



ulm university universität  
**uulm**

Universität Ulm | 89069 Ulm | Germany

**Fakultät für  
Ingenieurwissenschaften  
und Informatik**  
Institut für Datenbanken  
und Informationssysteme

# **Eine vergleichende Betrachtung verschiedener Ansätze zur Modellierung von Prozessvarianten**

Bachelorarbeit an der Universität Ulm

**Vorgelegt von:**

Gözde Ipek Bayrak

goezde.ipekbayrak@uni-ulm.de

**Gutachter:**

Manfred Reichert

**Betreuer:**

Andreas Lanz

2014

Fassung 17. Dezember 2014

© 2014 Gözde Ipekbayrak

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Satz: PDF-L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

## **Kurzfassung**

Die Modellierung von Geschäftsprozessen spielt eine wichtige Rolle bei der Realisierung von Anforderungen in Unternehmen. Bei der Geschäftsprozessmodellierung wird die Funktionalität und Struktur der Unternehmensprozesse in Form von Prozessmodellen dargestellt. In der Praxis existieren dabei häufig unterschiedliche Varianten eines Prozesses. Die adäquate Modellierung solcher Prozessvarianten ist mit heutigen Geschäftsprozessmodellierungssprachen jedoch nicht möglich. Insbesondere können in diesen Sprachen entweder alle Prozessvarianten in separaten Prozessmodellen oder mittels bedingten Verzweigungen in einem gemeinsamen Prozessmodell abgebildet werden. Beide Möglichkeiten verursachen jedoch erhebliche Nachteile, wie einen erhöhten Wartungsaufwand oder mögliche Inkonsistenzen, die unerwünscht sind.

Um Prozessvarianten umfassend abbilden zu können, wurden in den vergangenen Jahren verschiedene Ansätze entwickelt. Diese Arbeit beschäftigt sich mit den vier Ansätzen C-EPC, C-YAWL, PESOA und Provop. Zunächst werden die Eigenschaften jedes Ansatzes erklärt. Im Anschluss daran, werden diese Ansätze mittels ausgewählter Beispiele miteinander verglichen und kritisch betrachtet. Anschließend werden die Vor- und Nachteile der verschiedenen Ansätze diskutiert.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielsetzung . . . . .	2
1.3	Aufbau der Arbeit . . . . .	2
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Geschäftsprozess . . . . .	5
2.2	Geschäftsprozessmanagement . . . . .	7
2.3	Geschäftsprozessmodellierungssprachen . . . . .	7
2.3.1	EPC . . . . .	8
2.3.2	YAWL . . . . .	11
2.3.3	BPMN . . . . .	14
2.4	Prozessvarianten . . . . .	19
<b>3</b>	<b>Ansätze zur Abbildung von Prozessvarianten</b>	<b>21</b>
3.1	Verhaltensbasierter Ansatz . . . . .	22
3.1.1	C-EPC . . . . .	23
3.1.2	C-YAWL . . . . .	26
3.1.3	PESOA . . . . .	29
3.2	Strukturbasierter Ansatz . . . . .	32
3.2.1	Provop . . . . .	32
<b>4</b>	<b>Vergleich der Ansätze zur Modellierung variantenreicher Prozesse</b>	<b>37</b>
4.1	Prozessbeispiel 1 . . . . .	38

*Inhaltsverzeichnis*

4.2	Prozessbeispiel 2 . . . . .	46
4.3	Prozessbeispiel 3 . . . . .	55
4.4	Zusammenfassung der drei Prozessbeispiele . . . . .	67
<b>5</b>	<b>Zusammenfassung</b>	<b>71</b>

# 1

## Einleitung

### 1.1 Motivation

Die Arbeitsabläufe in Unternehmen werden oftmals in Prozessmodellen dokumentiert. Ein einzelnes Prozessmodell beschreibt üblicherweise einen bestimmten Ablauftyp und legt u.a. die durchzuführenden Aktivitäten, deren Ausführungsreihenfolge sowie die zu ihrer Ausführung erforderlichen Ressourcen fest [HBR08a]. In der Praxis existieren dabei häufig unterschiedliche Varianten eines Prozessmodells, die jeweils in einem bestimmten Szenario gültig sind. Daher wird die Verwaltung dieser Prozessvarianten in der Praxis immer wichtiger. Um Prozesse zu modellieren, gibt es im Allgemeinen verschiedene Modellierungssprachen wie beispielsweise BPMN, EPC und YAWL. Mithilfe dieser Sprachen können jedoch entweder alle Prozessvarianten in separaten Prozessmodellen, oder mittels bedingten Verzweigungen, in einem gemeinsamen Prozessmodell abgebildet

## *1 Einleitung*

werden. Beide Möglichkeiten verursachen allerdings erhebliche Nachteile wie einen erhöhten Wartungsaufwand oder mögliche Inkonsistenzen, die unerwünscht sind.

Um diese Nachteile zu überwinden und Prozessvarianten abbilden zu können, wurden in den vergangenen Jahren verschiedene Ansätze vorgeschlagen. Diese können entweder als verhaltensbasierter oder strukturbasierter Ansatz klassifiziert werden. Verhaltensbasierte Ansätze repräsentieren eine Obermenge der Prozessvarianten innerhalb des gleichen Modells. Es gibt hierbei zwei Möglichkeiten der Variationsdarstellung, das Blockieren/Verbergen-Konzept und die Konfiguration von Variationspunkten. Als Beispiele für diesen Ansatz können C-EPC, C-YAWL und PESOA genannt werden. Im Gegensatz dazu verwenden strukturbasierte Ansätze wie Provop eine Basisvariante des Prozesses sowie Änderungsoperationen wie Einfügen, Löschen, Modifizieren oder Verschieben von Aktivitäten, um Prozessvarianten abzuleiten.

### **1.2 Zielsetzung**

In der Praxis existieren häufig unterschiedliche Varianten derselben Prozesse. Während der Modellierung dieser Prozessvarianten treten erhebliche Probleme auf. Im Rahmen dieser Arbeit wird betrachtet, wie sich diese „Varianten-Problematik“ in realen Prozessen widerspiegelt. Anschließend wird untersucht und kritisch betrachtet, wie gut sich solche Varianten durch die aktuellen Ansätze C-EPC, C-YAWL, PESOA und Provop abbilden lassen. Dazu werden die Eigenschaften der einzelnen Ansätze erklärt und anhand ausgewählter Beispiele miteinander verglichen. Hierbei wird aufgezeigt, welche Probleme die einzelnen Ansätze lösen können und wo die Grenzen der Ansätze liegen.

### **1.3 Aufbau der Arbeit**

In Kapitel 2 werden die wichtigsten Grundlagen und Begriffe wie Geschäftsprozess, Modellierungssprachen und Prozessvarianten vorgestellt und eingeführt. In Kapitel 3 werden die Eigenschaften der verschiedenen Ansätze zur Modellierung von Prozessvarianten erklärt. Der Hauptteil dieser Arbeit bildet Kapitel 4. In diesem Kapitel werden

ausgewählte Beispielprozesse aus dem Umfeld der Studierendenverwaltung der Universität Ulm für die verschiedene Variante existieren, mit Hilfe der vier betrachteten Ansätze umgesetzt. Anschließend werden die entstandenen Prozessmodelle und Ergebnisse miteinander verglichen und die resultierenden Vor- und Nachteile analysiert und erläutert. Abschließend werden in Kapitel 5 alle Erkenntnisse zusammengefasst und in einen Gesamtzusammenhang gestellt.



# 2

## Grundlagen

### 2.1 Geschäftsprozess

Nach ISO 9000 ist ein Prozess ein System von in Wechselbeziehung oder Wechselwirkung stehenden Tätigkeiten, das Eingaben mit Hilfe von Mitteln in Ergebnisse umwandelt [ISO]. Einfacher gesagt, ist ein Prozess eine Abfolge von Aktivitäten, die in einem Beziehungszusammenhang stehen und aus einer Eingabe eine Ausgabe erzeugen. Prozesse sagen nichts darüber aus, wie umfangreich sie sind und welche Begrenzungen, Strukturen und Inhalte sie besitzen.

Ein Geschäftsprozess ist eine spezielle Art von Prozess. Er erzeugt Produkte oder Dienstleistungen für Kunden und bestimmt den betriebswirtschaftlichen sowie den finanziellen Erfolg eines Unternehmens.

Es existieren zahlreiche Interpretationen des Begriffs Geschäftsprozess. Hammer/ Champy definieren einen Geschäftsprozess als eine Menge von Aktivitäten, für die ein oder

## 2 Grundlagen

mehrere unterschiedliche Eingaben benötigt werden und die, für die Kunden, ein Ergebnis von Wert erzeugen [HC95].

Nach Schmelzer/Sesselmann bestehen Geschäftsprozesse aus Aktivitäten, verantwortlichen und beteiligten Stellen sowie Organisationseinheiten [SS08]. Durch Geschäftsprozesse wird es Unternehmen ermöglicht, ihre Aktivität darauf auszurichten, Kundenanforderungen zu erfüllen und Geschäftsziele zu erreichen. Ein Geschäftsprozess setzt sich aus einzelnen Komponenten zusammen. Dazu gehören die Anforderungen der Kunden, Eingaben, Wertschöpfung, Ergebnisse, Geschäftsprozessverantwortliche als auch Ziel- und Messgrößen, um die Prozessleistung zu messen [SS08].

Laut Gierhake sind Geschäftsprozesse betriebliche Abläufe, die sich entlang einer Wertschöpfungskette identifizieren lassen [Gie98]. Außerdem sind sie unmittelbar auf den Erfolg eines Unternehmens am Markt ausgerichtet und sind durch eine messbare Eingabe, eine Wertschöpfung und eine messbare Ausgabe gekennzeichnet [Gie98].

Nach Allweyer ist ein Geschäftsprozess eine Folge von sequentiell oder parallel ablaufenden Funktionen, die das Ziel haben, eine betriebliche Aufgabe zu erfüllen [All05]. Die erzielte Leistung ist die Erbringung von Informations- und/oder Materialtransformation. Anhand dieser Interpretationen können die Eigenschaften von Geschäftsprozessen folgendermaßen zusammengefasst werden.

- Ein Geschäftsprozess besteht aus einer Menge von Aktivitäten, die in einem logischen Zusammenhang stehen.
- Ein Geschäftsprozess ist strukturiert. Er besitzt eine logische und eine zeitliche Anordnung.
- Ein Geschäftsprozess ist kundenorientiert. Die Kundenanforderungen werden immer berücksichtigt.
- Ein Geschäftsprozess schöpft Werte und besitzt immer eine messbare Eingabe sowie eine messbare Ausgabe.
- Ein Geschäftsprozess besitzt einen Verantwortlichen.

## 2.2 Geschäftsprozessmanagement

Im Geschäftsleben existieren in einem Unternehmen zahlreiche Geschäftsprozesse. Die erfolgreiche Ausführung dieser Prozesse beeinflussen den Erfolg des Unternehmens direkt. Um die Geschäftsziele zu erreichen, Kundenanforderungen zu erfüllen und um sich gegen die Konkurrenz durchsetzen zu können, müssen die Geschäftsprozesse in einem Unternehmen verwaltet werden. Dies bezeichnet man als Geschäftsprozessmanagement. Das Geschäftsprozessmanagement hat dementsprechend in den letzten Jahren an Bedeutung gewonnen.

Das auch englisch als Business Process Management (BPM) bezeichnete Geschäftsprozessmanagement wird in der Literatur unterschiedlich definiert.

Die European Association of BPM (EABPM) definiert Geschäftsprozessmanagement als einen systematischen Ansatz der das Ziel hat sowohl automatisierte als auch nicht automatisierte Prozesse zu erfassen, zu gestalten, auszuführen, zu dokumentieren, zu messen, zu überwachen und zu steuern [FR10]. Damit sollen die mit der Unternehmensstrategie abgestimmten Ziele erreicht werden.

Nach Sesselmann und Schmelzer ist „Geschäftsprozessmanagement ein integriertes System aus Führung, Organisation und Kontrolle, das eine zielgerichtete Steuerung der Geschäftsprozesse ermöglicht. Es ist auf die Erfüllung der Bedürfnisse der Kunden und anderer Interessengruppen ausgerichtet und trägt wesentlich dazu bei, die strategischen und operativen Ziele des Unternehmens zu erreichen“ [SS08].

Zusammenfassend stellt das Geschäftsprozessmanagement sicher, dass die Geschäftsprozesse eines Unternehmens sowohl die strategischen Ziele als auch die Kundenziele erfüllen [SS08].

## 2.3 Geschäftsprozessmodellierungssprachen

Modellierungssprachen sind der Ausgangspunkt zur Visualisierung betrieblicher Strukturen und Abläufe in Unternehmen. Sie bilden damit die Basis für die Beschreibung von Prozess- und Datenmodellen, die grundlegend für die Entwicklung von Informationssystemen sind [HN09]. Jede Modellierungssprache verfügt über eine festgelegte Syntax,

## 2 Grundlagen

die über eine Grammatik oder ein Metamodell beschrieben werden kann [HM08]. Es existieren zahlreiche Modellierungssprachen. Jedoch wird in diesem Abschnitt nur auf die Eigenschaften der drei Sprachen EPC, YAWL und BPMN eingegangen, da sie für diese Arbeit relevant sind.

