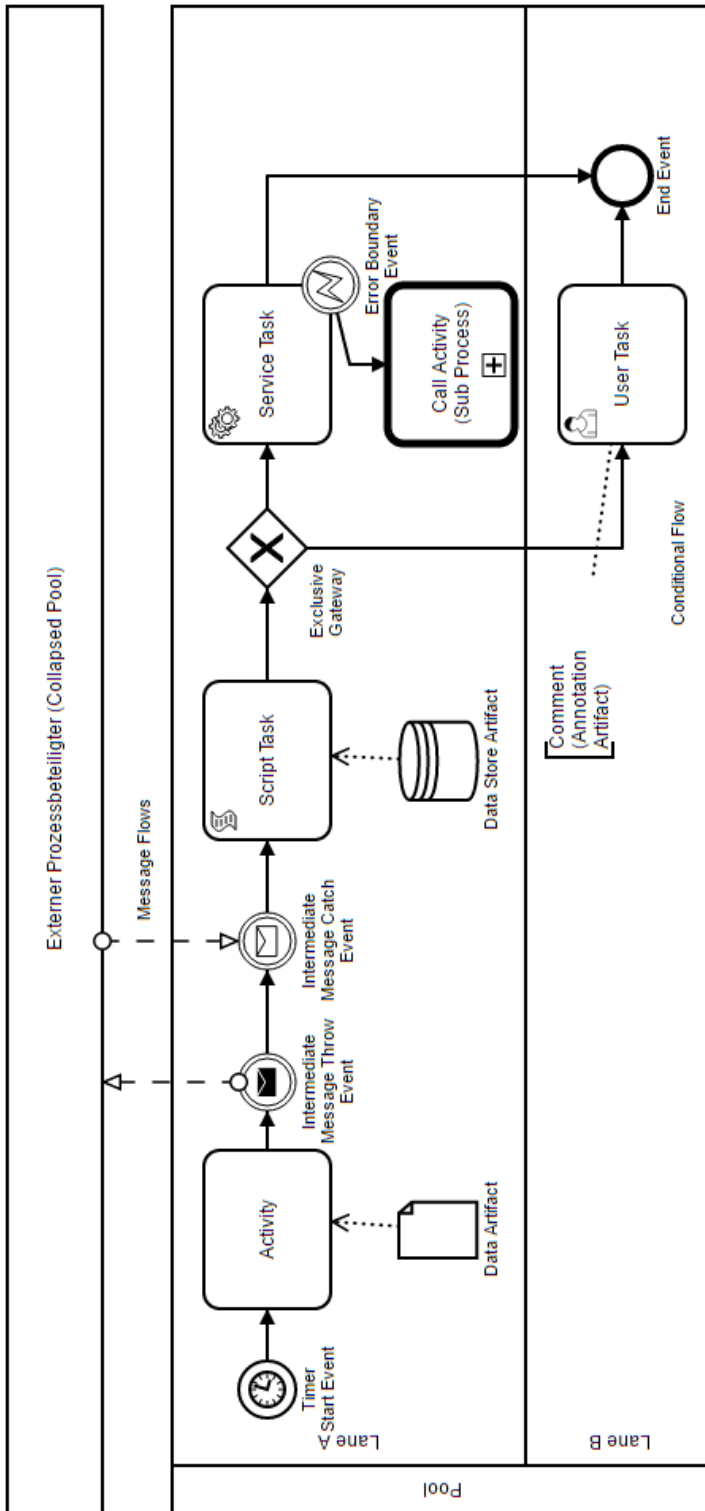


Abbildung 1: Kernelemente der BPMN 2.0 am Beispiel

Ausführbare Prozesse im Rahmen von Workflows

Ausführbare Prozesse sind solche, die von Computern unterstützt ausgeführt werden können. BPMN bietet es z.B. an, Activities zu typisieren (s. [7], S. 31 f.). Solche verfeinerten, typisierten Activities sind beispielsweise ein User Task, der explizit einem Mitarbeiter zur Bearbeitung zugewiesen wird, oder ein Service Task, der automatisierte Aufgaben darstellt, die nicht durch einen menschlichen Benutzer bearbeitet werden, sondern von einem verknüpften Programm bearbeitet werden.

BPMN 2.0 "unterstützt die Darstellung von Arbeitsabläufen (Prozessen) im Rahmen des Geschäftsprozessmanagements und die rechnergestützte Ausführung (Workflow-Management) der Prozesse" (s. [5], S 85 ff.). Erreicht wird dies durch die Konfiguration von Schnittstellen- oder Dateninformationen in den jeweiligen Prozesselementen, die zur Ausführung vorgesehen sind (vgl. [8], S. 198 ff.). Somit kann beispielsweise durch einen Script Task eine Variable initialisiert werden, die als Grundlage für eine Bedingung eines Gateways dient. Die Ausführung von Prozessen erfolgt als Workflow.

Ein Workflow ist ein "formal beschriebener, ganz oder teilweise automatisierter Geschäftsprozess", der Details über "die zeitlichen, fachlichen und ressourcenbezogenen Spezifikationen" beinhaltet, die "für eine automatische Steuerung des Arbeitsablaufes auf der operativen Ebene erforderlich sind". Die angestoßenen Arbeitsschritte werden durch menschliche Benutzer oder Anwendungsprogramme ausgeführt.

Ein Workflow beschreibt noch nicht den sich in der konkreten Ausführung befindlichen Prozess, sondern kann als Blaupause für angestoßene Prozesse verstanden werden: "Von dem Workflow als Typ oder Schema eines teil- bzw. automatisierten Arbeitsablaufes zu unterscheiden ist eine Workflow-Instanz, die eine konkrete Ausführung des Workflows bezeichnet" (vgl. [5], S 41). Prozesse werden in der Regel sehr oft durchlaufen, unterscheiden sich aber beispielsweise durch unterschiedliche Bewegungs- und Stammdaten, die verwendet werden. Die Prozesse, welche sich in der Ausführung befinden, werden als Prozessinstanz bezeichnet. Sie können durch Eintreten eines Startereignisses im Prozess durch ein Workflow-System instanziiert und gesteuert ausgeführt werden.

Workflow-Systeme steuern den Ablauf eines Prozesses und die benötigten Ressourcen: "Workflow-Systeme (...) sind Softwaresysteme, die aktiv daran beteiligt sind, innerhalb einer Organisation den Informationsfluss zu steuern und Arbeitsaufgaben durchzuführen. Es handelt sich um Anwendungen, die es Mitarbeitern oder Gruppen von Mitarbeitern ermöglichen, in einem strukturierten und unstrukturierten Umfeld automatisch eine Reihe von einmaligen oder wiederkehrenden Abläufen so zu steuern, dass die Ziele des Unternehmens erreicht werden" (s. [3], S. 267). Hierbei liegt der Fokus auf Schnittstellen zwischen Aktivitäten, die für eine Integration und Koordination verschiedener Arbeitsplätze über Abteilungsgrenzen hinweg definiert werden müssen. Auch Bearbeitungszeiten im Ganzen stehen verstärkt im Fokus. Ein primäres Ziel ist eine verringerte Durchlaufzeit eines gesamten Prozesses (vgl. [3], S. 265 ff.).

Workflow-Systeme unterstützen zusätzlich die Erreichung der folgenden Ziele: Den Prozess transparenter zu machen, Termine einzuhalten, auf Veränderungen flexibel reagieren zu können, die Anzahl

der durchzuführenden Arbeitsschritte zu minimieren, Arbeitsschritte zu parallelisieren und die Aufgabenkomplexität zu verringern (vgl. [3], S. 265 ff.).

Um einen Prozess ausführbar zu machen, wird in der Regel ein fachliches Ausgangsmodell als Ideal-ausprägung des Geschäftsprozesses verfeinert. Dies geschieht durch Zerlegung des Prozesses auf der Makro-Ebene in einzelne Aktivitäten auf der Mikro-Ebene, die von einer spezifischen Ressource ausgeführt werden sollen. Hierbei entstehen auch verschiedene Sichten auf einen Prozess: Während fachliche Sichten die Geschäftsstrategie und eine übersichtliche Darstellung der Arbeitsschritte zur Erzielung des Wertes für den Kunden im Fokus haben, benötigt eine technische Sicht, die zur Ausführungsreife entwickelt wird, um beispielsweise an ein Workflow-System übergeben werden zu können, eine formale Beschreibung und weitere technische Elemente aus der BPMN-Palette.

Die formale Beschreibung umfasst die zeitlich, fachlich und ressourcenbezogene Spezifikation, um eine automatische Steuerung zu ermöglichen (vgl. [5], S. 41 ff.). BPMN 2.0 verknüpft diese beiden Aspekte durch die Vereinigung grafischer und technischer Notationsteile. Darüber hinaus bietet sie die Möglichkeit, spezifische Erweiterungen in Form von beispielsweise industriespezifischen Activities zu definieren.

BPMN bietet den fachlichen und technischen Beteiligten von Geschäftsprozessen durch seine standardisierten Kernelemente eine gute Basis zur Zusammenarbeit verschiedener Akteure und bleibt dabei dennoch für fachliche Anwender überschaubar und nützlich (vgl. [3], S. 215 ff.)

Die Zusammenarbeit verschiedener Prozessbeteiligter, die Evolution von Prozessen, daraus resultierende verschiedene Sichten auf einen Prozess und typische Veränderungen sind Thema des nächsten Kapitel über Geschäftsprozessmanagement und dem Lebenszyklus von Geschäftsprozessen.

2.2 Lebenszyklus des Geschäftsprozessmanagements

In diesem Kapitel wird vor dem Hintergrund eines Lebenszyklusmodells des Geschäftsprozessmanagements gezeigt, welche Akteure an der Entwicklung von Prozessen beteiligt sind, welche Sichten auf Prozesse es gibt, was die Unterschiede zwischen den Sichten sind und welche daraus resultierenden, typische Veränderungen auftreten.

Das Geschäftsprozessmanagement (GPM) hat neben der Verwaltung von Prozessen vor allem die kontinuierliche Verbesserung und Optimierung (auch als Geschäftsprozessoptimierung (GPO) bekannt) bestehender Prozesse eines Unternehmens zur Aufgabe. Im Vordergrund stehen hierbei Qualität, Zeit, Kosten und Kundenzufriedenheit(vgl. [3], S. 86 ff.; [1], S. 253 ff.).

GPM wird auch betrieben, um Barrieren zwischen IT und Business zu überwinden. Oft besteht keine optimale technische Unterstützung der Geschäftsprozesse eines Unternehmens, vor allem in informationsgetriebenen Geschäftsprozessen, wie sie beispielsweise überwiegend bei Finanzdienstleistern anzutreffen sind. Ein Austausch und eine Abstimmung zwischen Business und IT sind teilweise nur in geringem Maße umgesetzt.

Diese Diskrepanz, die sich negativ auf die Wettbewerbsfähigkeit eines Unternehmens auswirkt, hat zu einer Entwicklung hin zu enger Abstimmung zwischen Fachbereichen und IT in Hinsicht auf Strate-

gien und Prozessen geführt. Angestrebt wird dabei eine gezielte, organisationsübergreifende und integrierte Aufgabenverteilung durch Prozessorganisation, beispielsweise zur Verringerung von Leer- und Wartezeiten in und zwischen Aktivitäten und somit als Grundlage der Optimierung von Durchlaufzeiten. Traditionell stand durch die Spezialisierung von Funktionen die Optimierung der Kapazitätsnutzung im Vordergrund.

Neben der Geschäftsprozessoptimierung gibt es den Ansatz des Business Process Reengineering (BPR). Durch BPR werden bestehende Prozesse nicht verfeinert und optimiert, sondern von Grund auf neu gestaltet. Hierbei soll unter anderem eine Abstimmung zwischen Business und IT von Anfang an erfolgen, um die Lücke zwischen Business und IT möglichst zu minimieren und eine technische Infrastruktur optimal zur Bewältigung der Unternehmensaufgaben zu nutzen (vgl. [4], S. 410 f.; [1], S. 258 f.).

GPM und BPR "haben gemeinsam, dass durch strukturiert modellierte Prozesse Transparenz über die Verflechtung der einzelnen Arbeitsschritte geschaffen werden kann" (vgl. [3], S. 86 ff.). Im Rahmen des Geschäftsprozessmanagements wird durch Modelle auch Transparenz über die bestehende Prozesslandschaft eines Unternehmens gefördert.

Diese Transparenz und der geschaffene Überblick ist bei der Optimierung und Neuentwicklung von Geschäftsprozessen wichtig, da so Anknüpfungspunkte für neue Prozesse an bestehenden Prozessen und abgebildeten Funktionen, sowie etwaige Lücken leichter identifiziert werden können.

Je umfangreicher die Prozesslandschaft eines oder mehrerer kooperierender Unternehmen ist, desto stärker ist eine flexible Strukturierung und Konfiguration der Prozesse nötig, was zur Folge hat, dass die Abhängigkeiten zwischen den Prozessen und Koordinations- und Planungsaufwand höher werden (vgl. [11], S. 377 ff.).

Lebenszyklus des Geschäftsprozessmanagements

Im Folgenden wird ein zweistufiger Workflow-Lebenszyklus mit drei Teilzyklen (vgl. [5], S. 58 ff.) vorgestellt, der als typisch für die Arbeitsweise im Rahmen des Geschäftsprozessmanagements angesehen werden kann.

Während beim einstufigen Modell ein Workflow-Modell direkt erstellt wird, leitet sich bei der zweistufigen Vorgehensweise das Workflow-Modell aus einem fachlichen Geschäftsprozessmodell ab. Dieser Ansatz ermöglicht es, verschiedene fachliche und technische Sichten auf einen Prozess zur Verfügung zu stellen und wird in der Praxis oft bevorzugt. Der Lebenszyklus sieht drei teilweise ineinandergreifende Teilzyklen vor, die einem iterativen Vorgehen folgen (s. Abbildung 2).

Die Ausgangsbasis des ersten Teilzyklus ist die Ermittlung und Modellierung von Ist-Prozessen. Bestehende Geschäftsprozesse werden analysiert, modelliert und neu strukturiert. Eine Strategieentwicklung und Optimierung finden hier auf fachlich-konzeptioneller Ebene statt. Das Ergebnis des ersten Teilzyklus sind angestrebte (ideale) Soll-Geschäftsprozessmodelle, die in den zweiten Teilzyklus überführt werden.

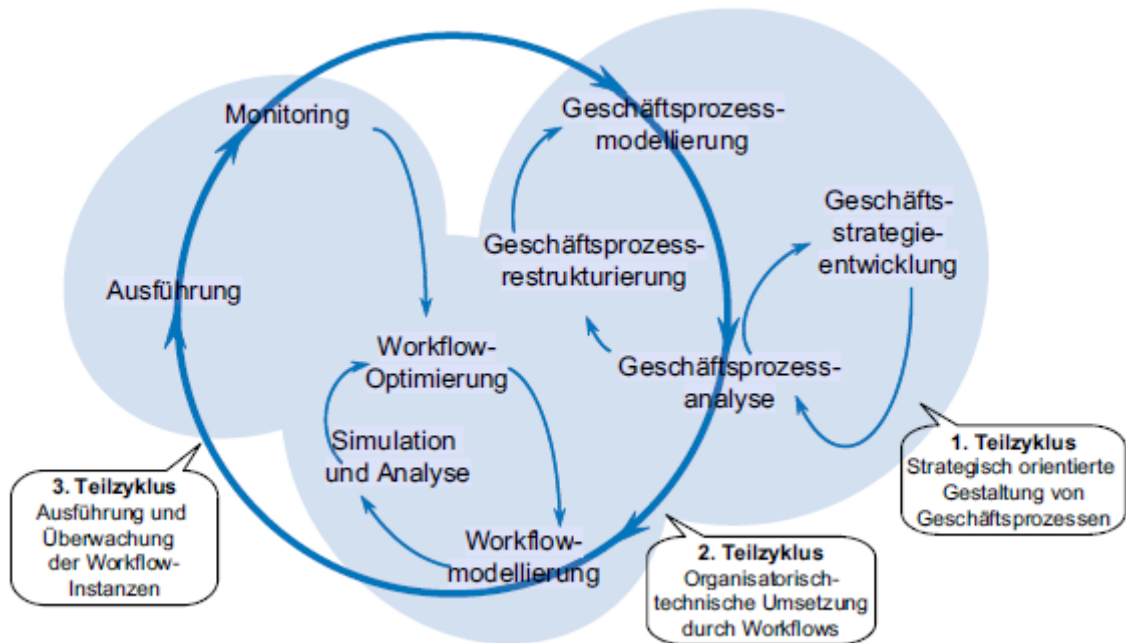


Abbildung 2: Lebenszyklus des Geschäftsprozessmanagements, übernommen aus [1], S. 59

Im zweiten Teilzyklus werden die Soll-Prozessmodelle soweit verfeinert, dass sie auf der operativen Workflow-Ebene verwendet werden können. Der angestrebte Detaillierungsgrad lässt die automatische Ausführung und eine Simulation zur Analyse der Workflow-Modelle zu. Das Ergebnis des zweiten Teilzyklus sind somit detaillierte, ausführbare Workflow-Modelle, die durch (rechnergestützte) Simulation auf Schwachstellen (wie beispielsweise Verklemmungen im Prozess), erwartete Durchlaufzeiten und Kosten analysiert werden können.

Der dritte Teilzyklus sieht die Optimierung der im zweiten Teilzyklus erstellten, ausführbaren Workflow-Modelle vor. Durch inkrementelle Änderungen und Optimierung der Modelle werden z.B. Abweichungen zwischen erwarteten Durchlaufzeiten und gemessener Durchlaufzeiten analysiert und verringert. Dieser Teilzyklus überwacht die Ausführung des Prozesses und sieht eine mögliche Rückkopplung zu den ersten beiden Teilzyklen vor.

Zusammengefasst kann der Lebenszyklus eines Geschäftsprozessmodells somit grob in drei Phasen aufgeteilt werden. In der ersten Phase werden die Prozesse aus fachlicher Sicht relativ detailliert, was technische Aspekte anbelangt, als Idealablauf geplant und modelliert. In der zweiten Phase werden die Modelle der Planungsphase durch Anreicherung und Konfiguration der verwendeten Modellelemente in ausführbare Modelle transformiert und durch Simulation einer ersten Analyse unterzogen. In der dritten Phase werden die ausführbaren Modelle während der Ausführung überwacht und in Abhängigkeit von Ist- und Soll-Werten weiterer Optimierung unterzogen (vgl. [10]).

Den Teilzyklen und Phasen können verschiedene Rollen und somit Akteure zugeordnet werden, die unterschiedliche Ziele verfolgen und hierfür verschiedene Sichten auf einen Prozess benötigen. Die Akteure und Sichten werden im folgenden Kapitel vorgestellt.

2.2.1 Beteiligte Akteure und verschiedene Sichten auf einen Prozess

Am Geschäftsprozessmanagement im Allgemeinen und der Entwicklung vom fachlichen zum ausführbaren Geschäftsprozessmodell im Speziellen sind in der Regel verschiedene Akteure oder Stakeholder beteiligt, die unterschiedliche Ziele verfolgen, verschiedene Sichten auf einen Prozess und Detaillierungsgrade eines Prozessmodells benötigen und denen verschiedene Rollen im bereits vorgestellten Lebenszyklusmodell zufallen (vgl. [12], S. 136 ff.; [8], S. 11).

Allgemein kann zwischen Analysten und Designern, die Geschäftsprozesse auf einer fachlichen Ebene entwerfen und verfeinern und technischen Architekten und Entwicklern, die Geschäftsprozesse an die technische Infrastruktur anpassen und durch IT implementieren, unterschieden werden (vgl. [3], S. 109 f.). Diese zwei Gruppen können weiter nach ihren Zielen und abgeleiteten Aufgaben differenziert werden (vgl. [13], S. 8 f.).

Auf fachlicher Ebene gibt es Business Analysten. Sie erstellen fachliche Modelle, die durchaus schon simulationsfähig sind. Durch sie entsteht eine Business-Sicht. Die Modelle werden angereichert durch organisatorische Elemente, ausgedrückt durch Pools und Lanes, sowie der ersten Zuweisung von Ressourcen durch Typisierung der Modellelemente (automatisch oder durch Benutzer ausgeführte Arbeit). Sie setzen Geschäftsregeln um und legen Kennzahlen (Zeit, Kosten) fest.

Unterstützung erhält der Business Analyst durch IT Systemanalysten, die Modelle an bestehende IT anpassen und korrigieren. Sie detaillieren und typisieren Aktivitäten, gestalten Sequenzflüsse, erstellen Use Cases für Tasks, Gateways, Entscheidungen und Ereignisse. Das fachliche Modell wird schrittweise in ein technisches Modell transformiert, die Business-Sicht entwickelt sich weiter zur IT-Sicht, die von IT Architekten und Entwicklern weiter verfeinert wird, bis ein operatives Modell entsteht.

IT Architekten beschreiben Serviceschnittstellen, Integrationsmethoden, erstellen Dokumentationen und Deployment-Pläne. Sie können auch BPEL zum Einsatz bringen, um das Zusammenspiel von (Web-)Services zu beschreiben.

IT Entwickler programmieren in BPEL, Java und anderen Sprachen, erstellen Tests der entwickelten Programme und stellen kompilierte Programme bereit.

Rollen im Kontext des Prozesslebenszyklusmodells

Im Kontext des Geschäftsprozessmanagements werden weitere Rollen beschrieben, die sich mit den bisher vorgestellten Akteuren überschneiden (vgl. [5], S. 4 f.). Von den sieben in [5] aufgeführten Rollen, die vom Chief Process Officer bis zum Prozessauditor reichen, sollen die relevantesten kurz beschrieben werden und den Teilzyklen des vorgestellten zweistufigen Lebenszyklusmodells zugeordnet werden (s. Abbildung 3).

Ein Process Owner oder Prozessmanager ist für die operative Steuerung und Modifikation von ihm zugeteilten Geschäftsprozessen verantwortlich. Er erstellt Zieldefinition und ist der Garant für die Zielerreichung und Leitung von Prozessmitarbeitern (ausführender Mitarbeiter).

