

**Institut für Architektur von  
Anwendungssystemen**

Universitätsstraße 38

D-70569 Stuttgart

Bachelorarbeit Nr. 167

**Unifizierte Service-  
Komposition für BPMN 2.0**

**Olaf Hoffeld**

Studiengang: Softwaretechnik

Prüfer: Prof. Dr. Frank Leymann

Betreuerin: Dipl.-Inf. Katharina Görlach

Beginn am: 14.08.2014

Beendet am: 13.02.2015

CR-Nummer: D.3.1, D.3.2, F.3.2, F.4.2, H.4.1



## Vorwort und Danksagung

Die vorliegende Bachelorarbeit entstand an der Universität Stuttgart während meiner Zeit des Studiums des Software Engineerings.

Mein herzlicher Dank gilt:

Herrn Prof. Dr. Frank Leymann, dem Direktor des Instituts für Architektur und Anwendungssysteme für die Möglichkeit zur Erstellung meiner Bachelorarbeit.

Frau Dipl.-Inf. Katharina Görlach, meiner Betreuerin, die obwohl in den Endzügen ihrer Doktorarbeit vertieft, immer ein offenes Ohr für meine Fragen hatte und mich in jeder erdenklichen Weise bei der Bearbeitung des Themas und der Implementierung unterstützt hat.

Meiner Freundin Lisa Fürbringer, die unermüdlich Korrektur gelesen hat und stets bereit war sich in ein fremdes Fachgebiet einzuarbeiten, um hilfreiche Denkanstöße und Tipps zu geben.

Nicht zuletzt einem anonym bleibenden Industriepartner aus der Automobilbranche, der großes Interesse an den Ergebnissen der Arbeit zeigt und mir den Einblick in eine Vielzahl von Geschäftsprozessen ermöglicht hat.



## Abstract

Die vorliegende Arbeit beschreibt die Konzeption, Entwicklung und Test einer Software, welche mit der Geschäftsprozessmodellierungssprache Business Process Model and Notation 2.0 (BPMN 2.0) modellierte Geschäftsprozesse vollautomatisch in ein grammatik-basiertes unifiziertes Modell umwandelt.

Dies geschieht im Kontext der Forschungsarbeit eine Meta-Modellierungssprache, das Unified Model (UM), zu entwickeln, die in der Lage ist eine beliebige Modellierungssprache am Rechner auszuführen. Dadurch lassen sich Aspekte wie die Ausführungsgeschwindigkeit und -effizienz verschiedener Engines, welche die Geschäftsprozessmodelle ausführen, unabhängig von der Modellierungssprache miteinander vergleichen.

Im Rahmen dieser Arbeit wurden die Sprachkonstrukte der BPMN 2.0 in eine UM-konforme Grammatik überführt. Hierfür wurde untersucht, ob sich aufgrund von Gemeinsamkeiten mit der partiell verwandten und bereits in das UM übersetzten Modellierungssprache BPEL (Business Process Execution Language) einige Sprachkonstrukte aus dieser Übersetzung adaptieren lassen. Für alle Modellierungselemente wurde eine Übersetzung entwickelt, die im UM funktionsgleich mit dem ursprünglichen BPMN 2.0-Modellierungselement ist.

Zuletzt wurde eine Software in Java entwickelt, welche Geschäftsprozesse in BPMN 2.0 in UM-konforme Grammatiken umwandelt. Abschließend wurde die Software auf Korrektheit und Performanz hin getestet.

## Inhaltsverzeichnis

<b>Vorwort und Danksagung.....</b>	<b>3</b>
<b>Abstract.....</b>	<b>5</b>
<b>1 Einleitung.....</b>	<b>8</b>
1.1 Problemstellung und Zielsetzung.....	8
1.2 Vorgehensweise.....	10
<b>2 Das Unified Model für Service-Kompositionen.....</b>	<b>12</b>
<b>3 BPMN 2.0.....</b>	<b>14</b>
3.1 Entstehung und Zielsetzung.....	14
3.2 Einführung in BPMN 2.0.....	15
3.2.1 Fluss-Objekte.....	16
3.2.1.1 Aktivitäten.....	16
3.2.1.1.1 Aufgabentypen.....	17
3.2.1.1.2 Markierungen.....	18
3.2.1.2 Ereignisse.....	19
3.2.1.3 Gateways.....	21
3.2.2 Verbindende Objekte.....	23
3.2.3 Daten.....	23
3.2.4 Schwimmbahnen.....	23
3.2.5 Artefakte.....	24
<b>4 Übersetzung von BPMN in das UM.....</b>	<b>25</b>
4.1 Adaption von Bestandteilen aus der bestehenden Übersetzung von BPEL in das UM.....	25
4.2 Übersetzung einzelner Sprachelemente.....	29
4.2.1 Tokenfluss.....	29
4.2.2 Parallele Ausführung.....	31
4.2.3 Sequentielle Ausführung.....	33
4.2.4 Inklusives Gateway.....	34
4.3 Übersetzungsalgorithmus.....	37
<b>5 Implementierung der Software.....</b>	<b>40</b>
5.1 Anforderungen.....	40
5.2 Entwurf.....	40
5.3 Testfälle.....	42
<b>6 Schlussbetrachtung.....</b>	<b>44</b>
<b>Quellenverzeichnisse.....</b>	<b>46</b>
<b>Ehrenwörtliche Erklärung.....</b>	<b>49</b>

## Abbildungsverzeichnis

Abbildung 1: Ausführung von Geschäftsprozessen mit Hilfe des UMs.....	10
Abbildung 2: Geschäftsprozess mit zwei Web-Services und zugehörige Produktionsregeln.....	13
Abbildung 3: Beispiel für einen BPMN 2.0-Geschäftsprozess.....	15
Abbildung 4: Aufgabentypen in BPMN 2.0.....	17
Abbildung 5: Markierungen in BPMN 2.0.....	18
Abbildung 6: Gateways in BPMN 2.0.....	21
Abbildung 7: Darstellung des Übersetzungsansatzes von BPMN 2.0 über BPEL in das UM.....	26
Abbildung 8: Terminierungsverhalten in BPMN 2.0.....	28
Abbildung 9: Tokenfluss in einem Prozess mit Inklusivem Parallelen Gateway.....	30
Abbildung 10: Modellierung der Tokenweiterleitung im UM.....	31
Abbildung 11: Paralleler Sequenzfluss.....	32
Abbildung 12: Sequentielle Ausführung.....	33
Abbildung 13: Aufsplittendes Inklusives Gateway.....	35
Abbildung 14: Kombinatorische Explosion von Produktionsregeln.....	37
Abbildung 15: Veranschaulichung des Übersetzungsalgorithmus.....	39
Abbildung 16: Komponenten der Übersetzungsssoftware.....	41
Abbildung 17: Testprozess zur Überprüfung der Korrektheit der Gateways.....	43

## Tabellenverzeichnis

Tabelle 1: Ereignisse in BPMN 2.0.....	21
Tabelle 2: Vergleich des Terminierungsverhalten zwischen BPMN 2.0 und BPEL.....	28

# 1 Einleitung

## 1.1 Problemstellung und Zielsetzung

Aus der heutigen Unternehmenswelt sind Geschäftsprozesse nicht mehr wegzudenken. Angefangen bei Kleinunternehmen bis hin zu Großkonzernen und Regierungsorganisationen haben sie sich sowohl im technischen als auch im kaufmännischen Bereich als essenzielles Organisationswerkzeug fest etabliert<sup>1</sup>. Der Begriff Geschäftsprozess ist in diesem Zusammenhang weit auszulegen und beinhaltet sowohl die operativen Handlungsabläufe in einem Unternehmen wie sie meist von Personen durchgeführt werden, als auch automatische von Maschinen durchgeführte Aktionen.

Ein typischer Geschäftsprozess ist beispielsweise die Warenannahme in einem Industriebetrieb. Warenladungen können vom Schichtleiter des Betriebs während der Öffnungszeiten angenommen werden. Hierzu fährt der Fahrer des Warentransporters in den Zu- und Entladebereich und übergibt dem Schichtleiter einen Lieferschein. Dieser überprüft mit Hilfe einer unternehmenseigenen Software, ob der Auftrag auch tatsächlich vergeben wurde und gleicht die Auftragsposten mit denen des Lieferscheins ab. Stimmen allen Angaben überein, weist der Schichtleiter die Lagermitarbeiter an das Fahrzeug zu entladen und die Waren im Lager einzusortieren. Währenddessen quittiert der Schichtleiter dem Fahrer des Transporters den Wareneingang mit einer Unterschrift und vermerkt in der Software den Zeitpunkt der Entladung und Einlagerung der Waren.<sup>2</sup>

Genauso vielfältig wie die Einsatzgebiete von Geschäftsprozessen ist auch die Anzahl der dafür verwendeten Modellierungssprachen. Ein skizzenhafter Entwurf eines Geschäftsprozesses, welcher zur Kommunikation eines Sachverhaltes an ein Whiteboard geschrieben wurde, kann bereits eine eigene Modellierungssprache darstellen.<sup>3</sup> Auf diese Weise entstehen häufig Modellierungssprachen für den Gebrauch innerhalb eines Unternehmens. Bei Bedarf nach höherer Präzision oder Funktionsvielfalt können diese vom Unternehmen beliebig erweitert werden.<sup>4</sup> Zudem gibt es Modellierungssprachen, welche ausführlich dokumentiert und als Standards international anerkannt sind. Hierzu gehören unter Anderem die Sprachen Business Process Execution Language (BPEL)<sup>5</sup> und Business Process Model and Notation 2.0 (BPMN 2.0),<sup>6</sup> wobei letztere als ISO-Norm geführt wird.<sup>7</sup>

Eine weit verbreitete Modellierungssprache für Geschäftsprozesse, die Gegenstand dieser Arbeit ist, ist BPMN. In der überarbeiteten Version als BPMN 2.0 bezeichnet, verfügt die Sprache seit dem Jahr 2011 über die Möglichkeit der Ausführung, das heißt wohldefinierte operationale Semantik.<sup>8</sup>

---

1 Vgl. Schmelzer, H. J.; Sesselmann, W. (2008), S. 224 f.

2 Vgl. Leiter Warenannahme (2014)

3 Vgl. Abteilungsleiter Erprobung (2014)

4 Vgl. Schulungsunterlagen: Wertstromanalyse nach VSDiA (2014)

5 Vgl. OASIS Committee (Hrsg) (2007), <http://docs.oasis-open.org>

6 Vgl. OMG (Hrsg.) (2011), <http://www.omg.org>

7 Vgl. ISO/IEC 19510:2013 (2013)

8 Vgl. OMG (Hrsg.) (2011), <http://www.omg.org>

Ausführende Modellierungssprachen sind für Unternehmen sehr attraktiv, da sie es ermöglichen Geschäftsprozesse zu automatisieren, indem die Prozesse teilweise oder sogar vollständig von Computer ausgeführt werden können. Dies kann zu hohen Kosteneinsparungen führen. Allgemeine Problematik bei der großen Vielfalt an Modellierungssprachen ist die Interoperabilität zwischen diesen Sprachen und die Vergleichbarkeit der ausführenden Engines hinsichtlich ihrer Performanz.<sup>9</sup> Aufgrund der unterschiedlichen Modellierungselemente ist es im Regelfall nicht problemlos möglich einen Geschäftsprozess von einer Sprache in eine andere zu übersetzen.<sup>10</sup> Es gibt Sprachen, die sich bereits von der Sprachlogik her grundsätzlich voneinander unterscheiden. Zu dem kommen bei unterschiedlichen Modellierungssprachen auch verschiedene Modellierungselemente zum Einsatz. Selbst Sprachen, welche über vergleichbare Funktionalitäten verfügen, wie beispielsweise die Fehlerbehandlung oder den Kompensationsvorgang, setzen diese unterschiedlich um. Somit kommt es zu erheblichen Differenzen in der Ausführungslogik, sodass für viele Modellierungselemente letztlich keine Äquivalente in der anderen Modellierungssprache existieren, sondern bestenfalls vergleichbare Funktion angeboten werden.<sup>11</sup>

Insbesondere ohne manuelles Eingreifen gestaltet sich die Übersetzung eines Geschäftsprozesses in eine fremde Modellierungssprache schwierig. Unternehmen wird der Umstieg auf modernere Modellierungssprachen erschwert, da die bisherigen Geschäftsprozesse häufig bereits modelliert wurden und große Mengen an Know-How enthalten, welche für das Unternehmen sehr wertvoll und nicht einfach wiederzubeschaffen sind.<sup>12</sup> Dieser Umstand kann bei der Zusammenarbeit oder Zusammenlegung mit einem anderen Unternehmen zu einem erheblichen Kostenfaktor führen. Insbesondere dann, wenn von den betroffenen Unternehmen unterschiedliche Modellierungssprachen verwendet wurden, die typischerweise nicht miteinander kompatibel sind. Auch innerhalb eines Unternehmens kann diese Problematik auftreten, wenn unterschiedliche Organisationseinheiten wie Abteilungen, Bereiche, Standorte und Business Units für ihre Prozesse verschiedene Modellierungssprachen verwendet haben.<sup>13</sup>

Eine Abhilfe hierfür bietet das Unified Model (UM), welches Geschäftsprozesse über eine formale Grammatik realisiert. Die Turingmächtigkeit dieser garantiert, dass grundsätzliche alle Prozessmodelle, die mit konventionellen Modellierungssprachen spezifiziert sind, mit dem UM repräsentiert werden können.<sup>14</sup> Einzige Voraussetzung ist eine Beschreibung der Übersetzung von der jeweiligen Modellierungssprache hin zum Meta-Modells des UMs. Ein Beispiel der möglichen Vereinheitlichung der Ausführung von Geschäftsprozessen, die mit dem UM erreicht werden kann, ist in Abbildung 1 dargestellt. Dort wird je ein Geschäftsprozess von den drei Modellierungssprachen BPMN 2.0, BPEL und ConDec<sup>15</sup> in das UM übersetzt und anschließend mit

<sup>9</sup> Vgl. Görlach K. (2014)

<sup>10</sup> Vgl. Recker, J. C., & Mendling, J. (2006), S. 521-532

<sup>11</sup> Vgl. Weidlich, M., Decker, G., Großkopf, A., & Weske, M. (2008), S. 265-282

<sup>12</sup> Vgl. Teamleiter Erprobung (2014)

<sup>13</sup> Vgl. Teamleiter Erprobung (2014a)

<sup>14</sup> Vgl. Görlach K. (2014a)

<sup>15</sup> Vgl. Pesic, M.; Van der Aalst, W. M. (2006, January), S. 169-180

einer Engine für das UM ausgeführt.

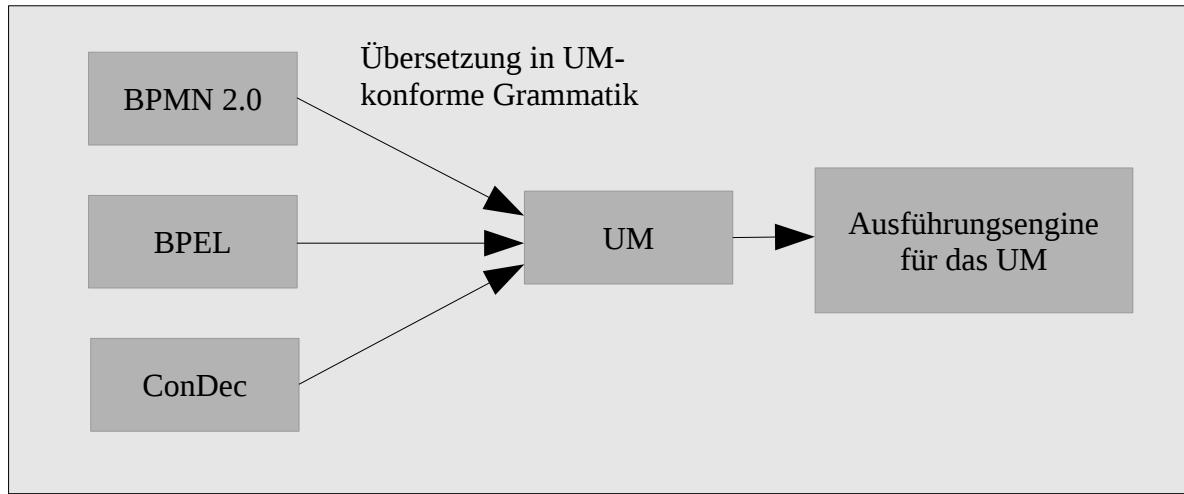


Abbildung 1: Ausführung von Geschäftsprozessen mit Hilfe des UMs<sup>16</sup>

Aktuell existieren Übersetzungen in das UM für die Modellierungssprachen ConDec<sup>17</sup> und BPEL<sup>18</sup>. Ziele dieser Arbeit sind die Erarbeitung einer Übersetzung von BPMN 2.0 in das UM sowie die Konzeption, Implementierung und Testung einer Software, welche einen BPMN 2.0-Prozess vollautomatisch in eine UM-konforme Grammatik übersetzt. Eine Übersicht der einzelnen Arbeitsschritte wird im nachfolgenden Unterkapitel gegeben.