### 2.3.1 EPC

EPC (Event-based Process Chain, Ereignisgesteuerte Prozessketten) ist eine grafische Modellierungssprache zur Darstellung von Geschäftsprozessen, die 1992 in einem Forschungsprojekt mit SAP am Institut für Wirtschaftsinformatik der Universität Saarland entwickelt wurde [tHvdAD05]. Sie ist zentraler Bestandteil der SAP-Referenzmodelle sowie der ARIS-Konzepte und bildet somit Grundlage für eine ganze Reihe modellgetriebener Ansätze für ein durchgängiges und werkzeuggestütztes Geschäftsprozessmanagement [NR02].

Die wesentlichen Grundelemente der EPC sind Ereignisse, Funktionen und Konnektoren (Verknüpfungsoperatoren), wie sie in Abbildung 2.1 dargestellt werden.

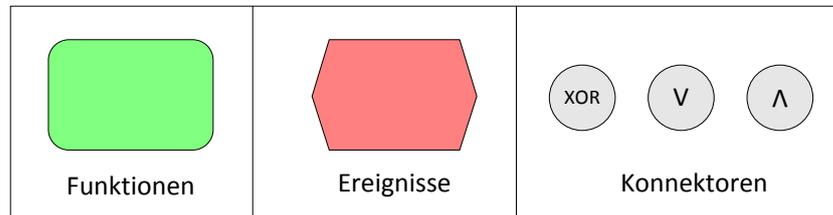


Abbildung 2.1: EPC Basiselementen [HKS93]

Funktionen beschreiben die Durchführung von Transformationsprozessen eine Eingabe hin zu einer Ausgabe zur Erreichung der Unternehmensziele. Sie werden als Rechteck mit abgerundeten Ecken dargestellt. Die Bezeichnung sollte immer das Objekt der Bearbeitung und ein Verb zur Kennzeichnung der Tätigkeit enthalten. Zwei Beispiele dafür sind „Kundenauftrag prüfen“ und „Fertigungsauftrag freigeben“ [HKS93].

Ereignisse lösen Funktionen aus und sind wieder um das Ergebnis von Funktionen. Sie werden durch ein Sechseck dargestellt. Die logische Verbindung zwischen Ereignissen

### 2.3 Geschäftsprozessmodellierungssprachen

und Funktionen werden dabei durch Pfeile dargestellt. Ereignisse repräsentieren die Ergebnisse der Zustandsänderung von Informationsobjekten, auf die mit Funktionen reagiert werden muss. Beispiel für Ereignisse sind „Kundenauftrag ist eingetroffen“ oder „Lieferantenangebot ist geprüft“ [HKS93].

Da innerhalb eines Prozesses auch Entscheidungen auf Basis von Bedingungen und Regeln getroffen werden müssen, existieren in der EPC Konnektoren. Diese können XOR (Entweder-oder), OR (oder) und AND (und) sein [SM97]. Dabei entspricht der AND-Operator einer Parallelisierung des Ablaufs, während der XOR-Operator zur Modellierung von Alternativen im Prozessablauf verwendet wird. Der OR-Operator lässt sowohl einen parallelen Ablauf des Prozesses als auch eine Entscheidung für einen der Pfade zu [KNS92].

Demnach kann eine EPC als gerichteter, zusammenhängender Graph mit einer Multiplizität von Eins aufgefasst werden, für den u. a. folgende Regeln gelten [KKS04]:

- Jede EPC beginnt und endet mit mindestens einem Ereignis.
- Funktionen besitzen genau eine eingehende und eine ausgehende Kante.
- Ereignisse haben genau eine eingehende und/oder genau eine ausgehende Kante.
- Funktionen und Ereignisse wechseln sich gegenseitig ab.
- Ereignisse können keine Entscheidungen treffen, das übernehmen die Funktionen.
- Konnektoren können mehrere eingehende Kanten und eine ausgehende Kante haben, oder sie können mehrere ausgehende Kanten und eine eingehende Kante haben.
- Konnektoren dürfen mit Konnektoren verbunden werden.
- Eine Verzweigung wird ggf. mit demselben Verknüpfungsoperator geschlossen, mit dem sie geöffnet wurde.
- Einem Ereignis darf kein öffnendes OR und XOR folgen.

Abbildung 2.2 zeigt einen Beispielprozess zur Anwendung der EPC. Dieser Prozess beschreibt den Ablauf einer Schadensdatenbearbeitung einer Versicherung. Zunächst meldet sich ein Kunde bei der Versicherung, um einen Schaden zu melden und ggf.

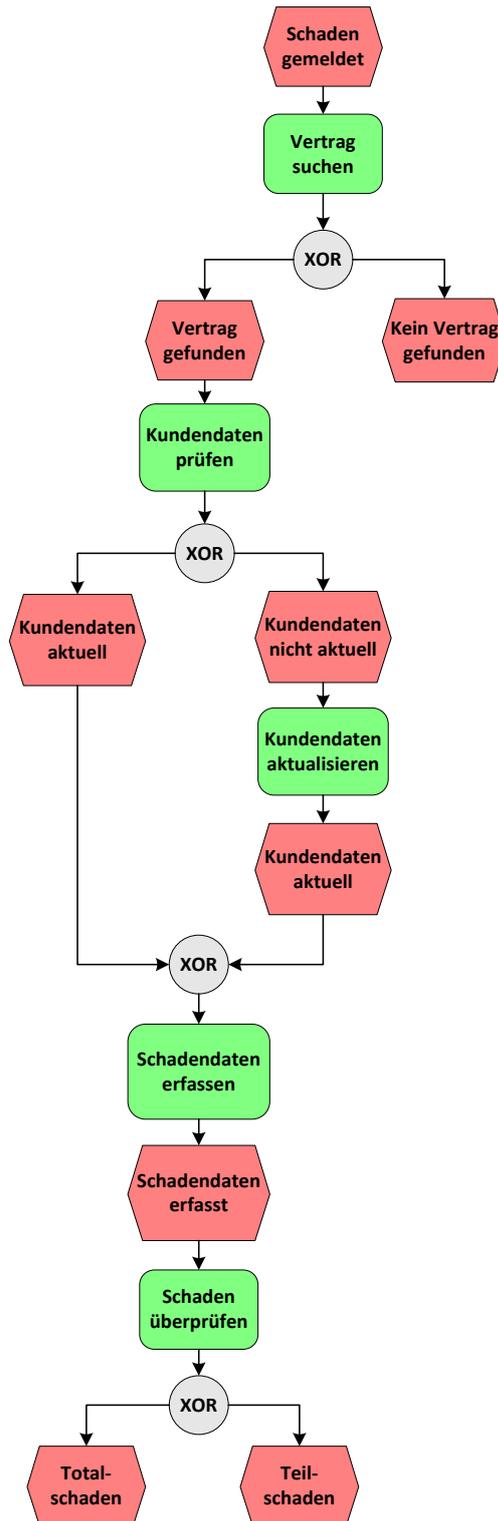


Abbildung 2.2: EPC Beispielprozess - Schadensdatenbearbeitung [Wie]

Leistungen der Versicherung in Anspruch zu nehmen. Die Versicherung überprüft, ob dieser Kunde überhaupt versichert bei ihr ist. Nur wenn diese Voraussetzung erfüllt ist, wird fortgefahren. Vor der Erfassung der Schadensdaten werden die Kundendaten aktualisiert, falls diese nicht mehr aktuell sind. Dann werden die Schadensdaten erfasst und anhand dieser Daten wird der Schaden überprüft. Schließlich wird eine Entscheidung getroffen, ob es sich um einen Totalschaden oder Teilschaden handelt.

Dieser Prozess besteht aus neun Ereignissen, fünf Funktionen und vier XOR-Knoten. Der Prozess fängt mit einem Ereignis an und kann mit einem von drei möglichen Ereignissen beendet werden. Wie in der Abbildung veranschaulicht ist, wechseln sich Funktionen und Ereignisse ab. Die Funktionen besitzen genau eine eingehende und eine ausgehende Kante. Ereignisse besitzen genau eine eingehende und/oder genau ausgehende Kante. Somit werden die oben genannten Regeln der EPC erfüllt.

### 2.3.2 YAWL

Yet Another Workflow Language (YAWL) wurde an der technische Universität Eindhoven entwickelt [tHvdA05]. Die Darstellung von YAWL basiert im Wesentlichen auf den mathematischen Grundlagen von Petri-Netze, die in den 1960er Jahren von Adam Petri entwickelt wurden [Pet62]. Petri-Netze sind ein universelles Werkzeug zur Analyse und Modellierung von dynamischen Abläufen. Sie sind besonders geeignet für nebenläufige Vorgänge, Kommunikationsprotokolle sowie Synchronisationsalgorithmen [Bau90].

Ein Workflow in YAWL wird mit „Workflow Netzen“ beschrieben. Diese basieren direkt auf Petri Netzen. Der wesentliche Unterschied ist, dass Workflow Netze immer einen Startpunkt und einen Endpunkt haben [tHvdA05]. YAWL erweitert Workflow-Netze um Mehrfach-Instanzen, zusammengefasste Aufgaben, OR-Joins, die Entfernung von Tokens und direkt verbundene Transitionen [tHvdA05]. Abbildung 2.3 zeigt die wichtigsten Modellierungselementen von YAWL [tHvdA05].

Die mittels YAWL beschriebene Prozessmodelle bestehen aus einer Reihe von Conditions und Tasks. Eine Condition ist eine Bedingung und speichert das Ergebnis einer Aufgabe. Sie wird normalerweise zwischen zwei Aufgaben verwendet. Zwei Aufgaben können jedoch auch direkt verbunden werden. In diesem Fall ist es so, als wäre zwi-

## 2 Grundlagen

schen diesen beiden Aufgaben eine „unsichtbare“ Condition eingebunden [tHvdA05]. Jedes Prozessmodell besitzt nur eine Input- und eine Output Condition, die jeweils den Start- bzw. Endpunkt definiert [tHvdA05]. Tasks in YAWL sind entweder „atomic“ oder „composite“. Während ein Atomic Task die einfachste Art einer Aufgabe definiert, stellt ein Composite Task eine Zusammensetzung von mehreren Teilaufgaben dar, welche in einem separaten Prozessmodell beschrieben ist. Beide Taskarten können mehrere Instanzen haben. Somit kann ein Geschäftsvorgang zur gleichen Zeit mehr als einmal durchlaufen werden. YAWL bietet eigenständige Modellierungselemente für AND-, XOR- und OR- Verzweigungen bzw. deren Zusammenführungen an. Das letzte Modellierungselement remove Tokens ermöglicht es, alle Marken aus dem umrandeten Bereich zu löschen. Dies ist unabhängig von der Anzahl der Tokens. Die Löschung findet statt, sobald der zugehörige Task ausgeführt wurde [tHvdA05].

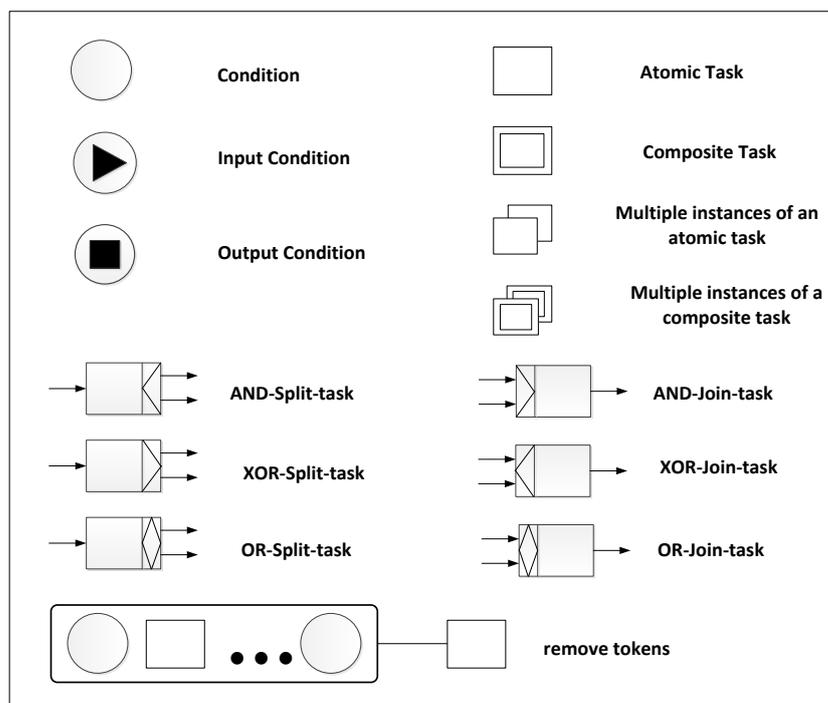


Abbildung 2.3: Modellierungselemente von YAWL [tHvdA05]

Abbildung 2.4 veranschaulicht die Eigenschaften von YAWL anhand eines Kreditantragsprozess [YAWb]. Das Kreditantragsverfahren beginnt, sobald ein Antragsteller einen