Prozess- und Workflowmodellierer arbeiten auf den bereits angesprochenen Ebenen. Der Prozessmodellierer arbeitet auf der betriebswirtschaftlichen Ebene und erstellt ausgehend von Ist-Analysen Soll-Prozesse auf einer hohen fachlichen Abstraktionsebene. Der Workflowmodellierer verfeinert die Soll-Prozesse der Prozessmodellierer in Hinsicht auf IT-Unterstützung und Implementierung in Workflow-Systeme. Er arbeitet auf einer niedrigen technischen Abstraktionsebene der (teil-)automatisierten Ausführung.

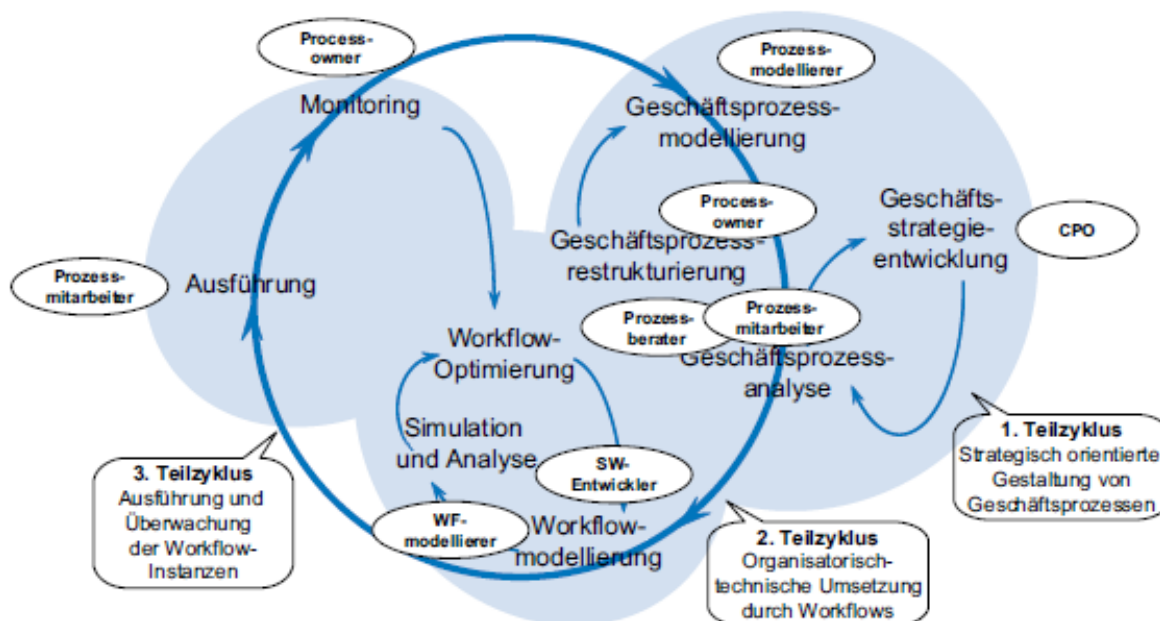


Abbildung 3: Rollen im Lebenszyklus des GPM, übernommen aus [1], S. 60

Idealtypisch werden die vorgestellten Rollen den Teilzyklen folgendermaßen zugeordnet (vgl. [5], S. 228 ff.): In Teilzyklus I sind unter anderem der Chief Process Officer, der Processowner und Prozessmodellierer tätig. Sie entsprechen den Akteuren Business Analyst und in Teilen dem IT Systemanalysten. In Teilzyklus II sind Workflowmodellierer und Softwareentwickler tätig, die den technischen Architekten und Entwicklern entsprechen. In Teilzyklus III ist wieder der Processowner zur Überwachung der Prozessausführung tätig. Ausführende Mitarbeiter sind in diesem Teilzyklus angesiedelt.

Die Unterschiede zwischen den so entstehenden Sichten (primär Business und IT) und typische Veränderungen auf dem Weg vom fachlichen zum technischen Modell werden im folgenden Kapitel näher betrachtet.

2.2.2 Unterschiede zwischen Sichten und typische Veränderungen

Durch die verschiedenen Akteure, die an der Geschäftsprozessmodellierung beteiligt sind, entstehen generell zwei Abstraktionsebenen eines Prozesses, bzw. zwei Sichten auf einen Prozess (vgl. Hervorhebungen in Abbildung 4; vgl. [8], S. 121 ff., S. 147 ff., S. 187 ff.). Auf der einen Seite die fachlich-konzeptionelle, auf der anderen die technische-ausführbare und somit operative Ebene bzw. Sicht (vgl. [5], S. 57 f.; [4], S. 565 ff.; [1], S. 272).

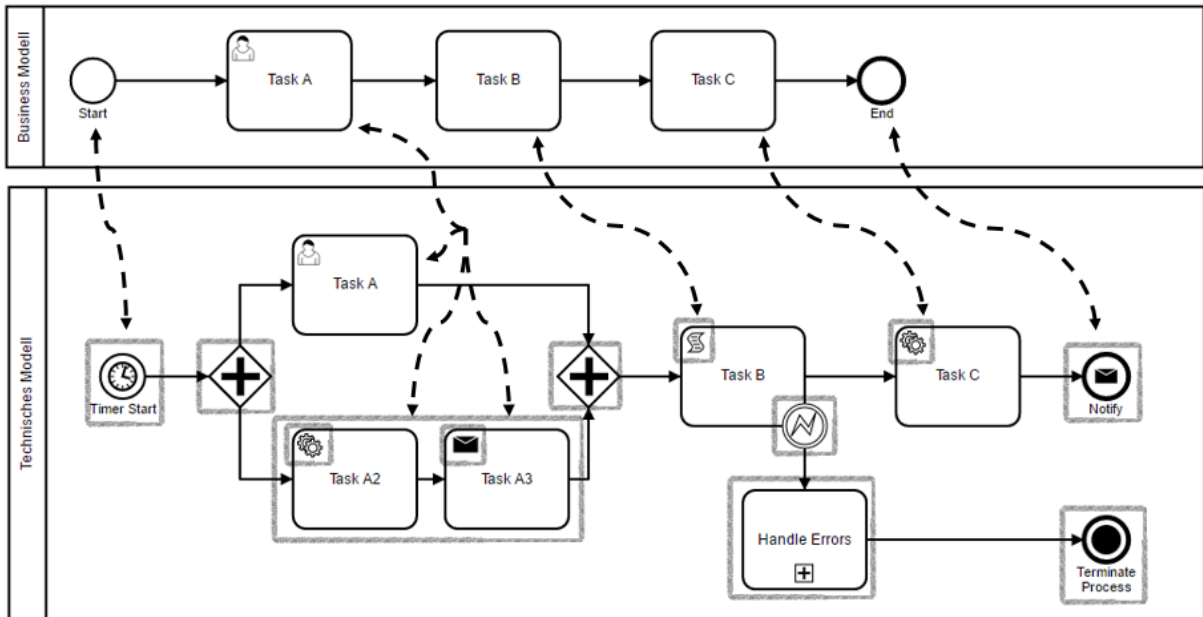


Abbildung 4: Beispiel einer Verfeinerung vom fachlichen zum technischen Modell (Veränderungen sind durch Rechtecke hervorgehoben, Korrespondenzen zwischen den Elementen durch gestrichelte Verbindungslinien)

Wird eine IT-Architekturansicht einbezogen, die eine Abbildung und Angleichung des Prozesses auf die vorhandene IT-Infrastruktur zum Ziel hat, entstehen in der Praxis drei verschiedene Prozessmodelle bzw. Sichten auf einen Prozess (vgl. [13], S. 9).

Daneben kann die bisher als höchste Abstraktionsebene anzusehende Business-Sicht nochmals in beschreibend (beispielsweise auch informell und durch Text) und analytisch (beispielsweise als BPMN-Modell) unterteilt werden. Die hinzugekommene, beschreibende Ebene stellt hierbei den Geschäftsprozess "sehr abstrakt dar" und "vermittelt einen Überblick über den Ablauf – normalerweise nur der Gutfall" und hat zum Ziel "in der Regel die Kommunikation der Geschäftsprozesse über Organisationseinheiten hinweg, beispielsweise zum oberen Management" (vgl. [14], S. 50 ff.; [8], S. 170 ff.).

Business- und IT-Sicht können nach Ziel, Gestaltungsebene und Detaillierungsgrad unterschieden werden, wie in Tabelle 1 aufgeführt. Hierbei ist primär zwischen dem "Was" (Business-Sicht, Geschäftsstrategie und Ziel) und dem "Wie" (IT-Sicht, Ausführung) zu unterscheiden (vgl. [5], S. 46 ff.).

Ein weiteres Unterscheidungsmerkmal zwischen den Sichten "ist die Ausführbarkeit durch einen menschlichen Aufgabenträger (Mitarbeiter) oder ein Computerprogramm" (vgl. [5], S. 46 ff.). Der Detaillierungsgrad eines Modells drückt sich auch im Grad der Formalität des Modells aus (vgl. [2], S. 211 ff.).

	Business-Sicht	IT-Sicht
Ziel	Analyse und Modellierung von Arbeitsabläufen unter Einbehaltung strategischer Ziele	Spezifikation der technischen Ausführung der Arbeitsabläufe
Gestaltungsebene	konzeptionelle Ebene, Verbindung zur Geschäftsstrategie	operative Ebene, Verbindung zur IT-Unterstützung
Detaillierungsgrad	von einem Mitarbeiter an einem Arbeitsplatz ausführbare Arbeitsschritte	konkretisierte Arbeitsschritte und Arbeitsverfahren, konkret zugeordnete personelle und technische Ressourcen

Tabelle 1: Abstraktionsebenen eines Prozessmodells differenziert nach Ziel, Gestaltungsebene und Detaillierungsgrad

Typische Änderungen an den Modellen

Generell können Änderungen in drei Gruppen unterteilt werden. Die erste Gruppe umfasst das Hinzufügen von Elementen (komplementär hierzu ist das Entfernen zu betrachten), oder die Modifikation der Informationen dieser. Die zweite Gruppe beinhaltet strukturelle Veränderungen, wie z.B. durch Verhaltensänderungen eines Prozessablaufs durch Restrukturierung der Sequenzflüsse. Die dritte Gruppe entsteht durch die Verfeinerung eines fachlichen Modells hin zum ausführbaren Modell durch das Hinzufügen von IT spezifischer Aktivitäten und Elemente, wie Fehlerereignisse (vgl. [15], S. 627 ff.). Im Anhang A sind typische Änderungen³ in einer Tabelle aufgelistet.

Typische Veränderungen lassen sich auch unter Hierarchieaspekten beschreiben (vgl. [13], S. 10 ff.). Eine Veränderung ist hierarchisch, wenn man beispielsweise die durch einen aufgespalteten Task entstehenden Blöcke in einen Subprozess kapseln kann. Nicht-hierarchisch sind Veränderungen, bei dem keine Hierarchie durch beispielsweise in den Prozess eingebettete Subprozesse aufgebaut wird.

Die Änderungen durch die beteiligten Akteure führen häufig zu Differenzen im Umfang der dargestellten Elemente in den Modellen, oder zu Verhaltensänderungen des Prozessablaufs und somit zu Inkonsistenzen. Für die Business-Sicht und somit den Geschäftsprozess sind seltener Elemente der IT-Sicht relevant:

(...) a process model, which is appropriate in a particular context, highlights certain aspects, whereas others are neglected or even ignored. ([16], S. 2)

Eine Ausnahme bilden hier in der IT-Sicht hinzugefügte Elemente, die nicht in der Business-Sicht berücksichtigt wurden und in diese übertragen werden müssen, um einen nötigen fachlichen Schritt in der Bearbeitung des Prozesses abzubilden. Im folgenden Kapitel wird auf das Thema vertikaler Inkonsistenzen und Möglichkeiten zur Entdeckung und Beseitigung dieser eingegangen.

³ Für eine weitere detaillierte Darstellung durch Beispiele in BPMN, siehe [13], S. 10 ff.

2.3 Vertikale Konsistenz

Im folgenden Kapitel soll der Begriff der vertikalen Konsistenz im Kontext zu den bereits vorgestellten Aspekten des Geschäftsprozessmanagements und der Geschäftsprozessmodellierung und beteiligter Akteure und typischer Veränderungen entwickelt werden.

2.3.1 Definition und Abgrenzung zu horizontaler Konsistenz

Konsistenz kann generell als Zustand der Widerspruchsfreiheit von Artefakten wie Modellen definiert werden. Das in Datenbanksystemen und verteilten Systemen umgesetzte ACID-Paradigma (Atomicity, Consistency, Isolation und Durability) soll analog helfen, Konsistenz zu definieren:

***Consistency** Eine Transaktion hinterlässt nach Beendigung einen konsistenten Datenbasiszustand. Anderenfalls wird sie komplett (siehe Atomarität) zurückgesetzt. Zwischenzustände, die während der TA-Bearbeitung entstehen, dürfen inkonsistent sein, aber der resultierende Endzustand muss die im Schema definierten Konsistenzbedingungen (z.B. referentielle Integrität) erfüllen (s. [17], S. 273)*

Der Prozess der Verfeinerung eines fachlichen Geschäftsprozessmodells hin zu einem technischen Workflowmodell ist hierbei als die Transaktion (oder mehrere im Rahmen des Lebenszyklus eines Prozessmodells) anzusehen. Die referentielle Integrität sei die Abbildung der in der Business-Sicht vorgegebenen Ablaufschritte und Geschäftsregeln.

Das Resultat ist somit zunächst die Abbildung aller fachlichen Vorgaben der Business-Sicht in der IT-Sicht. Der konsistente Endzustand ist die Transformation des fachlichen Modells in ein ausführbares Modell unter Berücksichtigung aller Restriktionen und Vorgaben des Ausgangsmodells, insbesondere in der Modellierung aller fachlichen und geschäftsrelevanten Elemente.

In der Praxis und in Bezug zur Geschäftsprozessmodellierung ist der Begriff der Konsistenz schwer zu fassen und subjektiv:

(...) stakeholders do not have a definitive notion of consistency and think that at the end of the day they are always capable of regaining an 'agreement' with respect to a consistency level, which is not really 'strict'. The benefit of working collaboratively with process models indicates that inconsistencies need to be "managed", that is detected, analysed, recorded and possibly resolved (...) Interestingly, it is generally accepted that it is hard for people to agree on the meaning of consistency between business and IT models (...) (s. [13], S. 17)

Horizontale und vertikale Konsistenz

Horizontale Konsistenz beschreibt die Konsistenz zwischen Artefakten auf der gleichen Abstraktionsebene. Verschiedene Versionen eines fachlichen Geschäftsprozessmodells, die sich auf der gleichen Ebene befinden, können beispielsweise durch Weiterentwicklungen durch verschiedene Business Analysten Überlappungen aufweisen, die es zu vereinheitlichen und somit konsistent zu machen gilt.

Vertikale Konsistenz hingegen erweitert das Konzept um Abstraktionsebenen bzw. verschiedene Sichten. Sie beschreibt die Konsistenz zwischen zwei Modellen auf verschiedenen Abstraktionsebenen oder zwei verschiedenen Modelltypen. Ein Beispiel hierfür ist die Konsistenz von Modellen eines Geschäftsprozesses, gesehen und gestaltet aus den in den vorangegangenen Kapiteln beschriebenen verschiedenen Sichten bzw. Abstraktionsebenen.

In dieser Arbeit wird der Begriff der vertikalen Konsistenz primär als Konsistenz zwischen zwei Prozessmodellen auf verschiedenen Abstraktionsebenen (wie in den Modellen eines Geschäftsprozesses aus Business- und IT-Sicht) hinsichtlich der Abbildung geschäftsrelevanter Elemente und ihrer Funktion (in beiden Modellen) unter Berücksichtigung der in Kapitel 2.2.2 aufgeführten typischen Veränderungen verstanden.

Für die Prozessmodellierung hat sich in der Praxis als besonders wichtig herausgestellt, dass jede geschäftsrelevante Aktivität sowohl in der Businesssebene, als auch in der Implementierungsebene zu finden sein muss. Rein technische Aktivitäten sollten hingegen nicht in der Business-Sicht enthalten sein. Differenzen des Abdeckungsgrades der geschäftsrelevanten Elemente in Bezug auf die Aktivitäten und Verhalten der Prozesse in Bezug auf Ausführungspfade stellen Inkonsistenzen dar (vgl. [13], S. 2 ff.).

Konsistenz kann durch Auswertung von Konsistenzregeln geprüft werden. Hierzu zählen logikbasierte Ansätze, Modellprüfung, spezialisierte Modellanalyse und menschenorientierte, kollaborative Explorationsansätze (vgl. [13], S. 3 f.). Diese Arbeit hat seinen Fokus dabei auf einen menschenorientierten Ansatz, der im Konzept weiter ausgeführt wird.

Bedeutung von vertikaler Konsistenz

Ein mögliches Konsistenzmanagement wird in [13] als bestehend aus der Herstellung und Erhaltung von Konsistenz zwischen Softwareartefakten, wie Modellen, Quellcode oder Dokumentation und Testfällen, welche durch verschiedene Stakeholder-Gruppen erstellt und verwendet werden beschrieben (vgl. [13], S. 2 ff.).

Die an der Modellierung im Rahmen eines mehrstufigen Ansatzes beteiligten Akteure arbeiten auf verschiedenen Abstraktionsebenen und haben verschiedene Ansichten und Annahmen über Prozesse und deren Ausgestaltung, was zu Inkonsistenzen zwischen den Modellen führt.

Vertikale Inkonsistenzen zwischen der Business-Sicht und IT-Sicht können die Qualität, Durchlaufzeiten und Kosten eines Prozesses und die angestrebte Kundenzufriedenheit negativ beeinflussen. Sei es durch Verzögerungen bei der Entwicklung, wenn beispielsweise eine geschäftsrelevante Aktivität nicht berücksichtigt, oder spät erkannt wurde und nachträglich in das fachliche Modell integriert werden muss, was wiederum Auswirkungen auf die Modellierung des technischen Modells hat.

Oder sei es durch nicht berücksichtigte, geschäftsrelevante Aktivitäten zum Zeitpunkt des dritten Teilzyklus des vorgestellten Lebenszyklusmodells des Geschäftsprozessmanagements. Dies kann zu finanziellen Einbußen im operativen Geschäft führen, wenn beispielsweise Prozessinstanzen angehalten, manuell neu gestartet oder bis auf weiteres gestoppt werden müssen. Wird in diesem Fall nur das

operative Modell um die fehlende Aktivität ergänzt (sozusagen als "Hotfix"), ohne die Änderung auf das fachliche Modell zu übertragen, können weitere Inkonsistenzen und Probleme bei weiteren Iterationen entstehen (vgl. [13], S. 17).

Vor- und Nachteile vertikaler Inkonsistenzen

Positiv gesehen können Inkonsistenzen helfen, verschiedene Ansichten und Ziele der Prozess-Stakeholder aufzuzeigen. Absichtlich können Inkonsistenzen eingeführt werden, um Prozessaspekte hervorzuheben, die einer genaueren Betrachtung und Verbesserung des Prozesses unterzogen werden sollten.

Inkonsistenzen können aber wie bereits ausgeführt zu Entwicklungsverzögerungen, erhöhten Kosten und zu Fehlern und Versagen im operativen Geschäft führen. Sie sollten dementsprechend so schnell wie möglich entdeckt und beseitigt werden, oder zumindest dokumentiert und verwaltet (vgl. [13], S. 2 ff.).

Wie dem Problem vertikaler Inkonsistenzen entgegengewirkt werden kann, wird im nächsten Abschnitt behandelt. Weiterführend hierzu werden verschiedene Ansätze im Kapitel über verwandte Arbeiten dargestellt.

Korrespondenzen als Ausgangsbasis einer einfachen Prüfung

Korrespondenzen stellen eine Verbindung zwischen den Elementen zweier Modelle auf verschiedenen Abstraktionsebenen her und sind durch die gleiche Funktionalität der verbundenen Elemente definiert (vgl. auch [18], S. 11 ff.):

A correspondence is a connection between two model elements in the primary and secondary business process model. In general, a model element in the primary process model is connected by a correspondence to another model element in the secondary process model, if both corresponding model elements represent the same functionality.
(s. [19], S. 15 ff.)

Für diese Arbeit ist im Zusammenhang das *primary model* als die Business-Sicht und das *secondary model* als die IT-Sicht auf einen Prozess zu verstehen. Weiter können Korrespondenzen als bidirektionale Relation zwischen einem oder mehreren Elementen eines Modells und einem oder mehreren Elementen des anderen Modells definiert werden (vgl. [20], S. 8 f.) Abbildung 4 zeigt solche Korrespondenzen.

Es werden folgend fünf valide Korrespondenztypen definiert werden (vgl. [18] - [20]), die für das Konzept als Grundlage dienen werden. Sie werden in Tabelle 2 aufgeführt. Die linke Seite der Typenbezeichnung (n-) stellt hierbei beispielsweise ein fachliches Modell und die Business-Sicht, die rechte Seite (-n) ein technisches Modell und die IT-Sicht dar (fachlich-technisch).

Typ	Beschreibung
1-1	Element x eines fachlichen Modells und Element y des technischen Modells stellen die gleiche Funktionalität dar.
1-0	Es existiert für Element x eines fachlichen Modells kein Element y des technischen Modells mit der gleichen Funktionalität.
0-1	Es existiert für Element y eines technischen Modells kein Element x des fachlichen Modells mit der gleichen Funktionalität.
1-n	Es existieren für Element x eines fachlichen Modells n Elemente des technischen Modells, die zusammen die Funktionalität von x darstellen.
n-1	Es existiert für mehrere Elemente eines fachlichen Modells ein Element x des technischen Modells, welches die Funktionalitäten der n Elemente des fachlichen Modells umfassen.