## 1.2 Vorgehensweise

In **Kapitel 2** werden die Grundlagen des UMs erläutert, welches im Rahmen dieser Arbeit die Zielsprache der Übersetzung von BPMN 2.0 ist. Ausführlich dargestellt wird die Modellierungssprache BPMN 2.0 in **Kapitel 3** hinsichtlich ihrer Funktionsweise und ihrer Modellierungselemente. Zudem wird ein kurzer Überblick über deren Entstehung gegeben. An die Grundlagenteile ankünfend wird die Übersetzung der Modellierungssprache BPMN 2.0 in das UM in **Kapitel 4** behandelt. Es werden ausschnittsweise einige der Modellierungselemente, stellvertretend für alle, isoliert für sich betrachtet und in das UM übersetzt. Dabei wird versucht auf die bereits bestehende Übersetzung von BPEL in das UM zurückzugreifen und geprüft, welche Elemente dieser Übersetzung sich, gegebenenfalls mit Änderungen, übernehmen lassen. Anschließend wird ein Übersetzungsalgorithmus entwickelt, welcher BPMN 2.0-Geschäftsprozesse in das UM überführen kann.

**Kapitel 5** behandelt die Implementierung der Übersetzungssoftware. Dabei werden die konkreten Anforderungen an die Software erfasst, sowie deren Betriebsumgebung und -parameter festgelegt. Daran anknüpfend wird ein grundlegender Überblick über die Architektur der Software gegeben,

---

16 Eigene Darstellung

17 Vgl. Liang, S. (2013)

18 Vgl. Shao, B. (2013)

bevor die Testfälle und deren Resultate dargelegt werden. Die Arbeit wird durch eine Schlussbetrachtung in **Kapitel 6** abgerundet, in welcher eine Zusammenfassung der wesentlichen Ergebnisse erfolgt und ein kurzer Zukunftsausblick anschließt.

## 2 Das Unified Model für Service-Kompositionen

Das Unified Model wurde mit dem Ziel konzipiert ein ausführbares Meta-Modell für alle Geschäftsprozesssprachen zu entwickeln.<sup>19</sup> Die Intention dabei war die Vergleichbarkeit von unterschiedlichen Modellierungssprachen und den verschiedenen, dafür angebotenen, ausführenden Engines hinsichtlich Kriterien wie der Performanz untereinander herzustellen<sup>20</sup>. Einzige Voraussetzung für die Verwendung einer Modellierungssprache mit dem UM ist das Vorhandensein einer Übersetzung der jeweiligen Sprache in das UM.

Das UM beschreibt eine formale Grammatik  $G=(V, \Sigma, P, S)$ , welche mit Hilfe von Terminal- und Nichtterminalsymbolen sowie Produktionsregeln<sup>21</sup> eine konkrete Instanz eines Geschäftsprozesses und dessen Ausführung repräsentiert.<sup>22</sup>

Darüber hinaus besitzen die Nichtterminalsymbole die Eigenschaft der Dimensionalität – sie können entweder als ein-, zwei- oder dreidimensionale Symbole vorkommen. Die erste Dimension legt den Namen des Nichtterminals fest. Zweidimensionale Nichtterminale sind zusätzlich noch mit einer Service-Operation verknüpft, deren Eingabe- und Ausgabeparameter, sofern vorhanden, an dieser Stelle ebenfalls beschrieben sind. In der dritten Dimension werden den Rückgabewerten der Service-Operationen eindimensionale Nichtterminale zugeordnet. Damit lassen sich unterschiedliche Produktionsregeln in Abhängigkeit von den Ergebnissen einer Service-Operation anwenden.<sup>23</sup>

Eine im Geschäftsprozess modellierte Aufgabe wird im UM durch ein entsprechendes Nichtterminal wiedergegeben. Die Ausführung der Aufgabe beziehungsweise der Aufruf des Web-Services wird durch den Aufruf der Produktionsregel modelliert. Diese überführt das die Aufgabe darstellende Nichtterminal in ein Terminal. Ein Beispiel dazu ist in nachstehender Abbildung 2 dargestellt.

---

19 Vgl. Görlach, K., Leymann, F., & Claus, V. (2013)

20 Vgl. Görlach, K. (2015)

21 Für eine ausführliche Erläuterung formaler Grammatiken, der Bedeutung von Terminal- und Nichtterminalen sowie von Produktionsregeln sei an dieser Stelle auf einschlägige Literatur verwiesen, z.B.: Schöning, U. (2008)., S. 13 f. oder Horn, C. (2001)., S.79 f.

22 Vgl. Görlach, K., Leymann, F., & Claus, V. (2013)

23 Vgl. Görlach, K. (2013), S. 24

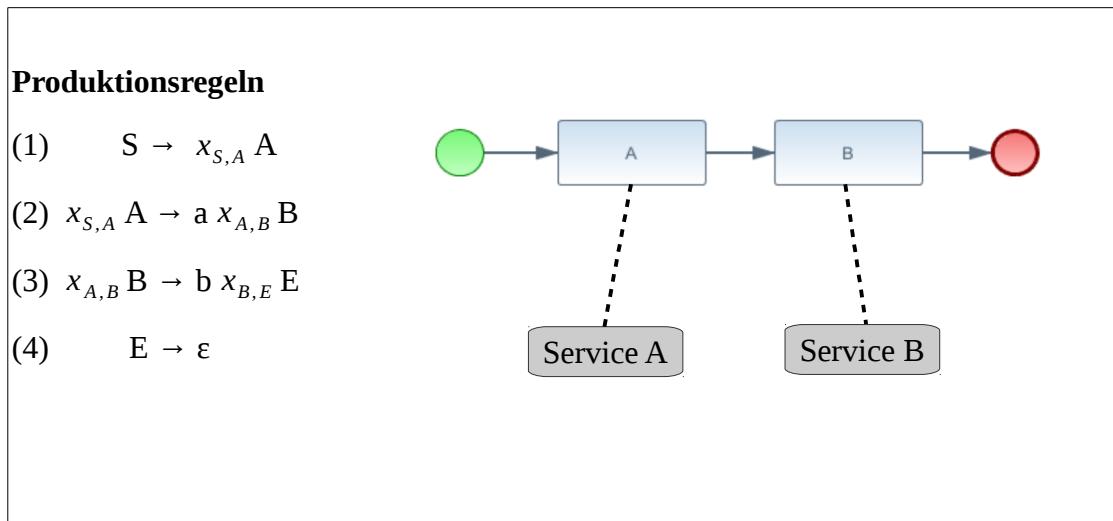


Abbildung 2: Geschäftsprozess mit zwei Web-Services und zugehörige Produktionsregeln<sup>24</sup>

Der modellierte Geschäftsprozess besteht aus den aufeinanderfolgenden Web-Services A und B. Sie werden jeweils durch Aufruf der Produktionsregeln (2) respektive (3) ausgeführt, wenn die Nichtterminalsymbole A und B, welche die Web-Services repräsentieren, in entsprechende Terminalssymbole überführt werden. Der Lesbarkeit wegen verwenden diese die selben Buchstaben wie die auszuführenden Aufgaben, lediglich in Kleinschreibweise.

Ein Geschäftsprozess des UMs wird auf Datenebene als \*.grammar-Datei im Format der Extensible Markup Language (XML) gespeichert. Darin enthalten sind das Startsymbol des Geschäftsprozesses, eine Auflistung aller verwendeten Terminalen und Nichtterminalen, sowie die Definition deren Dimensionen.<sup>25</sup>

Ausgeführt werden kann eine solche \*.grammar-Datei mit Hilfe der Unified Engine (UE), einer Ausführungsumgebung, welche ähnlich einer Turingmaschine funktioniert. Die UE schreibt das Startsymbol der Grammatik auf ein Turingband und wendet die in der \*.grammar-Datei definierten Produktionsregeln so häufig hintereinander an, bis ausschließlich Terminalssymbole auf dem Turingband verbleiben, womit die Ausführung des Geschäftsprozesses beendet ist.<sup>26</sup>

Für die Produktionsregeln und deren Anwendung werden vom UM eine Reihe von Regeln vorgeschrieben. Eine vollständige Auflistung sowie eine ausführlichere Beschreibung dieser Restriktionen, findet sich in dem Paper „A Generic Transformation of Existing Service Composition Models to a Unified Model“ von Katharina Görlach.<sup>27</sup>

24 Vgl. Görlach, K. (2014b)

25 Vgl. Görlach, K. (2014b)

26 Vgl. Görlach, K., Leymann, F., & Claus, V. (2013)

27 Vgl. Görlach, K. (2013), S. 8 - 13

## 3 BPMN 2.0

Da weite Teile dieser Arbeit auf einem grundlegenden Verständnis von BPMN 2.0 aufbauen, wird in den folgenden Abschnitten näher auf diese Modellierungssprache eingegangen.

Zunächst wird ein kurzer Überblick zur Entstehungsgeschichte und Verbreitung von BPMN vermittelt. Anschließend erfolgt eine ausführliche Beschreibung der Modellierungssprache und deren Sprachkonstrukte.

### 3.1 Entstehung und Zielsetzung

Im Jahr 2001 stellte Stephen A. White die erste Version von BPMN vor. Bereits drei Jahre später wurde die Modellierungssprache von der Object Management Group (OMG)<sup>28</sup>, einer Vereinigung zur Durchsetzung von Standards im Bereich der Geschäftsprozessmodellierung, vorgestellt und somit in den Status eines offiziellen Standards erhoben. Im Januar 2011 erschien die überarbeitete Version BPMN 2.0, welche den Standard um eine Vielzahl neuer Features ergänzt hat. Dazu zählen insbesondere die Möglichkeit das BPMN-Modell erweitern zu können, sowie eine formale Definition der Sprachkonstrukte. Seit Mitte des Jahres 2013 wird BPMN 2.0 als ISO-Standard geführt.<sup>29</sup> Der Kerngedanke bei der Konzeption von BPMN war die Erstellung einer Geschäftsprozessmodellierungssprache, welche leichtgewichtig genug ist, um sie schnell erlernen als auch einfach nachvollziehen zu können. Gleichermassen soll sie für Kaufleute zugänglich sein, die an einer skizzenhaften Prozessbeschreibung interessiert sind, als auch für technisch orientierte Personen, welche anschließend mit der Umsetzung und Ausarbeitung des vorab skizzierten Geschäftsprozesses zu einem ausführbaren Modell beauftragt sind.<sup>30</sup>

Die Hauptanwender von Geschäftsprozessen sind Unternehmen und große Organisationen. In diesen werden Prozesse im Regelfall durch Kaufleute festgelegt, die den Blick für Ereignisse, Tätigkeiten und Abläufe haben, die im Rahmen der operativen Geschäftstätigkeit anfallen und deren genauen Abläufe abhängig von bestimmten Bedingungen und Ereignissen sind. Beispielsweise erfordert die Bearbeitung eines Bestellvorgangs die Tätigkeit der Überprüfung der Bonität des Kunden anhand seines Schufa-Wertes<sup>31</sup>. Solche Geschäftsprozesse laufen heutzutage in erheblichem Maße computergestützt ab. Die für den Anwender und im Regelfall auch Modellierer des Geschäftsprozesses nicht sichtbaren und im Hintergrund verwendeten Technologien müssen ebenfalls modelliert und aufeinander abgestimmt werden. Dies ist die Aufgabe des technisch orientierten Modellierers, welcher für die Ausgestaltung und Umsetzung, der von den Kaufleuten vorgegebenen Geschäftsabläufe zuständig ist.<sup>32</sup> Diese müssen ausführbar sein, das heißt so weit

---

28 Damals noch Business Process Management Initiative (BPMI), später übernommen von der OMG.

29 Vgl. ISO/IEC 19510:2013 (2013)

30 Vgl. hierzu und im Folgenden Teamleiter Erprobung (2014b)

31 Der *Schufa-Wert* ist der gemittelte Werte der verschiedenen Schufa-Scores und ist ein in der Geschäftswelt weit verbreiteter Indikator für den vorraussichtlichen Zahlungsausfall einer Forderung.

Vgl. Projektleiter Jahresbericht (2014)

32 Vgl. Projektleiter Erprobung (2014c)

spezifiziert werden, dass eine Maschine in der Lage ist den Geschäftsprozess, gegebenenfalls auch unter Berücksichtigung von menschlichem Input, durchzuführen.

## 3.2 Einführung in BPMN 2.0

BPMN 2.0 ist eine grafische Sprache zur Beschreibung und Dokumentation von Geschäftsprozessen und Handlungsabläufen, welche typischerweise in Form eines Diagramms dargestellt werden.<sup>33</sup> Abbildung 3 zeigt einen ausgewählten BPMN 2.0-Geschäftsprozess, bei welchem auch ohne Kenntnisse von BPMN 2.0 intuitiv erkenntlich ist, welcher Zusammenhang hierbei modelliert wurde.

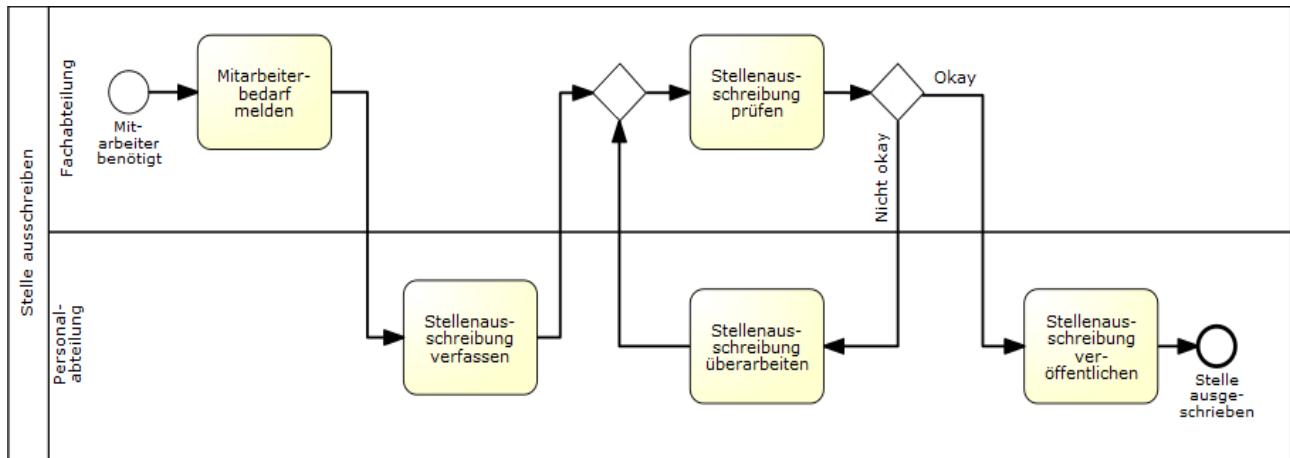


Abbildung 3: Beispiel für einen BPMN 2.0-Geschäftsprozess<sup>34</sup>

Der Prozess „Stelle ausschreiben“ beginnt, sobald eine Fachabteilung einen Mitarbeiter benötigt. Zuerst meldet diese den Mitarbeiterbedarf. Im nächsten Schritt wird die Personalabteilung eine Stellenausschreibung verfassen, woraufhin diese von der Fachabteilung geprüft wird. Wird die Stellenausschreibung für „nicht in Ordnung“ befunden, so wird sie zur Überarbeitung zurück an die Personalabteilung gegeben und anschließend, nach deren erfolgter Überarbeitung, von der Fachabteilung erneut geprüft. Ist die Stellenausschreibung im ersten Anlauf oder nach mehreren Durchläufen schlussendlich in Ordnung, muss die Personalabteilung sie noch veröffentlichen, womit der Prozess „Stelle ausschreiben“ endet.