### 2.3 Geschäftsprozessmodellierungssprachen

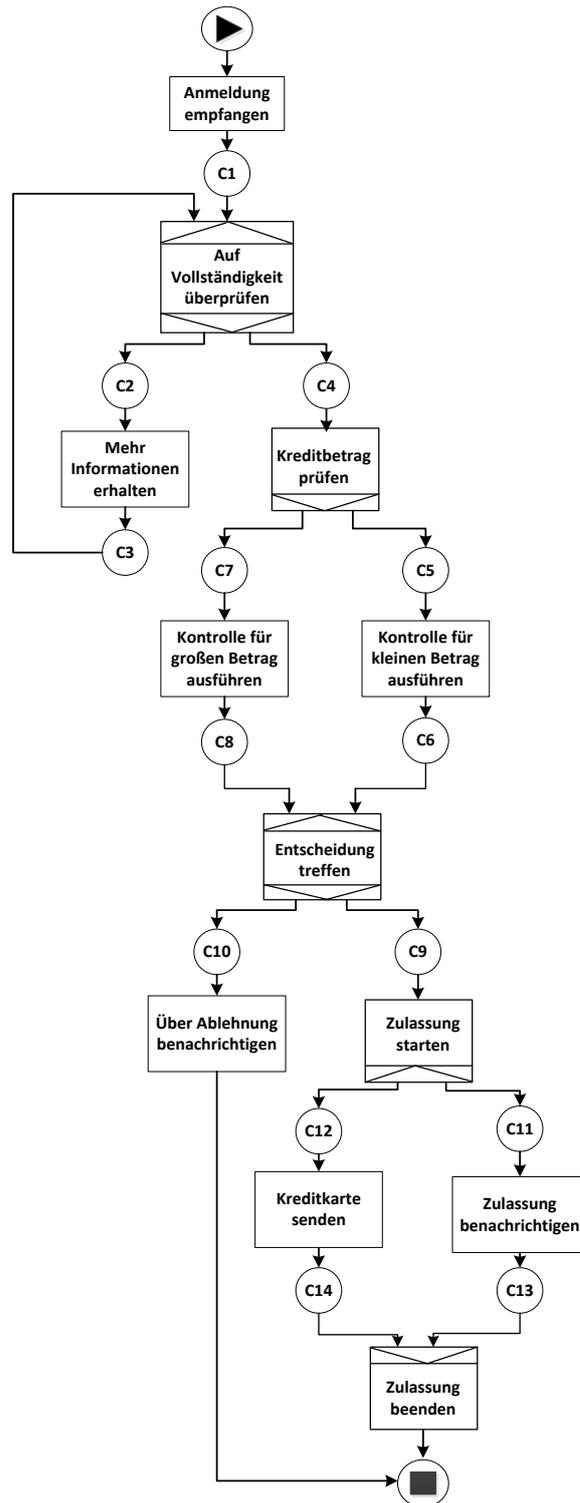


Abbildung 2.4: Kreditantragsprozess mit YAWL [YAWb]

## 2 Grundlagen

Antrag mit einer vorgeschlagenen Höhe einreicht. Bei Eingang eines Antrags prüft ein Kreditsachbearbeiter, ob diese vollständig ist. Wenn nicht, fordert der Sachbearbeiter zusätzliche Informationen an. Sobald er diese Informationen erhalten hat, kann er fortfahren. Für eine komplette Anmeldung führt der Sachbearbeiter weitere Prüfungen durch, um die Kredithöhe anhand des Einkommens und der Kredit-Geschichte des Antragstellers abzuwägen. Der gültige Antrag wird dann zu einem Manager weitergeleitet, der entscheidet, ob dieser akzeptiert oder abgelehnt wird. Im Falle der Annahme wird der Antragssteller darüber informiert und erhält anschließend eine Kreditkarte. Wird der Antrag abgelehnt, so wird der Antragssteller ebenfalls über die Entscheidung informiert und der Prozess endet.

### 2.3.3 BPMN

Die Business Process Modelling Notation (BPMN) wurde in der ersten Fassung 2004 von der Business Process Management Initiative (BPMI) veröffentlicht [FR10]. Als vorrangige Ziel der BPMN wird genannt, dass Prozessmodelle von den unterschiedlichsten Stakeholdern im Prozessmanagement problemlos verstanden und akzeptiert werden sollen [Whi04]. Von Anfang an war die Zielsetzung, eine standardisierte, grafische Prozessnotation bereitzustellen, die auch für die Prozessautomatisierung verwendet werden kann. 2005 übernahm die Object Management Group (OMG) die BPMI und somit auch die weitere Entwicklung der BPMN [FR10]. Mit der Übernahme durch die OMG begann auch der weltweite Siegeszug der BPMN, da schon allein die Standardisierung für viele Unternehmen einen großen Anreiz für den Umstieg darstellte. 2011 wurde die Version 2.0 durch die OMG veröffentlicht [OMGa].

Abbildung 2.5 stellt die verwendeten Symbole in BPMN dar. Diese werden den fünf Kategorien Flussobjekte, verbindende Objekte, Artefakte, Teilnehmer und Daten zugeordnet. Zu den Flussobjekte gehören Aktivitäten, Ereignisse und Gateways, mit denen das Verhalten von Geschäftsprozessen beschrieben wird. Aktivitäten geben an, was erledigt werden muss, damit der Prozess die gewünschte Leistung erbringen kann [FR10]. Sie werden als Rechtecke mit abgerundeten Kanten dargestellt. Eine weitere Art von Aktivitäten sind Sub-Prozesse. Ein Sub-Prozess wird durch ein kleines Plus-Zeichen in

## 2.3 Geschäftsprozessmodellierungssprachen

der unteren Mitte des Rechtecks unterschieden [OMGa]. Subprozesse sind Aktivitäten, die ganze Prozesse kapseln. Die dahinterliegenden Prozesse können an anderer Stelle definiert werden. Auf diese Weise können Abstraktionen geschaffen sowie die Übersichtlichkeit von Modellen verbessert werden [FR10].

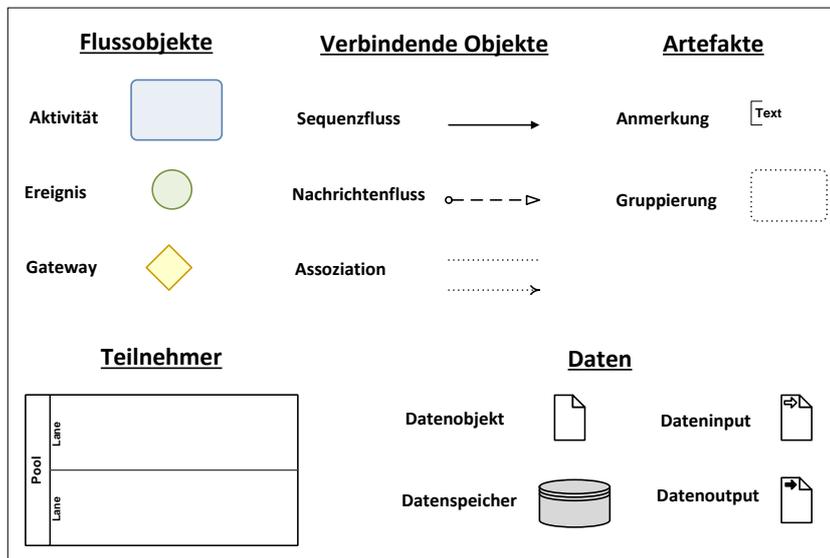


Abbildung 2.5: BPMN Basiselementen [FR10]

Ereignisse werden durch einen Kreis dargestellt und zeigen, ob vor, während oder am Ende des Prozesses etwas Beachtenswertes passiert. Sie beeinflussen den Ablauf des Prozesses und haben üblicherweise eine Ursache sowie ein Ergebnis [Whi04, FR10]. Prinzipiell gibt es drei Arten von Ereignissen: Startereignis, Zwischenereignis und Endergebnis. Von diesen Ereignistypen existieren noch diverse Spezialisierungen, welche die Aussagekraft verstärken oder Voraussetzung für die Verwendung an einer bestimmten Stelle in einem Prozess sind [FR10]. Abbildung 2.6 zeigt diese wichtigsten Ereignistypen. Ein Gateway wird durch ein Diamantensymbol dargestellt und wird verwendet, um die Divergenz und Konvergenz des Sequenzflusses zu steuern. Somit können Entscheidungen sowie die Gabelungen, Zusammenführungen und Synchronisation von Pfaden beschrieben werden. Das Symbol innerhalb des Gateways gibt die Art des Gateways an [Whi04]. Abbildung 2.7 stellt die meist verwendeten Gateways dar.

## 2 Grundlagen

	Startereignisse	Zwischenereignisse	Endereignisse
Nachrichten			
Zeit			
Fehler			
Terminierung			
Abbruch			

Abbildung 2.6: BPMN Ereignisse [OMGa]

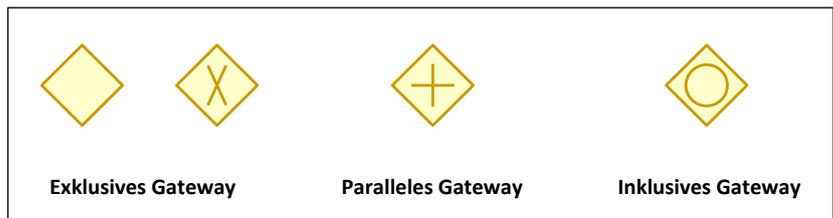


Abbildung 2.7: BPMN Gateways [OMGa]

Die Flussobjekte werden in einem Diagramm miteinander verbunden, um das Grundgerüst eines Geschäftsprozesses zu beschreiben. Hinzu werden die verbindenden Objekte verwendet. Hierzu gehören der Sequenzfluss, Nachrichtenfluss und Assoziation [Whi04]. Ein Sequenzfluss wird durch eine durchgezogene Linie mit einer geschlossenen Pfeilspitze dargestellt. Er wird verwendet, um zu zeigen, in welcher Reihenfolge die Aktivitäten in einem Prozess ausgeführt werden sollen. Ein Nachrichtenfluss wird durch eine gestrichelte Linie mit einer offenen Pfeilspitze dargestellt und wird verwendet, um einen Nachrichtenaustausch zwischen zwei separaten Teilnehmern an zu zeigen [Whi04]. Eine Assoziation wird durch eine gestrichelte Linie dargestellt. Sie wird verwendet, um sowohl die Eingabe- und Ausgabedaten von Aktivitäten zu beschreiben, als auch um Daten, Texte und andere Artefakte mit Flussobjekten zu verknüpfen [Whi04]. Datenobjekte

### 2.3 Geschäftsprozessmodellierungssprachen

werden im Rahmen der Prozessausführung benötigt, um Informationen erzeugen, verarbeiten und ablegen zu können. Weiterhin gibt es Artefakte (wie z.B. Gruppierungen und Anmerkungen), die zusätzliche Informationen zum Prozess liefern, aber keinen direkten Einfluss auf die Ausführungsreihenfolge der Flussobjekte haben [FR10].

Um in einem Prozessmodell erkennbar zu machen, wer für die Ausführung von Aktivitäten verantwortlich ist, bietet BPMN die Modellierungselemente Pool und Lane. Ein Pool stellt einen Prozessbeteiligten dar. Weiterhin gibt es die Möglichkeit, Pools zu strukturieren und weiter zu untergliedern, wofür Lanes verwendet werden. So kann zum Beispiel ein Unternehmen als Pool und seine Abteilungen als Lanes dargestellt werden [Whi04]. In Abbildung 2.8 zeigt einen in BPMN modellierten Beispielprozess. Das Prozessmodell stellt den Versandprozess von einem Hardware-Händler dar. In diesem Beispiel wird der Hardware-Händler durch einen Pool und die drei Rollen für Lagerarbeitskraft, Sachbearbeiter und Logistikmanager durch entsprechende Lanes dargestellt. Das einfache Starterereignis „Waren zu verschicken“ zeigt an, dass Waren versendet werden müssen. Gleich nach der Instanziierung des Prozesses, gibt es zwei Schritte, die parallel ausgeführt werden. Dazu wird ein Parallel-Gateway verwendet. Während der Sachbearbeiter sich entscheiden muss, ob es sich um eine normale Post oder spezielle Sendung handelt, kann der Lagerarbeiter bereits mit dem Verpacken der Waren beginnen. Für die Auswahl der Versandart wird das exklusive Gateway verwendet. Wenn eine spezielle Sendungsart benötigt wird, dann holt sich der Sachbearbeiter Angebote bei verschiedenen Transportunternehmen ein. Sobald er ein Transportunternehmen ausgewählt hat, gibt er diesem den Auftrag und füllt die Frachtpapiere aus. Falls jedoch die Ware mit der Post zu versenden ist, so prüft der Sachbearbeiter, ob eine zusätzliche Versicherung erforderlich ist und wenn ja, dann muss diese vom Logistikmanager abgeschlossen werden. Auf jeden Fall muss der Sachbearbeiter ein Frachtpapier bzw. ein Postetikett ausfüllen. Entsprechend wurde hier das inklusive Gateway verwendet. Somit wird ein Zweig immer ausgeführt, und falls es erforderlich ist, kann der zweite Zweig parallel dazu ebenfalls auch ausgeführt werden. Vor der Ausführung der letzten Aufgabe muss das parallele Gateway synchronisiert werden. Dann kann die Lagerarbeitskraft die notwendigen Papiere dieses zum Paket hinzufügen und zum Abholungsbereich senden. Mit dem Endereignis „Ware ist abholbereit“ wird der Prozess beendet.