Tabelle 2: Korrespondenztypen und ihre Beschreibung

Bei den Korrespondenzen ist bei den Typen 1-0 und 0-1 darauf zu achten, dass diese Unterschiede zwischen den Modellen bzw. den enthaltenen Elementen darstellen und nicht die Abwesenheit eines Elements gleicher Funktionalität. Sie können beispielsweise zur Markierung von Elementen, die nur für die IT-Sicht relevant sind (beispielsweise eines Logging-Tasks), verwendet werden.

Korrespondenzen des Typs 1-n und n-1 sind für Referenzmodellierung, wie sie bei der Verfeinerung eines fachlichen Modells zu einem technischen stattfindet, besonders interessant. Elemente der Business-Sicht werden häufig in mehrere Elemente in der IT-Sicht zerlegt, die das Ausgangselement in seiner Funktionalität abdecken. Korrespondenzen helfen dabei, Änderungen nachzuvollziehen und entsprechend handeln zu können. Sie können außerdem domänenspezifisches Wissen dokumentieren.

Eine 0-0 Korrespondenz ist nicht sinnvoll, da keine Relation zwischen Elementen beschrieben wird. 0-n oder n-0 können durch Korrespondenzen der Typen 0-1 oder 1-0 dargestellt werden. Der Typ m-n erscheint als wenig verbreitet; auf der hohen Abstraktionsebene der Business-Sicht angesiedelte Elemente sollten in der Regel durch Zerlegung in 1-n Korrespondenzen münden.

Einfache Prüfung der Konsistenz durch Abdeckungsgrad fachlicher Elemente

Durch Korrespondenzen kann eine einfache Prüfung auf Konsistenz vollzogen werden. Hierzu wird der Abdeckungsgrad der Elemente der Business-Sicht ermittelt. Sind alle Elemente des fachlichen Modells Teil einer Korrespondenz der Typen 1-1, 1-n oder n-1, ist diese Konsistenz erreicht, da alle geschäftsrelevanten Aktivitäten und Elemente im technischen Modell abgebildet sind.

Berechnung von Korrespondenzen

Korrespondenzen können prinzipiell auf zwei Arten automatisiert berechnet werden, durch statische und verhaltensbasierte Ansätze. Der statische Ansatz erfolgt durch typographische und linguistische

Aspekte der Namen der Modellelemente. Ein Prozessmodell kann für die statische Berechnung zusätzlich in eine Baumstruktur überführt werden. Die Ebene oder Tiefe, auf der sich Elemente befinden, sowie die Position im Sequenzfluss wird bei der Berechnung der Korrespondenz berücksichtigt. Der verhaltensbasierte Ansatz berücksichtigt die Semantik der Modellelemente, wie beispielsweise den Element-Typen (vgl. [19], S. 16 f.).

Initiale Korrespondenzen des Typs 1-1 können durch Klonen eines (fachlichen) Ausgangsmodells in eine andere (niedrigere) Abstraktionsebene erstellt werden. Die eindeutigen Identifikationscodes der Elemente, die in der BPMN 2.0 Spezifikation definiert sind, bilden hierbei die Basis der 1-1 Korrespondenzen. Das geklonte Modell wird anschließend (zu einem ausführbaren Modell) verfeinert.

Weiterführend können Korrespondenzen zwischen Aktivitäten bzw. Elementen eines Modells und Korrespondenzen zwischen Subprozessen unterschieden werden (vgl. [18], S. 11 ff.).

Im folgenden Kapitel wird im Rahmen der verwandten Arbeiten näher darauf eingegangen, wie Korrespondenzen berechnet werden können und für welche Anwendungsszenarien sie benutzt werden können. Das prototypische Werkzeug dieser Arbeit stellt auf die visuelle und manuelle Verwaltung und Manipulation von Korrespondenzen ab.

2.3.2 Verwandte Arbeiten

Dieses Kapitel stellt eine Auswahl an verwandten Arbeiten zum Thema vertikaler Konsistenz im Kontext von Geschäftsprozessmodellen vor und beleuchtet die vorgestellten Ansätze.

Feger (2006): Business Process Merging and Refactoring

Vor dem Hintergrund des Business-Driven Development (BDD), bei dem Anforderungen an die IT durch Geschäftsprozesse dargestellt seien und eine enge Konformität der IT-Lösung mit dem Geschäftsmodell erreicht werden sollte, wird in [18] ein Ansatz zur Verschmelzung zweier Geschäftsprozesse entwickelt. Der Ansatz berücksichtigt nicht nur ein horizontales Szenario, um beispielsweise zwei verschiedene Versionen eines fachlichen Modells zu verschmelzen, sondern auch ein vertikales Szenario zur Vereinigung eines fachlichen (*analysis*) mit einem technischen (*design*) Modell.

Die Verschmelzung helfe dabei, bei einer Zusammenschließung von Firmen Prozesse zu vereinigen, oder Legacy-Prozesse mit neuen Prozessen zu vereinigen, oder um Versionen des gleichen Prozesses zu vereinigen.

Verschiedene Operationen und Prozesstransformationen der Geschäftsprozessmodelle werden vorgestellt, um teilautomatisiert aus zwei Geschäftsprozessmodellen ein vereinigtes Modell zu erzeugen. Der Ansatz basiert auf Korrespondenztypen, die zwischen Tasks und Subprozessen unterscheiden.

Gerth (2007): Business Process Merging

In [19] wird ein weiterer Ansatz für das Verschmelzen unterschiedlicher Versionen eines Geschäftsprozesses vorgestellt. Hierbei werden Unterschiede zwischen verschiedenen Versionen eines Geschäftsprozesses entdeckt und Operationen für die Beseitigung der Unterschiede (durch Verschmelzen) unter Berücksichtigung von bestehenden Abhängigkeiten im Modell angeboten.

Korrespondenzen dienen auch hier zur Abbildung von Verbindungen zwischen Modellelementen in Geschäftsprozessmodellen.

Der Ansatz behandelt ein Versionsszenario und erstellt mit Hilfe eines statischen Ansatzes Korrespondenzen auf Basis von typographischer und linguistischer Aspekte (bezogen auf den Namen).

Die Initialisierung der Korrespondenzen geschieht durch Klonen des Geschäftsprozessmodells, wodurch 1-1 Korrespondenzen zwischen den Elementen der Modelle entstehen. Nach der Bearbeitung einer Version bzw. eines Modells werden die Korrespondenzen aktualisiert.

Es werden *Single Entry Single Exit* (SESE) Regionen zur vereinfachten und hierarchischen Erfassung von Unterschieden zwischen Modellen eingeführt. SESE-Regionen können weitere SESE-Regionen verschachtelt umfassen, sie überlappen nicht und sind daher disjunkt. Eine SESE-Region kann als durch Gateways eingeschlossene Elemente bzw. Regionen angesehen werden.

Der Ansatz verfolgt zwei Phasen für die Verschmelzung: Zunächst werden alle Unterschiede zwischen zwei Versionen eines Prozessmodells erfasst. Hieraus wird ein Change-Log mit Operationen zur Auflösung der Unterschiede erstellt und automatisiert abgearbeitet. Wie in [18] wurde eine prototypische Implementierung in Form eines Eclipse Plug-Ins als Erweiterung des IBM WebSphere Business Modeler entwickelt.

Weidlich et al. (2009): Towards Vertical Alignment of Process Models – A Collection of Mismatches

Vor dem Hintergrund verschiedener Abstraktionsebenen von Prozessmodellen werden in [16] auftretende Unterschiede aus fünf Sichten beleuchtet, wobei die Unterschiede nicht nur im Zusammenhang mit der angewendeten Abstraktionsebene entstehen. Die gefundenen Unterschiede seien Ausdruck des jeweiligen Fokus und der unterschiedlichen Ebenen, auf denen gearbeitet wird. Die Arbeit nimmt eine Kategorisierung und informelle Beschreibung der Unterschiede vor (siehe Anhang B für eine Aufbereitung).

Weidlich et al. (2009): Vertical Alignment of Process Models – How Can We Get There?

In [21] werden drei Herausforderungen für vertikale Konsistenz (*vertical alignment*) herauskristallisiert: Charakteristika von Unterschieden (ähnlich und teilweise basierend auf [16]), semantische Unschärfe bei der Beschreibung von Modellkorrespondenzen und die Definition von oder Vorstellungen über Konsistenz.

Es werden potentielle Anwendungsmöglichkeiten der Techniken aus dem Themenbereich der Prozessintegration kritisch betrachtet. Daraus werden Forschungsfragen abgeleitet, die als Grundlage eines

Frameworks für *model alignment* dienen sollen; durch hohe und niedrige Abstraktionsebenen entstehe eine zu überwindende "Business-IT-Gap". Der Fokus der Arbeit liegt hierbei auf BPMN als Notationssprache und nicht beispielsweise auf dem Problem, BPMN mit BPEL abzustimmen.

Die Arbeit sieht drei Use Cases als Herausforderungen an:

1. *validation*: Modell auf hoher Abstraktionsebene als Ausgangspunkt zur Validierung eines Modells auf niedrigerer Ebene
2. *inter-model analysis*: Verschiedene Modelle auf niedriger Ebene als das Ausgangsmodell bilden Ablauf des Ausgangsmodells ab
3. *change propagation*: Veränderungen sollen sich auf assoziierte Modelle durchschlagen und diese aktualisieren, um konsistent zu bleiben

In dem in Kapitel 3 vorgestellten Konzept der vorliegenden Arbeit findet sich besonders Punkt 1 im Rahmen der einfachen vertikalen Konsistenzprüfung wieder.

Eine automatische *change propagation*, die automatische Übernahme von Änderungen eines Modells in das Modell einer anderen Abstraktionsebene, wird in [21] als relativ unrealistisch angesehen, aber die automatische Feststellung von Regionen, die von Veränderungen beeinflusst wurden, wird als ein Gewinn angesehen. Manuelle Korrespondenzerstellung sei in der Praxis allerdings sehr aufwendig, evtl. unrealistisch und müsse empirisch geprüft werden.

Branco et al. (2012): An Empirical Study on Consistency Management of Business and IT Process Models

In [13] wird eine empirische Studie im Rahmen eines *business-driven engineering* Prozesses der brasilianischen Bank Banco do Nordeste do Brasil S.A. (BNB) vorgestellt. Untersuchungsgegenstand der Studie ist das Konsistenzmanagement von Geschäftsprozessen in der Praxis. Die Studie soll dabei helfen, den Entwicklungsprozess vom fachlichen zum technischen Modell in der Praxis besser zu verstehen, Strategien zum Aufdecken und Beseitigen von Inkonsistenzen zu verstehen, sowie zu klären, wie Werkzeuge dabei helfen können.

Die Studie besteht primär aus zwei Teilen. Der erste Teil besteht aus einer Analyse von fünf GPM Projekten mit mehr als 70 Geschäftsprozessmodellen der BNB, inklusive der vollzogenen Veränderungen und Change-Logs mit über 1000 Change-Requests. Der zweite Teil besteht aus Befragungen, um konkrete Probleme des Konsistenzmanagements zu erfassen. Die Befragungen bestehen aus semi-strukturierten, informellen und offen gestalteten Interviews mit 9 fachlichen und IT Anwendern, sowie Fragebögen, die an 23 fachliche und IT Anwender gesendet wurden. Insbesondere Äußerungen zum Vorgehen zur Abgleichung von Modellen sind in das Konzept der vorliegenden Arbeit eingeflossen. So wünschte sich ein Interviewpartner eine visuelle Gegenüberstellung von Modellen verschiedener Abstraktionsebenen, statt diese großformatig auszudrucken.

Im Rahmen der Analyse konnten (wiederkehrende) Muster identifiziert werden, die in der Praxis und in den analysierten Modellen benutzt werden, um das fachliche Modell hin zum Implementierungs-

modell zu verfeinern (siehe Anhang A, B). In der Praxis fanden sich drei in der Prozessmodellierung verwendete Abstraktionsebenen *business*, *technical* und *implementation* wieder, auf denen parallel durch verschiedene Akteure gearbeitet wird.

Die Analyse der Prozessmodelle hinsichtlich ihrer Veränderungen und Konsistenz fand durch manuelle Identifikation korrespondierender Modellelemente und -fragmente statt. Die Korrelation zwischen Beschreibungen der Change-Requests und den veränderten Artefakten dienten als Basis für erstellte Korrespondenzen über mehrere Abstraktionsebenen hinweg und zur Auswertung typischer Veränderungen.

Als Schwächen der Studie wird der geringe Umfang an Befragten und die Beschränkung auf ein Unternehmen einer Branche gesehen. Daneben können durch die notwendige Übersetzung in/aus dem Portugiesischen Missverständnisse aufgetreten sein, sowie darüber hinaus subjektive Antworten getroffen worden sein, die aus Zeitmangel getroffen worden sein können.

Branco et al. (2012): Matching Business Process Workflows across Abstraction Levels

In [22] wird ein Algorithmus zur automatischen Erstellung von Korrespondenzen einzelner Aktivitäten zweier Prozesse über mehrere Abstraktionsebenen hinweg vorgestellt. Er basiert in Teilen auf den Ergebnissen der empirischen Studie zum Konsistenzmanagement in [13].

Die Evaluierung des Algorithmus erfolgt durch die Anwendung auf 26 Paarungen der Prozessmodelle der Abstraktionsebenen *business-technical* und *technical-implementation*. Der Algorithmus hat zwei Phasen, *attribute matching* und *structure matching*.

Zuerst werden Ähnlichkeiten durch Modellattribute wie Aktivitätstyp und Name festgestellt, welche dann auf Basis der Modellstruktur verfeinert werden. Die Annahme ist, dass es ein gleicher Prozess ist, aber verschiedene Abstraktionsebenen vorliegen.

Die Basis für das Matching ist die Repräsentation des Prozessmodells als *Process Structure Tree* (PST) und SESE-Regionen. Die Blattknoten im PST sind die Modellelemente, innere Knoten sind SESE-Regionen. Zur Erstellung von Korrespondenzen werden unter anderem aus der Linguistik bekannte Bigramme (beispielsweise zwei aufeinanderfolgende Satzfragmente; in diesem Falle Bestandteile des Modellelement-Textes oder Namens) verwendet.

Die *overall precision* (korrekt ermittelte Korrespondenzen) des Ansatzes liegt bei 93%, bei einem Recall von 70% (Abdeckung aller Verbindungen) für 622 manuell durch Domänenexperten erstellte Korrespondenzen, von denen 438 korrekt entdeckt wurden, 32 False-Positive und 184 False-Negative waren. Die Rate fiel bei Korrespondenzen der Typen 1:n und m:n geringer aus. Insgesamt gibt es eine Schwankung zwischen 87% und 96%.

Als Schwächen der Studie wurde festgestellt, dass mehr Tests nötig seien; der Algorithmus wurde nur durch BNB-Modelle getestet.

La Rosa et al. (2013): Business Process Model Merging

[23] stellt eine Methode zum schrittweisen Verschmelzen konfigurierbarer Geschäftsprozessmodelle zu einem konsolidierten Modell vor. Korrespondenzen dienen auch in diesem Ansatz für weitere Schritte eines dreistufigen Algorithmus.

Zunächst wird zu vergleichenden Elementen auf Basis ihrer Typen und Bezeichnung ein Ähnlichkeitsmaß zugewiesen. Dann werden Substitutionen, wie hinzugefügte und entfernte Elemente im festgestellten Korrespondenzbereich, ermittelt und diese Informationen zur Verschmelzung angewendet.

Ein konfigurierbares Prozessmodell beschreibt eine integrierte Repräsentation verschiedener Varianten eines Prozessmodells einer Domäne, aber verschiedener Einsatzorte, die sich beispielsweise durch Ausführungspfade verschiedener Zahlungsmöglichkeiten innerhalb eines Bestellprozesses ausdrücken. Die Konfigurationsmöglichkeiten werden durch Gateways abgegrenzt und definiert.

Änderungen können außerdem in einer Übersicht veranschaulicht dargestellt werden, um eine Nachvollziehbarkeit der Veränderungen zwischen den Input-Modellen zu ermöglichen. Dies kann der vertikalen Konsistenzerhaltung dienen. Werkzeuge zu konfigurierbaren Geschäftsprozessmodellen sind in den von den Autoren betreuten Projekten "apomore" und "Process Configuration.com" verfügbar ([24], [25]).

Apomore bietet Funktionen zur Prozessentdeckung aus Event Logs via Process Mining und zur Verschmelzung dieser an. In einem integrierten Signavio-BPMN-Editor können konfigurierbare Modelle erstellt werden, mit questionnaire models verknüpft und schließlich auf Basis der Konfigurationswahl erstellt werden⁴. Es können auch verschiedene Geschäftsprozessmodelle auf Basis von Event Logs generiert werden und grafisch miteinander verglichen werden⁵.

Process Configuration.com bietet Forschungsergebnisse und Erfahrungen im Gebiet konfigurierbarer Geschäftsprozesse an. Studien werden durch die QUT BPM Research Group (Brisbane, Australien) in Zusammenarbeit mit der TU/e BPM Research Group (Eindhoven, Niederlande) und der Software Engineering Group der Tartu Universität (Tartu, Estland) durchgeführt und Implementierungen als kostenfreie Werkzeuge angeboten. Die grundlegende Forschungsarbeit ist frei zugänglich.

Küster et al. (2015): Supporting different process views through a Shared Process Model

In [20] werfen verschiedene Sichten auf einen Prozess das Problem der Synchronisierung und Konsistenz bei Veränderungen der Modelle auf. Das vorgestellte *Shared Process Model (SPM)* soll als Ansatz zur Synchronisierung einer fachlichen und einer technischen Abstraktionsebene unter Beibehaltung der Modelle der verschiedenen Ebenen dienen. Die Arbeit stellt detaillierte Anforderungen, ein Lösungskonzept und die prototypische Implementierung als Machbarkeitsstudie des SPM vor. Die Ergebnisse aus [20] sind in [26] im Rahmen eines technischen Berichts ausführlicher dargestellt.

⁴ Für eine Vorstellung des Werkzeugs, siehe

<https://www.youtube.com/watch?v=DGeUyLImBsk&list=PLUF8mYs5Y5i5sHj6JMqDtPPvuAeJdBRHx>

⁵ Siehe <https://www.youtube.com/watch?v=XPqx99R74pc>

2. Grundlagen und Begriffsdefinitionen

Durch die Arbeit soll die folgende Frage geklärt werden: Wie können Sichten auf denselben Prozess, aber auf verschiedenen Abstraktionsebenen, konsistent gehalten werden und wie können die Änderungen automatisch propagiert werden? Es wird eine, in Abgrenzung zur BNB Studie, neue Kategorisierung von Veränderungen vorgestellt, die in Tabelle 3 dargestellt ist.

Das Lösungskonzept sieht eine Business- und IT-Sicht vor. Durch *get / put* Operationen wird das SPM mit einem Repository synchronisiert. Durch als *public* oder *private* definierte Änderungen wird bestimmt, welche Elemente synchronisiert und gegebenenfalls in die andere Abstraktionsebene propagiert werden; beispielsweise kann ein IT-Akteur einen business-relevanten Task in das Business-Level synchronisieren und nur für IT relevante Tasks als *private* nicht synchronisieren bzw. propagieren.

Es werden drei Statuszustände hinsichtlich der Konsistenz der Modelle bzw. Sichten definiert: *business conformance*, *IT conformance* und *business-IT*, welche durch Stakeholder geprüft werden sollen.

Anfängliche Korrespondenzen werden durch Klonen (vgl. [19]) des Ausgangsmodells auf Business-Ebene initialisiert. Im Rahmen der Prozessentwicklung vorgenommene private Änderungen betreffen nur eine Sicht, öffentliche Änderungen müssen propagiert werden. Aus der BNB Studie wird der Aspekt der Abdeckung aller Business-Level Tasks als Basis für eine Konsistenz aufgegriffen. Daneben wird die Erhaltung des Ablaufverhaltens und Reihenfolge der Business-Sicht beachtet.

Kategorie	Beschreibung
komplimentäres Implementationsdetail	Ausführungsdetails werden für eine Aktivität hinterlegt, ohne dass sich die Position oder die Funktion der betroffenen Aktivität verändert
Formalisierung und Umbenennung	lokale, nicht-strukturelle Änderungen; Spezifizierung von Typen, z.B. vom unspezifischen Task zum Service Task, Human Task etc.; analog vom nichttypisierten Zwischenereignis zum spezifischen Ereignis; Umbenennungen auf IT Level
Verfeinerung des Verhaltens und Refactoring des Ablaufs	hierarchische Verfeinerung / Subsummierung / Refakturierung; Aktivität auf hoher Abstraktionsebene wird in Subprozess mit gleichen In- und Outputverhalten transformiert oder in Call Activity ausgelagert
Hinzufügen und Entfernen von Aktivitäten	manche Business-Level-Tasks sind nicht auf ausführbarer Ebene vorhanden oder nötig; nur für IT-relevante Aktivitäten hinzugefügte Aktivitäten wie Script Task oder Aktivitäten zur dauerhaften Speicherung von Daten
zusätzliches Verhalten	Beispielsweise Behandlung von technischen Fehlern, z.B. um fehlgeschlagene Serviceaufrufe oder Timeouts zu behandeln
Korrektur und Überarbeitung des Sequenzflusses	auf Input-Output bezogene Reihenfolgeänderungen, die den Datenfluss beeinflussen

Tabelle 3: Neue Kategorisierung von Veränderungen

Die Arbeit definiert zwei Konsistenzbegriffe, starke und schwache Konsistenz.

1. Starke Konsistenz: Nach Klonen stark konsistent, danach durch z.B. Einführung von Boundary Events und die entstehenden neuen Sequenzpfade ist die Konsistenz nicht mehr stark; private Elemente müssen in bestehende Ablaufreihenfolge eingebettet sein und dürfen nicht zu neuem Verhalten führen
2. Schwache Konsistenz: Private, nur für IT relevante Boundary Events und private Fehlerbehandlungspfade sind erlaubt

Das prototypisch implementierte Werkzeug bietet Operationen als Funktionsbündel für Veränderungen an, die in einem Änderungslog gespeichert als Skript zur Ausführung während der Synchronisation verwendet werden. Sie basieren teilweise auf den in [22] dargelegten Methoden. Zusammengesetzte Änderungen wurden als notwendig erachtet, da eine zu feine Granularität der Änderungen als unbrauchbar erachtet wurde, vor allem in Bezug auf teilautomatisierte Operationen, die Intervention durch menschliche Benutzer erfordern (wie die Bestimmung eines *insertion-points* für hinzugefügte Aktivitäten).