Jeder Prozess in BPMN 2.0 beginnt mit einem Startereignis, das ein Markierungselement, ein sogenannter *Token*, in den Prozess schickt.<sup>35</sup> Dieser bewegt sich entlang der verbindenden Objekte und zeigt den derzeitigen Stand im Prozess an. Erreicht ein Token ein Flussobjekt, so wird dieses ausgeführt. Bestimmte Flussobjekte können ein Token vermehren oder auch wieder entfernen. In einem Prozess können so mehrere Tokens gleichzeitig aktiv sein. Erst wenn der Prozess keine Tokens mehr enthält, ist er beendet.

33 Vgl. OMG (Hrsg.) (2011), <http://www.omg.org>

34 Entnommen aus: Allweyer, T. (2009), S. 16

35 Vgl. hierzu und im Folgenden OMG (Hrsg.) (2011), <http://www.omg.org>

Ein *Prozess* ist ein in sich geschlossener Handlungsaublauf, welcher in BPMN 2.0 durch die beschriebenen Modellierungselemente dargestellt werden kann. Typischerweise dient er der Erreichung eines konkreten Ziels und ist entsprechend benannt, wie der Beispielprozess „Stelle ausschreiben“ in Abbildung 3. Bei komplexen Prozessen kann es sinnvoll sein, auf bereits modellierte Vorgänge zurück zu greifen. So wäre es beispielsweise denkbar, dass der Prozess „Firmenniederlassung gründen“ mehrfach den bereits modellierten Prozess „Stelle ausschreiben“ aufruft. In diesem Zusammenhang wäre der Prozess „Stelle ausschreiben“ ein *Teilprozess* des Prozesses „Firmenniederlassung gründen“.

Der beschriebene BPMN 2.0-Prozess ist relativ einfach. Allerdings enthält er einige der nachfolgend aufgeführten fünf Basiselemente, aus denen ein BPMN 2.0-Prozess aufgebaut sein kann:

1. *Fluss-Objekte* repräsentieren die Geschehnisse im Prozess und dienen der Steuerung des Prozessablaufs.
2. Über *Verbindende Objekte* werden Relationen und Reihenfolgen unter den Fluss-Objekten definiert.
3. *Daten* visualisieren verwendete, modifizierte und erstelle Informationen.
4. *Schwimmbahnen* gliedern den Prozess nach den einzelnen Akteuren.
5. *Artefakte* dienen der genaueren Beschreibung des Prozesses und seiner Bestandteile.

Diese Elemente können jeweils in weitere Kategorien aufgeteilt sein und sind ausführlich im weiteren Verlauf des Kapitels erläutert.

Für eine darüber hinausgehende Beschreibung der einzelnen Sprachbestandteile und deren Funktionsweisen sei an dieser Stelle auf die BPMN 2.0-Spezifikation der OMG verwiesen.<sup>36</sup> Eine Vielzahl praktischer Anwendungsfälle und Beispiele dazu findet sich auch in einschlägiger Literatur zu BPMN 2.0, wie beispielsweise von Freund, J., & Rücker, B.<sup>37</sup>, Kocian, C.<sup>38</sup> und Sadowska, M.<sup>39</sup>

### 3.2.1 Fluss-Objekte

Fluss-Objekte sind die tatsächlichen Geschehnisse im Prozess. Sie entspringen stets einer der drei Kategorien - Aktivitäten, Ereignisse oder Gateways – welche im Nachfolgenden näher beschrieben sind:

#### 3.2.1.1 Aktivitäten

Die kleinsten in BPMN 2.0 bekannten Arbeitspakete sind Aufgaben. Sie stehen im Diagramm des

---

36 Vgl. OMG (Hrsg.) (2011), <http://www.omg.org>

37 Vgl. Freund, J., & Rücker, B. (2014)

38 Vgl. Kocian, C. (2011)

39 Vgl. Sadowska, M. (2011)

Geschäftsprozesses stellvertretend für die Handlungen und Aktionen, die im Rahmen einer Aufgabe im tatsächlichen Geschäftsprozess durchgeführt werden.

Aufgaben können über sogenannte Aufgabentypen sowie Markierungen um eine Reihe von Eigenschaften erweitert werden, wobei erstere die Art und Weise der Aufgabenausführung spezifizieren und letztere den Typus der Aufgabe festlegen. Auf diese beiden Arten von Zusätzen wird im Nachfolgenden näher eingegangen.

### 3.2.1.1.1 Aufgabentypen

Eine Auflistung aller in BPMN 2.0 vorhandenen und nachfolgend ausgeführten Aufgabentypen findet sich in nebenstehender Abbildung 4.<sup>40</sup>

#### Senden

Eine Senden-Aufgabe verschickt eine Nachricht an einen Empfänger, welcher nicht im Prozess selber modelliert ist.

#### Empfangen

Analog zur Senden-Aufgabe wartet eine Empfangen-Aufgabe auf die Ankunft einer Nachricht von einem Sender, welcher nicht im Prozess selber modelliert ist.

#### Benutzer

Eine solche Aufgabe erfordert stets eine Person, welche die Aufgabe unter Zuhilfenahme einer (Software-)Anwendung erledigt.

#### Manuell

Eine manuell auszuführende Aufgabe besteht ausschließlich aus Arbeitspaketen, welche ohne Einsatz einer Software-Anwendung durch eine Person durchgeführt werden.

#### Geschäftsregel

Speziell für die Verbindung einer Aufgabe mit einer Business Rules Engine<sup>41</sup> wurde der Aufgabenzusatz der Geschäftsregel entwickelt. Er deutet an, dass die jeweilige Aufgabe einen Input an eine Business Rules Engine versendet oder von ihr erhält.



Abbildung 4: Aufgabentypen  
in BPMN 2.0

40 Entnommen aus: Humboldt-Universität zu Berlin (Hrsg.) (2011)

41 *Business Rules* sind im Vorfeld klar definierte operative Vorgänge, wie beispielsweise die Entschädigung eines Firmenkunden, falls dessen Internetverbindung für mehr als eine Stunde im aktuellen Kalenderjahr ausgefallen ist. Derartige Geschäftsvorfälle und deren Behandlung können im Vorfeld festgelegt und in sogenannten Business Rules hinterlegt werden. *Business Rules Engines* sind Software-Komponenten, welche Business Rules automatisiert auf Eintritt prüfen und dann ausführen können. Mit Hilfe dieser Technologien lassen sich häufig wiederkehrende und einwandfrei beschreibbare Geschäftsvorfälle automatisiert abwickeln.

Vgl. Fowler, M. (Hrsg.) (2009)

## Service

Aufgaben, welche an externe Services, wie beispielsweise einen Web-Service, delegiert werden, sind in BPMN 2.0 mit dem Zusatz der Service-Aufgabe versehen.

## Skript

Eine Aufgabe vom Typ „Skript“ ruft zu Beginn ihrer Ausführung ein Skript auf und endet umgehend nach dessen Ausführung.

### 3.2.1.1.2 Markierungen

Die in Abbildung 5<sup>42</sup> dargestellten Markierungen und deren Bedeutung werden in diesem Kapitel näher erläutert. Zu beachten ist, dass eine Aktivität auch mit mehreren Markierungen gleichzeitig versehen sein kann.

#### Teilprozess

Hinter einer Aufgabe, welche mit dem Symbol eines Teilprozesses versehen ist, verbirgt sich ein vollständiger Prozess mit mehreren Aufgaben. Der Übersichtlichkeit wegen lassen sich einzelne Prozessschritte auf diese Weise zusammenfassen.

#### Schleife

Das Schleifen-Symbol bedeutet, dass die Aufgabe unter bestimmten Bedingungen so lange wiederholt wird, wie die Schleifen-Bedingung erfüllt ist.

#### Parallele Mehrfachausführung

Eine Aufgabe, welche mit drei parallelen, vertikalen Strichen versehen ist, wird mehrfach instanziert und entsprechend oft ausgeführt. Im Gegensatz zu einer Schleife wird dabei nicht die selbe Instanz mehrfach ausgeführt, sondern es werden jeweils eigene Instanzen der Aufgabe umgesetzt.

#### Sequentielle Mehrfachausführung

Analog zur Parallelens Mehrfachausführung werden bei der Sequentiellen Mehrfachausführung ebenfalls mehrere Instanzen der selben Aufgabe erstellt. Diese werden jedoch sequentiell ausgeführt.

#### Ad-Hoc

Alle in einer als „Ad-Hoc“ gekennzeichneten Aktivität enthaltenen Aufgaben können in beliebiger Reihenfolge beliebig häufig ausgeführt werden. Sobald eine Aufgabe beendet wurde, wird eine

	Teilprozess
	Schleife
	Parallele Mehrfachausführung
	Sequentielle Mehrfachausführung
	Ad-Hoc
	Kompensation

Abbildung 5: Markierungen

in BPMN 2.0

---

42 Entnommen aus: Humboldt-Universität zu Berlin (Hrsg.) (2011)

Abbruchbedingung für das Ad-Hoc-Konstrukt geprüft. Ist die Abbruchbedingung erfüllt, wird die Ad-Hoc-Aktivität beendet. Andernfalls wird eine beliebige, weitere, darin enthaltene Aufgabe ausgeführt. Über Sequenzflüsse und Gateways werden Aufgaben und deren Bearbeitungsreihenfolge innerhalb eines Prozesses strukturiert.

## Kompensation

In bestimmten Fällen ist es notwendig bereits erfolgreich durchgeföhrte Tätigkeiten wieder rückgängig zu machen. Da dies je nach Komplexität und Anzahl der Tätigkeiten größere Eingriffe in den Prozess erfordern kann, bietet BPMN 2.0 speziell hierfür das Sprachkonstrukt der Kompensation an. Entsprechend markierte Aufgaben dienen der Rückabwicklung bisher erledigter Aufgaben.

### 3.2.1.2 Ereignisse

Ein weiteres zentrales Konzept der Modellierungssprache BPMN 2.0 sind Ereignisse. Sie beschreiben Geschehnisse und Handlungen, die eintreten können und auf welche der Prozess reagieren muss. Ereignisse treten im Verlauf des Prozesses ein und dieser kann, abhängig vom eingetretenen Ereignis, darauf reagieren. BPMN 2.0 kennt eine große Bandbreite verschiedener Ereignisse, welche sich jeweils einer der drei folgenden Kategorien von Ereignissen zuordnen lassen:

#### Startereignisse

Sie sind im Sinne des Prozessflusses die Quelle, aus welcher der Prozess beginnt. Entsprechend haben Startereignisse auch keine eingehenden, sondern ausschließlich ausgehende Pfade.

#### Zwischenereignisse

Ereignisse, welche nach dem Start eines Prozesses und vor dessen Ende eintreten, werden Zwischenereignisse genannt. Sie können eingesetzt werden, um den Prozess dynamisch auf neu eintretende Situationen reagieren zu lassen.

#### Endereignisse

Analog zum Startereignis als Quelle des Prozessflusses stellt das Endereignis dessen Senke dar. Ein Prozess wird mit Erreichen des Endereignisses beendet, sofern sich keine Token mehr in dem Prozess oder dem Teilprozess befinden. Ausgenommen davon ist das terminierende Endereignis, welches den Prozess unabhängig von dessen Ausführungsstatus, sofort beendet.

Beim Einsatz von Ereignissen gibt es gewisse Einschränkungen. So können einige Ereignisse beispielsweise nicht innerhalb des Prozessflusses selbst verwendet werden, sondern werden an Prozesse oder Teilprozesse geknüpft, sodass der Sequenzfluss lediglich an die verbundenen Ereignisse, beispielsweise zur Fehlerbehandlung, übergeben wird. Eine vollständige Auflistung und Beschreibung aller Ereignisse, die in BPMN 2.0 vorhanden sind, ist in nachfolgender Tabelle 1 dargestellt.

Ereignis	Beschreibung
Blanko	Mit dem Blanko-Ereignis wird ein Prozess begonnen. Dieses Ereignis wird verwendet, wenn es sich um einen globalen Prozess handelt oder wenn ein Teilprozess modelliert wird. In diesem Fall wird bei Aufruf des Teilprozesses automatisch das Blanko-Ereignis aufgerufen und somit der Teilprozess selbst verwendet werden, sondern werden an Prozesse oder Teilprozesse geknüpft, sodass der Sequenzfluss lediglich an die verbundenen Ereignisse, beispielsweise zur Fehlerbehandlung, übergeben wird. Eine vollständige Auflistung und Beschreibung aller Ereignisse, die startet.
Nachricht	Eine eingehende Nachricht wurde empfangen.
Zeituhr	Bei Eintritt einer bestimmten Uhrzeit oder eines Datums wird das Ereignis erzeugt.
Bedingung	Dieses Ereignis ist mit einer Bedingung verbunden. Wenn diese Bedingung wahr wird, dann tritt das Ereignis ein.
Fehler	Bei Eintritt eines Fehlers wird der das Ereignis beinhaltende Prozess unterbrochen. Anknüpfend an dieses Ereignis können weitere Aktivitäten folgen, etwa zur Fehlerbehebung.
Kompensation	Bei Aufruf des Kompensationsereignisses wird der aktuell ausgeführte Prozess zu Ende gebracht, bevor anschließend eine Reihe von Aktivitäten durchgeführt werden, welche den betroffenen Prozess rückgängig machen. Die Aktionen, die für die Rückabwicklung des Prozesses durchzuführen sind, müssen vom Modellierer festgelegt werden.
Eskalation	Bei Eintritt dieses Ereignisses wird die weitere Ausführung des Prozesses an einen Teilprozess übergeben.
Abbruch	Das Abbruch-Ereignis führt zum sofortigen Stopp der Prozessausführung.
Terminierung	Der Eintritt dieses Ereignisses führt zum sofortigen Abbruch aller im Prozess beinhalteten Aktivitäten und beendet diesen ohne Aufruf einer Kompensation oder weiterer Ereignisbehandlung.
Link	Mit einem Link-Ereignis ist stets ein weiteres, gleichnamiges Link-Ereignis verknüpft. Beide Ereignisse fungieren wie ein Prozessübergreifender Sequenzfluss.
Signal	Dieses Ereignis wartet auf den Empfang eines Signals. Im Gegensatz zu Nachrichten werden Signale nicht an einen bestimmten Empfänger versandt, sondern können von allen Signalereignissen empfangen werden.

Ereignis	Beschreibung
Mehrfach	Vergleichbar mit dem Ereignis-basierten Gateway ist dieses Ereignis mit weiteren Ereignissen verknüpft. Beim Eintritt des ersten verknüpften Ereignisses tritt das Mehrfach-Ereignis ein und der Prozess wird auf dem Pfad begonnen oder fortgesetzt, mit dem das eingetretene Ereignis verknüpft ist.
Mehrfach Parallel	Das Mehrfache Parallelle Ereignis verhält sich ähnlich dem Mehrfach-Ereignis. Der Unterschied besteht darin, dass auf den Eintritt aller mit ihm verknüpften Ereignisse gewartet wird, bevor das Mehrfach Parallelle Ereignis eintritt.

Tabelle 1: Ereignisse in BPMN 2.0<sup>43</sup>

### 3.2.1.3 Gateways

Gateways dienen dazu die Ablauffreihenfolge des Prozesses zu steuern und erlauben es, Stellen im Prozess festzulegen, an denen ein Sequenzfluss anhand von bestimmten Bedingungen geteilt, umgeleitet oder mit anderen Flüssen zusammengeführt wird. Da Gateways in fast jedem BPMN-Prozess enthalten sind, stellen sie ein zentrales Element der Modellierungssprache da. Die unterschiedlichen Arten von Gateways sind in nachfolgender Abbildung 6 veranschaulicht.