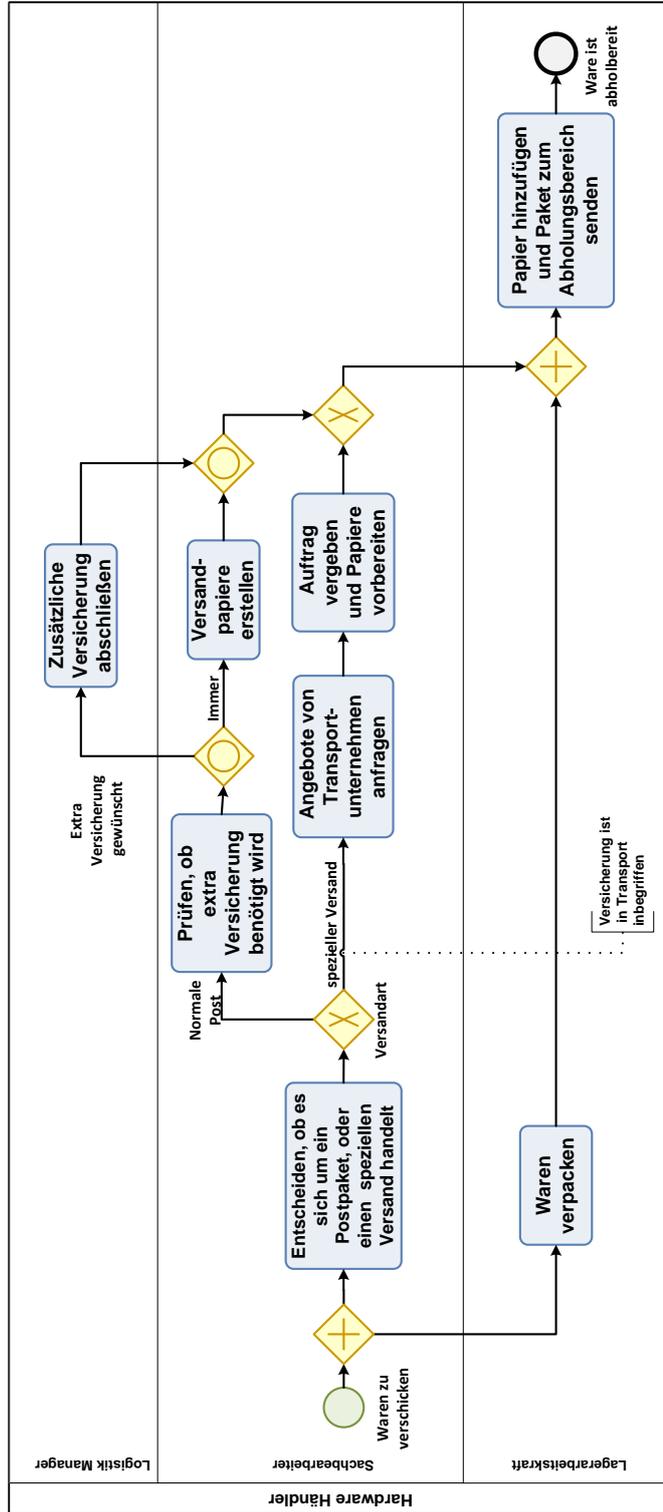


Abbildung 2.8: BPMN Beispielprozess [OMGb]

## 2.4 Prozessvarianten

In der Praxis hat sich gezeigt, dass Geschäftsprozesse je nach ihrem Anwendungskontext häufig in unterschiedlichen Varianten auftreten. Diese Prozessvarianten verfolgen alle das gleiche oder ein ähnliches Geschäftsziel, unterscheiden sich aber durch verschiedene Rahmenbedingungen, wie Vorschriften in verschiedenen Ländern oder Regionen, Kundenkategorien, oder strategischen Ausrichtungen [Hal09, HBR08a].

Zur Darstellung von Prozessvarianten existieren zwei grundsätzliche Ansätze: den Single-Modell Ansatz und Multi-Modell Ansatz [HBR10].

Beim Multi-Modell Ansatz werden alle Varianten in separaten Prozessmodellen realisiert. Somit existiert für jede Variante ein Prozessmodell. Jedoch gibt es keinen Hinweis darauf, ob die einzelnen Modellteile variantenspezifisch oder für alle Modelle gemeinsam sind. Wenn die meisten Modellteile der verschiedenen Prozessvarianten identisch oder ähnlich sind, verursacht dies hochreduzante Modelle [HBR10].

In der Praxis ist die Anzahl der Prozessvarianten häufig sehr groß. Diese große Zahl der Varianten erschwert die Modellierung und erhöht den Wartungsaufwand erheblich. Insbesondere verursachen die Änderungen an gemeinsamen Teilen hohe Aufwände, weil Basisprozessänderungen für jede Variante separat durchgeführt werden müssen. Das ist sowohl zeitaufwendig als auch fehleranfällig [HBR10].

Der Multi-Modell Ansatz macht es für Prozessdesigner zusätzlich schwierig, die Modelle der Prozessvarianten zu analysieren, zu vergleichen, zu vereinheitlichen und die vielfältigen Varianten innerhalb eines gemeinsamen IT-Systems zu implementieren. Folglich stellt es für sinnvolles Variantenmanagement keine adäquate Lösung dar, alle Prozessvarianten in separaten Modellen darzustellen [HBR10].

Im Gegensatz dazu fasst der Single-Modell Ansatz mehrere Varianten eines Prozesses mittels bedingter Verzweigung in einem einzigen Modell zusammen. Die unterschiedlichen Ausführungspfade im Modell stellen dabei die unterschiedlichen eine besondere Variante dar. In diesem Fall ist es aber schwer in einem großen Prozessmodell die einzelnen Varianten zu begreifen und es ist teuer das Modell auszuführen. Zusätzlich kann der Benutzer normale und variantenspezifische Verzweigungen nicht mehr unterscheiden. Deswegen sollten in diesem Fall die Verzweigungen entsprechend markiert werden,

## *2 Grundlagen*

je nachdem, ob es sich um eine normale oder eine variantenspezifische Verzweigung handelt. Die Varianten sind zudem weder transparent noch explizit definiert. Als Folge sind die Varianten den IT-Systemen unbekannt und alle Verzweigung werden im Single-Ansatz folglich als „normalen Verzweigung“ behandelt [HBR10].

Zusammenfassend ist weder der Multi-Modell Ansatz noch der Single-Modell Ansatz geeignet, um Prozessvarianten einwandfrei darstellen zu können.

# 3

## **Ansätze zur Abbildung von Prozessvarianten**

Um die vorhandene Variabilität in einem Geschäftsprozessmodell richtig darstellen zu können, sollte definiert werden, welche Teile des Modells nach einem bestimmten Kontext variieren können, welche Alternativen für diese Teile existieren, und nach welchen Bedingungen diese Alternativen gewählt werden können [TZW<sup>+</sup>12]. Heutige Geschäftsprozessmodellierungssprachen sind nicht ausreichend, um diese Fragen zu behandeln und um Prozessvarianten umfassend zu verwalten. Deswegen wurden in den letzten Jahren verschiedene Ansätze zur Modellierung Varianten behaftete Prozesse vorgeschlagen. Diese können entweder den verhaltensbasierten oder strukturellen Ansätze zugeordnet werden. In diesem Kapitel werden diese Ansätze und die Vorschläge, die nach diesen Ansätzen entwickelt wurden, näher definiert und erläutert.

### 3.1 Verhaltensbasierter Ansatz

Der verhaltensbasierte Ansatz repräsentiert eine Obermenge der Prozessvarianten in einem sogenannten Referenzprozessmodell. Dieses Modell erfasst sowohl die Gemeinsamkeiten als auch die Unterschiede der möglichen Varianten. In dieser Arbeit werden drei grundsätzliche Ansätze betrachtet, wie ein Referenzprozessmodell gestaltet werden können.

Als erste Möglichkeit werden konfigurierbare Knoten zur Verfügung gestellt. Mit Hilfe dieser Knoten können Variationspunkte des Referenzprozessmodells und mögliche Konfigurationsalternativen dargestellt werden. Darüber hinaus können diesen konfigurierbaren Knoten Konfigurationseinschränkungen hinzugefügt werden, um mögliche, unerwünschte Kombinationen von Prozessvariationen zu vermeiden [RW12]. Als Beispiel für diesen Ansatz wird in Abschnitt 3.1.1 die Modellierungssprache C-EPC erklärt. Die zweite Möglichkeit ist das Blockieren oder Verstecken der Kontrollknoten und der ausführbaren Pfade, die nicht zur aktuell gewählten Variante gehören. Bei der Blockierung wird die Ausführung einer Aktivität gesperrt und somit kann der Knoten nicht ausgeführt werden. Nachfolgeknoten von blockierten Knoten können nur dann ausgeführt werden, wenn ein anderer oder direkter Pfad vom Startknoten aus zu diesem Nachfolgeknoten existiert. Wiederum ermöglicht das Verstecken eine Abstraktion des Referenzprozessmodells, d.h. bei der Ausführung wird eine versteckte Aktivität nicht berücksichtigt und übersprungen [RW12]. Allerdings bleibt der entsprechende Pfad insgesamt ausführbar. Die Modellierungssprache C-YAWL basiert auf diesem Ansatz und wird im Abschnitt 3.1.2 erklärt.

Die letzte Möglichkeit ist der annotationsbasierte Ansatz. Diese Vorgehensweise bietet keine Sprache zur Darstellung und Konfiguration der Referenzprozessmodelle, sondern verbessert die Anpassung der Variationen an den Prozess. Die Orte, in einem Prozessmodell, an denen Variabilitäten auftreten können, werden als Variationspunkte markiert und mögliche Variationen im Prozessmodell zusätzlich gezeigt [LRDtH08]. PESOA ist eine Modellierungssprache, die auf diesem Ansatz basiert und wird in Abschnitt 3.1.3 erklärt.

## 3.1.1 C-EPC

C-EPC [RvdA07] ist eine Erweiterung der Modellierungssprache EPC, die in Abschnitt 2.3.1 erläutert wurde. Sie ermöglicht mit ihren neuen Konstrukten mehrere Prozessvarianten innerhalb eines einzigen Prozessmodells darzustellen. Diese Darstellung wird durch die Konfiguration der Funktionen und Konnektoren realisiert. Im Modell werden die konfigurierbaren Funktionen und Konnektoren dabei mit einem dickeren Rahmen hervorgehoben [RvdA07].

Konfigurierbare Funktionen werden als ON, OFF oder OPT konfiguriert werden. Ist eine Funktion als ON konfiguriert, so wird diese in den entsprechenden Variante ausgeführt. Ist eine Funktion als OFF konfiguriert, wird diese in der entsprechende Variante versteckt und diese Aufgabe bei der Ausführung übersprungen. Die Konfiguration OPT ist eine Kombination der beiden vorherigen Optionen. Hier wird während der Laufzeit entschieden, ob die Aufgabe ausgeführt oder übersprungen werden soll [LR09]. Abbildung 3.1 veranschaulicht, wie aus diesen Optionen die entsprechenden Varianten abgeleitet werden können.

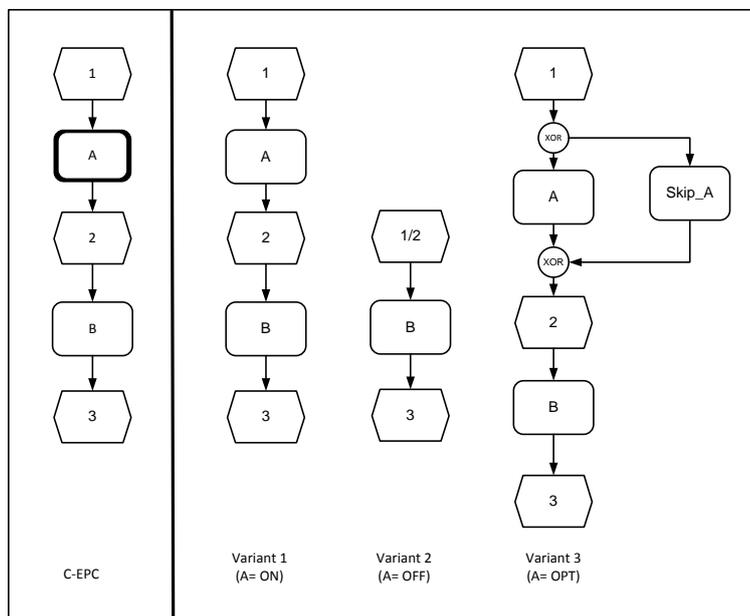


Abbildung 3.1: Konfigurationen einer Funktion [RvdA07]

### 3 Ansätze zur Abbildung von Prozessvarianten

Im Gegensatz der konfigurierbare Funktionen lassen sich konfigurierbare Konnektoren auf gleichen oder weniger ausdruckskräftige Konnektortypen abbilden. Folglich kann ein konfigurierbares OR auf ein reguläres OR, XOR, AND oder einen seiner abgehenden / ankommenden Sequenzflüsse SEQn abgebildet werden. Wenn der Konnektor vom Typ Split ist, gibt SEQn eine seiner ausgehenden Kanten an. Falls der Konnektor vom Typ Join ist, gibt SEQn eine der eingehenden Kanten an. Ein konfigurierbarer XOR kann zu einem regulären XOR oder einem seiner ausgehenden/ankommenden Sequenzflüsse gesetzt werden. Ein konfigurierbarer AND-Konnektor kann nur zu einem regulärem AND-Konnektor mit einer reduzierten Auswahl an Pfaden abgebildet werden. Tabelle 3.1 zeigt die genauen Beschränkungen für die Konfiguration der Konnektoren [RvdA07].

Konfigurationsknoten \ Abbildung	OR	XOR	AND	SEQ
OR	X	X	X	X
XOR		X		X
AND			X	

Tabelle 3.1: Beschränkungen für die Konfiguration der Konnektoren

Um die Konfigurationen im Prozess besser zu regeln, werden sogenannte Requirements (Anforderungen) und Guidelines (Richtlinien) verwendet. Sie werden durch Marker dargestellt, die an beteiligten Knoten befestigt werden. Requirements formalisieren die Beschränkungen zwischen den Alternativen von konfigurierbaren Knoten. Guidelines bieten Vorschläge an, mit denen Konfigurationen besser unterstützt werden sollen [LRDtH08, LR09].