2.4 BPMN-Modellierungswerkzeug bpmn-js

Bpmn-js⁶ ist ein webbasiertes Open Source Werkzeug zur Darstellung und Modellierung von Geschäftsprozessmodellen auf Basis von BPMN 2.0. Entstanden ist es Anfang 2014 aus dem Camunda-Umfeld heraus. Camunda⁷ ist eine Open Source Workflow-Engine, die neben der Ausführung von Geschäftsprozessen auf Basis von BPMN 2.0 auch Funktionen des Geschäftsprozessmanagements anbietet.

2.4.1 Architektur

Bpmn-js besteht aus verschiedenen modular aufgebauten Komponenten und ist in JavaScript geschrieben. Die Kernkomponenten sind diagram-js und bpmn-moddle. Diagram-js ist für die grafische Darstellung eines BPMN-Modells durch SVG-Elemente im Browser und Manipulation eines Modells zuständig. Bpmn-Moddle ist für die formelle Spezifikation eines BPMN-Modells verantwortlich und dient außerdem der Validierung eines BPMN-Modells im XML-Format. Zusammen bilden sie mit verschiedenen Erweiterungen und Features bpmn-js⁸.

Diagram-js stellt neben grafischen Benutzerschnittstellen zur Modellierung intern verschiedene Services bereit. *Canvas* dient der grafischen Darstellung und Manipulation eines BPMN-Modells. *Canvas* bietet auch die Möglichkeit, *Overlays* für Annotationen, oder zur farblichen Gestaltung von Modell-Elementen einzusetzen. Für den Darstellungsbereich des Modells wird eine *Canvas* in ein div-Element einer HTML-Seite eingebettet. *Overlays* können CSS-Definitionen verwenden⁹.

⁶ Bpmn-js ist zu erreichen unter <http://bpmn.io/toolkit/bpmn-js/> und <https://github.com/bpmn-io/>

⁷ Camunda ist zu erreichen unter <https://camunda.org/> und <https://github.com/camunda>

⁸ Siehe <http://bpmn.io/toolkit/bpmn-js/walkthrough/#bpmn-js-internals>

⁹ Siehe <https://github.com/bpmn-io/bpmn-js-examples/tree/master/colors#option-1-colors-via-css-styling>

3. Konzept

In diesem Kapitel wird das Konzept für ein Werkzeug (Tool) zur Unterstützung vertikaler Konsistenz zwischen zwei Prozessmodellen unterschiedlicher Abstraktionsebenen vorgestellt. Das Werkzeug soll dabei helfen, die vorgenommenen Schritte durch verschiedene Beteiligte des Modellierungsprozesses im Rahmen der Verfeinerung eines fachlichen Modells hin zu einem technischen Modell durch Korrespondenzen nachvollziehen zu können.

Es bietet hierzu einen visuellen Direktvergleich und die Modellierung von zwei sich auf unterschiedlichen Abstraktionsebenen befindlichen Modellen an. Die Elemente der jeweiligen Modelle werden hierzu durch Korrespondenzen verknüpft. Die visuelle Gegenüberstellung dient außerdem als Diskussionsgrundlage verschiedener Modellierungsbeteiligter.

Korrespondenzen zwischen den sich auf verschiedenen Ebenen befindlichen Modell-Elementen werden initial durch Klonen eines Prozessmodells erstellt. Im weiteren Verlauf der Modellierung und Verfeinerung werden diese mit Hilfe der grafischen Ansichten der beiden assoziierten Prozessmodelle manuell erstellt und manipuliert. Als Basis dient hierbei die eindeutige ID der jeweiligen Modell-Elemente.

Die erstellten Korrespondenzen dienen als Basis für eine einfache Konsistenzanalyse, die auf den Abdeckungsgrad geschäftsrelevanter Modellelemente abstellt¹³. Der Abdeckungsgrad drückt sich durch die Relation der Elemente im fachlichen Modell zu Umsetzung der Elemente im technischen Modell aus. Sind alle fachlichen Elemente Teil einer Korrespondenz und somit im technischen Modell verwirklicht, ist eine einfache Konsistenz von 100% gegeben.

Mit Hilfe farblicher Einfärbungen der Modell-Elemente kann visuell geprüft werden, wie die fachlichen Elemente im technischen Modell umgesetzt und verfeinert wurden. Bisher nicht umgesetzte Elemente im fachlichen Modell werden schnell identifiziert. Elemente, die Teil einer Korrespondenz sind, können ebenso identifiziert werden. Neben der visuellen Prüfung kann eine Liste aller Elemente und ihrer etwaigen Zugehörigkeit zu einer Korrespondenz erstellt werden.

3.1 Architektur

Die Architektur ist webbasiert gestaltet und sieht einen Client und einen Webserver vor. Das Tool läuft beim Client im Webbrowser und ist JavaScript und PHP basiert. Der Client kommuniziert mit dem Server und tauscht Daten per AJAX-Aufrufe, JSON-Datenformate und PHP-Skripte aus (s. Abbildung 6).

¹³ Primär sind dies alle Activities und Tasks, Gateways und Events. Konnektoren werden nicht berücksichtigt, da eine manuelle Korrespondenzerstellung und Verwaltung dieser durch das Tool sehr aufwendig wäre.

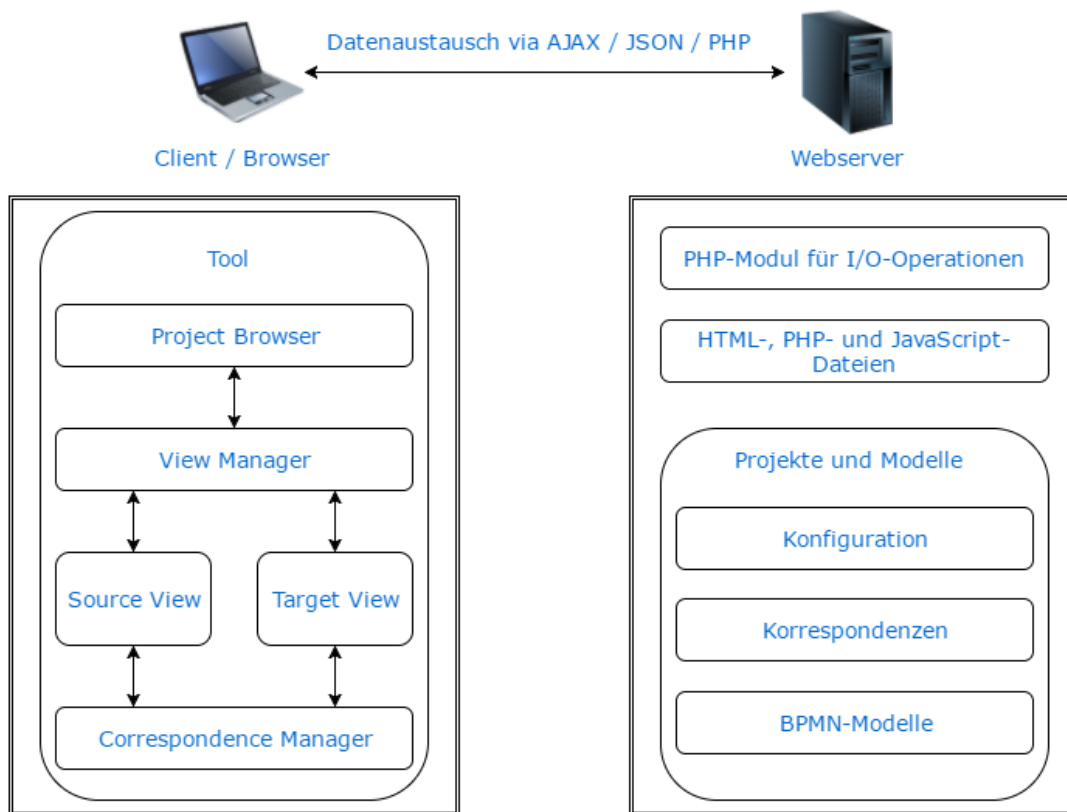


Abbildung 6: Top-Level Architektur des Tools

3.1.1 Zusammenspiel der Komponenten

Das beim Client laufende Tool beinhaltet einen Project Browser, der zur Verwaltung von Modellierungsprojekten dient. Durch ihn werden Projekte, Prozesse und Modelle verschiedener Abstraktionsebenen verwaltet. Die Modelle werden in den View Manager geladen und im Source und Target View zur Modellierung dargestellt. Der Correspondence Manager verwaltet die verknüpften Korrespondenzen der zwei geladenen Modelle.

Der Webserver dient primär dem Datenaustausch und Dateioperationen durch ein PHP-Modul. Der Server ist nötig, da der Client bzw. der Webbrowser nur begrenzte Zugriffsrechte auf das Dateisystem hat. Diese Lösung bietet den verschiedenen Beteiligten des Modellierungsprozesses Zugriff auf das Tool, beispielsweise durch einen zentralen Webserver.

Neben den benötigten Dateien des Tools (HTML- und JavaScript-Dateien, Modelle, Korrespondenzdaten), die an den Client übertragen werden, verwaltet und speichert der Server Projektstrukturen, Konfigurations- und Korrespondenz-Dateien, sowie BPMN-Modelle.

Das User Interface des Hauptbildschirms besteht aus vier Teilen (s. Abbildung 7). Einen zentralen Platz nehmen die Modell-Ansichten Source View und Target View ein. Sie stellen die Modelle grafisch übereinander dar und erlauben die Modellierung durch Kontrollelemente zum Manipulieren des Modells (vgl. Abbildung 27).

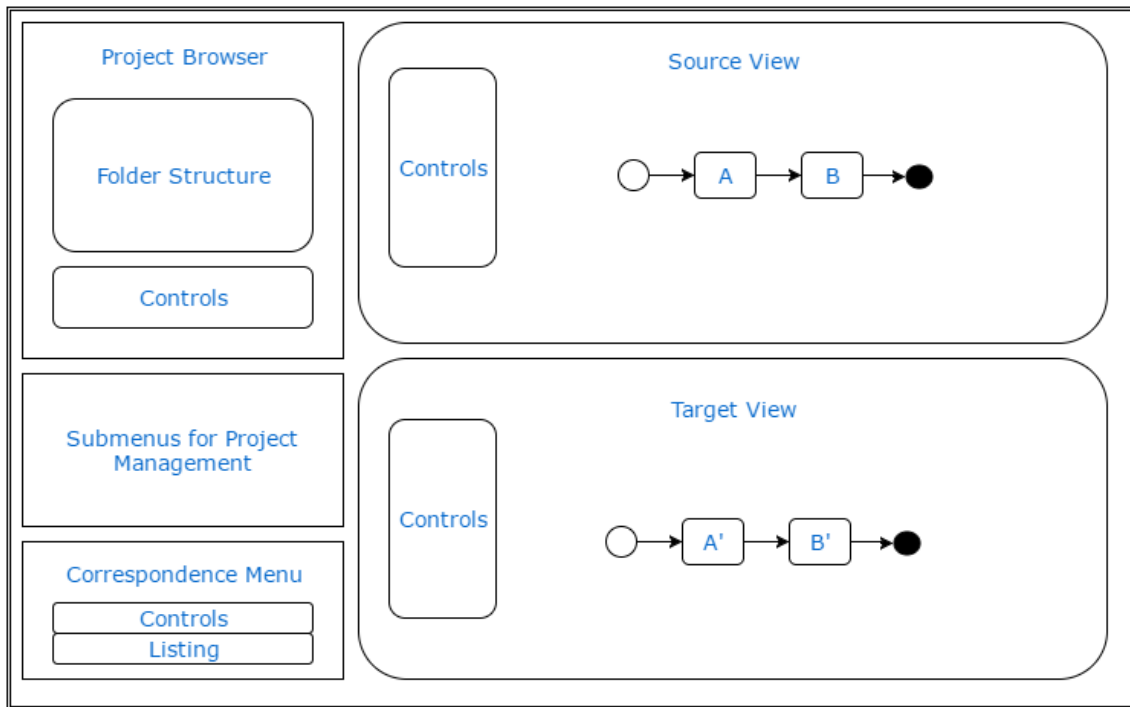


Abbildung 7: Mockup des User Interface des Tools

Auf der linken Seite befindet sich ein Project Browser, der die vorhandenen Projekte, Prozesse und Modelle in einer Baumstruktur anzeigt und darunter angesiedelt Kontrollelemente anbietet. In Abhängigkeit der ausgewählten Ebene im Projektbaum (Projekt, Prozess, Abstraktionsebene und Modell) werden Untermenüs eingeblendet, die zur Verwaltung der Projekte dienen. Verschiedene Masken dienen zum Einrichten neuer Projekte, sowie für den Upload von Modellen im Rahmen von eingerichteten Prozessen und Abstraktionsebenen. Die verfügbaren Modelle werden in Kombination zueinander durch Assoziationsdateien geladen, welche die Verknüpfung von beispielsweise einem fachlichen und einem technischen Modell beinhalten.

Unter dem Project Browser befinden sich ein Korrespondenzmenü und Kontrollelemente für die Korrespondenzverwaltung. Unter diesem Teil der grafischen Benutzerschnittstellen werden Informationen über die Korrespondenzen und die (einfache) vertikale Konsistenz in einer Liste aufgeführt.

Wird in einer der Views ein Element ausgewählt, werden Informationen über das Element im Kontext der bestehenden Korrespondenzen angezeigt. Vorhandene Korrespondenzen werden in den Views farblich hervorgehoben. Grün werden Korrespondenzen des Typs 1-1 hervorgehoben, gelb die Typen 1-n und n-1, blau die Typen 1-0 und 0-1, sowie rot die Elemente, die keiner Korrespondenz zugeordnet sind. Daneben wird die Ansicht angepasst, indem die Sicht auf ausgewählte Elemente fokussiert wird.

3. Konzept

Neben Informationen über die ausgewählten Elemente wird die Möglichkeit gegeben, ein Element zu fokussieren. An dieser Stelle werden auch Informationen zu einer einfachen vertikalen Konsistenz ausgegeben.

3.1.3 Klassenmodell des Werkzeugs

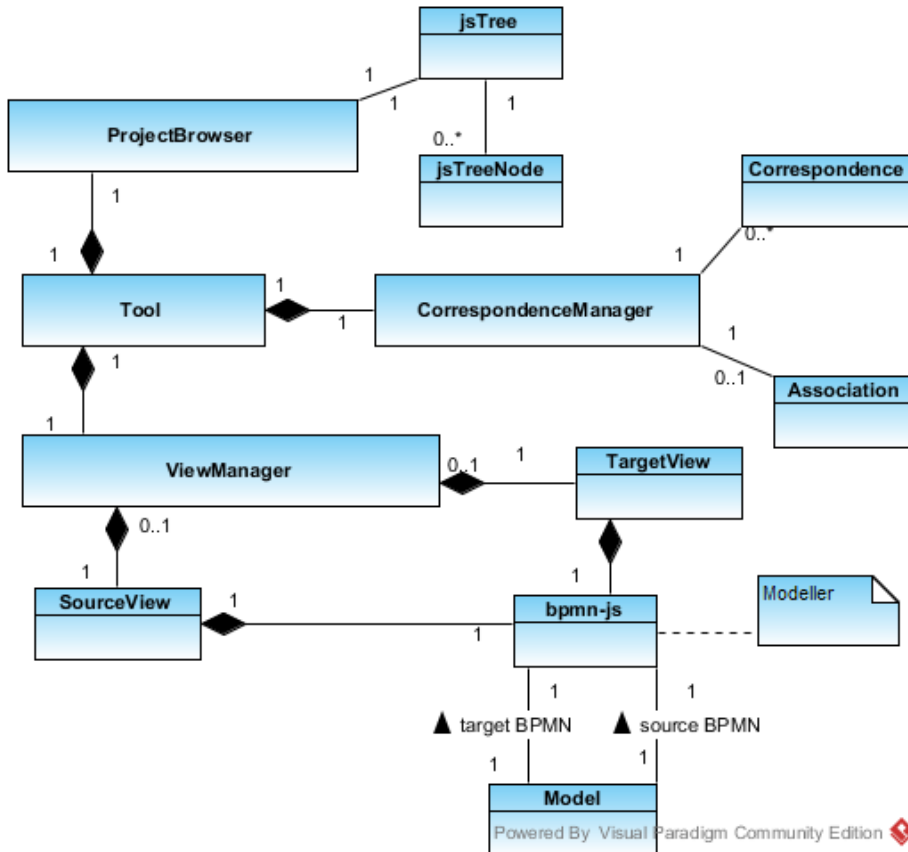


Abbildung 8: Klassenmodell des Tools

Das Klassenmodell¹⁴ des Tools (s. Abbildung 8) beinhaltet die Komponenten der bisher vorgestellten Architektur und zeigt ihre Abhängigkeiten auf. Das zentrale Element ist hierbei die Klasse Tool, von denen alle weiteren Komponenten abhängen. Sie dient primär als Namensraum für die weiteren Komponenten und zur Initialisierung der weiteren Klassen bzw. Komponenten. Hinzugekommen zu den vorgestellten Komponenten sind die Komponenten jsTree¹⁵ und jsTreeNode zur Darstellung der Projekte als Baumstruktur im Rahmen des ProjectBrowser. Es wird außerdem das Modellierungswerkzeug in Form von bpmn-js dargestellt. Die Aufgaben der Klassen Tool, ViewManager, ProjectBrowser, CorrespondenceManager und Views werden folgend beschrieben.

¹⁴ Für mehr Details sei auf den kommentierten Quellcode unter <https://github.com/azensen/ba-tool> oder auf der CD zur Arbeit verwiesen.

¹⁵ jsTree ist ein kostenloses Open Source Plug-In für jQuery zur Darstellung von Baumstrukturen und zu finden unter <https://www.jstree.com/>

ProjectBrowser

Die Klasse ProjectBrowser bietet Methoden für die Schaltflächen unterhalb der Anzeige der Projekte, Prozesse, ihrer Abstraktionsebenen und Modelle an. Sie ist zuständig für die Anzeige der Projektstruktur und von Untermenüs und Masken der gleichnamigen Komponente, sowie dem Laden und Speichern der Modelle.

ViewManager

Diese Klasse interagiert stark mit bpmn-js, welches als Source und Target View zur Darstellung und Manipulation von zwei Modellen dient. Sie verwaltet primär alle grafischen Aspekte und Funktionen des Tools, von der Auswahl von Elementen, bis zur fokussierten Anzeige und farblichen Hervorhebung von Korrespondenzen und in ihnen enthaltenen Elementen in der zugehörigen Ansicht. Außerdem generiert sie Informationen über vertikale Konsistenz und Korrespondenzen. Sie dient als Einstiegspunkt für die Erstellung und Manipulation von Korrespondenzen.

CorrespondenceManager

Die Klasse CorrespondenceManager ist für die Verwaltung und Manipulation von Korrespondenzen verantwortlich. Sie verwaltet eine Korrespondenzliste und bietet Methoden zum Laden, Ändern und Speichern dieser Liste an. Außerdem verwaltet sie die Assoziationsdaten, über die zwei Modelle miteinander verknüpft sind.

SourceView und TargetView

Diese Komponenten sollen durch das in Kapitel 2.4 vorgestellte Modellierungswerkzeug bpmn-js realisiert werden. Bpmn-js soll hierzu im Modellierungsmodus geladen werden. Die Source View stellt hierbei die Ansicht des Ausgangsmodells oder fachlichen Modells dar. Das dargestellte Modell dient als Ausgangspunkt für eine einfache vertikale Konsistenzprüfung. Die Target View stellt das über eine Assoziation verknüpfte, verfeinerte technische Modell dar.

3.2 Ausgewählte Use Cases

Abbildung 9 gibt eine Übersicht über erstellte Use Cases. Sie beinhalten die Projekt- und Prozessverwaltung, eine einfache Prüfung auf vertikale Konsistenz und Use Cases zur Korrespondenzerstellung (siehe Anhang C). Die Use Cases werden im Rahmen des Arbeitsvorgehens mit dem Tool in Kapitel 4.2 unter Beachtung der Anforderungen aus dem nachfolgenden Kapitel aufgegriffen.