Abbildung 6: Gateways in BPMN 2.0<sup>44</sup>

Jedes Gateway kann entweder ein aufsplittendes oder zusammenführendes Gateway sein. Aus einem eingehenden Sequenzfluss erzeugt ein aufsplittendes Gateways mehrere ausgehende Sequenzflüsse, während zusammenführende Gateways genau umgekehrt operieren.

#### Exklusives Gateway



Als einziges Gateway kann das Exklusive Gateway durch zwei unterschiedliche Symbole dargestellt werden, welche jedoch die gleiche Bedeutung haben.

**Aufsplittend:** Bei diesem Gateway wird ein eingehender Sequenzfluss in zwei ausgehende Sequenzflüsse unterteilt. Welcher der beiden möglichen Pfade dann tatsächlich eingeschlagen wird, entscheidet das Gateway über eine vom Modellierer festzulegende „Entweder-Oder-Bedingung“. Im Regelfall steht diese Bedingung, der Nachvollziehbarkeit wegen, direkt neben dem Gateway.

**Zusammenführend:** Ein auf einem beliebigen Pfad eingehender Token wird von dem Gateway

43 Vgl. hierzu und Folgende OMG (Hrsg.) (2011), <http://www.omg.org>

44 Vgl. hierzu und Folgende OMG (Hrsg.) (2011), <http://www.omg.org>

direkt auf den ausgehenden Pfad weitergeleitet.

### Paralleles Gateway



Aufsplittend: Wenn das Gateway aktiviert wird, versendet es auf allen ausgehenden Pfaden jeweils ein Token.

Zusammenführend: Das Gateway wartet mit dem Versenden des Tokens auf dem ausgehenden Pfad so lange bis auf allen eingehenden Pfaden ein Token angekommen ist.

### Inklusives Gateway



Aufsplittend: Dieses Gateway kann Token auf den verschiedenen ausgehenden Pfade versenden, wobei jeder ausgehende Pfad mit einer Bedingung verbunden ist. Ein Token wird nur versand, wenn die Bedingung zu „wahr“ ausgewertet wird.

Zusammenführend: Analog zum Parallelen Gateway wartet das Inklusive Gateway, bis auf allen eingehenden Pfaden, auf denen ein Token hätte eingehen können, ein Token empfangen wurde, bevor es aktiviert und ein Token auf den ausgehenden Pfaden verschickt wird.

### Komplexes Gateway



Aufsplittend: Jeder ausgehende Sequenzfluss ist mit einer eigenen Bedingung versehen, bei dessen Erfüllung ein Token auf diesem Pfad versandt wird. Wird das Gateway aktiviert, dann werden alle Bedingungen in einer zufälligen Reihenfolge geprüft.

Zusammenführend: Diese Art von Gateway erlaubt das Festlegen eines beliebigen logischen Ausdrucks, nach dessen Erfüllung das Gateway aktiviert wird, beispielsweise nach Erhalt eines Tokens auf drei von fünf eingehenden Pfaden.

### Ereignis-basiertes Gateway



Das Gateway existiert nur in der aufsplittenden Variante. Es ist stets mit mehreren nachfolgenden Ereignissen verknüpft. Sobald eines davon eintritt, wird das Gateway aktiviert und verschickt ein Token auf den Pfad, welcher mit dem eingetretenen Ereignis verknüpft ist.

### Exklusives Ereignis-basiertes Gateway



Die Funktionsweise dieses Gateways entspricht der des Ereignis-basierten Gateways. Der einzige Unterschied besteht darin, dass das Exklusive Ereignis-basierte Gateway nur am Anfang eines Prozesses stehen darf und den Zweck hat diesen zu initiieren.

### Paralleles Ereignis-basiertes Gateway



Analog zum Exklusiven Ereignis-basierten Gateway wird auch das Parallele Ereignis-basierte Gateway nur zum Starten eines Prozesses verwendet und steht somit stets am

---

Anfang eines Prozesses. Der Unterschied ist jedoch, dass erst wenn alle mit dem Gateway verknüpften Ereignisse eingetreten sind, der Prozess gestartet wird und auf allen ausgehenden Pfaden ein Token verschickt wird.

### 3.2.2 Verbindende Objekte

Verbindende Objekte verknüpfen die Fluss-Objekte eines Prozesses in eine Richtung miteinander und legen so eine Reihenfolge über die Menge der Fluss-Objekte fest. Sie werden mit Hilfe von Pfeilsymbolen visualisiert. BPMN kennt Verbindende Objekte aus folgenden vier Kategorien, welche jeweils unterschiedliche Arten von Objekten miteinander verknüpfen, um unterschiedliche Gegebenheiten im Prozess wiederzugeben:

1. Für die Verbindung von Aktivitäten und Gateways sind **Sequenzflüsse** zuständig. Sie legen die Reihenfolge fest, in welcher der Prozess durchlaufen wird.
2. **Nachrichtenflüsse** dienen der Verbindung von Nachrichten zwischen Versendern und Empfängern.
3. **Assoziationen** verbinden Sprachelemente von BPMN 2.0 mit Beschreibungen.
4. **Daten-Assoziationen** geben die Übergabe von Informationen zwischen Datenobjekten und Sequenzflüssen wieder.

### 3.2.3 Daten

Daten sind Informationen, die während des Ablaufs des Prozesses genutzt, generiert, verarbeitet oder gespeichert werden. Typischerweise entstammen sie Dokumenten, Gesprächen oder Nachrichten. Um diese darzustellen, verwendet BPMN 2.0 vier Arten von Daten:

1. **Datenobjekte** beschreiben die Informationen und Dokumente, die eine Aktivität als Eingabe benötigt oder als Ergebnis produziert. Sie können mit Aktivitäten oder Sequenzflüssen verknüpft werden.
2. **Dateneingaben** stellen Eingaben von Informationen dar, welche beispielsweise zur Bewältigung einer Aufgabe benötigt werden.
3. **Datenausgaben** zeigen die von Aufgaben produzierten Informationen an.
4. **Datenspeicher** dienen der Ablage oder Änderung von Informationen, welche über das Ende des Prozesses hinaus existieren.

### 3.2.4 Schwimmbahnen

Schwimmbahnen ordnen Teile eines Prozesses einer Abteilung in einem Unternehmen, einer Abteilung oder einem Teilnehmer zu. So kann der Zusammenhang bestimmter Elemente eines Prozesses zu einer Organisationseinheit erkenntlich werden. Dazu können sogenannte **Bahnen** und **Pools** genutzt werden, wobei letztere stets einen Pool in weitere Unterkategorien aufgeteilt werden kann.

**Bahnen** können sowohl eigenständig als auch innerhalb eines Pools auftreten und stellen ein eigenes physisches Objekt oder eine organisatorische Gruppierung, wie beispielsweise eine Fachabteilung in einem Unternehmen, dar. Eine Bahn kann dabei mehrere ihr zugehörige Aktivitäten umfassen oder lediglich als Kommunikationspartner anderer Geschäftsprozesse dienen.

**Pools** können verwendet werden, um mehrere Bahnen zu einer größeren Einheit zusammenzuschließen. Dadurch ist es möglich Organisationsstrukturen feingranularer wiederzugeben und beispielsweise die einzelnen Fachabteilungen eines Unternehmens zu einer Unternehmenseinheit zusammenzufassen.

### 3.2.5 Artefakte

Artefakte geben nähere Informationen zu einzelnen Bestandteilen eines Geschäftsprozesses und können an beliebigen Stellen im Prozess vorkommen. Es wird unterschieden zwischen **Textmarkierungen** und **Gruppen**.

**Textmarkierungen** enthalten zusätzliche Informationen über Elemente oder Teile des Prozesses und können insbesondere bei komplexen Zusammenhängen das Verständnis erleichtern.

**Gruppen** bestehen aus einer Menge von Objekten, welche der selben Kategorie entstammen und dienen der Verdeutlichung ihres Zusammenhangs. Sie haben, ebenso wie die Textmarkierungen, keinen Einfluss auf die Ausführung eines Geschäftsprozesse.

## 4 Übersetzung von BPMN in das UM

Nachdem im vorigen Kapitel Grundlegendes zur Modellierungssprache BPMN 2.0 erläutert wurde, erfolgt in diesem Kapitel die Überführung von BPMN 2.0 in eine UM-konforme Grammatik. Die folgenden Abschnitte beschreiben wie Sprachkonstrukte der bereits existierenden Übersetzung von BPEL in das UM für die Übersetzung von BPMN 2.0 in das UM wiederverwendet werden können. Anschließend befasst sich die Arbeit mit der Übersetzung der nach der Übernahme aus BPEL noch verbleibenden, nicht übersetzten Sprachelemente.

### 4.1 Adaption von Bestandteilen aus der bestehenden Übersetzung von BPEL in das UM

Für die Übersetzung von BPMN 2.0 in das UM scheint es sinnvoll auf bereits bestehende Übersetzungen von anderen Modellierungssprachen in das UM zurückzugreifen, beziehungsweise einzelne Bestandteile zu adaptieren und nicht alle Sprachelemente von BPMN 2.0 in das UM zu übersetzen. Hierfür eignet sich insbesondere BPEL, da einerseits die Übersetzung von BPEL in das UM schon existiert und andererseits BPMN 2.0 erheblich stärker mit BPEL als mit dem UM verwandt ist. Zudem erscheint es kostengünstiger zu den zu übersetzenden Sprachelementen in BPMN 2.0 äquivalente Sprachelemente in der bereits in das UM übersetzten Sprache BPEL zu finden, anstatt diese Sprachelemente erneut zu übersetzen. BPEL und BPMN 2.0 sind zwei grundverschiedene Modellierungssprachen, welche Gemeinsamkeiten aufweisen, aber auch aus unterschiedlichen Sprachkonstrukten bestehen. Somit ist bereits im Voraus absehbar, dass nur ein Teil der Sprachelemente aus der Übersetzung von BPEL in das UM übernommen werden kann.<sup>45</sup> Eine Veranschaulichung des Übersetzungsansatzes findet sich in nachfolgender Abbildung 7.

---

45 Vgl. Görlach K. (2014c)

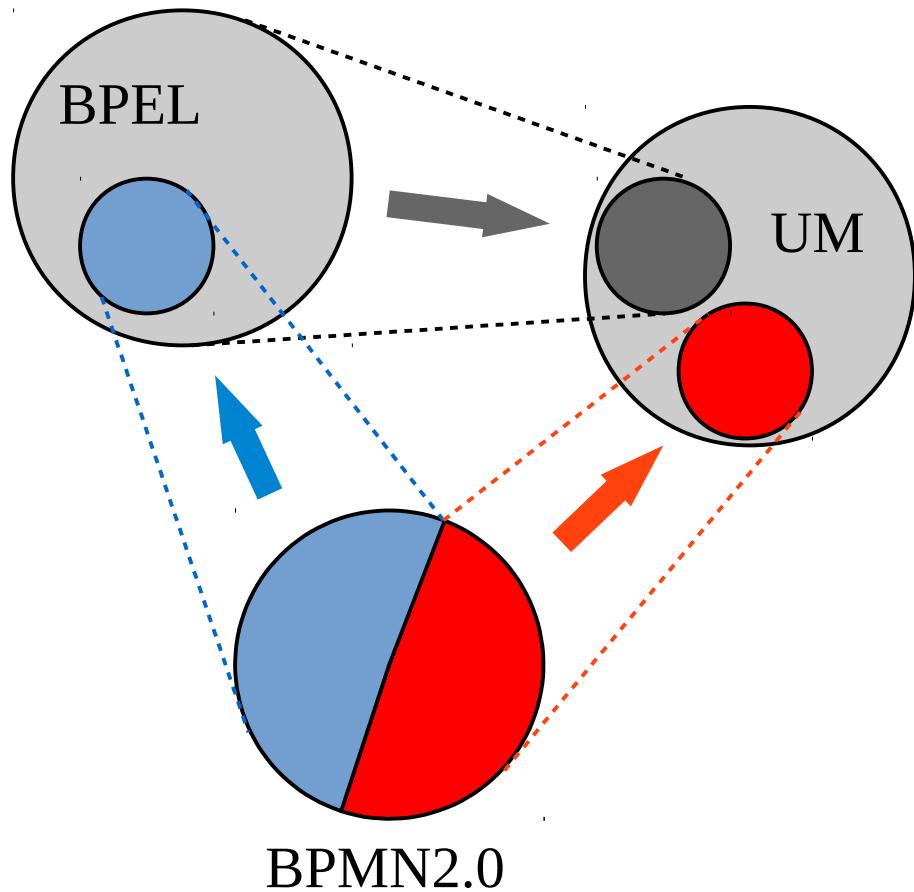


Abbildung 7: Darstellung des Übersetzungsansatzes von BPMN 2.0 über BPEL in das UM<sup>46</sup>

Der dunkelgrau markierte Bereich beschreibt die Menge der Sprachelemente, welche von BPEL in das UM übersetzt wurden. Der blau markierte Bereich beschreibt den Versuch aus Modellierungselementen von BPMN 2.0 Sprachkonstrukte in BPEL nachzubauen. Konstrukte für welche das gelingt, können auf die bereits bestehende Übersetzung von BPEL in das UM zurück greifen. Zuletzt beschreibt der rot markierte Bereich die Menge der Modellierungselemente die direkt in das UM übersetzt werden mussten.

Bei der Übersetzung kann zwischen drei Gruppen von Modellierungselementen unterschieden werden. Die erste Gruppe besteht aus den Sprachelementen, die sich in BPMN und BPEL nicht voneinander unterscheiden. Diese können beinahe eins zu eins übernommen werden und es sind höchstens kleine Anpassungen erforderlich. Hierzu zählen beispielsweise die Logik der sequentiellen und parallelen Ausführung von Aufgaben, sowie einige Elemente der Kontrollflussteuerung, wie die Verarbeitung von Konditionen und if-then-else-Konstrukten. Außerdem kann die Schleifenlogik übernommen werden.

Die zweite Gruppe enthält die Menge der Sprachkonstrukte in BPMN 2.0, zu denen es keine

---

46 Eigene Darstellung

direkten Äquivalente in BPEL gibt, deren Verhalten sich aber durch die Verknüpfung mehrerer BPEL-Sprachelemente miteinander simulieren lässt. Hierbei besteht eine gewisse Unschärfe in der Abgrenzung, da das Verhalten der meisten BPMN 2.0-Elemente mit beliebig hohem Aufwand und großen Prozessen durch BPEL-Konstrukte dargestellt werden kann.<sup>47</sup> Als Abgrenzungskriterium soll hierbei gelten, ob es ohne unverhältnismäßigen Aufwand möglich ist, die Funktion des ursprünglichen Sprachelementes an seinem übersetzten Pendant abzulesen. Auch hier verbleibt durch die subjektive Bewertbarkeit eine gewisse Restunschärfe.

Die Idee der Übersetzung von BPEL in BPMN 2.0 respektive BPMN 2.0 in BPEL ist nicht neu und wird bereits in der Spezifikation von BPMN 2.0 ausführlich behandelt.<sup>48</sup> Tatsächlich wird seit vielen Jahren an einer Übersetzung gearbeitet und insbesondere Unternehmen haben ein großes Interesse daran einen höheren Grad an Vergleichbarkeit und Interoperabilität in ihren Geschäftsprozessen zu erreichen.<sup>49</sup>

Bisherige Forschungsergebnisse haben allerdings ergeben, dass es keine funktionsgleiche Übersetzung zwischen BPMN 2.0 und BPEL gibt,<sup>50</sup> da sich BPEL und BPMN 2.0 in elementaren Konzepten, wie der Ereignis-, Terminierungs-, Ausnahme- und Kompensationsbehandlung wesentlich voneinander unterscheiden. Dies führt so weit, dass für genannte Funktionen keine funktionsgleiche Überführung zwischen den Modellierungssprachen möglich ist.