Abbildung 3.2 zeigt einen mit C-EPC modellierten Prozess vor und nach der Konfiguration. Der Prozess beschreibt eine Reisebuchung. Er besitzt zwei konfigurierbare OR-Knoten und vier Requirements. Bei Requirement 1 wird Pfad SEQ1a ausgeführt und nur ein Flug gebucht. Wenn nur ein Hotel gebucht werden soll, wird SEQ1b nach Requirement 2 ausgeführt. Das Requirement 3 wandelt den OR-Knoten durch eine Konfiguration in einen AND-Knoten um. Somit werden Flug und Hotel zusammen gebucht. Im Beispiel wird nach der Konfiguration Requirement 3 dargestellt. Um einen fehlerfreien Prozess realisieren zu können, muss darauf geachtet werden, dass die Art des ersten

und zweiten OR-Knoten nach Konfigurationen immer gleich bleibt. Dies wird hier durch Requirement 4 sicher gestellt.

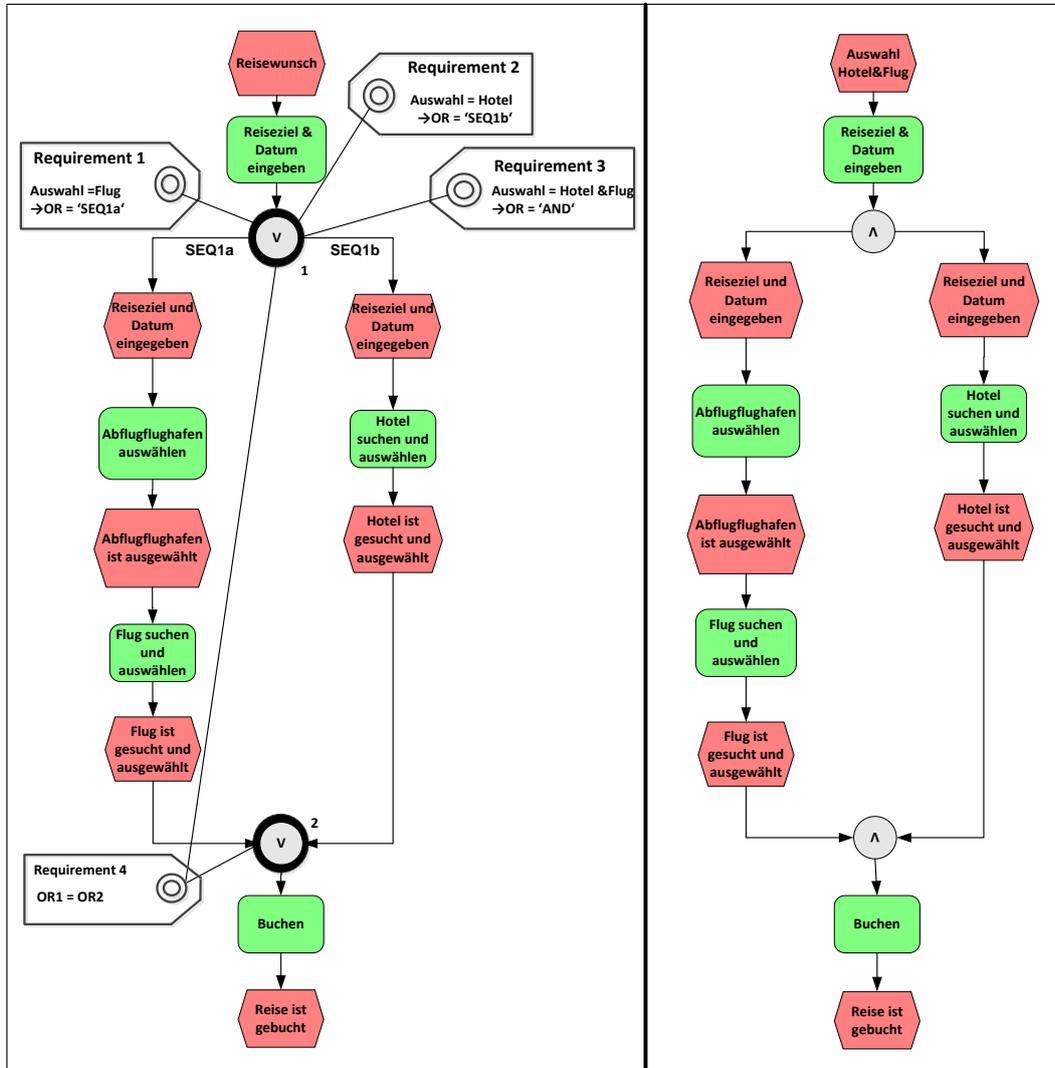


Abbildung 3.2: C-EPC Prozess vor und nach der Konfiguration

### 3.1.2 C-YAWL

C-YAWL (Configurable YAWL) [tHvdAAR09] ist eine Erweiterung der Prozessmodellierungssprache YAWL, die bereits in Abschnitt 2.3.2 erklärt wurde. C-YAWL erweitert YAWL um sogenannte Ports, die als Variationspunkte benutzt werden [LRDtH08, LRGDvdA08]. Mit Hilfe des Versteckungs- und Blockierungskonzeptes werden damit die Prozessvarianten realisiert.

In C-YAWL werden der Eingangs- und der Ausgangsport zur Konfiguration verwendet. Ein Eingangs-Port kann als aktiviert, blockiert oder versteckt konfiguriert werden. Wenn dieser Port aktiviert ist, wird die Aufgabe ausgeführt. Wenn er blockiert ist, wird das Auslösen dieser Aufgabe verhindert. Falls er versteckt ist, wird die Ausführung der Aufgabe übersprungen, aber der nachfolgende Pfad bleibt immer noch ausführbar. Ein Ausgangsport kann mit genau dem selben Prinzip aktiviert oder blockiert werden [LRDtH08].

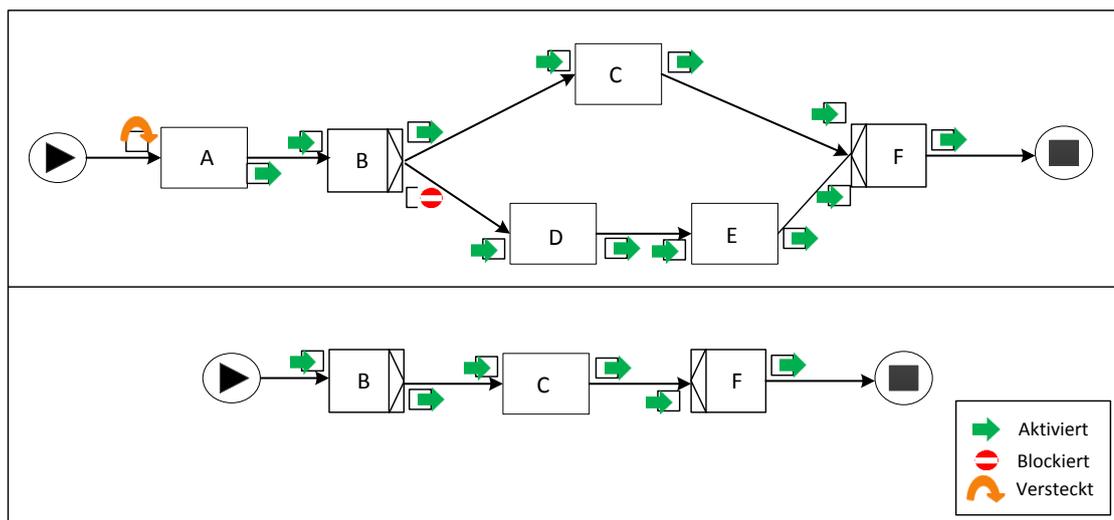


Abbildung 3.3: Konfiguration durch Blockieren und Verstecken [YAWa]

Abbildung 3.3 zeigt, wie Konfigurationen durch das Verstecken und Blockieren funktioniert. Wenn keine Konfigurationen für diesen Prozess vorhanden sind, können entweder die Aufgaben (A, B, C, F) oder (A, B, D, E, F) ausgeführt werden. In Abbildung 3.3 sind der Eingangsport von Aufgabe A als versteckt und ein Ausgangsport des XOR-Split B als blockiert markiert. Das heißt, zuerst wird Aufgabe A ohne Ausführung einfach übersprun-

gen. Jedoch beeinflusst dies nicht die Ausführung der anderen Aufgaben. Danach wird einer der Ausgangsports von B blockiert. Somit ist der einkommende Pfad der Aufgabe D nicht ausführbar. Deswegen können Aufgabe D und Aufgabe E nicht mehr bearbeitet werden. Zum Schluss bleiben nur die Aufgaben (B, C, F) nach den Konfigurationen ausführbar. Die entsprechende Variante ist im unteren Teil von Abbildung 3.3.

Je nach Typ des Konnektors und Anzahl der ein- bzw. ausgehenden Pfade ändert sich die Anzahl der Eingangs- bzw. Ausgangsports [tHvdAAR09]. Tabelle 3.2 zeigt die konfigurierbaren Konnektoren, Anzahl der Ports und Konfigurationsoptionen. „n“ steht für die Anzahl der ein- bzw. ausgehenden Pfade für jeden Konnektor.

Konfigurierbare Element	Anzahl der Ports	Konfigurationsoptionen
AND-join	1	aktiviert, blockiert, versteckt
XOR-join	n	aktiviert, blockiert, versteckt
OR-join	1	aktiviert, blockiert, versteckt
AND-split	1	aktiviert, blockiert
XOR-split	n	aktiviert, blockiert
OR-split	$2^n - 1$	aktiviert, blockiert

Tabelle 3.2: Konfigurationsoptionen für einen Konnektor mit n eingehenden/ausgehenden Pfaden [tHvdAAR09]

Abbildung 3.4 zeigt einen Beispielprozess aus dem Gesundheitsbereich [ATP11, WSR09]. Der Prozess wird durch die Aufnahme des Patienten gestartet. Dann wird die Anamnese und die klinische Untersuchung des Patienten durchgeführt. Nach dieser Untersuchung werden verschiedene Tests (Röntgen, MRT und Sonographie) in beliebiger Reihenfolge bearbeitet. Hierbei existieren folgende Varianten: Wenn der Patient an einem Kreuzbandriss leidet, werden die Aktivitäten „initiale Behandlung & Operation Planung“ und „operative Behandlung“ durchgeführt. Wenn der Patient einen Herzschrittmacher trägt, muss die Aktivität „MRT-Untersuchung“ übersprungen werden. Falls der Patient an einem Erguss in seinem Knie leidet, muss eine Punktion gemacht werden. Wenn es aufgrund gesetzlicher Bestimmungen notwendig ist, wird der Patient über die Behandlung informiert. Ansonsten wird diese Aufgabe übersprungen. Die rechte Seite von Abbildung 3.4 zeigt die entsprechende Variante für den Fall.

### 3 Ansätze zur Abbildung von Prozessvarianten

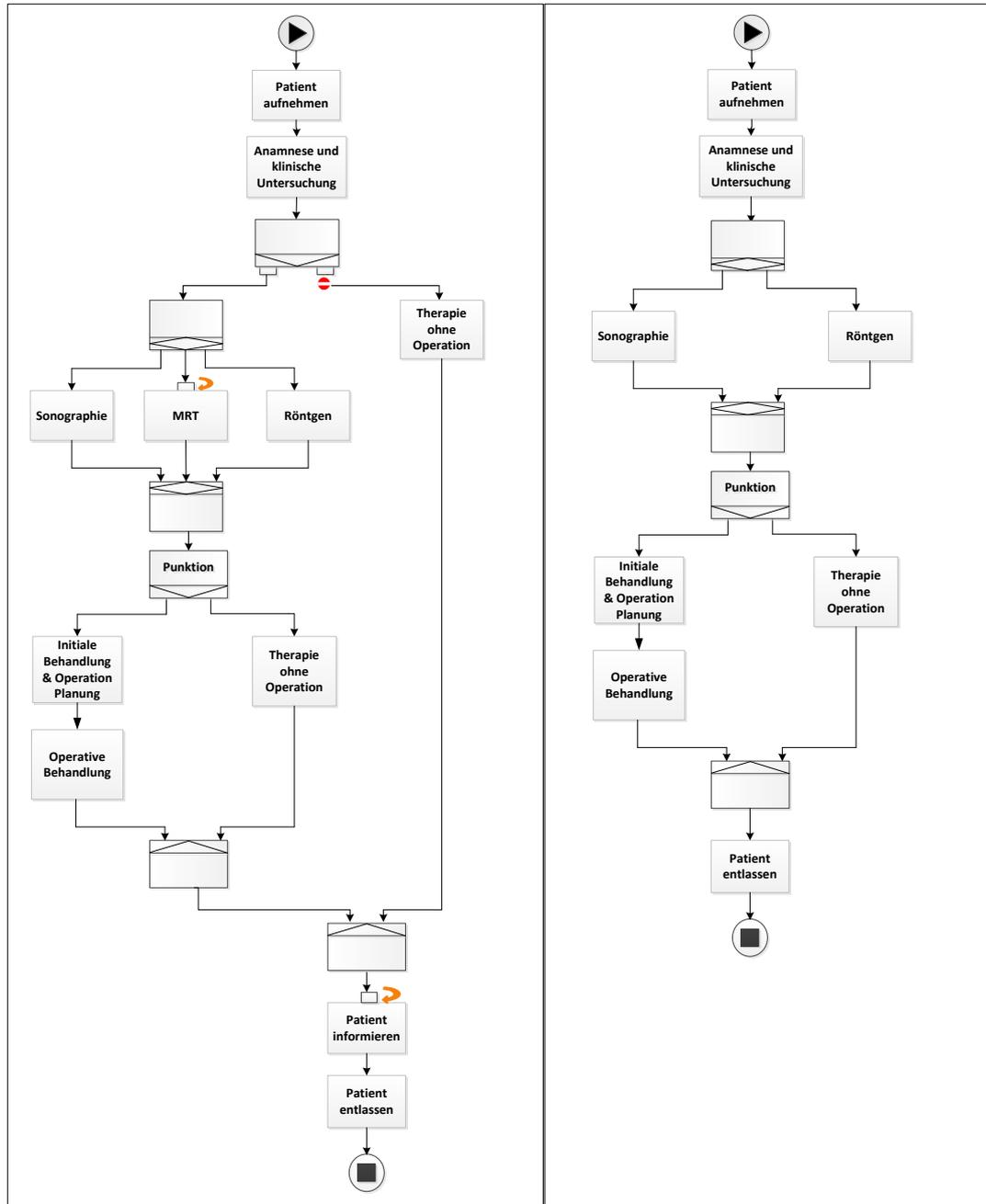


Abbildung 3.4: Beispielprozess mit C-YAWL [ATP11, WSR09]

#### 3.1.3 PESOA

PESOA ist ein Kooperationsprojekt, das von einer Gruppe aus Unternehmen (Daimler-chrysler AG, Delta Software Technology GmbH, ehotel AG und das Fraunhofer IESE), Wissenschaftlern des Hasso-Plattner-Instituts und der Universität Leipzig durchgeführt wird [ATP11].