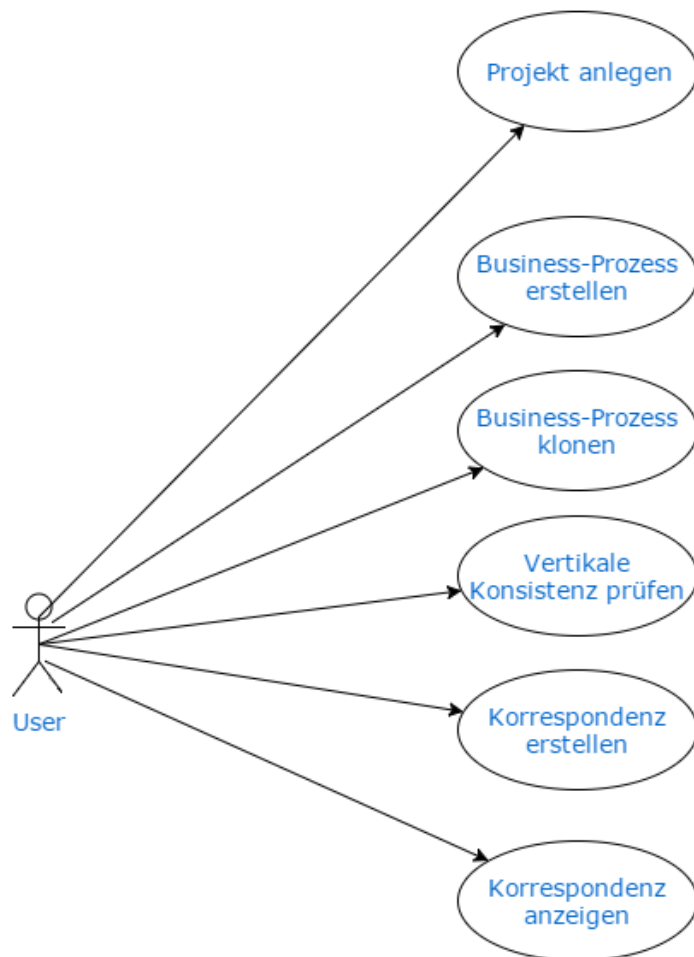


Abbildung 9: Use Cases

3.3 Anforderungen

Die Anforderungen an das Tool umfassen die Verwaltung von Projekten, Prozessen und Modellen, die Anzeige, Erstellung und Manipulation von Korrespondenzen, sowie eine einfache Konsistenzprüfung auf Basis der relevanten fachlichen Elemente, die in Korrespondenzen enthalten sind. Die Umsetzung der Anforderungen wird in Kapitel 4.2 als Vorgehensweise mit dem Tool dargestellt. Insbesondere bestehen folgende funktionale Anforderungen (s. Tabelle 4):

Nr.	Beschreibung
R1	Es sollen Projekte, Prozesse und Modelle auf verschiedenen Abstraktionsebenen verwaltet werden können, inklusive der Möglichkeit eines Datei Up- und Downloads der bearbeiteten Modelle. Die Modellierung zweier Modelle nebeneinander soll möglich sein.
R2	Es sollen, zwischen den Elementen zwei sich auf unterschiedlichen Abstraktionsebenen befindlichen Modellen eines Prozesses, Korrespondenzen durch Klonen eines Modells einer Abstraktionsebene auf eine andere zur Erstellung initialer Korrespondenzen erstellt werden können.
R3	Korrespondenzen zwischen zwei nebeneinander angezeigten Modellen verschiedener Abstraktionsebenen sollen durch das Tool angezeigt werden können, enthaltene Elemente farblich hervorgehoben werden und die Korrespondenzen durch das Tool editierbar sein.
R4	Es sollen weiterführende Informationen zu bestehenden Korrespondenzen in Abhängigkeit zu ausgewählten Elementen bereitgestellt werden und auf ausgewiesene Elemente einer Korrespondenz fokussiert werden können.
R5	Es sollen alle vorhandenen Elemente der verknüpften Modelle entsprechend ihrer Zugehörigkeit zu etwaigen Korrespondenzen farblich hervorgehoben werden können. Relevante Elemente seien zunächst alle Tasks und ihre Typisierungen, Events, Nachrichtenflüsse und Gateways. Nicht relevant seien der Prozess als Element des BPMN-Markups selbst, sowie Standardsequenzflüsse zur Verbindung der einzelnen Elemente (Konnektoren).
R6	Es soll eine einfache Konsistenzprüfung ermöglicht werden, bei der die Relation der geschäftsrelevanten Elemente des fachlichen Modells zu den im technischen Modell abgedeckten Elementen im Rahmen der Korrespondenzen ausschlaggebend ist.

Tabelle 4: Funktionale Anforderungen

3.4 Datenmodelle und Formate

Die verwendeten Datenmodelle basieren auf der JavaScript Simple Object Notation (JSON)¹⁶. JSON basiert auf der Notationsweise von Objekten in JavaScript und bietet sich daher als einfach zu verarbeitendes Datenaustauschformat an und wird von gängigen Webtechnologien unterstützt.

Neben den folgend vorgestellten Modellen und Formaten sei auf eine als .pcfg-Datei zu speichernde Projektkonfiguration hingewiesen, die ausgewählte Abstraktionsebenen ebenfalls im JSON-Format speichert. Die für den Prototyp vorgegebenen Abstraktionsebenen sind Business, IT-Architecture und IT-Execution. Diese werden unter anderem zum Klonen von Modellen einer Ebene auf eine andere Ebene verwendet.

3.4.1 Korrespondenzen

Das Schema für Korrespondenzen enthält als Felder den Korrespondenztypen, sowie die eindeutigen IDs der enthaltenen BPMN-Elemente für die Source- und Target-Views. Das folgende Beispiel (s. Abbildung 10) stellt eine Korrespondenz des Typen 1-n in JSON-Schreibweise dar:

```
{
  "corrType": "onemany",
  "source": "Source-ID-1",
  "target": [ "Target-ID-1", "Target-ID-2" ]
}
```

Abbildung 10: Korrespondenz des Typs 1-n in JSON-Schreibweise

Das Attribut "corrType" gibt hierbei den Korrespondenztypen an. Zulässig sind die in Kapitel 0 vorgestellten Korrespondenztypen, die als String codiert sind (*onezero*, *zeroone*, *oneone*, *onemany*, *manyone*). Mehrere Elemente, wie bei den Typen 1-n und n-1, werden als Array notiert. Gespeichert werden Korrespondenzdateien mit dem Namen des Ausgangsmodells und der Endung .corr, neben dem als .bpmn-Datei gespeicherten Ausgangsmodell (Source View).

3.4.2 Assoziation

Die als .asso-Datei gespeicherten Assoziationen werden wie die Korrespondenzen in JSON-Schreibweise gespeichert. Durch die Assoziation werden die Modelle unterschiedlicher Abstraktionsebenen miteinander verknüpft und in die entsprechenden Ansichten des Hauptbildschirms geladen. Es werden Modellnamen und relative Pfade, ausgehend vom Ordner html/web, in dem die index.html liegt, gespeichert. Die Ordnerstruktur des Projekt-Ordners ist hierbei folgendermaßen gestaltet: projects/Projekt-Name/Abstraktionsebene/Prozess-Name/Prozess-Modell. Die .asso-Datei befindet sich auf der gleichen Ebene wie die .corr-Datei.

¹⁶ Für eine ausführlichere Behandlung siehe <http://www.json.org/js.html>. Im Rahmen des Prototyps und wegen der geringen Komplexität wird auf eine formale Beschreibung des Schemas verzichtet.

4. Prototypische Implementierung

In diesem Kapitel wird die Umsetzung des im vorherigen Kapitel vorgestellten Konzepts beschrieben. Es wird im Rahmen der verwendeten Technologien die Umsetzung des Front- und Backend dargestellt und die Anwendung des Tools im Kontext der Use Cases beschrieben. Die Nutzung des Tools für eine einfache vertikale Konsistenzprüfung wird hierbei gesondert behandelt.

4.1 Verwendete Technologien und Umsetzung der Architektur

Grundlage für die Implementierung des Konzepts sind Webtechnologien, insbesondere JavaScript und jQuery auf Seite des Anwenders in einem modernen Webbrowser (entwickelt wurde mit Chrome 52.x, Firefox 43.x), sowie ein Webserver mit PHP-Modul auf Serverseite zur Verarbeitung von Zugriffen auf das Dateisystem.

Zur Darstellung und Manipulation von BPMN 2.0 Modellen und zur Umsetzung der Anforderungen wird das in Kapitel 2.4 vorgestellte Modellierungswerkzeug bpmn-js als Seed-Bundle verwendet.

4.1.1 Aufbau des Frontend

Das Frontend¹⁷ umfasst die im Konzept dargelegten Komponenten und ihren Aufbau. Sie ist durch dynamisierte HTML-Seiten realisiert. Für den Project Browser wird die auf jQuery aufbauende Bibliothek jsTree verwendet. Zur Erstellung der Projektstruktur wird durch ein PHP-Skript der Projektordner ausgelesen und in einem jsTree-spezifischen JSON-Format an jsTree übergeben. jsTree baut daraufhin aus HTML-Elementen die Baumstruktur auf.

In Abhängigkeit des gewählten Knotens oder Blattes im Baum stehen die Kontrollelemente unterhalb des Projektbrowsers zur Verfügung, um beispielsweise ein neues Projekt einzurichten. Im für Untermenüs vorgesehenen Bereich unter dem ProjectBrowser wird jeweils eine Maske zur Eingabe der erforderlichen Daten eingeblendet.

Wird im Baum ein Prozessmodell ausgewählt, werden zunächst automatisch die verfügbaren Korrespondenzen mit Hilfe der Assoziationsdatei per AJAX-Request geladen (siehe auch Konsole des Webbrowsers für weitere Informationen). Anschließend lassen sich die verknüpften Modelle in die entsprechenden Ansichtsbereiche mit Hilfe der Kontrollelemente und via AJAX-Request laden. Es steht außerdem das Korrespondenzmenü zur Verwaltung der Korrespondenzen zur Verfügung.

Ein implementierter Editiermodus, der durch Auswahl der "Edit Mode"-Checkbox aktiviert werden kann, fungiert als Weiche. Der Editiermodus erlaubt die Manipulation der Modelle, Elemente sowie die Erstellung und Manipulation von Korrespondenzen.

Im Normalmodus wird versucht, für ein ausgewähltes Element einer der beiden Ansichten eine Korrespondenz zu finden, in dem das Element enthalten ist. Schaltflächen, die bpmn-js neben selektierten Elementen als Kontextmenü bereitstellt, werden im normalen Modus nicht eingeblendet.

Die Schaltflächen des Korrespondenzmenüs erlauben die Manipulation von Korrespondenzen, die farblich hervorgehobene Anzeige aller Elemente zweier Elemente und die Anzeige der Informationen

¹⁷ Das Frontend ist durch die index.html im Ordner ba-tool/web/html/ erreichbar.

zur vertikalen Konsistenzprüfung. Unterhalb dieser Schaltflächen des Korrespondenzmenüs werden Informationen zu ausgewählten Elementen, Korrespondenzen und zur einfachen vertikalen Konsistenzprüfung angezeigt. Die aufgeführten Elemente können im entsprechenden View durch ein Kontrollelement neben den Informationen fokussiert werden.

4.1.2 Aufbau des Backend

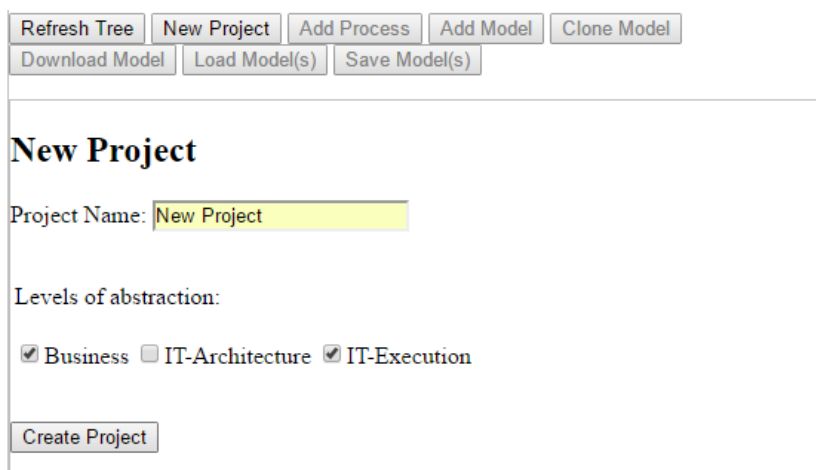
Zu Entwicklungszwecken wurde eine XAMPP-Installation¹⁸ und der mitgelieferte Apache-Webserver benutzt. Die Installation stellt unter anderem ein PHP-Modul bereit, um die verwendeten PHP-Skripte im Rahmen von JSON-Anfragen zu verarbeiten, sowie die benötigten Daten, Dateien und HTML-Seiten dem Client verfügbar zu machen. Das Backend dient primär dem Datenaustausch. Auf eine Datenbank wird im Rahmen des Prototyps verzichtet.

4.2 Arbeitsvorgehen mit dem Werkzeug

Folgend wird ein Arbeitsvorgehen mit dem Werkzeug von der Erstellung eines Projekts bis zur vertikalen Konsistenzprüfung zweier Modelle beschrieben. In diesem Rahmen werden die Anforderungen und Use Cases des Konzepts aufgegriffen. Der Einstiegsbildschirm des Werkzeugs befindet sich in *ba-tool/web/html/index.html*. Der Ordner für Projekte ist *ba-tool/projects/*.

4.2.1 Ein Projekt einrichten und einen Prozess anlegen (Anforderung R1)

Befinden sich im Ordner *ba-tool/projects* keine Projektdateien, ist der Project Browser leer. Um ein neues Projekt einzurichten, kann die entsprechende Maske (s. Abbildung 11) über den Menüpunkt "New Project" aufgerufen werden. Neben dem Projektnamen können die vordefinierten Abstraktionsebenen Business, IT-Architecture und IT-Execution gewählt werden. Die Schaltfläche "Create Project" erstellt die Ordnerstruktur für das neue Projekt und Ordner für die ausgewählten Abstraktionsebenen.



Refresh Tree New Project Add Process Add Model Clone Model
Download Model Load Model(s) Save Model(s)

New Project

Project Name: New Project

Levels of abstraction:

Business IT-Architecture IT-Execution

Create Project

Abbildung 11: Eingabemaske für ein neues Projekt

Hierzu werden die Eingaben per AJAX-Request an ein PHP-Skript gesendet. In der im Projektordner abgelegten Datei *project.pcfg* werden die ausgewählten Abstraktionsebenen für einen späteren Zugriff

¹⁸ XAMPP ist eine benutzerfreundliche Bündelung von Webserver, PHP- und Perl-Modul sowie Datenbank und zu erreichen unter <https://www.apachefriends.org/de/index.html>

weiterer Operationen und Masken im JSON-Format hinterlegt. Nach erfolgreicher Erstellung des Projekts erscheint eine Meldung und der Project Browser aktualisiert sich und zeigt das neue Projekt an.

Anschließend kann ein neuer Prozess innerhalb des Projekts angelegt werden. Hierzu wird zunächst die Baumstruktur des neuen Projekts entweder durch einen Doppelklick, oder durch einen Klick auf den kleinen Pfeil neben dem Namen aufgeklappt. Dann wird die gewünschte Abstraktionsebene für den neuen Prozess durch einen Klick ausgewählt. Der nun verfügbare Schaltfläche "Add Process" zeigt nun eine Maske für einen neuen Prozess an (s. Abbildung 12). In der Maske wird der Name des neuen Prozesses eingegeben und das BPMN 2.0 Modell im .bpmn-Format angegeben.

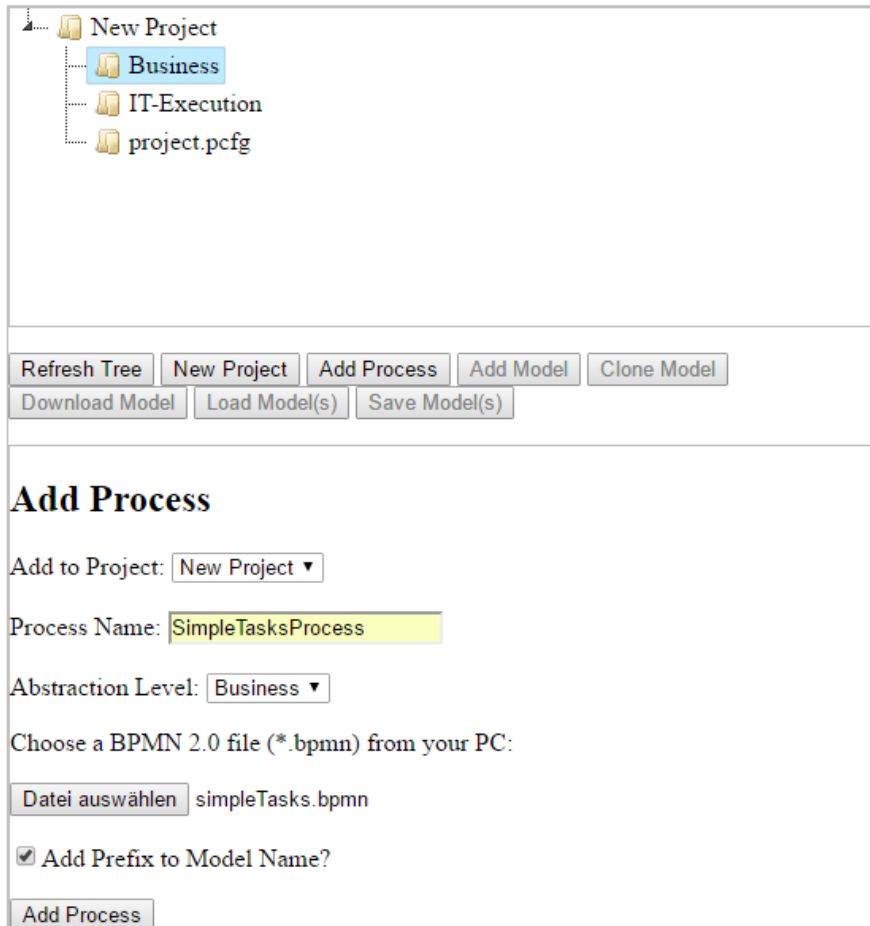


Abbildung 12: Einen neuen Prozess zu einem Projekt hinzufügen

Es kann gewählt werden, ob dem Modellnamen die ausgewählte Abstraktionsebene als Präfix vorangestellt werden soll¹⁹. Durch den Schaltfläche "Add Process" wird ein Ordner mit dem Prozessnamen innerhalb der gewählten Abstraktionsebene erstellt und die Modelldatei gesichert.

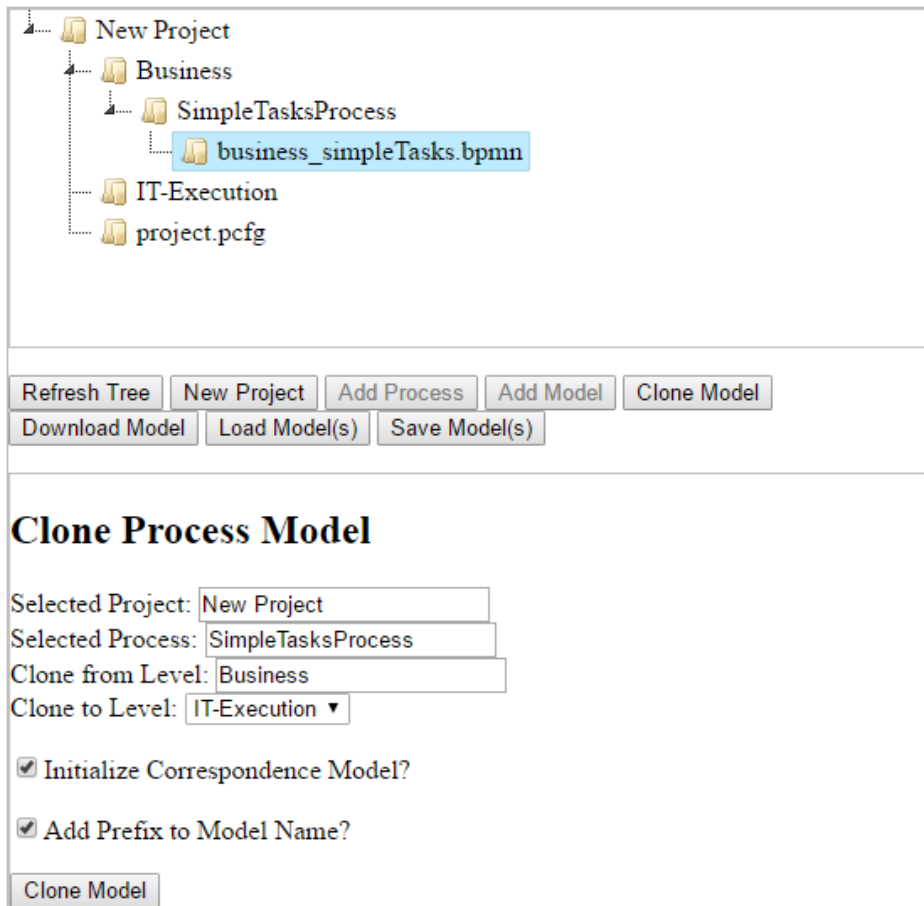
4.2.2 Klonen eines Modells und Initialisieren der Korrespondenzen (Anforderung R2)

Das gespeicherte Modell der Business-Ebene kann anschließend auf eine andere Ebene geklont werden (s. Abbildung 13). Zum Klonen wird in der Baumstruktur des angelegten Projekts die .bpmn-Datei

¹⁹ Wird beispielsweise eine Modelldatei "SimpleTasks.bpmn" gewählt und in die Ebene "Business" geladen, so wird die Modelldatei als "business_SimpleTasks.bpmn" gespeichert.

ausgewählt. Die Schaltfläche "Clone Model" wird verfügbar gemacht und zeigt eine Maske zum Klonen des ausgewählten Modells an.

Beim Klonen können zwischen den Elementen des Ausgangsmodells und des Klonmodells initiale 1-1 Korrespondenzen erstellt werden. Hierzu wird die Checkbox "Initialize Correspondence Model?" ausgewählt. Dem Dateinamen des geklonten Modells kann außerdem das Präfix der neuen Ebene vorangestellt werden. Das bisherige Präfix (in diesem Falle "business") wird entsprechend ersetzt.



Das verwendete PHP-Skript (*ba-tool/web/php/cloneModel.php*, function *createCorrespondences*) ist

Abbildung 13: Klonen eines Modells

so gestaltet, dass für die initialen 1-1 Korrespondenzen alle Flow-Elemente außer Konnektoren, der Prozess-Pool, Participants und Collaborations, Lanes und Labels für die initialen Korrespondenzen in Betracht gezogen werden. Die farbliche Hervorhebung nicht in Korrespondenzen enthaltener Elemente wie Pools und Participants führt zu grafischen Fehlern und Überlappungen in bpmn-js, welche die Ansicht und Navigation stark beeinträchtigen können.

4.2.3 Laden und Speichern von Modellen (Anforderung R1)

Zum Laden zweier Modelle wird neben der initialen Korrespondenzdatei die Assoziationsdatei mit den entsprechenden Informationen über die relativen Dateipfade zu den source- und target-.bpmn-Dateien benötigt. Die Assoziationsdatei wird neben den initialen Korrespondenzen im Rahmen des Klonierungsprozesses wie im vorherigen Abschnitt beschrieben erstellt.