Dies lässt sich an der Gegenüberstellung des Terminierungsverhaltens von BPEL und BPMN erkennen, wie es in Tabelle 2 dargestellt ist.

---

47 Vgl. Weidlich, M., Decker, G., Großkopf, A., & Weske, M. (2008)

48 Vgl. hierzu und Folgende OMG (Hrsg.) (2011), <http://www.omg.org>

49 Vgl. Projektleiter Erprobung (2014)

50 Vgl. Weidlich, M., Decker, G., Großkopf, A., & Weske, M. (2008)

BPMN 2.0	BPEL
Terminierung beendet den Prozess und alle darin enthaltenen Instanzen und Aktivitäten sofort. Es findet keine Ereignisbehandlung oder Kompensation statt.	Verwendet auf Scope-Ebene beziehungsweise Prozessebene eine Reihe von Handlern auf, welche für die Behandlung von Fehlern, Terminierung, Kompensation und Ereignissen die im Rahmen des Scope bzw. des Prozesses auftreten verantwortlich sind. Bei Aufruf der Terminierung wird der Prozess, bei verschachtelten Prozessen angefangen auf der innersten Ebene nach aussen heraus beendet. Erst wenn alle Aktivitäten innerhalb des Scope beendet sind, wird die Kontrolle an den Terminierungs- Handler übergeben, welcher dann die mit ihm verbundenen Terminierungstätigkeiten ausführt. Bei Bedarf kann der Terminierungs Handler auch auf Kompensierungsaktivitäten zurück greifen. Terminierungen werfen grundsätzlich keine Fehler in BPEL und können diese entsprechend auch nicht propagieren.

Tabelle 2: Vergleich des Terminierungsverhalten zwischen BPMN 2.0 und BPEL<sup>51</sup>

Eine mögliche Wiedergabe des Terminierungsverhaltens in BPEL mit Hilfe von BPMN schlagen Weidlich, Großkopf und XY, wie in Abbildung 8 modelliert, vor.

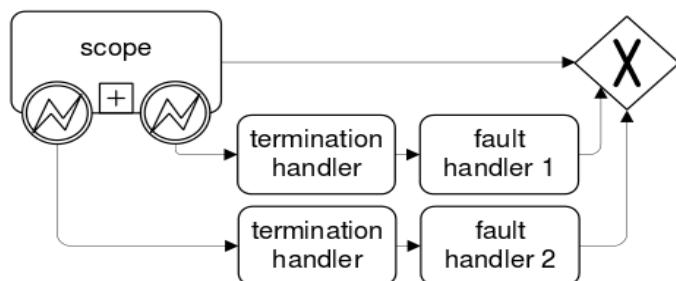


Abbildung 8: Terminierungsverhalten in BPMN 2.0<sup>52</sup>

Hier wird versucht über Fehlerereignisse in BPMN2.0 das Termination Handling in BPEL nachzubauen. Nach geworfenem Fehler wird der Sequenzfluss zuerst an eine Aktivität weitergegeben, welche den Termination-Handler nachstellt und anschließend weiter an den Fault-Handler gereicht. Jeder in BPEL definierte Handler wird versucht über ein eigenes Fehlerereignis in

51 Eigene Darstellung

52 Enthalten in: Weidlich, M., Decker, G., Großkopf, A., & Weske, M. (2008)

BPMN 2.0 wiederzugeben. Dennoch ergeben sich hier in der Ausführung Unterschiede, so dass von der Idee der Überführung Abstand genommen werden kann.

Zuletzt beschreibt die dritte Gruppe die Menge der Sprachelemente in BPMN 2.0, zu denen es weder Äquivalente in BPEL gibt, noch diese sinnvoll durch BPEL-Konstrukte ausgedrückt werden können. Elemente dieser Gruppe müssen zwangsläufig einzeln in das UM übersetzt werden.

## 4.2 Übersetzung einzelner Sprachelemente

Inhalt dieses Kapitels ist die isolierte Übersetzung einzelner Sprachelemente von BPMN 2.0 in das UM. Es werden beispielhaft einige zentrale Konzepte, die für die Übersetzung relevant sind, vorgestellt, sowie deren Herleitung erläutert.

### 4.2.1 Tokenfluss

BPMN 2.0 determiniert die Ausführungsreihenfolge eines Geschäftsprozesses mit Hilfe von Tokens, welche auf den ausgehenden Pfaden von Fluss-Objekten versandt werden.<sup>53</sup> Die erfolgreiche Ausführung einer Aufgabe endet stets mit dem Versand eines Tokens auf dem ausgehenden Pfad der Aufgabe. Ein Pfad verbindet stets zwei Fluss-Objekte miteinander. Die Anzahl der auf einem Pfad versandten Tokens ist eine elementare Information für den Prozessablauf. Insbesondere einige fortgeschrittene Modellierungselemente in BPMN 2.0, wie das Inklusive und das Komplexe Gateway, sind bei ihrer Verwendung auf die Information angewiesen, ob auf den in sie eingehenden Pfaden jeweils bereits ein Token eingegangen ist, definitiv kein Token mehr eingehen wird oder die Auswertung noch nicht abgeschlossen ist. Da es sich dabei um eine Information handelt, welche aufgrund der Verwendung von Konditionen und Ereignissen unter Umständen erst zur Laufzeit bestimmt werden kann, muss der Tokenfluss in der Grammatik explizit modelliert werden.

Eine Veranschaulichung dieser Problematik ist anhand eines BPMN 2.0-Prozesses in Abbildung 9 dargestellt.

---

<sup>53</sup> Vgl. hierzu und Folgende OMG (Hrsg.) (2011), <http://www.omg.org>

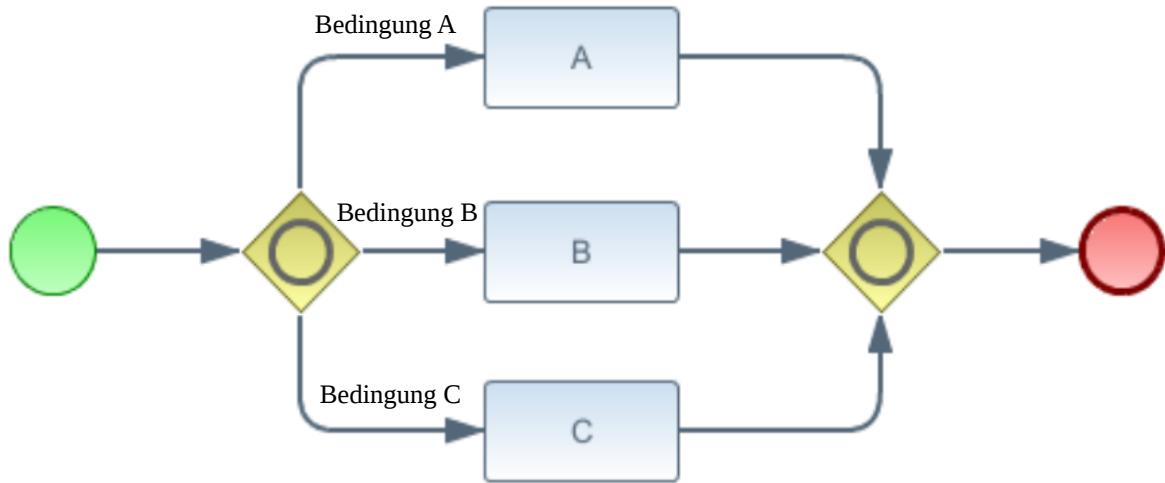


Abbildung 9: Tokenfluss in einem Prozess mit Inklusivem Parallelen Gateway<sup>54</sup>

Nach Prozessbeginn wird der Prozessfluss über ein Inklusives Gateway aufgesplittet, wobei die Pfade, anhand der Bedingungen A, B und C evaluiert werden.<sup>55</sup> Abhängig von den Ergebnissen der Bedingungsauswertung werden Tokens auf den Pfaden versandt. Das schließende Inklusive Gateway wartet in der Zwischenzeit auf den Eingang eines Tokens auf allen Pfaden, auf denen ein Token auch eingehen kann. Welche Pfade das sind, kann ausschließlich zur Laufzeit bestimmt werden, da dies abhängig von der Auswertung der einzelnen Bedingungen ist. Somit muss diese Information stets entlang der Pfade geführt werden. Dieser Umstand hat ebenfalls wesentliche Unterschiede in der Übersetzung von BPMN 2.0 in das UM zur Übersetzung von BPEL in das UM zur Folge.

In der Grammatik wird ein Token durch ein Terminalsymbol modelliert. Dieses wird durch ein „x“ dargestellt, welches, tiefer gestellt, sowohl den Namen des Quell- als auch des Ziel-Fluss-Objektes , getrennt durch ein Komma, enthält. Das Terminalsymbol  $x_{A,B}$  gibt somit an, dass es sich um ein Token handelt, welches sich ausgehend von einem Fluss-Objekt mit dem Namen A, auf einem Pfad befindet, welcher als nächstes ein Fluss-Objekt mit dem Namen B erreicht.

Sollte auf einem Pfad mit Sicherheit kein Token mehr versandt werden können, wird dies durch das Terminalsymbol f dargestellt, welches analog zur Tokendarstellung, tiefer gestellt, wieder Quell- und Ziel-Fluss-Objekt beschreibt.

In Abbildung 10 wird veranschaulicht wie ein Token durch Aktivitäten weitergeleitet wird.

54 Eigene Darstellung

55 Vgl. hierzu und Folgende OMG (Hrsg.) (2011), <http://www.omg.org>

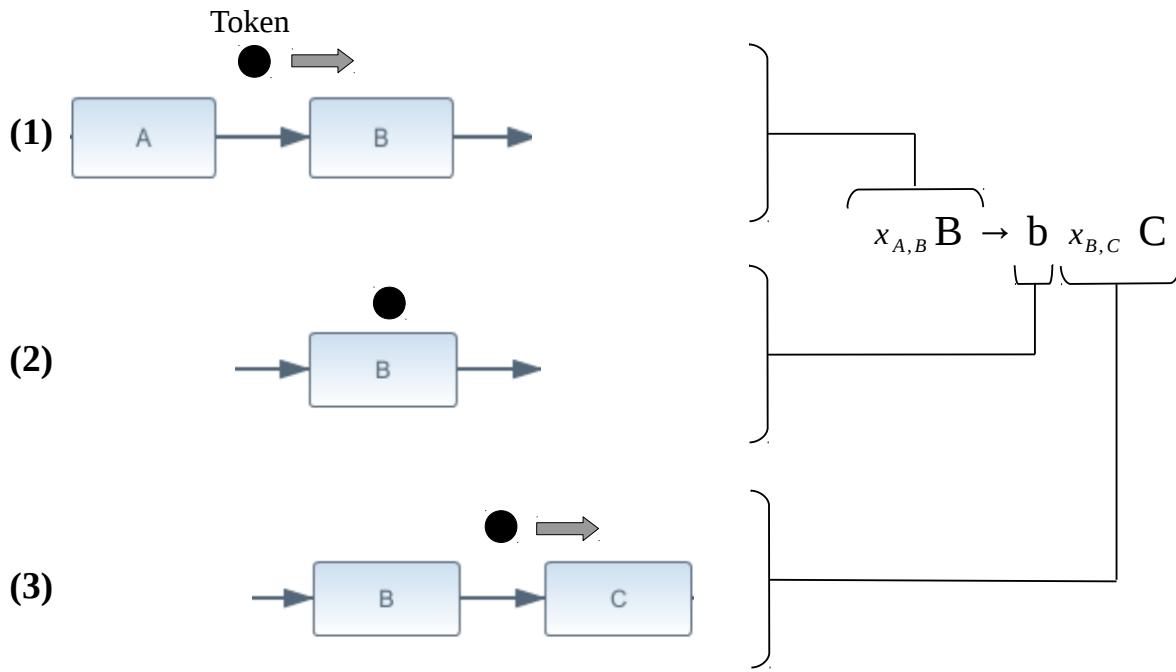


Abbildung 10: Modellierung der Tokenweiterleitung im UM<sup>56</sup>

In Schritt (1) befindet sich das Token auf dem Pfad von Aktivität A zu B. Im UM wird dieser Zustand in den Produktionsregeln dadurch dargestellt, dass das Token  $x_{A,B}$  gefolgt von der Aktivität B auf der linken Seite der Produktionsregel steht. Die Ausführung der Aktivität B erfolgt in Schritt (2). Im UM entspricht der Ausführung dieser Aktivität der Übergang von der linken Seite auf die rechte Seite der Produktionsregel. Nach erfolgreicher Ausführung von Aktivität B, gibt diese das Token in Schritt (3) auf ihrem ausgehenden Pfad weiter. Da dieser in Aktivität C mündet, wird, um die erfolgte Ausführung der Aktivität auszudrücken, auf der linken Seite der Produktionsregel zuerst das Terminal b geschrieben. Diesem folgt das Terminalsymbol  $x_{B,C}$ , welches den ausgehenden Token repräsentiert. Zuletzt wird das Nichtterminalsymbol C auf die rechte Seite der Produktionsregel geschrieben, da das ausgehende Token diese Aktivität als nächstes erreichen wird.

Grundsätzlich ist es nicht notwendig, den Tokenfluss an jeder Stelle im Prozess explizit zu modellieren, da nur wenige Modellierungselemente von BPMN 2.0 diesen in genannter Ausführlichkeit erfordern. Es wäre möglich, den Tokenfluss implizit zu gestalten und die ausführliche Variante nur in der Umgebung der Modellierungselemente zu verwenden, welche diese tatsächlich benötigen. Um eine konsistente Übersetzung über die gesamte Grammatik hinweg zu gewährleisten, wird auf diese Vereinfachung verzichtet.

#### 4.2.2 Parallele Ausführung

Ein typisches Kennzeichen vieler, in der Praxis vorkommenden Geschäftsprozesse ist die

<sup>56</sup> Eigene Darstellung

Modellierung parallel bearbeitbarer Aufgaben.<sup>57</sup> Für die Ausführung bedeutet dies, dass die parallel auftretenden Aufgaben entweder tatsächlich parallel, das heißt gleichzeitig, oder sequentiell, dann allerdings in beliebiger Reihenfolge, ausgeführt werden.

Das UM unterstützt dieses Ausführungsverhalten nativ, da alle auf dem Turingband stehenden Symbole über die C-Interpretation von Produktionsregeln<sup>58</sup> in beliebiger Reihenfolge und auch mit gegebenenfalls dazwischenstehenden anderen Symbolen ausgeführt werden können. Eine Übersetzung dieses Sprachelements ist somit nicht erforderlich. In Abbildung 11 ist die parallele Ausführung dreier Tätigkeiten innerhalb eines BPMN 2.0-Beispielprozesses dargestellt.