Das Ziel dieses Projektes ist es nicht, eine Sprache für die Darstellung und Konfiguration von Prozessmodellen zu schaffen, sondern die Anpassung von prozessorientierten Softwaresystemen zu verbessern [LR09].

PESOA besitzt verschiedene Variabilitätsmechanismen wie Parametrisierung, Vererbung, Design Pattern und Datenkapselung verbergen. Diese Mechanismen wurden im Rahmen des Projekts in verschiedene Modellierungssprachen wie UML-Aktivitätsdiagramme, UML State Machines, BPMN und Matlab/Simulink übertragen [PSWW05]. Diese Arbeit beschränkt sich jedoch darauf, wie diese Mechanismen in BPMN umgesetzt werden und zur Modellierung von Prozessvarianten benutzt werden können.

Um ein sogenanntes „variantenreiches Geschäftsprozessdiagramm“ zu realisieren, sind nach [PSWW05] drei Ergänzungen zu den Standard-Geschäftsprozessdiagrammen notwendig. Die erste Ergänzung ist eine Kennzeichnung der Stellen, an welchen Variabilität auftritt. Zweitens sollten mögliche Lösungen in der Darstellung aufgezeigt werden. Drittens sollte der vorhandene Variabilitätsmechanismus verwendet werden, um Lösung für die mögliche Variation abzuleiten [PSWW05].

Mittels dem aus UML bekannten Stereotypenkonzept können in BPMN Variationspunkten identifiziert werden. Die Orte, bei welchen eine Prozessmodellvariabilität auftreten kann, werden mit dem Stereotyp Variationspunkte markiert [LR09, PSWW05]. Der Name des Stereotyps wird in kursiver Schrift, zwischen zwei spitzigen auf jeder Seite platzierten Klammern geschrieben. Zur Erfüllung des Ziels von variantenreichen Prozessdiagrammen ist es ausreichend, den Variationspunkt als Stereotype «VarPoint» zu deklarieren. Dieser Stereotyp kann auch grafisch durch einen Marker in Form eines Puzzle-Stücks an der Unterseite einer Aktivität ausgedrückt werden. Wenn diese grafische Darstellung verwendet wird, kann die textuelle Bezeichnung weggelassen werden [PSWW05].

Für bessere Verständlichkeit werden die Stereotypen mit Eigenschaftswerten darge-

### 3 Ansätze zur Abbildung von Prozessvarianten

stellt. Ein Eigenschaftswert wird unter einem Stereotyp in geschweiften Klammern als Schlüssel-Wert Paar geschrieben werden. Der Stereotyp VarPoint hat zwei vordefinierte Eigenschaftswerte, Funktion und Typ. Der Wert der Typ-Eigenschaft kann „optional“, „abstrakt“, „null“ und „default“ sein [PSWW05]. Anstatt der Typ-Eigenschaft kann «VarPoint» auch als «Optional», «Abstract», «Null» und «Default» spezialisiert werden. In dem Fall wird der zusätzliche Eigenschaftswert nicht mehr benötigt [PSWW05].

Wie die Variationsmechanismen genutzt werden können, wird anhand des in Abbildung 3.5 dargestellten Beispiels erklärt.

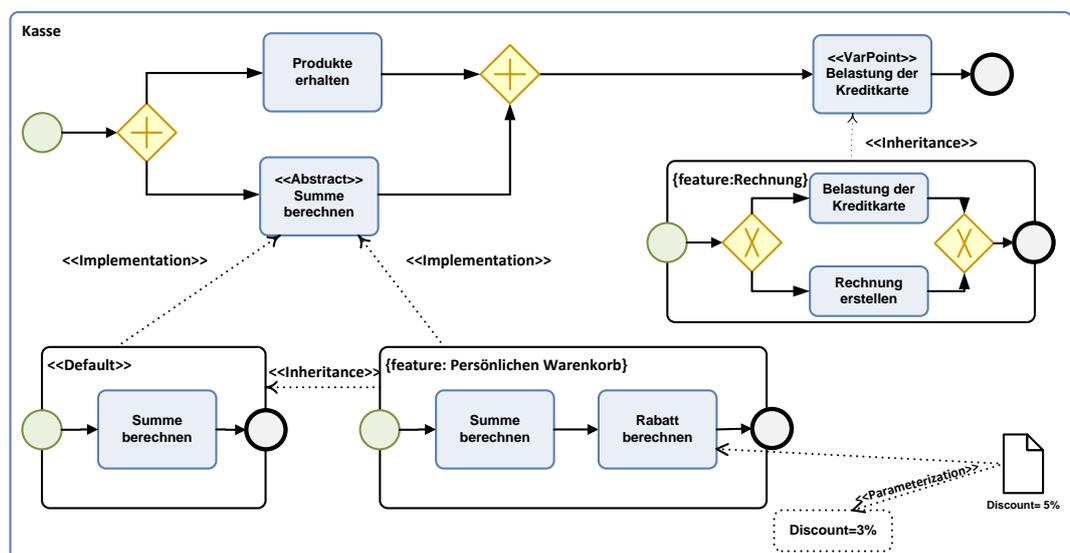


Abbildung 3.5: Beispielprozess mit PESOA [PSWW05]

Eine der Variabilitätsmechanismen bei PESOA ist die Verkapselung von Teilprozessen. Ein BPMN Teilprozess kann sich hinter einer invarianten Schnittstelle als alternativer Teilprozess verstecken. Dadurch wird mit einer Schnittstelle wie eine Menge von Eingangs- und Ausgangsereignissen definiert. Diese Schnittstellenaktivität wird dem Stereotyp «Abstract» markiert [PSWW05]. In Abbildung 3.5 besitzt die Aktivität „Summe berechnen“ den Stereotype «Abstract». Hier gibt es zwei verschiedenen Varianten. Entweder wird der Sub-Prozess mit «Default»-Wert gewählt und nur die Summe berechnet oder der Sub-Prozess mit dem Eigenschaftswert „persönliches Warenkorb“ wird gewählt und die

Summe sowie der Rabatt berechnet.

Ein anderer Variabilitätsmechanismus ist die Parametrisierung. Jedes BPMN-Attribut kann parametrisiert werden, um Variationen abzubilden. Für eine graphische Darstellung wird das Attribut neben das Element geschrieben und von einer Gruppierungsbox umgeben. Wenn die Verbindung zwischen dem Attribut und dem Element falsch interpretiert werden kann, sollte zusätzlich eine Assoziation verwendet werden. Die Assoziation wird mit dem Stereotyp «Parametrisierung» markiert [PSWW05]. Im Beispielprozess kann der Rabatt, je nach Entscheidung des Verkäufers, von 3% auf 5% erhöht werden.

Die nächste Variabilitätsmethode ist die Vererbung (Inheritance). Sie verändert einen vorhandenen Teilprozess durch das Hinzufügen oder Entfernen von Elementen unter Beziehung bestimmter Regeln. Eine Assoziation von der Kind- Aktivität bzw. Kind-Subprozess zu der Vater-Aktivität wird hierzu mit dem Stereotyp «Inheritance» markiert [PSWW05]. Im Beispiel in Abbildung 3.5 ist die Vererbungsbeziehung zwischen den Aktivitäten veranschaulicht. Die Aktivität „Belastung der Kreditkarte“ ist als Variationspunkt markiert. Mit der Vererbung wird ein neuer Subprozess als Erweiterung der Standardaktivität angeboten. Somit werden zwei Varianten dargestellt, um die Rechnung zu bezahlen.

Zur Verwendung der Verkapselung- und Vererbungskonzepte können die sogenannten Design Pattern implementiert werden. Somit wird hier keine neue Notation benötigt. In Abbildung 3.5 wird die zusätzliche Vererbungsbeziehung zwischen „Summe und Rabatt berechnen“ und „Summe berechnen“ zur bereits erwähnten Verkapselungseigenschaft der Aktivität „Summe Berechnen“ hinzugefügt. Somit wird das Design Pattern erhalten. Mit Hilfe vom Erweiterungspunkten können optionale Variationspunkte realisiert werden. Dazu sind die Verkapselung mit einem Null-Teilprozess kombiniert. Eine Aktivität wird mit dem Stereotyp «Null» markiert und optionale Implementierungen werden mit als «Extension» markierten Assoziationen verbunden. Gibt es nur eine optionale Variante, wird diese Aktivität mit einem Stereotyp «Optional» markiert. Damit bietet PESOA eine flexible Prozessmodellierungsmöglichkeit.

## 3.2 Strukturbasierter Ansatz

Der strukturbasierte Ansatz verwendet eine Basisvariante der unterschiedlichen Prozessvarianten, sowie Änderungsoperationen wie Einfügen, Löschen oder Verschieben, um die einzelnen Prozessvarianten daraus abzuleiten. Um die Basisvariante an die möglichen Prozessvarianten anpassen zu können, werden Variationspunkte aufgesetzt. Die notwendigen Änderungen für die einzelnen Varianten werden auf Basis dieser Variationspunkten mit Hilfe von Änderungsoperationen realisiert [RW12]. Im folgenden Abschnitt wird PROVOP als Vertreter für den strukturellen Ansatz präsentiert.

### 3.2.1 Provop

Provop (PROZessVarianten mittels OPtionen) ist ein operativer Ansatz, der an der Uni Ulm entwickelt wurde [Hal09]. Er ermöglicht die Variantenvielfalt von Prozessen in einem gemeinsamen Prozessmodell abzubilden. Als Ausgangspunkt wird ein festgelegter Prozess verwendet. Von diesem Prozess können die Varianten auf Basis von Variationspunkten und mittels der variantenspezifischen Abweichungen modelliert werden [HBR08b].

Als Ausgang wird ein Basisprozess festgelegt, von dem unterschiedliche Prozessvarianten, durch Konfiguration, abgeleitet werden. Es gibt verschiedene Strategien, die bei der Auswahl des Basisprozesses helfen. Beispielsweise kann als Basisprozess ein domänenspezifischer Standard- oder Referenzprozess verwendet werden. Oder, wenn eine Prozessvariante häufiger verwendet wird als eine andere, kann diese als Basisprozess ausgewählt werden. Als andere Möglichkeit kann aus der Schnittmenge der Prozessvarianten ein Basisprozess gebildet werden. Welche Strategie ausgewählt wird, um einen Basisprozess zu definieren, ist abhängig von der Modellierung und vom Szenario der vorliegenden Prozessumgebung [Hal09, HBR10].

Nach der Auswahl des Basisprozesses werden sogenannte Aufsetzpunkte definiert, die im Basisprozess die Variationspunkte anzeigen. Sie werden als schwarze Rauten mit einem Bezeichner dargestellt und können von Änderungsoperationen referenziert werden [Hal09]. Aufsetzpunkte können sowohl beim Ursprung bzw. beim Ziel

einer Kontrollfluss-Kante als auch an Ein- und Ausgängen von Knoten positioniert werden [Hal09].

Änderungsoperationen werden dazu verwendet, ausgehend vom Basisprozessmodell die einzelnen Prozessvariante abzuleiten. Die wichtigsten Änderungsoperationen sind INSERT, DELETE und MOVE von Prozessfragmenten sowie MODIFY von Attributwerten einzelner Prozesselemente [Hal09, RW12]. Abbildung 3.6 zeigt die Symbole dieser Änderungsoperationen. Während Insert-Operationen es ermöglichen, dass ein Prozessfragment einem Basisprozess hinzugefügt wird, entfernen Delete-Operationen Prozessfragmente aus dem Basisprozessen. Move-Operationen ändern die Ausführungsreihenfolge der Prozessfragmente, und die Attribute der Prozesselemente werden mit Hilfe der Modify-Operationen geändert. Durch die Kombination mehrerer Änderungsoperationen sind zudem komplexe Anpassungen des Basisprozesses möglich [Hal09].