Die verknüpften Modelle können durch Auswahl des Ausgangsmodells im Project Browser durch die Schaltfläche "Load Models" in die entsprechenden Views geladen werden. Gespeichert werden können die Modelle durch die Schaltfläche "Save Models". Die Korrespondenzen werden ebenfalls gespeichert. Ebenso können im Project Browser ausgewählte Modelle per "Download Model" heruntergeladen werden. Ein Modell kann, beispielsweise im Rahmen eines RoundTrips mit einem externen Modellierungswerkzeug, durch die Schaltfläche "Add Model" zu einer Abstraktionsebene hinzugefügt und mit gleichem Namen ersetzt werden.

4.2.4 Anzeigen, Erstellen und Manipulieren von Korrespondenzen (Anforderung R3, R4)

Sind die verknüpften Modelle geladen, können die in der .corr-Datei hinterlegten Korrespondenzen angezeigt werden. Zuvor kann außerdem geprüft werden, ob die an Hand ihrer Id in der .corr-Datei gespeicherten Elemente noch vorhanden sind. Dies geschieht durch die Schaltfläche "Make correspondences consistent". Es werden alle Elemente in den Korrespondenzen durchlaufen und geprüft, ob diese noch in den Modellen existieren. Ist ein Element in einem der Modelle nicht mehr verfügbar, wird die Korrespondenz entsprechend korrigiert.

Zum Anzeigen der Korrespondenzen bestehen drei Möglichkeiten. Die erste Möglichkeit besteht darin, in einer der Ansichten auf ein Element zu klicken. Das so ausgewählte Element wird entsprechend der Zugehörigkeit zu einer Korrespondenz farblich hinterlegt und enthaltene Elemente fokussiert (s. Abbildung 14). 1-1 Korrespondenzen sind grün, 1-n und n-1 gelb und 1-0 und 0-1 blau dargestellt²⁰.

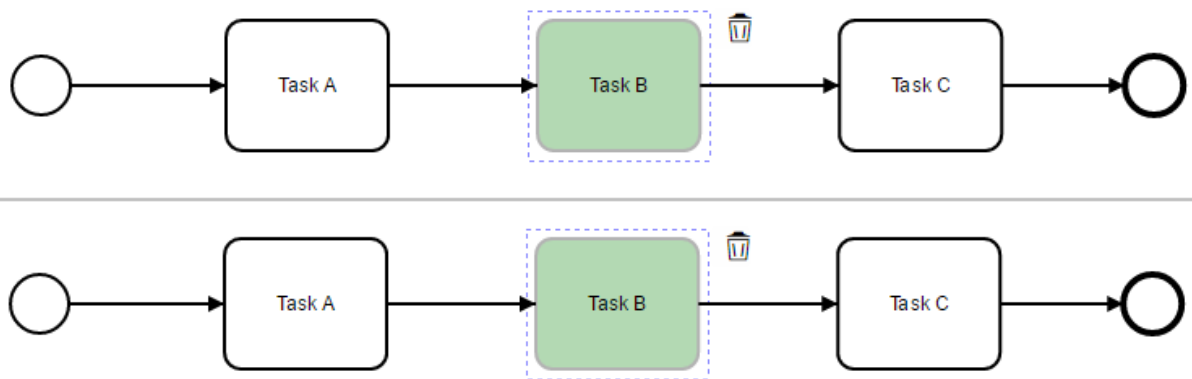


Abbildung 14: Eine farblich grün dargestellte 1-1 Korrespondenz

Informationen zur Korrespondenz werden unterhalb des Korrespondenzmenüs ausgewiesen. Neben den aufgeführten Elementen, die in der Korrespondenz enthalten sind, lässt sich durch die Schaltfläche "Focus View" das Element in der entsprechenden Ansicht fokussieren (s. Abbildung 15). Ist ein Element in keiner Korrespondenz hinterlegt, wird es rot eingefärbt. Die weiteren Möglichkeiten werden in 4.3.1 beschrieben.

²⁰ Angezeigt werden Korrespondenzen zu allen nicht ausgeschlossenen Elementen wie Pools, Lanes, Labels, Konnektoren.

Um eine Korrespondenz zu erstellen oder zu erweitern, kann zunächst ohne den "Edit Mode" zu aktivieren ein Element ausgewählt werden. Ist eine Korrespondenz für das Element enthalten, werden alle enthaltenen Elemente selektiert. Nun kann der Editiermodus aktiviert werden, um beispielsweise bei gedrückter Steuerungs-Taste ein weiteres Element einer angezeigten 1-1 Korrespondenz hinzuzufügen (s. Abbildung 16).

Correspondence Type: 1-1

Source Id(s)	Name	BPMN-Type	
Task_14h9162	Task B	Task	<input type="button" value="Focus View"/>
Target Id(s)	Name	BPMN-Type	
Task_14h9162	Task B	Task	<input type="button" value="Focus View"/>

Abbildung 15: Informationen über eine Korrespondenz mit Fokussierungs-Funktion

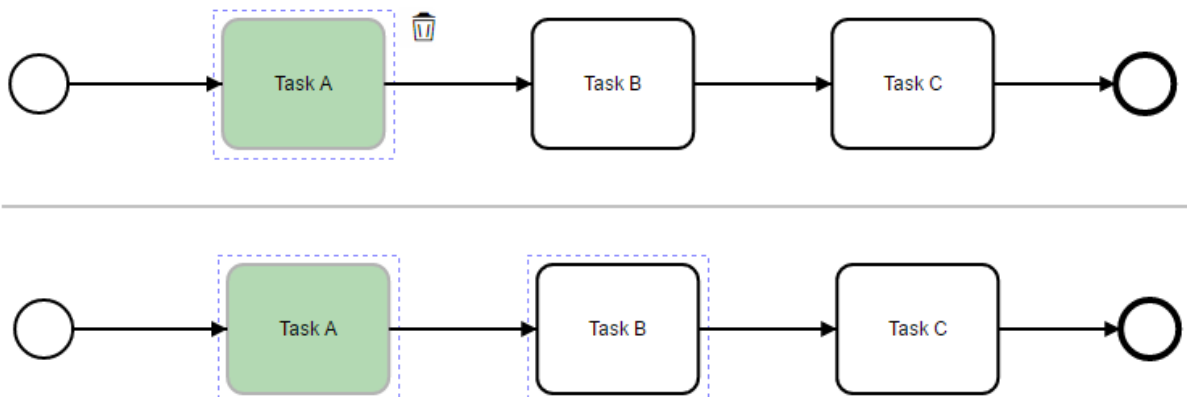


Abbildung 16: Erweiterte Selektion einer bestehenden 1-1 Korrespondenz

Anschließend wird die erweiterte Selektion durch die Schaltfläche "Turn selection into correspondence" in einer Korrespondenz gespeichert²¹.

Zuvor bestehende und durch die Auswahl betroffene Korrespondenzen werden entsprechend der Auswahl transformiert (s. Abbildung 17 und Abbildung 18). Gespeichert werden können die Korrespondenzen durch die Schaltfläche "Save correspondences".

Wird ein Element durch das Kontextmenü eines Elements gelöscht, werden die Korrespondenzen durchlaufen, das Element entfernt und die Korrespondenz entsprechend transformiert. Elemente, die Teil einer Korrespondenz sind, für die nach der Transformation kein source- oder target- Attribut vorhanden ist, werden nicht in eine 1-0 oder 0-1 Korrespondenz übertragen.

²¹ Korrespondenzen können nur mit zugelassenen Elementen erstellt werden. Primär sind dies alle Activities und Tasks, Gateways und Events.

4. Prototypische Implementierung

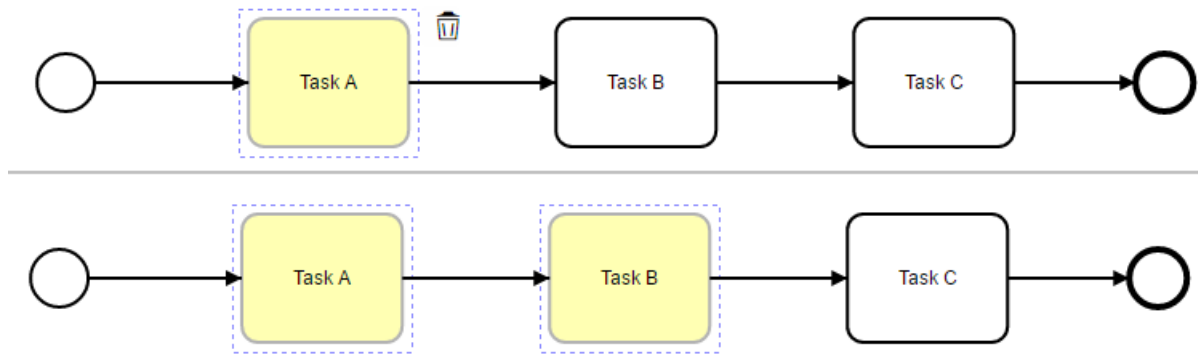


Abbildung 17: Neu erstellte 1-n Korrespondenz

4.3 Einfache vertikale Konsistenzprüfung

4.3.1 Visuelle Prüfung durch bestehende Korrespondenzen (Anforderung R5, R6)

Die zweite Möglichkeit zum Anzeigen der Korrespondenzen besteht durch die Schaltfläche "Highlight correspondences" (s. Abbildung 18). Es werden alle Elemente entsprechend der Zugehörigkeit zu einer Korrespondenz (oder keiner Zugehörigkeit) eingefärbt. Rot eingefärbte und somit nicht einer Korrespondenz zugehörige Elemente in der Source View müssen dem in Kapitel 2.3 vorgestellten Konsistenzbegriffs nach somit im Target View eingefügt werden und bzw. oder einer Korrespondenz hinzugefügt werden.

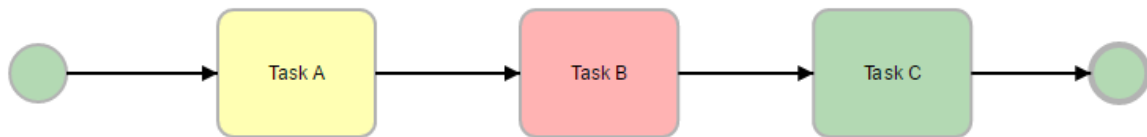


Abbildung 18: Highlighting der Elemente des Ausgangsmodells

Die dritte Möglichkeit zur Anzeige von Korrespondenzen besteht durch die Schaltfläche "Display table of matches". Hierbei werden alle Korrespondenzen mit ihrem Typ unterhalb des Korrespondenzmenüs aufgelistet. Neben den Ids, dem Namen und BPMN-Typen der enthaltenen Elemente gibt es die Möglichkeit, ein Element in der entsprechenden Ansicht zu fokussieren und zum Element zu navigieren (vgl. Abbildung 15).

4.3.2 Prüfung durch existierende Elemente

Basic Vertical Consistency Check

Vertical consistency matches for relevant element types based on source: 80%

Overall Elements	Relevant Elements	Matched Elements	Unmatched Elements
16	5	4	1

List of unmatched elements which are not in a correspondence:

Element Id	Element Name	Element Type	Focus
Task_14h9l62	Task B	Task	Focus View

Abbildung 19: Konsistenzprüfung auf Basis abgedeckter Elemente des Ausgangsmodells

Die Schaltfläche "Vertical Consistency Check" weist die Relation zwischen relevanter Elemente des Modells der Source View und der Elemente des Modells der Target View als Prozentwert an (vgl. Kapitel 0; s. Abbildung 19). Nicht durch eine Korrespondenz abgedeckte Elemente werden in einer

4. Prototypische Implementierung

Liste unterhalb des Korrespondenzmenüs aufgeführt und können fokussiert werden, um sie beispielsweise einer Korrespondenz hinzuzufügen.

5. Validierung des Prototyps durch Fallbeispiele

In diesem Kapitel soll das Tool anhand von zwei Beispielen auf seine Tauglichkeit und Schwachstellen geprüft werden. Die Beispielprozesse bzw. Modelle sind ein einfacher Urlaubsantrag und ein umfangreicheres Modells der Vorgänge an einem Geldautomaten, wie in [13] und [26] dargestellt.

5.1 Vorgehensweise

Die Modelle werden jeweils von der fachlichen in die technische Ebene überführt und verfeinert. Zur Erstellung von Korrespondenzen wird anfänglich die Klon-Funktion des Tools zur Erstellung initialer Korrespondenzen benutzt. Das fachliche Modell wird von der fachlichen auf die technische Abstraktionsebene geklont. Zur externen Modellierung wird das Modellierungswerkzeug Signavio²² in der Academics-Version benutzt, um einen Round-Trip zwischen bpmn-js und Signavio zu prüfen und das aufgekommene Problem neu generierter Ids durch bpmn-js im Rahmen von Typisierungsoperationen, wie in 5.2.3 beschrieben, zu umgehen.

Es werden zunächst die Ausgangsbasis und die vorgenommenen Veränderungen an den Modellen beschrieben. Dann werden die mit dem Werkzeug vorgenommenen Arbeitsschritte dargestellt, bevor die Anwendung des Tools validiert wird.

5.2 Fallbeispiel Urlaubsantrag

5.2.1 Ausgangsbasis und Veränderungen

Die Ausgangsbasis ist das fachliche Modell wie in Abbildung 20 dargestellt. Es wurden nichttypisierte Elemente verwendet. Neben dem Erfolgsfall "Urlaub wurde genehmigt" (oberer Sequenzpfad) wurde auch der Nichterfolg "Urlaub wurde abgelehnt" (unterer Sequenzpfad) modelliert.

Auslöser des Prozesses ist der Antrag auf Urlaub eines Mitarbeiters. Der Mitarbeiter gibt den Anfang und das Ende des gewünschten Urlaubszeitraums ein. Ist Resturlaub vorhanden, wird z.B. durch den Vorgesetzten auf Überschneidungen geprüft und der Antrag genehmigt, oder abgelehnt. Entsprechend wird der Antragsteller informiert. Wurde der Antrag genehmigt, wird zusätzlich die Vertretung des Mitarbeiters informiert.

Das verfeinerte technische Modell ist in Abbildung 21 dargestellt. Es wurden ein Pool und Lanes für den Mitarbeiter, das System und den Vorgesetzten eingeführt. Activities wurden zu User, Script, Service und Message Tasks typisiert. Die Prüfung auf Resturlaub und die Mitteilungen über eine Genehmigung oder Ablehnung des Urlaubsantrags finden automatisch statt. Außerdem wird automatisiert der Urlaubsantrag archiviert und der Resturlaub des Mitarbeiters angeglichen.

²² Signavio ist ein häufig verwendetes webbasiertes BPMN 2.0 Modellierungswerkzeug und zu erreichen unter <http://www.signavio.com/>

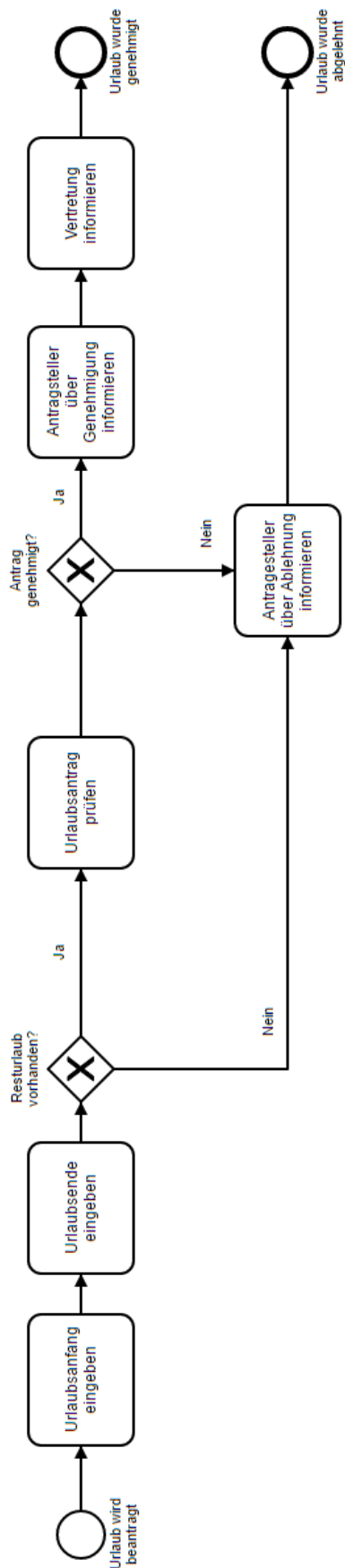


Abbildung 20: Fachliches Ausgangsmodell des Prozesses "Urlaubsantrag"

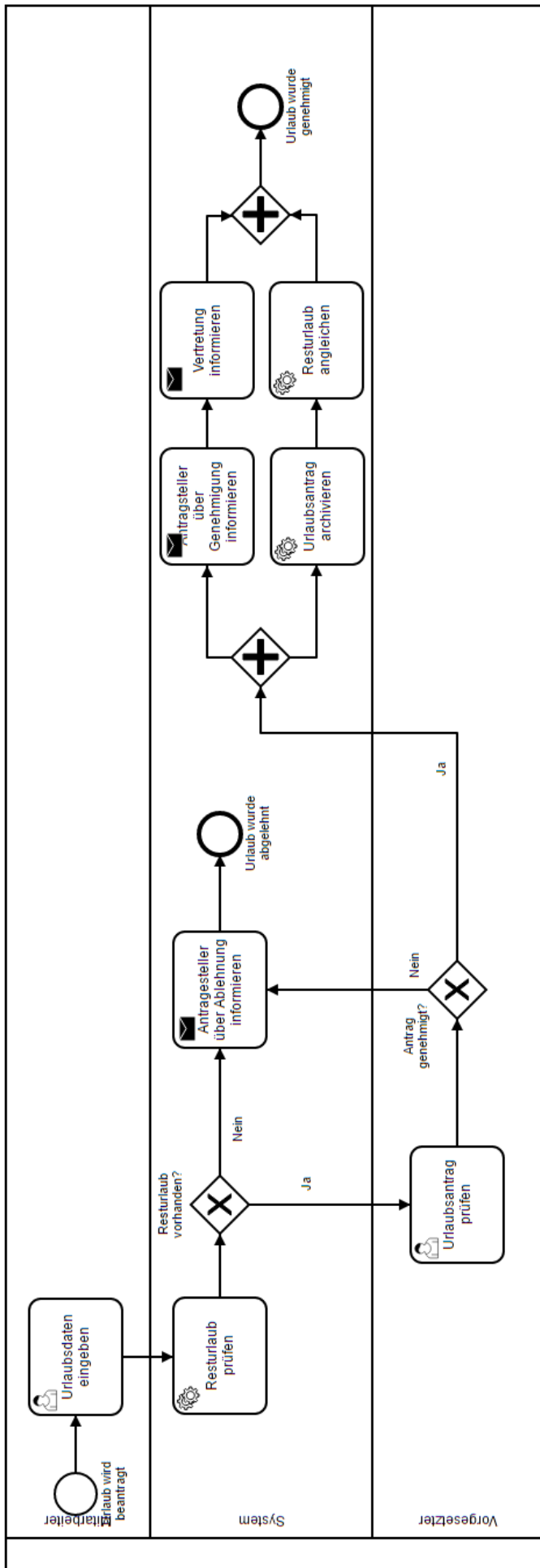


Abbildung 21: Verfeinertes, technisches Modell des Urlaubsantragsprozesses

5.2.2 Arbeitsschritte mit dem Werkzeug

Für den Urlaubsantrag wurden die in Kapitel 4.2 beschriebenen Schritte bis einschließlich des Klonens des Ausgangsmodells auf die IT-Execution-Ebene vollzogen. Dann wurden im Tool zunächst ein Pool mit drei Lanes für jeweils den Mitarbeiter und Antragsteller, das System und den Vorgesetzten modelliert. Die zwei Activities des fachlichen Modells nach dem Start Event wurden zu einem User Task zusammengelegt. Dies geschah mit der Absicht, die benötigten Daten dem Antragsteller in einer Maske zusammengefasst anzuzeigen. Die zwei exklusiven Gateways wurden beibehalten und in ihrer Position verändert den Lanes System und Vorgesetzter zugeordnet. Nach dem exklusiven Gateway "Antrag genehmigt?" wurde ein paralleler Gateway eingeführt. Die Benachrichtigung des Antragstellers und der Vertretung erfolgen durch Send Tasks. Die Archivierung des Urlaubsantrages und Angleichung des Resturlaubs auf dem Urlaubskonto des Mitarbeiters geschehen automatisiert durch Service Tasks.

Es muss eine Korrespondenz erstellt bzw. angepasst werden. Die Activity "Urlaubsende eingeben" muss in eine n-1 Korrespondenz mit der source "Urlaubsanfang eingeben", "Urlaubsende eingeben" und dem target "Urlaubsdaten eingeben" gefasst werden.

5.2.3 Validierung

Die Veränderungen durch die Verfeinerung waren geringen Ausmaßes mit lediglich zwei hinzugefügten Tasks und zwei parallelen Gateways. Ein Hauptteil bestand in der Typisierung von Activities zu Tasks.

In der aktuellen Version von bpmn-js gehen durch Typisierungsoperationen die Ids der Elemente verloren. Vor der Typisierung der Activities im Tool besteht eine vertikale Konsistenz von 91%. Die Activity "Urlaubsende eingeben" des fachlichen Modells ist keiner Korrespondenz zugeordnet und macht die 9% aus. Von 11 relevanten Elementen des fachlichen Ausgangsmodells sind 10 im technischen Modell abgedeckt.

Eine Konsistenz von 100% kann durch die Erstellung einer n-1 Korrespondenz mit den sources "Urlaubsanfang eingeben" und "Urlaubsende eingeben", sowie dem target "Urlaubsdaten eingeben" erzielt werden. Das Problem der Neugeneration von Ids nach Typisierungsoperationen ist den Entwicklern von bpmn-js bekannt und wurde als Issue aufgenommen²³. Es lässt sich festhalten, dass jedes neu hinzugefügte Element einer Korrespondenz zugeordnet werden muss.