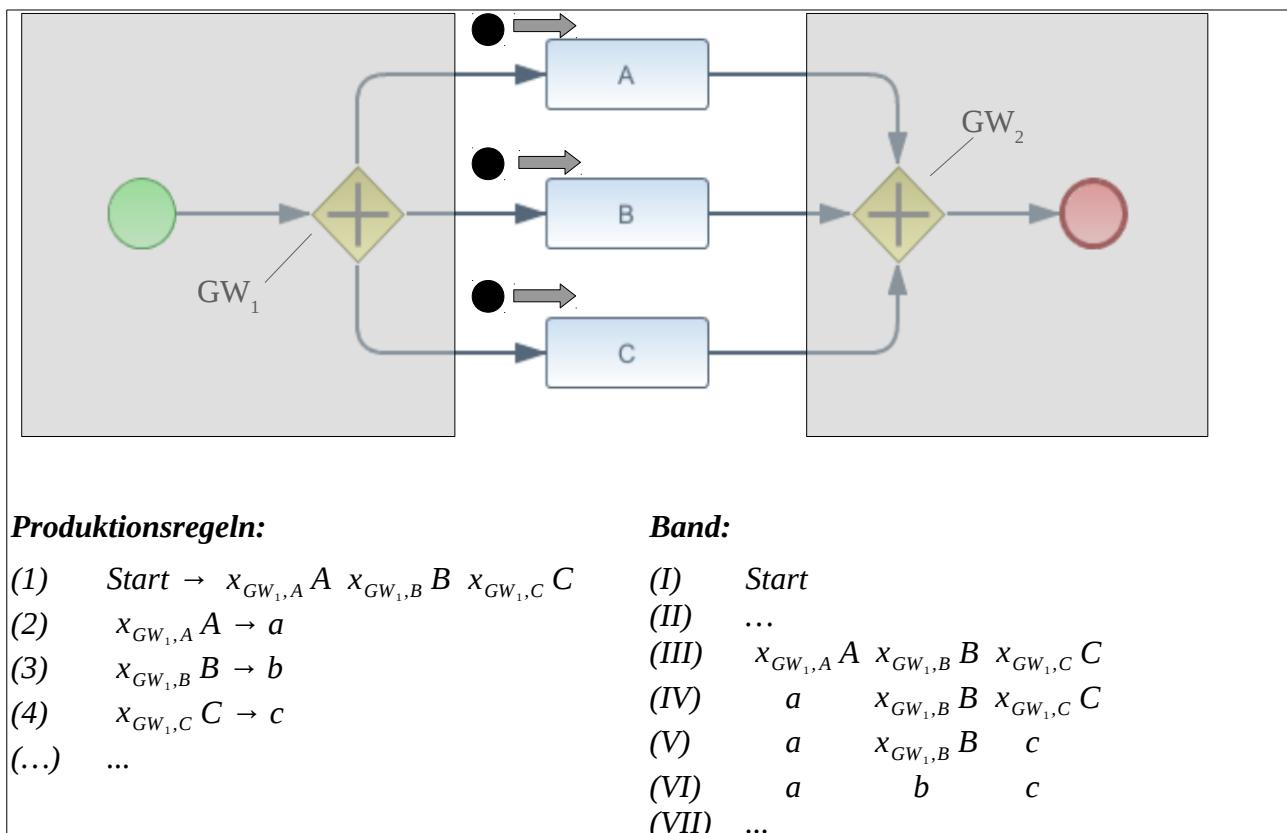


Abbildung 11: Paralleler Sequenzfluss<sup>59</sup>

Ausgehend von einem Startsymbol werden über ein Paralleles Gateway drei Tätigkeiten parallel zueinander ausgeführt, bevor ein weiteres Paralleles Gateway den Prozessfluss wieder synchronisiert und anschließend den Prozess beendet. Die grau hinterlegten Prozessteile, einschließlich der Gateways, können für diese Betrachtung unberücksichtigt bleiben, da sie für die Erläuterung der Parallelität nicht erforderlich sind. In der dargestellten Ausgangssituation befindet sich jeweils ein Token auf dem eingehenden Pfad zu den Tätigkeiten A, B und C. Das Band befindet sich im Zustand (III) und es können die Produktionsregeln (1), (2) sowie (3) ausgeführt werden. Die

57 Vgl. Wertstromanalyse (2014)

58 Vgl. Görlich, K., Leymann, F., & Claus, V. (2013)

59 Eigene Darstellung

Reihenfolge, in welcher die Produktionsregeln angewandt werden können, ist dabei beliebig. Damit wird die Parallelität der Ausführung sicher gestellt. In dem Beispiel in Abbildung 11 werden die Produktionsregeln in der Reihenfolge (2), (4) und (3) ausgeführt, welche zu den Zuständen (IV), (V) und (VI) führen.

#### 4.2.3 Sequentielle Ausführung

Die sequentielle Ausführung von Tätigkeiten, ist durch die strikte Einhaltung der Ausführungsreihenfolge gekennzeichnet. Im UM wird eine auszuführende Tätigkeit durch ein Nichtterminalsymbol und die Ausführung einer Tätigkeit durch den Übergang von einem Nichtterminalsymbol zu einem Terminalsymbol modelliert. Ein Beispiel hierfür ist in Abbildung 12 dargestellt.

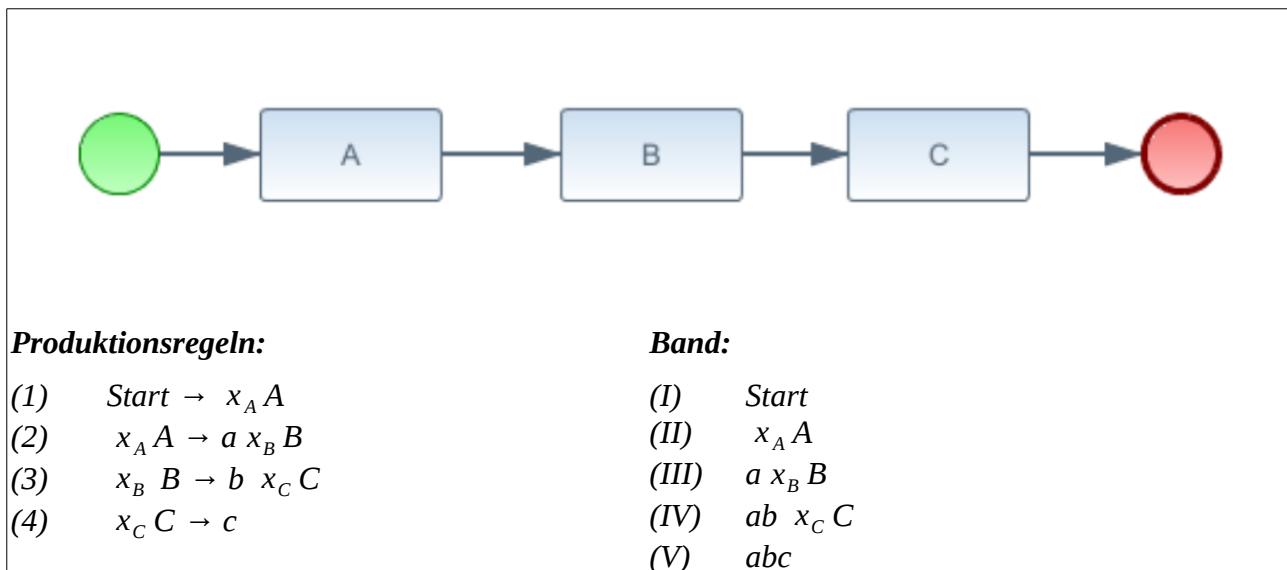


Abbildung 12: Sequentielle Ausführung<sup>60</sup>

Aufgrund der C-Interpretation von Produktionsregeln<sup>61</sup> dürfen für die sequentiell auszuführenden Tätigkeiten nicht gleichzeitig mehrere Produktionsregeln anwendbar sein, welche eine der Tätigkeiten ausführen können (1). Somit darf zu Beginn der Ausführung stets nur ein Token auf dem ausgehenden Pfad des Startsymbols versandt werden. Das Token wird anschließend vom Nichtterminalsymbol der ersten Tätigkeit der Sequenz konsumiert und die Tätigkeit wird ausgeführt (2). Die Anwendung der Produktionsregeln, welche eine Tätigkeit der Sequenz ausführt, muss ebenfalls die Ausführung der jeweils nachfolgenden Tätigkeit - direkt oder indirekt - ermöglichen. Um dies zu erreichen, erzeugt die Ausführung der in der Sequenz als nächstes stehenden Aufgabe im selben Schritt das Token für die nächste Tätigkeit in der Sequenz (3). Gleichzeitig wird das für die in der Sequenz nachfolgende Tätigkeit stehende Nichtterminalsymbol produziert.

60 Eigene Darstellung

61 Vgl. Görlach, K., Leymann, F., & Claus, V. (2013)

Prinzipiell wäre es möglich eine Sequenz, wie im Technischen Bericht von Katharina Görlachs beschrieben<sup>62</sup>, bereits im ersten Schritt der Ausführung vollständig zu erzeugen und auf das Band zu schreiben. Anschließend würde das Token schrittweise durch die Sequenz geleitet und die entsprechenden Aufgaben würden ausgeführt. Hierauf wurde bewusst verzichtet, da der Übersetzungsalgorithmus dadurch vereinfacht werden konnte.

#### 4.2.4 Inklusives Gateway

Aufsplittende Gateways verfügen lediglich über einen eingehenden Pfad und werden daher in jedem Fall ausgeführt, wenn sie ein Token empfangen. Alle Gateways haben stets mehrere ausgehende Pfade. Auf welchen dieser Pfade das Gateway Tokens im Rahmen seiner Ausführung verschickt, ist abhängig von der Art des Gateways und gegebenenfalls vorhandener Bedingungen, die ebenfalls vom Gateway ausgewertet werden. Somit ist zum Zeitpunkt der Ausführung bekannt, über welche ausgehende Pfade Tokens verschickt werden.<sup>63</sup>

Das aufsplittende Inklusive Gateway entspricht im Wesentlichen dem aufsplittenden Exklusiven Gateway. Bei der Ausführung des Gateways wird für jeden ausgehenden Pfad einzeln geprüft, ob die mit diesem Pfad verbundene Bedingung erfüllt ist. Nur wenn dies der Fall ist, versendet das Gateway ein Token auf dem jeweiligen Pfad.

Die Auswertung der Bedingung erfolgt dabei ähnlich über ein dreidimensionales Nichtterminal, welches die Bedingung an einen externen Service zur Auswertung übergibt. Die Bedingungsauswertung kann entweder zu „wahr“ oder zu „falsch“ ausgewertet werden und es wird abhängig vom Ergebnis der Auswertung ein Nichtterminalsymbol erzeugt, anhand dessen der weitere Verlauf des Tokenflusses modelliert wird. Abhängig von den Bedingungen kann es möglich sein, dass alle Bedingungen zu „falsch“ ausgewertet werden. In diesem Fall würde die Prozessausführung zum Stillstand kommen, da von dem Gateway keine Tokens mehr ausgehen können. Um diesem Umstand vorzubeugen, wurde der sogenannte Standardfluss eingeführt. Er fungiert wie ein gewöhnlicher ausgehender Pfad des Gateways. Allerdings wird über ihn ausschließlich dann ein Token versandt, wenn alle Bedingungen der anderen Pfade zu „falsch“ ausgewertet wurden.

Die Übersetzung des aufsplittenden Inklusiven Gateways in das UM ist in Abbildung 13 dargestellt.

---

62 Vgl. Görlach, K. (2013), S. 26

63 Vgl. hierzu und Folgende OMG (Hrsg.) (2011), <http://www.omg.org>

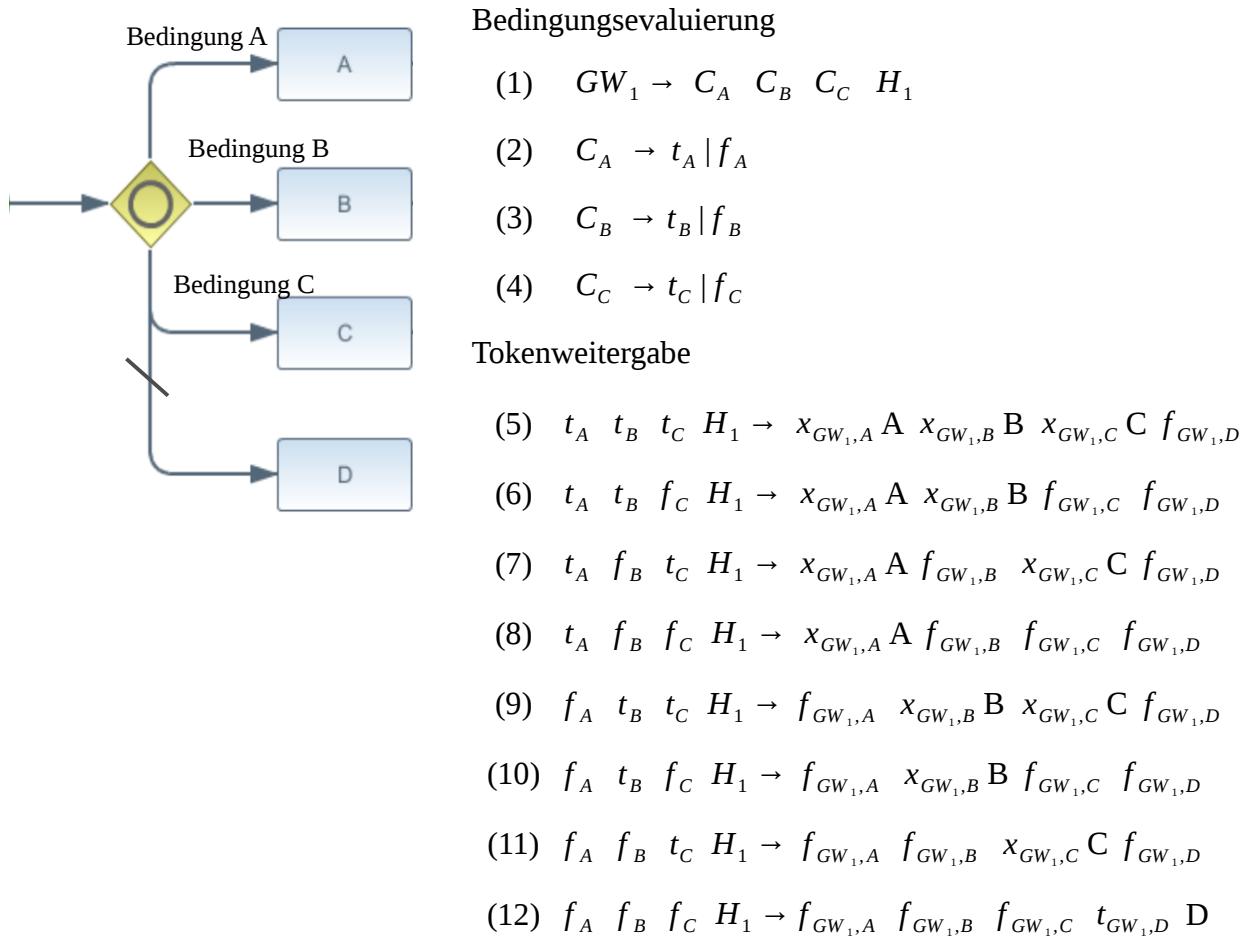


Abbildung 13: Aufsplittendes Inklusives Gateway

Die Ausführung erfolgt zweistufig. Zuerst werden in den Produktionsregeln (1) bis (4) alle Bedingungen ausgewertet und es wird geprüft, ob der Standardfluss eingeschlagen wird. Anschließend werden, basierend auf den Auswertungsergebnissen der Bedingungen, in den Produktionsregeln (5) bis (12) die Tokens auf den ausgehenden Pfaden versandt.

Mit der ersten Produktionsregel (1) werden die dreidimensionalen Nichtterminalsymbole  $C_A$ ,  $C_B$  und  $C_C$  generiert. Sie repräsentieren jeweils eine Bedingung des Inklusiven Gateways und werden in beliebiger Reihenfolge mit den Produktionsregeln (2) bis (4) ausgewertet. Je nachdem ob die Bedingung zu „wahr“ oder „falsch“ evaluiert, wird sie in ein entsprechendes Terminalsymbol überführt. Das Hilfssymbol  $H_1$  stellt sicher, dass die Ausführung des Gateways erst voranschreitet, wenn alle Bedingungen ausgewertet sind.

In den Produktionsregeln (5) bis (12) werden jeweils alle Kombinationsmöglichkeiten aus den beiden Pfadzuständen „auf dem Pfad ist ein Token eingegangen“ und „es kann auf diesem Pfad kein Token eingehen“ und der Menge der eingehenden Pfade in jeweils einer eigenen

Produktionsregel abgebildet. Die Anzahl der zur Abbildung dieses Umstandes benötigten Produktionsregeln entspricht damit  $2^n$ , wobei  $n$  die Anzahl der eingehenden Pfade ist. Somit kommt es selbst bei einer geringen Anzahl an eingehenden Pfaden zu einer kombinatorischen Explosion und es werden große Mengen an Produktionsregeln notwendig. Dies ist jedoch für die Grammatik nicht abträglich, da weder die Übersetzungsgeschwindigkeit noch die Lesbarkeit durch den Menschen Anforderungen an die Grammatik darstellen.<sup>64</sup> Generiert werden auf der rechten Seite der Produktionsregel jeweils die Tokens für die ausgehenden Pfade des Gateways. Auf jedem Pfad, dessen Bedingung zu „wahr“ ausgewertet wurde, wird ein Token versandt. Auf den übrigen ausgehenden Pfaden wird die Information weitergegeben, dass kein Token mehr auf diesem Pfad verschickt wird.