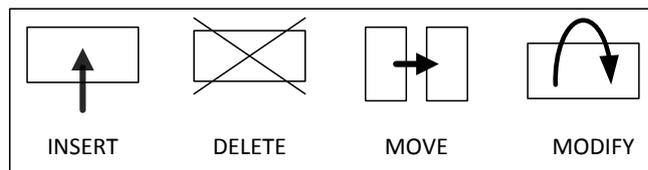


Abbildung 3.6: Symbole der Änderungsoperationen [Hal09]

Die erforderliche Anzahl der nötigen Änderungsoperationen, um Prozessvarianten von einem Basisprozess ableiten zu können, können sehr groß werden. Provop erlaubt das Gruppieren der Änderungsoperationen in sogenannte Änderungsoptionen. Dies ist zum Beispiel nützlich, wenn bestimmte Änderungsoperationen bei der Ableitung bestimmte Varianten immer zusammen auftreten [RW12].

Um die gewünschte Prozessvariante abzuleiten, können mehrere Änderungsoptionen auf ein Basisprozessmodell angewendet werden. Dies muss jedoch nicht bedeuten, dass beliebige Kombinationen der Änderungsoptionen zugelassen werden. Zum Beispiel können zwei Änderungsoptionen sich gegenseitig ausschließen, während manche Optionen andere Änderungsoptionen implizieren, um die syntaktische und semantische Richtigkeit der Prozessvariante sicherzustellen. Provop ermöglicht es diese semantischen und syntaktischen Beziehungen zwischen Änderungsoperationen explizit zu

### 3 Ansätze zur Abbildung von Prozessvarianten

erfassen. Beispiele für solche Einschränkungen der Optionen sind Implikationen, der wechselseitige Ausschluss und die Hierarchie. Wenn die Verwendung einer bestimmten Änderungsoption die gleichzeitige Anwendung einer anderen Änderungsoption erfordert, werden Implikationseinschränkungen definiert. Beim wechselseitigen Ausschluss darf eine Änderungsoption X nicht in Verbindung mit einer Änderungsoption Y (und umgekehrt) angewendet werden. Hierarchien ermöglichen Vererbung zwischen den Optionen [RW12]. Darüber hinaus erlaubt Provop kontextsensitive Prozesskonfigurationen. Das heißt, Provop ermöglicht die Konfiguration einer Prozessvariante nur durch die Anwendung der im aktuellen Kontext relevanten Optionen. Dies erfordert wiederum ein Modell, welches den Prozesskontext beschreibt. Bei Provop besteht ein solches Kontextmodell aus einer Reihe von Kontextvariablen. Jede Kontextvariable repräsentiert eine bestimmte Dimension des Prozesskontextes und wird durch einen Namen und einen Wertebereich definiert [Hal09, HBR10]. Um zahlreiche Wertekombinationen zu beurteilen, bietet Provop explizite Kontextregeln für jede Option an. Diese werden als einfache Wenn-Dann-Formeln spezifiziert [Hal09]. Der in Abbildung 3.7 dargestellte Beispielprozess zeigt, wie die Möglichkeiten des Provop-Ansatzes für die Modellierung von Prozessvarianten verwendet werden können. Im gezeigten Beispiel geht es um einen Prozess für ein Darlehen. Im Basisprozess stellt der Kunde zunächst einen Darlehensantrag. Der Bankangestellte überprüft, ob der Antragsteller Schulden hat. Wenn er Schulden hat, wird das Darlehen abgelehnt. Im anderen Fall bekommt der Antragsteller das Darlehen. Je nach Zweck des Darlehens ändert sich jedoch der Prozessablauf. Wenn das Darlehen, beispielsweise für des Bau eines Hauses beantragt wurde, muss zusätzlich die finanzielle Haftung des Antragstellers überprüft werden. Dazu wird Option 1 angewendet. Mit der Insert-Operation wird hier die zusätzliche Aufgabe zum Basisprozess hinzugefügt. Wenn der Antrag jedoch wegen des Studiums gestellt wird und der Antragsteller Schulden hat, dann bekommt er ein beschränktes Darlehen. Dazu wird Option 2 verwendet. Mit der Delete-Option wird die Aktivität „Darlehen genehmigen“ aus dem Basisprozess entfernt und stattdessen mit der Insert-Operationen die Aktivität „beschränktes Darlehen genehmigen“ hinzugefügt. Beide Optionen besitzen dabei voneinander unabhängige Kontextregeln (CTXTEXT RULE). Das heißt, es gibt keinen semantischen Zusammenhang zwischen diesen Optionen. Kontext-Regeln und Opti-

onsbedingungen sind größtenteils unabhängig. Da die gleichzeitige Anwendung der beiden Optionen jedoch nicht möglich ist, werden diese über der Optionsbedingungen als „wechselseitiger Ausschluss“ gekennzeichnet.

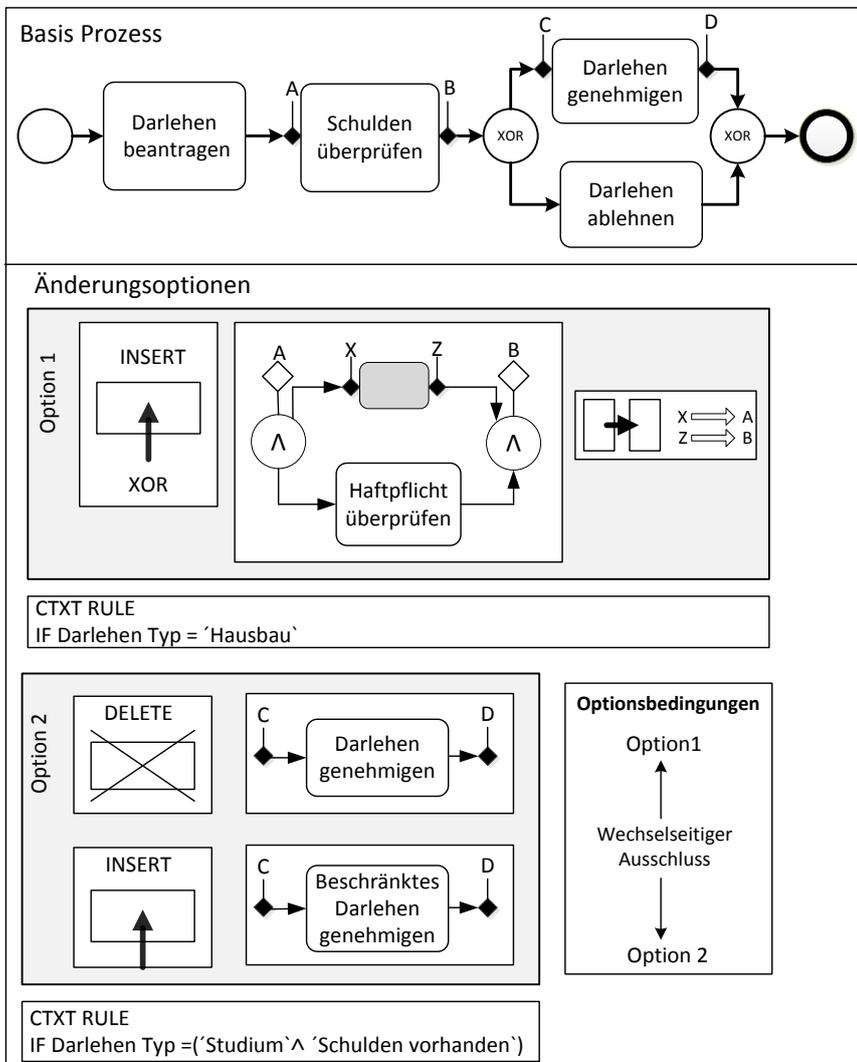


Abbildung 3.7: Beispielprozess mit Provop [Wor]



# 4

## **Vergleich der Ansätze zur Modellierung variantenreicher Prozesse**

Wie im vorherigen Kapitel erläutert wurde, bieten die Ansätze C-EPC, C-YAWL, PESOA und Provop die Möglichkeit, Prozessvarianten zu realisieren. In diesem Kapitel werden diese Ansätze anhand von drei Beispielprozessen der Universität Ulm miteinander verglichen und ihre Vor- sowie Nachteile diskutiert.

## 4.1 Prozessbeispiel 1

Im ersten Beispiel geht es um die Zeit- und Raumplanung an der Universität Ulm. In diesem Prozess gibt es drei Beteiligte, den Lehrveranstaltungsadministrator (LV-Koordinator), den dezentralen Raumverwalter und den zentralen Raumverwalter.

Die Raum- bzw. Zeitanfrage wird von einem LV-Koordinator gestellt. Falls diese Anfrage konkret ist und der Anfragende selbst Raumverwalter ist, so kann er die Raumverfügbarkeit selbst überprüfen. Sollte der Raum verfügbar sein, wird dieser direkt zugewiesen. Für unkonkrete Anfragen muss der zentrale Raumverwalter, unter Berücksichtigung des erforderlichen Konfliktmanagement, einen geeigneten Raum suchen und dessen Verfügbarkeit überprüfen. Wenn der Raum verfügbar ist, belegt der zentrale Raumverwalter den Raum, bestätigt dessen Belegung und sendet die Bestätigung an den LV-Koordinator. Wenn die Raum-/Zeitanfrage unkonkret ist und der Anfragende nicht selbst der Raumverwalter ist, muss die Anfrage für diesen Raum an den zuständigen Raumverwalter weitergeleitet werden. Falls hierfür der dezentrale Raumverwalter zuständig ist, muss er sich um die Überprüfung der Raumverfügbarkeit kümmern. Sofern der Raum verfügbar ist, belegt der dezentrale Raumverwalter den Raum, bestätigt die Belegung und sendet die Bestätigung zum Koordinator.

Der zentrale Raumverwalter führt die gleichen Aufgaben wie der dezentrale Raumverwalter aus. In allen Fällen wird die Anfrage wiederholt oder die Raumplanung abgebrochen, falls der Raum nicht verfügbar ist.

Abbildung 4.1 stellt das in BPMN modellierte originale Prozessmodell dar. In diesem Prozess existieren zwei Variationspunkte. Der erste Punkt entscheidet ob der Anfragende selbst ein Raumverwalter ist oder nicht. Der zweite Punkt entscheidet welcher Raumverwalter für den angefragten Raum zuständig ist. Anhand dieser Variationspunkte wird versucht die Prozessvarianten in den vier zuvor beschriebenen Ansätze zu modellieren. In der Abbildung 4.2 ist das C-EPC Prozessmodell dargestellt. Das Prozessmodell besteht aus einem konfigurierbaren XOR-Knoten und einem Requirement. Mit Hilfe dieses Requirements und des XOR-Knotens wird der erste Prozessvariationspunkt „Ist Anfragender Raumverwalter,“ realisiert. Damit wird der Pfad SEQ1a ausgeführt, wenn der Anfragende selbst der Raumverwalter ist, andernfalls wird Pfad SEQ1b ausgeführt.

# 4.1 Prozessbeispiel 1

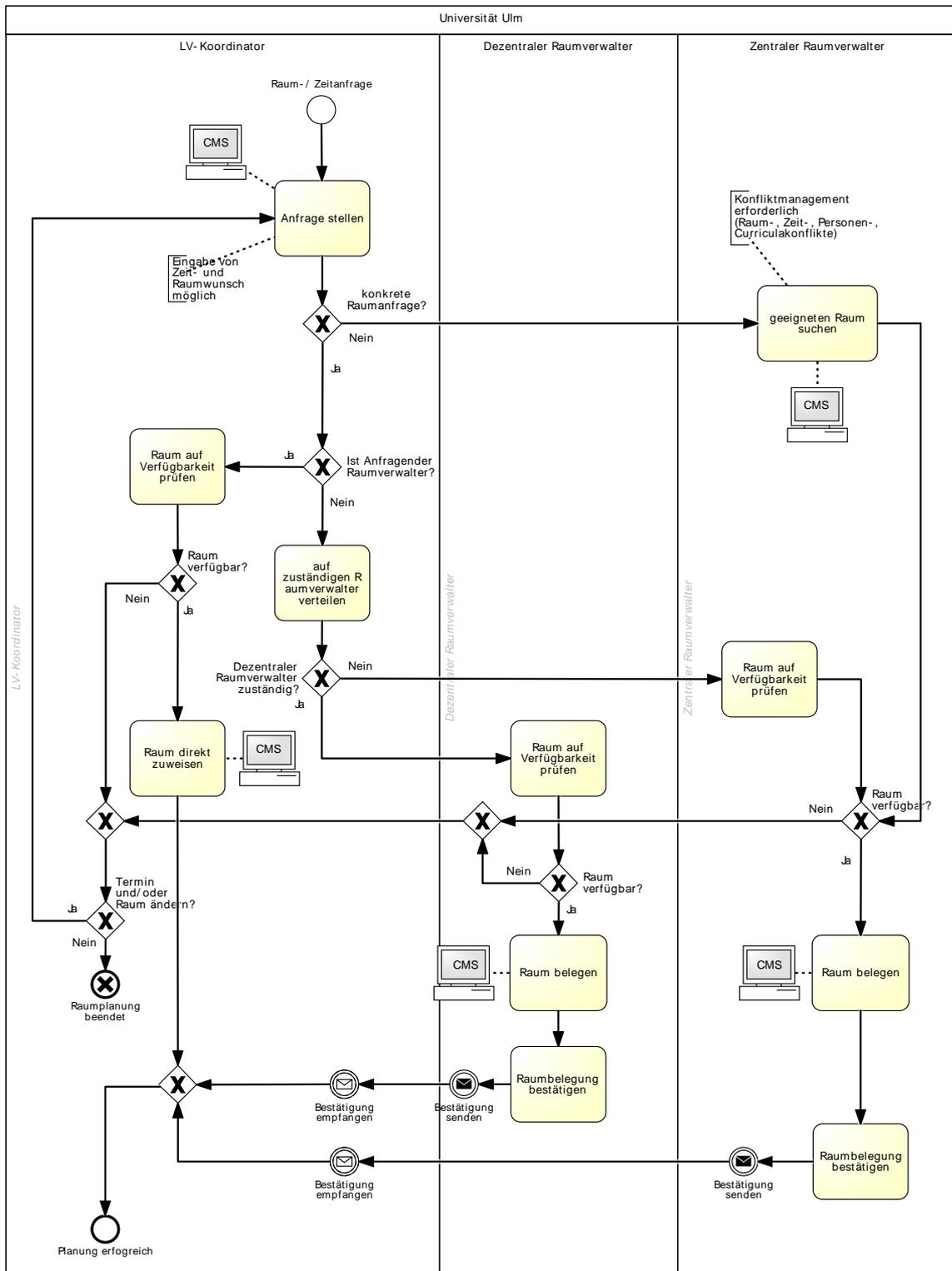


Abbildung 4.1: Zeit- und Raumplanung

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

Beim zweiten Variationspunkt sollte die Anfrage zum richtigen Raumverwalter verteilt und weitere Aufgaben vom dementsprechenden Raumverwalter ausgeführt werden, wenn der Anfragende kein Raumverwalter ist. Der zentrale und dezentrale Raumverwalter führen genau dieselben Aufgaben aus. Deswegen sind zusätzliche konfigurierbare XOR-Knoten und Requirements an dieser Stelle überflüssig. Insbesondere ist es in C-EPC schwierig, die eine Aktivität ausführende Rolle entsprechend der gewählten Variante zu konfigurieren.