Nach der Typisierung Prüfung der Korrespondenzen auf vorhandene Elemente (durch die Schaltfläche "Make correspondences consistent") durch bpmn-js beträgt die Konsistenz lediglich noch 45%. Von 11 relevanten Elemente sind nur noch fünf in Korrespondenzen enthalten.

Die Ids können durch die externe Verfeinerung mit Hilfe des Signavio Modellierungswerkzeuges beibehalten werden. In Signavio gehen die Ids bei Typisierungsoperationen nicht verloren. Eine weitere

²³ Für den aktuellen Status sei auf <https://github.com/bpmn-io/bpmn-js/issues/513> verwiesen.

Möglichkeit zum Erhalt von Element-Ids ist die aufwendige Beibehaltung von Ids durch manuelles Kopieren und Einfügen in einem Properties-Fenster, welches als Erweiterung von bpmn-js angeboten wird²⁴. Schneller und komfortabler sollte die Erstellung neuer Korrespondenzen sein.

5.3 Fallbeispiel ATM Machine

Ausgangsbasis für das fachliche technische Modell ist das in [13] vorgestellte Prozessmodell. Der Prozess stellt den Vorgang an einer Automated Teller Machine (ATM), einem Bankautomaten, dar. Grundlage war ein in der Praxis verwendetes, leicht vereinfachtes Modell einer brasilianischen Bank. Ein Kunde führt zunächst seine Bankkarte in die ATM ein und wählt anschließend eine Transaktion aus, bevor er seine persönliche Identifikationsnummer (PIN) eingibt, um fortzufahren und die Transaktion abzuschließen. Als Transaktionen sind Geldabheben ("Process Withdraw"), Kontostand anzeigen ("Consult Balance") und Umsätze anzeigen ("Consult Statement") als Activities modelliert. Der Kunde kann bis zu drei Mal einen falschen PIN eingeben (Prüfung durch Gateway "PIN is valid?"), bevor die Transaktion abgebrochen wird. Ist die Karte ungültig (Prüfung durch Gateway "Card is valid?"), wird der Prozess ebenso abgebrochen (Gateway "Attempts <=3?" und Activity "Cancel Transaction").

5.3.1 Ausgangsbasis und Veränderungen

Das fachliche Modell (s. Abbildung 22) beinhaltet nur wenige Typisierungen in Form von User Tasks und keine technischen Details. Das technische Modell (s. Abbildung 23) dagegen beinhaltet diverse typische Veränderungen (vgl. Kapitel 2.2.2, Anhang A und [13], S. 10 ff.), um technisches Verhalten abzubilden.

So wird das Einführen der Kundenkarte, welches zuvor im fachlichen Modell als Activity nach dem Start Event modelliert war, nun als Start Event des technischen Modells in Form eines Signal Events modelliert.

Anschließend werden durch einen hinzugefügten Script Task Parameter für die Transaktion initialisiert und die Kundenkarte mit Hilfe von asynchronen Send und Receive Tasks validiert. Für die Validierungsanfrage wurde eine mögliche Zeitüberschreitung von acht Sekunden in Form eines unterbrechenden, an den Receive Task angehängten Timer Event modelliert.

Tritt eine Zeitüberschreitung ein, wird die Transaktion beendet. Hierzu wurde ein Link Event "Cancel" modelliert, in das alle Abbruch-Fälle durch Timeouts oder den Kunden münden und zu einer BPMN-Transaktion führt. Außerdem werden Maßnahmen getroffen, um einen konsistenten Zustand des Kundenkontos zu bewahren und vor Betrugsversuchen zu schützen. Hierzu wird die BPMN-Transaktion verwendet.

Die Transaktionsauswahl durch den Kunden wurde um ein an den User Task "Customer Provides Transaction Details" angehängtes regelgesteuertes Ereignis ergänzt, um einen Abbruch durch den Benutzer zu modellieren. Bricht der Benutzer die Transaktionswahl oder die Eingabe des seines PINs ab, oder macht er falsche Eingaben, wird der Sequenzpfad zum Link Event und zur Transaktion verfolgt.

²⁴ Siehe <https://github.com/bpmn-io/bpmn-js-properties-panel>

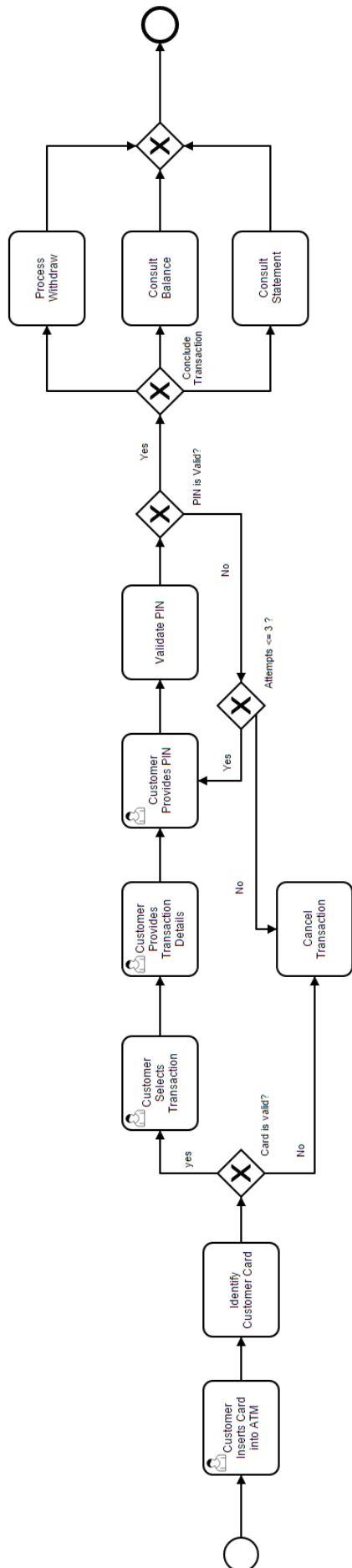


Abbildung 22: ATM Modell auf Business-Ebene

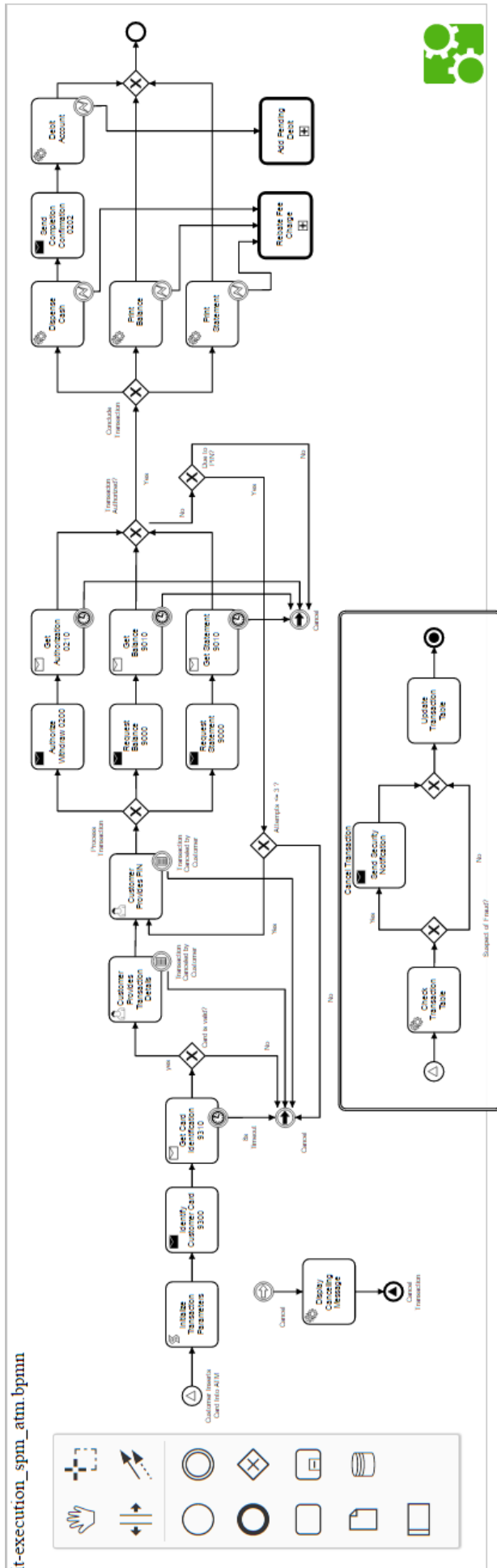


Abbildung 23: Technisches Modell des ATM Prozesses

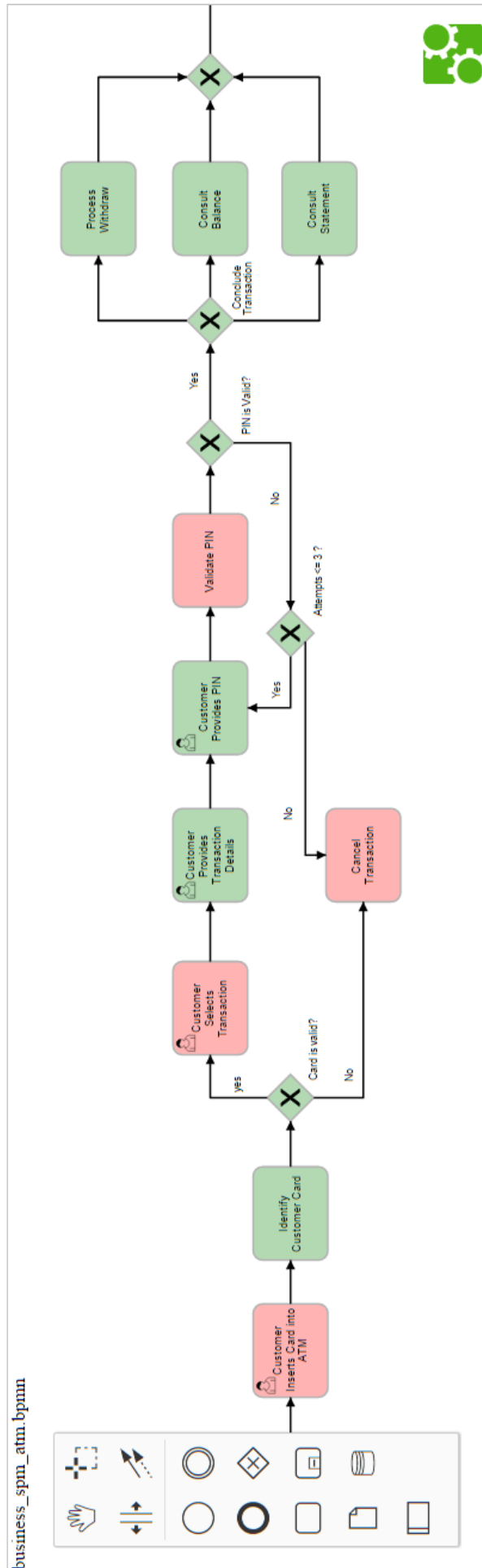


Abbildung 24: Anzeige der Korrespondenzen im fachlichen Modell nach Verfeinerung des geklonten Modells

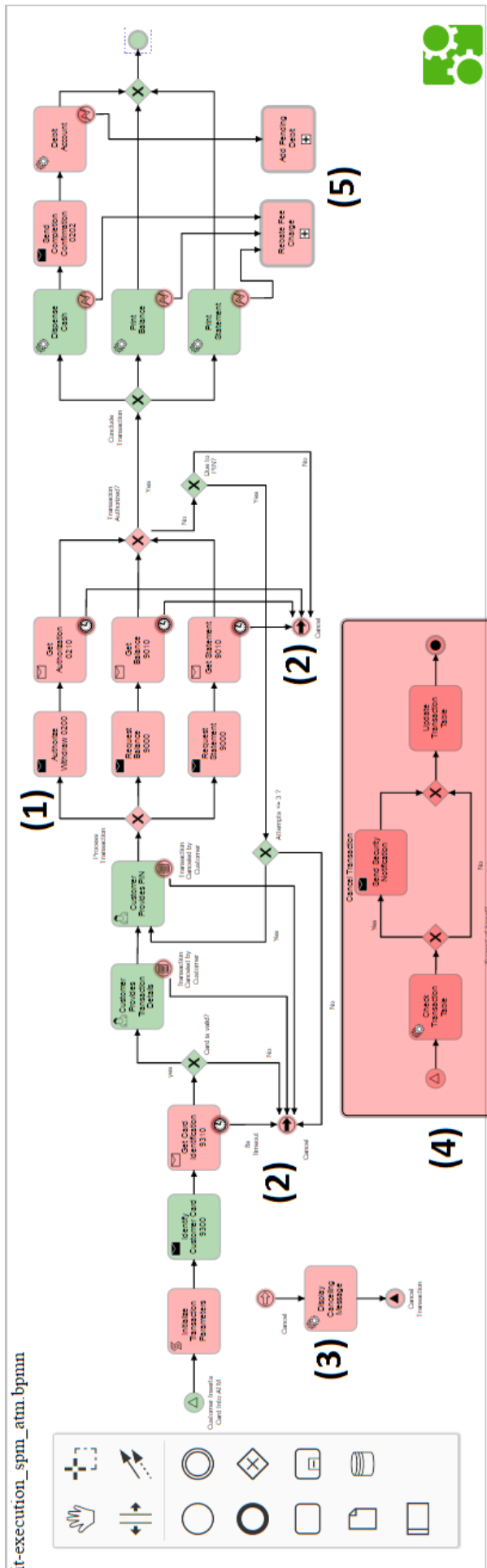


Abbildung 25: Anzeige der Elemente und ihrer Korrespondenzen im technischen Modell nach der Verfeinerung

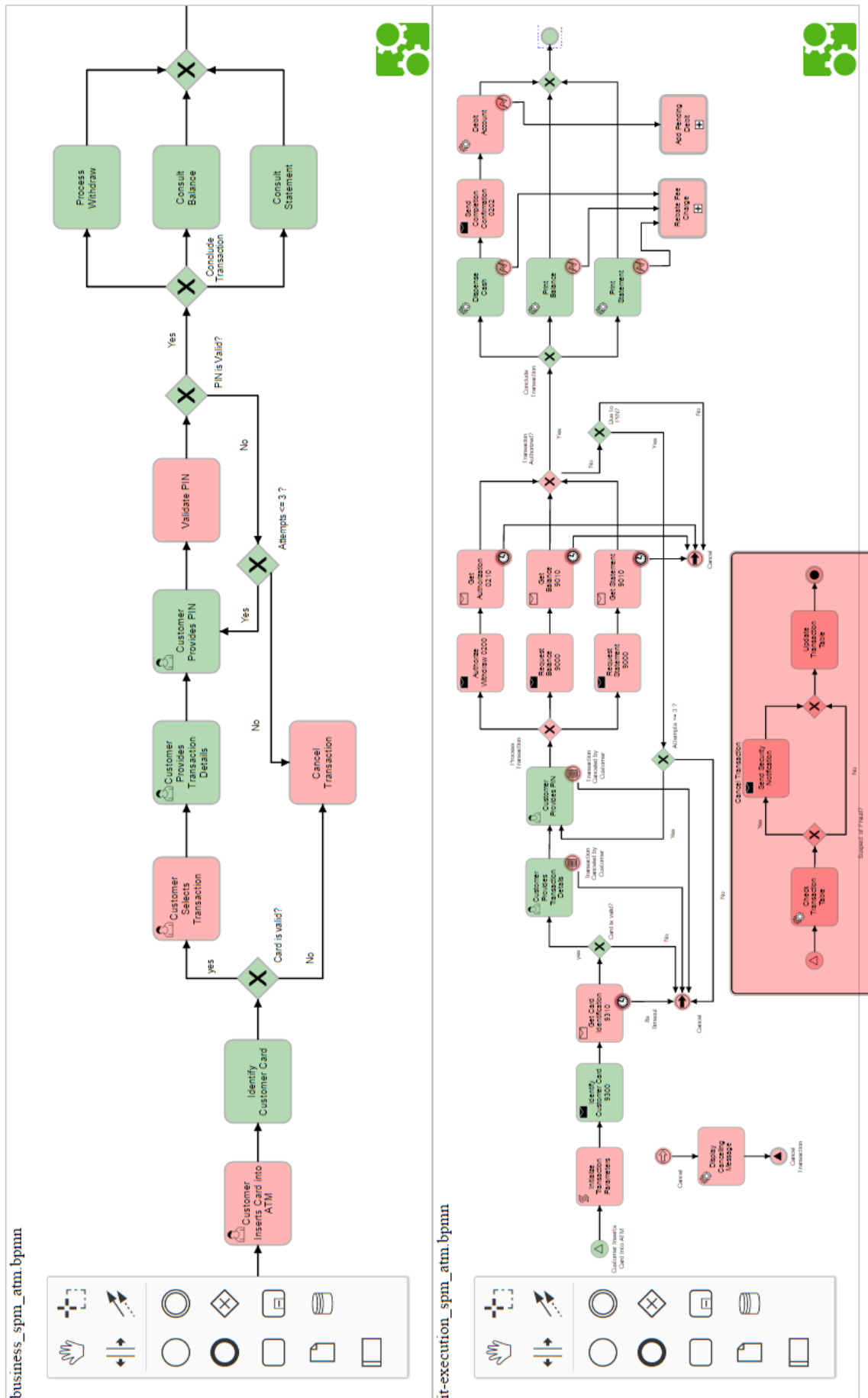


Abbildung 26: Visueller Direktvergleich nach Verfeinerung und ohne Anpassung der Korrespondenzen

Die Transaktionsauswahl wurde um eine nachgelagerte Autorisierung erweitert. Sie wird nach dem Gateway "Process Transaction" durch weitere Paare von Send und Receive Tasks modelliert (s. (1) in Abbildung 25). Ist die Transaktion autorisiert worden, wird der Sequenzpfad nach dem Gateway "Conclude Transaction" entsprechend der Auswahl durch den Kunden verfolgt.

Mögliche technische Fehler bei der Transaktionsabwicklung, die durch Service Tasks modelliert wurden ("Dispense Cash", "Print Balance" und "Print Statement"), sind durch angehängte fangende Error Events modelliert. Fehler werden in neu hinzugefügten, zusammengeklappten Call Activities behandelt (s. (2) in Abbildung 25).

Die Änderungen umfassen vor allem Umbenennungen von Activities (beispielsweise durch Hinzufügen der angesprochenen Protokolle zur Autorisierung der Kundenkarte), die Aufteilung fachlicher Elemente in zwei Elemente (Send und Receive Tasks), oder die Blockbildung (wie die Elemente zur Autorisierung der vom Kunden gewählten Transaktion, nachdem dieser seinen PIN eingegeben hat), Typisierungen der fachlichen Elemente und Fehlerbehandlung technischer Vorgänge durch angehängte Ereignisse.

Daneben wurden neue Gateways für die Transaktionsautorisierungen und Transaktionsabwicklung hinzugefügt und die Gateways "PIN is valid?" und "Attempts <=3?" des fachlichen Modells nachgelagert und verschoben. Für technische Ausnahme- und Fehlerbehandlung wurde außerdem eine BPMN-Transaktion eingeführt (s. (3), (4) in Abbildung 25).

5.3.2 Arbeitsschritte mit dem Werkzeug

Das fachliche Modell wurde zunächst in bpmn-js modelliert und im Tool im Rahmen eines neuen Projekts und Prozesses auf die Business-Ebene geladen. Dann wurde es auf die IT-Execution-Ebene geklont, um initiale Korrespondenzen zu erstellen. Um das Problem der durch Typisierungsoperationen in bpmn-js verloren gehenden Ids, wie in 5.2.3 beschrieben, zu umgehen, wurde das geklonte technische Modell extern mit Signavio verfeinert. Anschließend wurde das zuvor geklonte Modell im Projektordner ersetzt und die Korrespondenzen konsistent gemacht (vgl. Abbildung 26).

Der Arbeitsaufwand ist im Vergleich zum Urlaubsantragsbeispiel höher, wenn alle neu hinzugefügten Elemente im technischen Modell in Korrespondenzen enthalten sein sollen, wie in Abbildung 27 als eine Möglichkeit dargestellt. Hierbei sind die Send und Receive Task Paare in 1-n Korrespondenzen enthalten. Technische Elemente, wie angehängte Events, oder auch die nur im technischen Modell abgebildeten Call Activities (s. (5) in Abbildung 25), wurden in 0-1 Korrespondenzen erfasst.

5.3.3 Validierung

Nach der Verfeinerung blieb eine vertikale Konsistenz auf Basis der fachlichen Elemente von 76% bestehen (vgl. Abbildung 24). Nur vier Elemente des fachlichen Elements waren nicht mehr in Korrespondenzen enthalten. Sie sind nicht mehr enthalten, da Aufgaben zusammengelegt wurden, oder unter Berücksichtigung der technischen Infrastruktur durch Send und Receive Tasks modelliert wurden. So wurde die Eingabe der Kundenkarte direkt als Start Event modelliert, oder die Transaktionsauswahl durch den Kunden mit der Eingabe von Transaktionsdetails zusammengefasst. Die Validie-

Die Verfeinerung des PINs erfolgt im Rahmen eines neu hinzugefügten Blocks von Send und Receive Tasks (vgl. (1) in Abbildung 25). Die Activity "Cancel Transaction" wurde durch eine neue BPMN-Transaktion modelliert, in die die meisten Ereignisse zur Fehler- und Ausnahmebehandlung münden (vgl. (2), (3) und (4) in Abbildung 25). Insgesamt sind im technischen Modell 37 neue Elemente im Rahmen der Verfeinerung hinzugekommen, die nicht in Korrespondenzen enthalten sind oder erhalten blieben.

Der erzielte Konsistenzwert von 76% konnte durch beibehaltene Elemente und Typisierung der fachlichen Elemente im geklonten Modell erzielt werden. Da eine Vorlage existierte, die zur Modellierung des technischen Modells genutzt wurde, ist fraglich, wie hoch die erzielte Konsistenz ohne Vorlage und Berücksichtigung vorhandener Elemente zur Anpassung des fachlichen Modells gewesen wäre.