Ähnlich gestaltet sich das Verhalten bei der zusammenführenden Variante des Inklusiven Gateways. Dieses führt erst aus, wenn auf allen eingehenden Pfaden, auf denen ein Token hätte eingehen können, auch ein Token eingegangen ist. Insbesondere bei der Verwendung von Bedingungen ist die Information, auf welchen Pfaden Tokens versandt werden, erst zur Laufzeit verfügbar. Siehe hierzu das Beispiel aus Abbildung 9. Ob das zusammenführende Gateway in dem Beispiel auf ein Token auf dem untersten Pfad wartet, ist davon abhängig, ob die Bedingung C zuvor zu „wahr“ ausgewertet wurde. In diesem Fall wartet das Gateway auf diesem Pfad auf ein eingehendes Token. Die Abbildung dieses Verhaltens in Produktionsregeln erfolgt analog zu der Realisierung der Tokenweitergabe bei dem aufsplittenden Inklusiven Gateway und ist in Abbildung 14 dargestellt.

---

<sup>64</sup> Vgl. Görlach, K. (2015a)

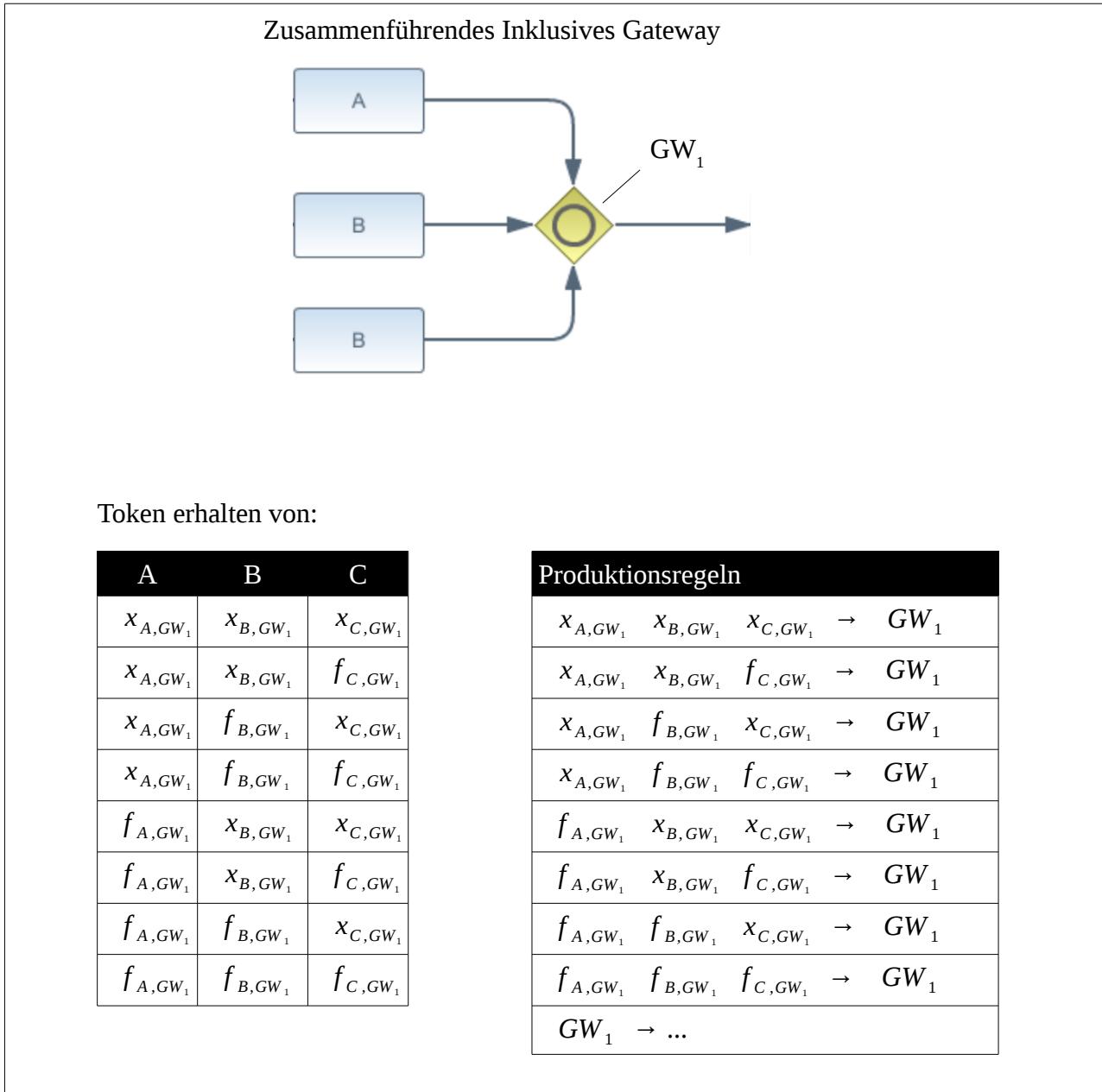


Abbildung 14: Kombinatorische Explosion von Produktionsregeln<sup>65</sup>

Die Produktionsregeln bilden auf ihrer linken Seite alle möglichen Kombinationen an Pfadzuständen ab, wobei eine Produktionsregel eine mögliche Zustandskombination darstellt. Somit ist es dem Gateway erst möglich auszuführen, wenn alle eingehenden Pfade verarbeitet wurden. Anschließend setzt die Prozessausführung fort.

## 4.3 Übersetzungsalgorismus

Die Übersetzung komplexer Prozesse geht, wie auch bei gesprochenen Sprachen, über die

<sup>65</sup> Eigene Darstellung

Übersetzung der isoliert betrachteten Wörter hinaus und muss den Kontext der Wörter im Rahmen eines Satzes beziehungsweise der Grammatik berücksichtigen, um ein richtiges Wort oder einen Satz der Sprache zu erzeugen. Hierzu wird ein Algorithmus entwickelt, welcher einen BPMN 2.0-Prozess durch Rückgriff auf die Übersetzung der einzelnen Modellierungselemente, vollständig und automatisch in das UM überführen kann. Ein wesentliches Ziel bei der Erarbeitung des Algorithmus ist die Eigenschaft, eine Grammatik mit nur einmaligem Durchlauf zu übersetzen und gleichzeitig eine UM-konforme Grammatik zu erzeugen. Die hierfür verwendete und den Geschäftsprozess repräsentierende Datenstruktur wird in Java durch das Eclipse Modeling Frameworks (EMF) und den BPMN 2.0 Modeler erstellt.

Diese Datenstruktur besteht aus einer Menge von BPMN 2.0-Modellierungselementen, im Folgenden Knoten genannt, die miteinander verknüpft sind und einen Graphen aufspannen. Bei jedem Besuch eines Knotens erstellt der Übersetzungsalgorithmus die entsprechenden Produktionsregeln ausschließlich für den aktuell besuchten Knoten, indem er sich der erarbeiteten Übersetzungsmuster bedient und diese in den Kontext des jeweiligen Knotens setzt. Ausgehend vom Startereignis durchläuft der Algorithmus mit Hilfe der Breitensuche alle Knoten im Geschäftsprozess. Von jedem besuchten Knoten aus werden all diejenigen Nachbarknoten untersucht, welche auf Pfaden liegen die sich maximal zwei Knoten von dem aktuell besuchten Knoten entfernt befinden. Das heißt diejenigen Objekte, welche über ein Fluss-Objekt mit dem jeweiligen Modellierungselement verbunden sind. Aufgrund der Realisierung des Tokenflusses, auch für den Fall, dass kein BPMN 2.0-Token weitergeleitet wird, reicht die vom besuchten Knoten ausgehende, lokal begrenzte Sichtweise auf die verbundenen Fluss-Objekte aus, um alle benötigten Produktionsregeln zu erzeugen. Zudem kommt bei dem Durchlauf durch den Geschäftsprozess ein Markierungsalgorithmus zur Anwendung, welcher ein Protokoll über die bereits besuchten Knoten führt. So wird verhindert, dass ein Knoten mehrfach besucht wird und Produktionsregeln mehrfach erzeugt werden. Dies kann in bestimmten Situationen eine fehlerhafte Grammatik erzeugen.

In nachstehender Abbildung 15 wird anhand eines Parallelen Gateways veranschaulicht, wie der Übersetzungsalgorithmus beim Durchlauf durch den Geschäftsprozess eine korrekte Grammatik erzeugt.

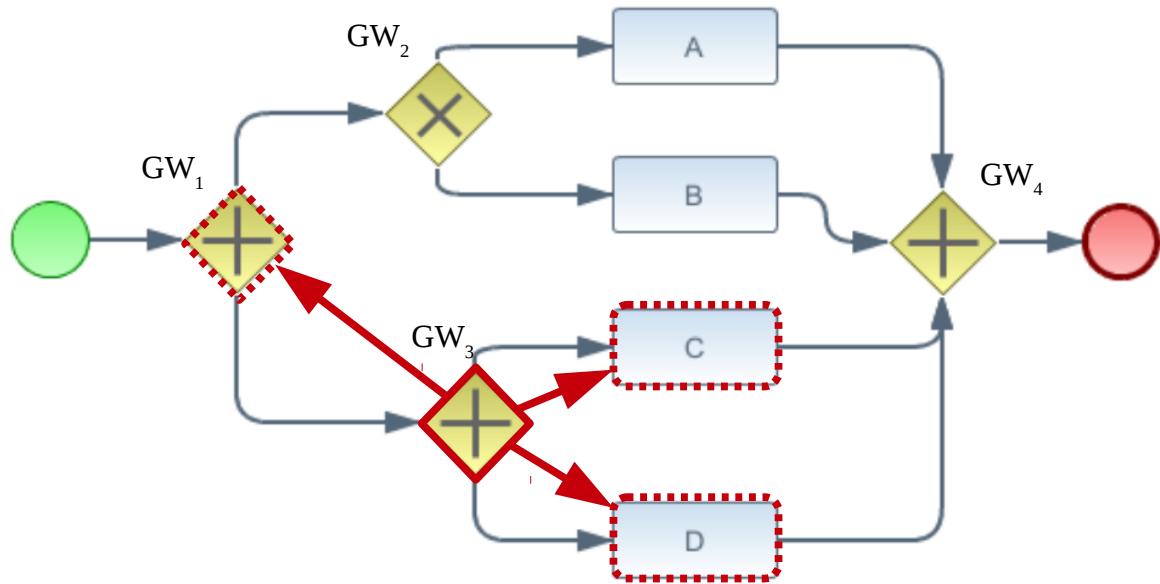


Abbildung 15: Veranschaulichung des Übersetzungsalgorithmus<sup>66</sup>

In dem Beispielprozess sei das durchgehend rot umrahmte Gateway ( $GW_3$ ) der aktuell vom Algorithmus besuchte Knoten. Seine direkten Nachbarknoten sind die Pfade  $<GW_3, C>$ ,  $<GW_3, D>$  sowie  $<GW_1, GW_3>$ . Über diese Pfade mit dem Gateway verbunden, sind das Gateway  $GW_1$  sowie die Aktivitäten C und D. Die allgemeine Übersetzung des Parallelen Gateways wird nun als Grundlage genutzt und mit den Spezifika der umliegenden Modellierungselemente zu einer Grammatik ergänzt. Für die Erzeugung der Grammatik greift der Algorithmus auf eine Reihe von unterstützenden Funktionen zurück. So führt er beispielsweise eine Liste mit allen, im Laufe des Prozesses verwendeten Tokens, welche die Terminierung des Prozesses unterstützen.

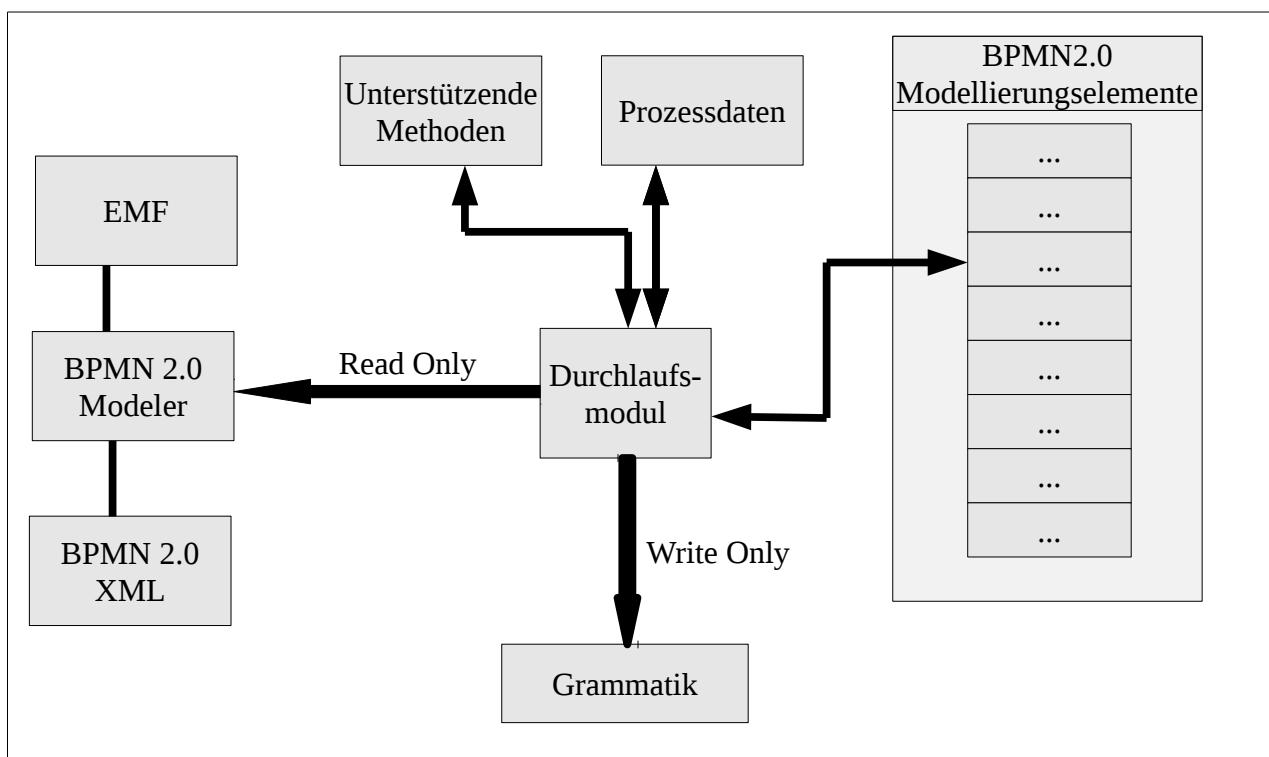
## 5 Implementierung der Software

### 5.1 Anforderungen

Die entwickelte Java-Software soll in der Lage sein, die XML-Datei eines in BPMN 2.0 modellierten Geschäftsprozesses mit Hilfe des EMFs und des BPMN 2.0 Modelers einzulesen, in das UM zu übersetzen und eine \*.grammar-Datei im XML-Format auszugeben.<sup>67</sup> Diese \*.grammar-Datei soll eine UM-konforme Grammatik des BPMN 2.0-Geschäftsprozesses enthalten. Insbesondere soll die Übersetzungslogik der einzelnen Modellierungselemente unkompliziert zu erweitern sein. Im Idealfall wird hierfür eine Klasse pro BPMN 2.0-Modellierungselement verwendet, in welcher die Übersetzungslogik weitgehend abgekapselt von der restlichen Software ist. Der Performanz der Übersetzung wird kein großer Stellenwert zugerechnet, da lediglich eine Übersetzung pro Geschäftsprozess notwendig ist, bevor der Geschäftsprozess beliebig oft mit der UE ausgeführt werden kann. Von der Korrektheit der BPMN 2.0-Geschäftsprozesse wird ausgegangen, eine vorherige Prüfung auf eventuelle Fehler soll nicht durchgeführt werden. Ausführen muss die Software unter Windows 7 mit installiertem Java Development Kit (JDK) in der Version 1.7.