Die Erstellung möglicher Korrespondenzen (vgl. Abbildung 27) warf Fragen auf. Es war nicht eindeutig, ob Send und Receive Tasks zusammen als n-Seite in einer 1-n Korrespondenz erfasst werden sollten, oder diese Elemente rein technische Aspekte darstellen und somit in 0-1 Korrespondenzen zu fassen seien. Außerdem ist die Trennung von Transaktionsauswahl und nachgelagerter Autorisierung nicht klar in Korrespondenzen zu fassen, auch wenn eine 1-n Korrespondenz nahe liegen mag. Eine weitere Frage entstand durch die vielen rein technischen Details und wie diese in Korrespondenzen zu fassen seien, oder überhaupt erfasst werden müssen. Hierzu sollte eine Art Wizard zur Unterstützung der Erstellung vieler einzelner 0-1 Korrespondenzen in Erwägung gezogen werden.

6. Fazit und Ausblick

Der entwickelte Prototyp des Werkzeugs hilft dabei, Inkonsistenzen zwischen zwei Modellen auf verschiedenen Abstraktionsebenen aufzudecken, zu diskutieren und zu beseitigen. Er setzt dabei auf die Modellierungssprache BPMN 2.0, welche fachliche und technische Aspekte vereint und die Modellierung ausführbarer Geschäftsprozessmodelle unterstützt.

Das Werkzeug unterstützt die Verwaltung von Modellen auf verschiedenen Abstraktionsebenen, die aus unterschiedlichen Sichten auf einen Prozess durch die an der Prozessmodellierung beteiligten Akteure entstehen. Eine webbasierte Verwaltung und Bearbeitung der Modelle und Korrespondenzen zwischen den Elementen zweier Modelle ermöglicht ein leichtgewichtiges Arbeiten an den Modellen in gängigen Webbrowsern und somit einen einfachen Zugang.

Die visuelle Gegenüberstellung eines fachlichen und technischen Modells ermöglicht es, Verfeinerungen kontrolliert vorzunehmen. Die Korrespondenzen der Elemente zweier Modelle und insbesondere ihre grafische Hervorhebung helfen dabei, durch die Verfeinerung der Modelle entstehende Änderungen nachvollziehbar zu machen und Inkonsistenzen zu kontrollieren und zu beseitigen.

Korrespondenzen können intuitiv durch die grafische Benutzeroberfläche des verwendeten Modellierungswerkzeugs erstellt werden. Weiterführende Informationen zu vorhandenen Korrespondenzen und in ihnen enthaltenen Elementen geben Aufschluss über die vorgenommenen Änderungen, machen diese nachvollziehbar und bieten eine simple Dokumentationsgrundlage für weitere Diskussionen zwischen den Modellierungsbeteiligten. Domänenspezifisches Wissen der an der Modellierung beteiligten Akteure kann erfasst werden.

Eine Fokussierungsfunktion neben den ausgewiesenen Elementen einer Korrespondenz, oder neben den Elementen, die keiner Korrespondenz angehören, ermöglicht eine komfortable Navigation zu den entsprechenden Elementen im entsprechenden Modell.

Eine einfache vertikale Konsistenzprüfung gibt einen Überblick über die Umsetzung der geschäftsrelevanten Elemente des Ausgangsmodells auf der fachlichen Ebene in das technische Modell einer niedrigeren Abstraktionsebene. Nicht modellierte, geschäftsrelevante Elemente können visuell durch eine farbliche Hervorhebung identifiziert werden und entsprechend behandelt werden.

Die Validierung des Werkzeugs warf die Frage auf, wie komfortabel und somit nützlich eine manuelle Erstellung von Korrespondenzen ist. Der Arbeitsaufwand steigt mit der Anzahl von Veränderungen und hinzugefügten Elementen eines verfeinerten Modells. Die initiale Erstellung von Korrespondenzen durch Klonierung des Ausgangsmodells und sorgfältige Beibehaltung bestehender Elemente in einer anschließenden Verfeinerung helfen dabei, den Arbeitsaufwand zu reduzieren. Eine Vereinigung des Konzepts dieser Arbeit mit den rechnergestützten Methoden, wie sie in den verwandten Arbeiten vorgestellt wurden, erscheint sinnvoll, um den Arbeitsaufwand weiter zu reduzieren. Das vorgestellte, prototypisch implementierte Werkzeug dieser Arbeit sollte weiterentwickelt und im Rahmen einer empirischen Studie in der Praxis im Verbund mit den anderen Methoden der vorgestellten verwandten Arbeiten erprobt werden.

Literaturverzeichnis

- [1] Abts, D, Müller, W (Hrsg) (2010): Masterkurs Wirtschaftsinformatik. Kompakt, praxisnah, verständlich - 12 Lern- und Arbeitsmodule. 1. Auflage. Vieweg + Teubner, Wiesbaden.
- [2] Harmon, P (2014): Business Process Change. A Business Process Management Guide for Managers and Process Professionals. 3. Auflage. Elsevier/Morgan Kaufmann, Amsterdam.
- [3] Schwarzer, B, Krcmar, H (2014): Wirtschaftsinformatik. Grundlagen betrieblicher Informationssysteme. Schäffer-Poeschel Verlag, Stuttgart, Germany.
- [4] Schmelzer, HJ, Sesselmann, W (2013): Geschäftsprozessmanagement in der Praxis. Kunden zufriedenstellen, Produktivität steigern, Wert erhöhen. 8. Auflage. Hanser, München.
- [5] Gadatsch, A (2012): Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. 7. Auflage. Imprint Vieweg+Teubner Verlag, Wiesbaden.
- [6] Ahlrichs, F, Knuppertz, T (2010): Controlling von Geschäftsprozessen. Prozessorientierte Unternehmenssteuerung umsetzen. 2. Auflage. Schäffer-Poeschel, Stuttgart.
- [7] Object Management Group Inc. (2011): Business Process Model and Notation (BPMN) Version 2.0. <http://www.omg.org/spec/BPMN/2.0>.
- [8] Freund, J, Rücker, B (2014): Praxishandbuch BPMN 2.0. 4. Auflage. Hanser, München.
- [9] Fettke, P (2008): Business Process Modeling Notation. WIRTSCHAFTSINFORMATIK, 50(6):504–507.
- [10] van der Aalst, W (2014): Geschäftsprozessmodellierung. Die „Killer-Applikation“ für Petrinetze. Informatik-Spektrum, 37(3):191–198.
- [11] Houy, C, Fettke, P, Loos, P, van der Aalst, Wil M.P., Krogstie, J (2011): Geschäftsprozessmanagement im Großen. WIRTSCHAFTSINFORMATIK, 53(6):377–381.
- [12] Allweyer, T (2009): Geschäftsprozessmanagement. Strategie, Entwurf, Implementierung, Controlling. 3. Auflage. W3L-Verl., Herdecke.
- [13] Branco, M, Xiong, Y, Krzysztof Czarnecki, Küster, JM, Voelzer, H (2012): An Empirical Study on Consistency Management of Business and IT Process Models.
- [14] Duggen, KN, Grass, A, Weilkiens, T, Weiß, C (2015): Basiswissen Geschäftsprozessmanagement. Aus- und Weiterbildung zum OMG-Certified Expert in Business Process Management 2 (OCEB2) - Fundamental Level. 2. Auflage. dpunkt, Heidelberg, Neckar.

- [15] Hutchison, D, Kanade, T, Kittler, J, Kleinberg, JM, Mattern, F, Mitchell, JC, Naor, M, Nierstrasz, O, Pandu Rangan, C, Steffen, B, Sudan, M, Terzopoulos, D, Tygar, D, Vardi, MY, Weikum, G, France, RB, Kazmeier, J, Breu, R, Atkinson, C (Hrsg) (2012): *Matching Business Process Workflows across Abstraction Levels*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [16] Matthias Weidlich, Gero Decker, Alistair Barros, and Mathias Weske (2009): *Towards Vertical Alignment of Process Models - A Collection of Mismatches*, BPT Technical Report 05-2009.
- [17] Kemper, A, Eickler, A (2006): *Datenbanksysteme. Eine Einführung*. 6. Auflage. Oldenbourg, München [u.a.].
- [18] Felix Feger (2006): *Business Process Merging and Refactoring in Business-Driven Development*. Diplomarbeit, Otto-Friedrich-Universität Bamberg.
- [19] Christian Gerth (2007): *Business Process Merging. An Approach based on Single-Entry-Single-Exit Regions*. Diplomarbeit, Universität Paderborn.
- [20] Küster, J, Völzer, H, Favre, C, Branco, MC, Czarnecki, K (2015): *Supporting different process views through a Shared Process Model*. *Software & Systems Modeling*.
- [21] Weidlich, M, Barros, A, Mendling, J, Weske, M (2009): *Vertical Alignment of Process Models – How Can We Get There?* In: van der Aalst, W, Mylopoulos, J, Sadeh, NM, Shaw, MJ, Szyperski, C, Halpin, T, Krogstie, J, Nurcan, S, Proper, E, Schmidt, R, Soffer, P, Ukor, R (Hrsg), *Enterprise, Business-Process and Information Systems Modeling*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [22] Castelo Branco, M, Troya, J, Czarnecki, K, Küster, J, Völzer, H (2012): *Matching Business Process Workflows across Abstraction Levels*. In: Hutchison, D, Kanade, T, Kittler, J, Kleinberg, JM, Mattern, F, Mitchell, JC, Naor, M, Nierstrasz, O, Pandu Rangan, C, Steffen, B, Sudan, M, Terzopoulos, D, Tygar, D, Vardi, MY, Weikum, G, France, RB, Kazmeier, J, Breu, R, Atkinson, C (Hrsg), *Matching Business Process Workflows across Abstraction Levels*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [23] La Rosa, M, Dumas, M, Uba, R, Dijkman, R (2013): *Business Process Model Merging*. *ACM Transactions on Software Engineering and Methodology*, 22(2):1–42.
- [24] Marcello La Rosa (2009 - 2015): *Apromore Initiative*. www.apromore.org.
- [25] Marcello La Rosa et al. (2008 - 2014): *Process Configuration.com*. <http://www.processconfiguration.com/>.
- [26] Küster, J, Völzer, H, Favre, C, Branco, MC, Czarnecki, K: *Supporting different process views through a Shared Process Model*. Technical Report RZ3823, IBM, 2013.

Anhang

Anhang A

Typische Veränderungen im Rahmen der Verfeinerung von Geschäftsprozessen

Neben dem Musternamen sind in der Tabelle die Ausprägungen und Auswirkungen auf die Sichten bzw. Abstraktionsebenen beschrieben (vgl. [13], S. 10 ff.).

Identifiziertes, gruppiertes Muster	Ausprägung(en)	Auswirkung(en)
add properties	Hinzufügen von beispielsweise Task- oder Gateway-Eigenschaften, wie z.B. Applikations-URLs und Protokollen für Services; Definition von Transaktionsverhalten	keine Änderung des Workflows, evtl. Anbieterspezifische Erweiterungen
add script task	für Variableninitialisierung, Business-Rules, Logging	keine korrespondierenden Elemente auf Business-Level
add protocol task	asynchrones Senden und Empfangen von Nachrichten bzw. Validierungsnachrichten auf Basis definierter Protokolle	keine korrespondierenden Elemente auf Business-Level
add boundary event	Benutzerverhalten in Bezug auf beispielsweise Abbruch eines Vorgangs oder einer Transaktion; Timeouts bei asynchronen Nachrichtenflüssen	häufig nicht in Business-Level abgebildet
add technical exception flow	Sequenzflüsse für technische Fehler / Ausnahmebehandlung	keine korrespondierenden Elemente auf Business-Level, oft in Zusammenhang mit Workflowänderungen
change activity name	z.B. Umbenennungen einer Aktivität in technischer Sicht, um Namensähnlichkeiten zu vermeiden	Name des Elements verändert sich
change activity type	Änderungen des Elementtyps, z.B. durch Automatisierung einer bisher als vom Benutzer ausgeführten Aktivität	Typänderung kann teilweise nur durch ein ersetzendes Element bewirkt werden (z.B. wenn eine Aktivität durch ein Start-Ereignis abgebildet wird)

(Fortsetzung)

suppress specification activity	Zusammenführen von mehreren Aktivitäten z.B. auf der Business-Ebene, die durch einen einzelnen IT-Task abgedeckt werden können; mehrere Benutzertasks zusammengeführt in einen Task (z.B. durch Abdeckung der Tasks in einer Maske)	Differenz der Anzahl der Elemente zwischen den Sichten
splitting tasks into blocks	beispielsweise eine Aufteilung eines technischen Tasks in gruppierte Blöcke, die abhängig von einer Bedingung eine bestimmte Transaktion und Service-Calls (send-receive) ausführen	Differenz der Elementanzahl der Sichten, evtl. Veränderung der Ausführungspfade und dem Ablaufverhalten
splitting workflow	Aufteilung des Hauptflusses in kleinere, vom Hauptfluss kontrollierte Flüsse, die sehr speziell sind, oder wiederverwendet werden können; durch Events ausgelöste Transaktionen, Subprozesse oder externe Prozesse	Evtl. Veränderung der Ausführungspfade und dem Ablaufverhalten
refactor gateway	allgemeine Verfeinerung oder Veränderung des Workflows (z.B. Hinzufügen von mehreren Bedingungen), um technisches Verhalten abzubilden	Veränderung der Ausführungspfade und dem Ablaufverhalten

Anhang B

Informelle Kategorisierung von Unterschieden aus verschiedenen Perspektiven

Folgend sind die in [16] eingenommenen Perspektiven und daraus resultierende Unterschiede dargestellt.

Perspektive	Beschreibung, Ursache(n) für Unterschiede
Prozessperspektive	Unterschied z.B. durch Aufteilung eines großen Prozesses in kleinere Prozesse, Abdeckungsgrad ist gleich, aber nicht die Aufteilung; verschiedene Grade an Korrespondenzen sind nötig
Aktivitätssicht	u.a. Abdeckungsgrad wie bei der Prozessperspektive, Aufteilung kann auch anders sein; Aktivitäten umfassen nicht vollständig oder nur partiell die zu erledigende Arbeit der korrespondierenden Aktivität; Aktivität ist nicht vorhanden
Sequenzflusssicht	unterschiedliche Ausführungspfade in den Modellen durch unterschiedliche Reihenfolge und veränderte Abhängigkeiten der Elemente in einem Sequenzfluss; unterschiedliche Entscheidungen und abweichend vom korrespondierenden Modell zusammenfließende Sequenzpfade
Datensicht	vertauschte Reihenfolge von In- und Output-Objekten und unterschiedlicher Zugriff auf Daten
Ressourcensicht	unterschiedliche Aufteilung der Ressourcen, partielle Äquivalenz der Ressourcen / Überlappungen, nicht-erfasste Ressourcen, zusätzliche Ressourcenzuweisung

Anhang C

Use Case Beschreibungen

Name:	<i>Projekt anlegen</i>
Primärer Nutzer:	(Business-)Analyst
Ziel (des Nutzers):	Anlegen eines neuen Projekts im System
Vorbedingung:	Projektbrowser verfügbar
Nachbedingung bei Erfolg:	Neues Projekt erfolgreich angelegt.
Beteiligte Benutzer:	-
Standardablauf	<ol style="list-style-type: none">1. Der Analyst ruft die „New Project“-Funktion auf.2. Das System öffnet eine Eingabemaske für den Analysten.3. Der Analyst füllt die Maske aus, indem er den Projekt-namen einträgt, sowie die anzulegenden Abstraktionsebenen auswählt.4. Das System legt das neue Projekt und eine Ordnerstruktur an und stellt es im Projektbrowser bereit.
Erweiterungen	-

Name:	<i>Business-Prozess erstellen</i>
Primärer Nutzer:	(Business-)Analyst
Ziel (des Nutzers):	Anlegen eines neuen Prozesses aus Business-Sicht innerhalb eines Projekts
Vorbedingung:	Projektbrowser verfügbar, angelegtes Projekt ausgewählt
Nachbedingung bei Erfolg:	Erfolgreiches Anlegen eines neuen Business-Prozesses, Zuordnung zum Projekt, Modell ist zugeordnet
Beteiligte Benutzer:	-
Standardablauf	<ol style="list-style-type: none">1. Der Analyst wählt im Project Browser die Abstraktionsebene für den zu erstellenden Prozess aus.2. Der Analyst wählt die Funktion "Add Process" aus.3. Das System öffnet eine Eingabemaske für die Daten des neuen Prozesses.4. Der Analyst gibt den Prozessnamen ein und wählt durch die Maske ein BPMN Modell aus.5. Der Analyst wählt die Option "Add Prefix to Model Name?" aus.6. Der Analyst bestätigt durch die Schaltfläche "Add Process" seine Eingaben.7. Das System legt das Modell im Ordner des Prozesses ab.
Erweiterungen	-

Name:	<i>Business-Prozess klonen</i>
Primärer Nutzer:	(Business-)Analyst
Ziel (des Nutzers):	Anlegen eines Prozesses aus IT-Sicht innerhalb eines Projekts auf Basis eines Business-Prozessmodells
Vorbedingung:	Projektbrowser verfügbar, angelegtes Projekt ausgewählt, Business-Prozess zum Klonen ausgewählt
Nachbedingung bei Erfolg:	Erstellte Kopie der BPMN-Datei, erstellte .corr und .asso Dateien für Korrespondenzen und Assoziation
Beteiligte Benutzer:	-
Standardablauf	<ol style="list-style-type: none"> 1. Der Analyst wählt ein Business-Prozessmodell im Projektbrowser aus. 2. Der Analyst ruft die „Clone Model“-Funktion auf. 3. Das System zeigt die Namen des Projekts, des Prozesses und des Abstraktionslevels an. 4. Der Analyst wählt die Ziel-Abstraktionsebene aus einer Liste der im Projekt verfügbaren Abstraktionsebenen an. 5. Der Analyst wählt die Optionen zur Initialisierung des Korrespondenzmodells und für den Modellpräfix aus. 6. Der Analyst bestätigt durch die Schaltfläche "Clone Model" seine Auswahl. 7. Das System erstellt im Ausgangspfad eine Assoziationsdatei, in der die Pfade zum Ausgangsmodell und Zielmodell festgehalten sind. 8. Das System erstellt eine Kopie des Business-Prozessmodells unter Verwendung des Levelprefix. 9. Das System erstellt ein Korrespondenzmodell für die assoziierten Business- und IT-Prozessmodelle und speichert es im Pfad des Ausgangsmodells. 10. Das System aktualisiert den Projektbrowser.
Erweiterungen	-

Name:	<i>Vertikale Konsistenz prüfen</i>
Primärer Nutzer:	(Business-, IT-)Analyst
Ziel (des Nutzers):	Prüfen, ob alle geschäftsrelevanten Elemente bzw. Aktivitäten in den Modellen einer Business- und IT-Sicht eines Prozesses enthalten sind
Vorbedingung:	Projektbrowser verfügbar, angelegtes Projekt ausgewählt, Business- und IT-Prozess verfügbar, Korrespondenzmodell verfügbar
Nachbedingung bei Erfolg:	Anzeige von Informationen über die vertikale Konsistenz und fehlende Elemente bzw. Korrespondenzen
Beteiligte Benutzer:	-

Standardablauf

1. Der Analyst wählt ein (Business-)Prozessmodell im Projekt-/Prozessbrowser aus.
2. Der Analyst ruft die „Vertical Consistency Check“-Funktion auf.
3. Das System ermittelt auf Basis des Korrespondenzmodells und den Business- und IT-Prozessmodellen die vertikale Konsistenz.
4. Das System gibt Informationen über die vertikale Konsistenz aus.

Erweiterungen

-

Name:

Korrespondenz erstellen

Primärer Nutzer:

(Business-, IT-)Analyst

Ziel (des Nutzers):

Eine Korrespondenz zwischen zwei Elementen eines Business- und eines IT-Prozessmodells erstellen

Vorbedingung:

Projektbrowser verfügbar, angelegtes Projekt ausgewählt, Business- und IT-Prozess verfügbar, Korrespondenzmodell verfügbar

Nachbedingung bei Erfolg:

Meldung über die erstellte Korrespondenz (Konsole)

Beteiligte Benutzer:

-

Standardablauf

1. Der Analyst wählt ein (Business-)Prozessmodell im Projekt-/Prozessbrowser aus und wählt die Funktion "Load Model(s)".
2. Das System zeigt das ausgewählte Prozessmodell und das assoziierte (zuvor geklonte) Prozessmodell an.
3. Der Analyst wählt in beiden Modellen die korrespondierenden Elemente aus.
4. Der Analyst wählt die „Korrespondenz erstellen“-Funktion aus.
5. Das System aktualisiert das Korrespondenzmodell mit IDs der gewählten Elemente.
6. Das System gibt eine Meldung über die erfolgreich erstellte Korrespondenz aus.

Erweiterungen

-

<i>Name:</i>	Korrespondenz anzeigen
<i>Primärer Nutzer:</i>	(Business-, IT-)Analyst
<i>Ziel (des Nutzers):</i>	Eine Korrespondenz zwischen zwei Elementen eines Business- und eines IT-Prozessmodells anzeigen
<i>Vorbedingung:</i>	Projektbrowser verfügbar, angelegtes Projekt ausgewählt, Business- und IT-Prozess verfügbar und geladen, Korrespondenzmodell verfügbar
<i>Nachbedingung bei Erfolg:</i>	Farbliche Anzeige der korrespondierenden Elemente, Information über korrespondierende Elemente
<i>Beteiligte Benutzer:</i>	-
<i>Standardablauf</i>	<ol style="list-style-type: none"> 1. Der Analyst wählt in einem Modell ein Element aus. 2. Das System prüft im Korrespondenzmodell, ob im assoziierten Prozessmodell das ausgewählte Element vorhanden ist. 3. Das System zeigt das ausgewählte und korrespondierende Element farblich an. 4. Das System zeigt eine Information über die vorhandene Korrespondenz aus.
<i>Erweiterungen</i>	-

Versicherung

„Ich versichere, dass ich die vorstehende Arbeit selbständig angefertigt und mich fremder Hilfe nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichtem oder nicht veröffentlichtem Schrifttum entnommen sind, habe ich als solche kenntlich gemacht.“