### 5.2 Entwurf

Der Aufbau der entwickelten Software ist dargestellt in Abbildung 16



<sup>67</sup> Vgl. hierzu und im folgenden Görlach, K. (2015b)

*Abbildung 16: Komponenten der Übersetzungssoftware*

Das EMF ist ein Framework, welches Java-Klassen für allgemeine Modellierungsaufgaben zur Verfügung stellt. Aufbauend darauf erweitert der BPMN 2.0 Modeler diese Klassen und realisiert damit ein Java-basiertes BPMN 2.0 Modellierungstool. Mit Hilfe dieses Tools und des zugehörigen Java-Gerüstes können BPMN 2.0 Geschäftsprozesse grafisch in einem Interface gezeichnet und in ein BPMN 2.0 konformes XML Dokument ausgegeben werden.

Das Durchlaufmodul realisiert den Algorithmus, welcher im Zuge der Übersetzung einen BPMN 2.0 Geschäftsprozess durchläuft. Hierfür greift das Modul auf das vom BPMN 2.0 Modeler erweiterte Java-Gerüst zurück. Um spätere Anpassungen zu vereinfachen wurde die Übersetzungslogik der einzelnen Modellierungselemente ausgegliedert. So können diese bei Bedarf auch später noch modifiziert werden, ohne dass größere Änderungen an dem Programm vorgenommen werden müssen. Jede der BPMN 2.0 Modellierungselemente-Klasse repräsentiert ein BPMN2.0 Modellierungselement und realisiert die Übersetzung des jeweiligen Elementes in das UM. Für bestimmte Operationen, wie beispielsweise die Terminierung werden Prozessdaten in einer eigenen Komponente gesammelt und verwaltet. Auf diese kann bei Bedarf zurückgegriffen werden. Ähnlich verhält es sich mit dem Modul „Unterstützende Methoden“, wobei dessen Methoden hauptsächlich Methoden zur Flusskontrolle verwaltet. Ausgehend von der Breitensuche des Übersetzungsalgorithmus werden die Übersetzungsmethoden der jeweils besuchten Knoten aufgerufen. Der Geschäftsprozess wird so sukzessive in eine UM-konforme Grammatik übersetzt.

## 5.3 Testfälle

Die Überprüfung der Korrektheit der Übersetzung gestaltet sich allgemein schwierig. Eine Prüfung mit Hilfe von Testfällen ist geeignet einzelne Aspekte eines Prozesses beziehungsweise der Übersetzung zu überprüfen, kann allerdings lediglich die Korrektheit von Teilaспектen sicherstellen und ist kein Garant für eine korrekte Übersetzung. Da BPMN 2.0 in der Lage ist einen Geschäftsprozess auszuführen, wäre es prinzipiell denkbar, die Ausführung des ursprünglichen BPMN 2.0-Prozesses der Ausführung des übersetzten Prozesses im UM gegenüber zu stellen. Differenzen in der Ausführung könnten hier als Indikatoren für einen Fehler in der Übersetzung gewertet werden.

Alle Modellierungselemente wurden einzeln in einem Testprozess verwendet, welcher ausschließlich das jeweilige Modellierungselement auf seine Funktionen hin überprüft. Somit ließen sich die Funktionalitäten der jeweiligen Modellierungselemente mit typischen Use Cases überprüfen. Da ausserdem die Überprüfung der Terminierung von besonderem Interesse war wurden ausgewählte Beispielprozesse eines Industriepartners mit BPMN 2.0 modelliert und unter verschiedenen Bedingungen auf Terminierung geprüft. In Abbildung 17 ist beispielsweise einer der größeren Testprozesse dargestellt, welcher zur Korrektheitsprüfung der Gateways verwendet wurde.

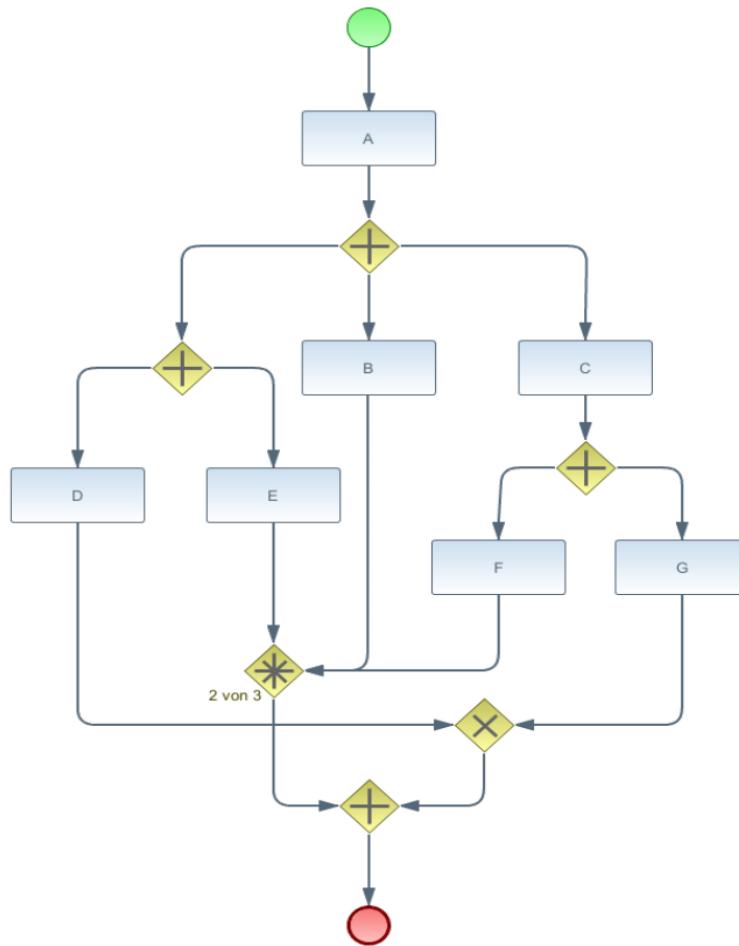


Abbildung 17: Testprozess zur Überprüfung der Korrektheit der Gateways.

Als Testumgebung wurden zwei Rechnern verwendet, auf denen jeweils die Tests abliefern. Einmal auf einem älteren Notebook aus dem Jahre 2007, sowie auf einem moderneren Office Rechner aus dem Jahre 2011. Die Rechner besitzen folgenden Spezifikationen:

Lenovo Thinkpad X61 Tablet

Prozessor: Intel Core2 Duo L7500, 2\*1,6 GHz

Arbeitsspeicher: 3GB

Dell Vostro 460

Prozessor: Intel Core i3-2100, 2\*3,10 GHz

Arbeitsspeicher: 8GB

Beide Testgeräte wurden jeweils mit einem leeren Datenträger ausgestattet, auf denen die Betriebssysteme Windows 7 Professional sowie Ubuntu 14.04.1 LTS, welche derzeit mit zu einigen den verbreitetsten Distributionen der Windows- respektive Linux-Betriebssysteme gehören,<sup>68</sup> neu installiert und eingerichtet. Anschließend wurden die Entwicklungs- und Ausführungsumgebung auf den Rechnern eingerichtet und die Tests wurden durchgeführt.

<sup>68</sup> Vgl. Unsigned Integer Limited (Hrsg.) (2001)

## 6 Schlussbetrachtung

Für die Beschreibung, Dokumentation und Ausführung von Geschäftsprozessen in der heutigen Zeit sind Modellierungssprachen unabdingbar geworden. Aufgrund der großen Vielfalt an verwendeten Modellierungssprachen, die zueinander nicht kompatibel sind, kommt es in der Unternehmenswelt zu erheblichen Schnittstellenproblemen. Das UM ist als unifizierende Meta-Modellierungssprache entwickelt worden, die in der Lage ist alle konventionellen Modellierungssprachen auszuführen. Hierfür ist eine Übersetzung der jeweiligen Modellierungssprache in das UM erforderlich.

Für die Modellierungssprachen BPEL und ConDec existieren bereits jeweils eine Übersetzung und eine Software, welche Geschäftsprozesse in das UM übertragen können. BPMN 2.0 ist zwar eine weit verbreitete Modellierungssprache<sup>69</sup>, die international als ISO-Standard geführt wird, für die jedoch noch keine Übersetzung in das UM existierte. Für die Übersetzung wurde im Rahmen dieser Arbeit versucht auf bisherigen Ergebnissen der bestehenden Übersetzung von BPEL in das UM aufzubauen und diese zu adaptieren. Aufgrund der Verschiedenheit der Modellierungssprachen konnten jedoch nur wenige Modellierungselemente auf diese Weise übernommen werden. Alle anderen Modellierungselemente mussten einzeln in das UM übersetzt werden. Einige wesentliche Konzept bei der Übersetzung wurden am Beispiel von Gateways erläutert.

Für die Automatisierung der Übersetzung wurde ein Algorithmus entwickelt, welcher BPMN 2.0-Geschäftsprozesse durchlaufen kann und dabei eine Übersetzung der Prozesse in das UM vornimmt. Dieser Algorithmus wurde im Rahmen einer Java-Software erarbeitet, die in der Lage ist einen beliebigen BPMN 2.0-Prozess, der im XML-Format vorliegt, in eine UM-konforme Grammatik zu überführen und die zugehörige \*.grammar-Datei zu generieren. Die Software wurde anhand mehrerer Testfälle auf Korrektheit hin untersucht und hat die getesteten BPMN 2.0-Prozesse korrekt übersetzt.

Das UM greift einen leichtgewichtigen und vergleichsweise einfach umsetzbaren Lösungsansatz zu der aktuellen, größeren Problematik von Unternehmen auf, Geschäftsprozesse, unabhängig von den jeweils zugrunde liegenden Modellierungssprachen verwenden und ausführen zu können. Im Gegensatz zu administrativ aufwendigen Lösungen, wie dem Umstieg auf eine andere Modellierungssprache, bietet der Ansatz des UMs den entscheidenden Vorteil, dass bisherige Geschäftsprozesse in den ursprünglich verwendeten Modellierungssprachen weiterhin benutzt werden können. Somit wird der Benutzer nicht zum Umstieg auf eine neue Modellierungssprache gezwungen. Der Wechsel zu einer anderen Modellierungssprache stellt insbesondere für Großunternehmen eine erhebliche Hemmschwelle dar, da mit dem Umstieg auf eine andere Modellierungssprache bisherige Prozesse meist nicht mehr weiterbenutzt werden können. Dies kann zu einem beachtlichen Know-How-Verlust und hohen Kosten für das Unternehmen führen.

Da bereits Übersetzungen für die sehr weit verbreiteten und in der Praxis häufig von Unternehmen

---

69 Vgl. Murzek, M., Rausch, T., & Kühn, H. (2013)

verwendeten Modellierungssprachen BPEL und nun auch für BPMN 2.0 existieren, ist es denkbar, dass das Interesse an dieser Technologie auch weiterhin zunehmen wird und somit auch die Anzahl der Übersetzungen von weiteren Modellierungssprachen in das UM ansteigen wird.

## Quellenverzeichnisse

### Literaturverzeichnis

- Allweyer, T. (2009).** *BPMN 2.0-Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung.* BoD–Books on Demand.
- Freund, J., & Rücker, B. (2014).** *Praxishandbuch BPMN 2.0.* Carl Hanser Verlag GmbH Co KG.
- Schmelzer, H. J.: Sesselmann, W. (2008).** Geschäftsprozessmanagement in der Praxis. Carl Hanser Verlag GmbH Co KG
- Görlach, K. (2013).** A generic transformation of existing service composition models to a unified model. *University of Stuttgart, Technical Report, 1.*
- Görlach, K., Leymann, F., & Claus, V. (2013).** Unified Execution of Service Compositions. *Proceedings of the 6th IEEE International.*
- Horn, C., (2001).** *Lehr- und Übungsbuch Informatik.* Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2001.
- ISO/IEC 19510:2013 (2013)** Information technology – Object Management Group Business Process Model and Notation
- Liang, S. (2013).** Unifizierte Service-Komposition für ConDec.
- Kocian, C. (2011).** *Geschäftsprozessmodellierung mit BPMN 2.0: Business Process Model and Notation im Methodenvergleich.* Hochschule für Angewandte Wissenschaften, Fachhochschule Neu-Ulm.
- Murzek, M., Rausch, T., & Kühn, H. (2013).** BPMN als Bestandteil der BPMS-Modellierungsmethode. In *Prozessmanagement für Experten* (pp. 93-113). Springer Berlin Heidelberg.
- Pesic, M.; Van der Aalst, W. M. (2006, January).** A declarative approach for flexible business processes management. In *Business Process Management Workshops* (S. 169-180). Springer Berlin Heidelberg.
- Recker, J. C.; Mendling, J. (2006).** On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages. In *The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium* (S. 521-532). Namur University Press.
- Shao, B. (2013).** Unified service-composition for BPEL.

**Sadowska, M. (2011).** BPMN 2.0 für Einsteiger-Methoden zu Prozessdesign und Prozessdarstellung.

**Schöning, U. (2008).** Theoretische Informatik kurzgefasst, Spektrum Taschenbuch, 5. Auflage, 2008.

**Weidlich, M., Decker, G., Großkopf, A., & Weske, M. (2008).** BPEL to BPMN: The myth of a straight-forward mapping. In *On the Move to Meaningful Internet Systems: OTM 2008* (S. 265-282). Springer Berlin Heidelberg.

## Gesprächsverzeichnis

**Abteilungsleiter Erprobung (2014)**, persönliches Gespräch

**Görlach K. (2014)**, persönliches Gespräch

**Görlach K. (2014a)**, persönliches Gespräch

**Görlach K. (2014b)**, persönliches Gespräch

**Görlach K. (2014c)**, persönliches Gespräch

**Görlach, K. (2015)**, persönliches Gespräch

**Görlach, K. (2015a)**, persönliches Gespräch

**Görlach, K. (2015b)**, persönliches Gespräch

**Leiter Warenannahme (2014)**, persönliches Gespräch

**Projektleiter Erprobung (2014)**, persönliches Gespräch

**Projektleiter Jahresbericht (2014)**, Besprechung

**Teamleiter Erprobung (2014)**, persönliches Gespräch

**Teamleiter Erprobung (2014a)**, Besprechung

**Teamleiter Erprobung (2014b)**, persönliches Gespräch

**Teamleiter Erprobung (2014c)**, persönliches Gespräch

## Firmeninterne Quellen

**Schulungsunterlagen: Wertstromanalyse nach VSDiA (2014)**

**Wertstromanalyse (2014)**, Workshop zur Prozessoptimierung

## Internetquellen

**Fowler, M. (Hrsg.) (2009): RulesEngine**  
<http://martinfowler.com/bliki/RulesEngine.html>  
(Stand 18.01.2015)

## Unifizierte Service-Kompositionen für BPMN 2.0

### 6 Schlussbetrachtung

---

**Humboldt-Universität zu Berlin (Hrsg.) (2011):** BPMN2.0 – Business Model and Notation

[www.bpmn.de/images/BPMN2\\_0\\_Poster\\_DE.pdf](http://www.bpmn.de/images/BPMN2_0_Poster_DE.pdf)

(Stand 04.02.2015)

**OASIS Committee (Hrsg.) (2007):** Web Service Business Process Execution Language

<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>

(Stand: 10.01.2015)

**OMG (Hrsg.) (2011):** Business Process Model and Notation (BPMN)

<http://www.omg.org/spec/BPMN/2.0/PDF/>

(Stand 10.01.2015)

**Unsigned Integer Limited (Hrsg.) (2001):** DistroWatch Page Hit Ranking

<http://distrowatch.com/dwres.php?resource=popularity>

(Stand 02.02.2015)

## Ehrenwörtliche Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Olaf Hoffeld, Stuttgart, den 13.02.2015