Abbildung 4.3 stellt die Modellierung des Prozesses mit C-YAWL dar. Beim XOR-Split „Anfragender Raumverwalter?“ existieren zwei Ports. Der erste Port ermöglicht es C5 zu aktivieren und C6 zu blockieren. Umgekehrt ermöglicht es der zweite Port C5 zu blockieren und C6 zu aktivieren. Das heißt, wenn der Anfragende Raumverwalter ist, wird C5 aktiviert und C6 blockiert, ansonsten wird C5 blockiert und C6 aktiviert. Somit kann der erste Variationspunkt des Prozesses realisiert werden. Um des zweiten Variationspunkt realisieren zu können, könnte ein konfigurierbares XOR-Split „ist dezentraler Raumverwalter zuständig“ nach dem Task „auf zuständigen Raumverwalter verteilen?“ in den Prozess eingefügt werden. Dadurch wäre es möglich durch Blockieren und Aktivieren der Ports, diese Variation darzustellen. Dies ist jedoch wenig sinnvoll, da an diese Stelle lediglich die gleiche Aufgaben von verschiedenen Personen ausgeführt werden sollen. Daher ist der zweite Variationspunkt bei C-YAWL ebenfalls nicht darstellbar.

Abbildung 4.4 zeigt, wie dieser Prozess mit den Möglichkeiten der PESOA realisiert werden kann. Normalerweise besteht dieser Prozess aus drei Beteiligten, nämlich aus dem Lehrveranstaltungsleiter, dem zentralen Raumverwalter und dem dezentralen Raumverwalter. Um mit den Möglichkeiten der PESOA richtig umgehen zu können, werden der zentrale und der dezentrale Raumverwalter zu einer Rolle Raumverwalter zusammengefügt, da es mit PESOA schwierig ist, Lane-übergreifende Prozesssteile in einer Prozessvariante zusammen darzustellen. In dem Prozess existieren zwei Variationspunkte, die mit dem Stereotyp «Abstract» markiert sind. Der erste Variationspunkt ist „Raum auf Verfügbarkeit prüfen“ und der zweite Variationspunkt ist „Raum belegen“. Diese Variationspunkte werden, je nach Kontext, mit Hilfe der entsprechenden Implementierung erweitert. Mittels dieser Features können die gewünschten Prozessvariationen dargestellt werden. PESOA bietet für dieses Beispiel eine kompakte Darstellung.

#### 4.1 Prozessbeispiel 1

Abbildung 4.5 stellt den Prozess mit Hilfe des Provop-Ansatzes dar. Es gibt einen Basisprozess, zwei Optionen mit entsprechenden Kontextregeln und sowie Abhängigkeiten zwischen den Optionen. Als Basisprozess wird die Prozessvariante ausgewählt, bei der die Raumanfrage konkret und der Raumanfragende der Raumverwalter ist. Der Basisprozess besitzt zwei Aufsetzpunkte, die als A und B markiert sind. Sie zeigen die möglichen Konfigurationspunkte in dem Prozess. Mit Hilfe der Optionen, und je nach Kontextregel, werden die Variationen realisiert. Wenn die Raumanfrage konkret und der Anfragende kein Raumverwalter ist, wird Option 1 ausgeführt. D.h. die zwischen den Aufsetzpunkten A und B stehenden Aufgaben werden gelöscht und an diese Stelle neue Aufgaben hinzugefügt. Option 2 bietet mit der Änderungsfunktion Modify die Möglichkeit an, die gleiche Aufgabe für unterschiedliche Personen zu spezifizieren. Somit kann festgestellt werden, wer für die Ausführung dieser Aufgaben zuständig ist. Mit dieser Eigenschaft bietet Provop im Vergleich zu den anderen drei Ansätzen die beste Lösung, um eine zu realisieren.

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

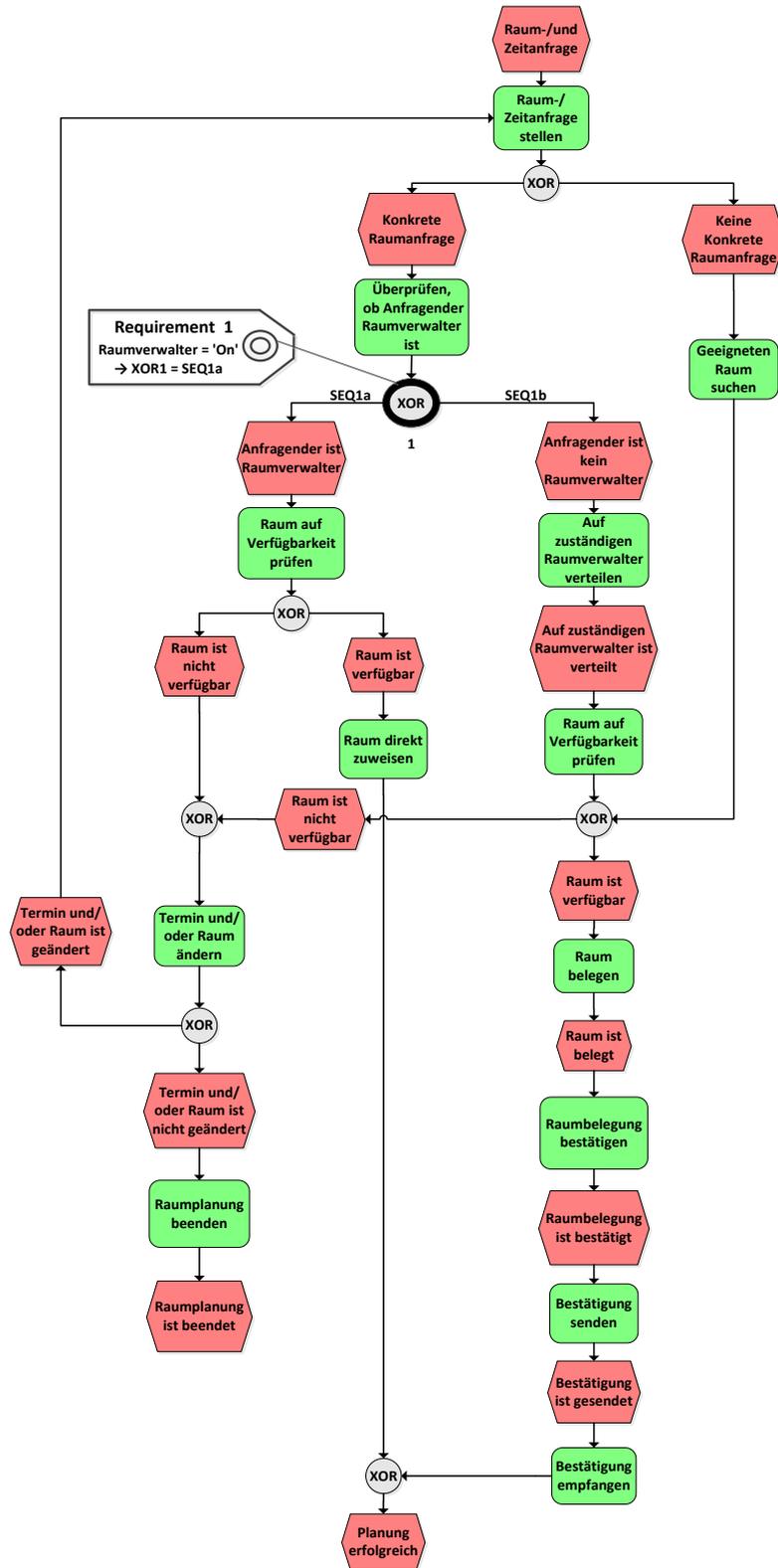


Abbildung 4.2: Zeit- und Raumplanung mit C-EPC

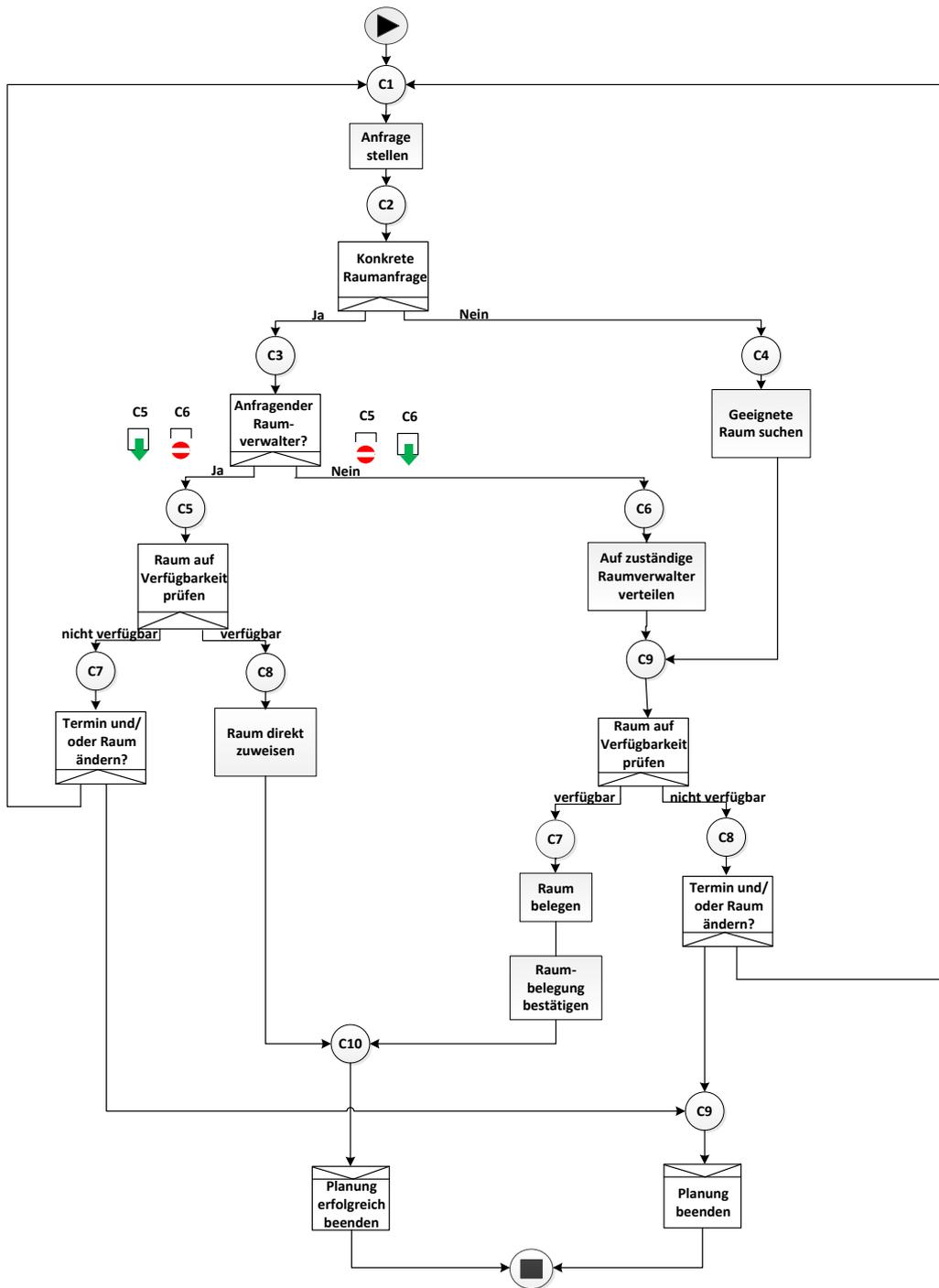


Abbildung 4.3: Zeit- und Raumplanung mit C-YAWL

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

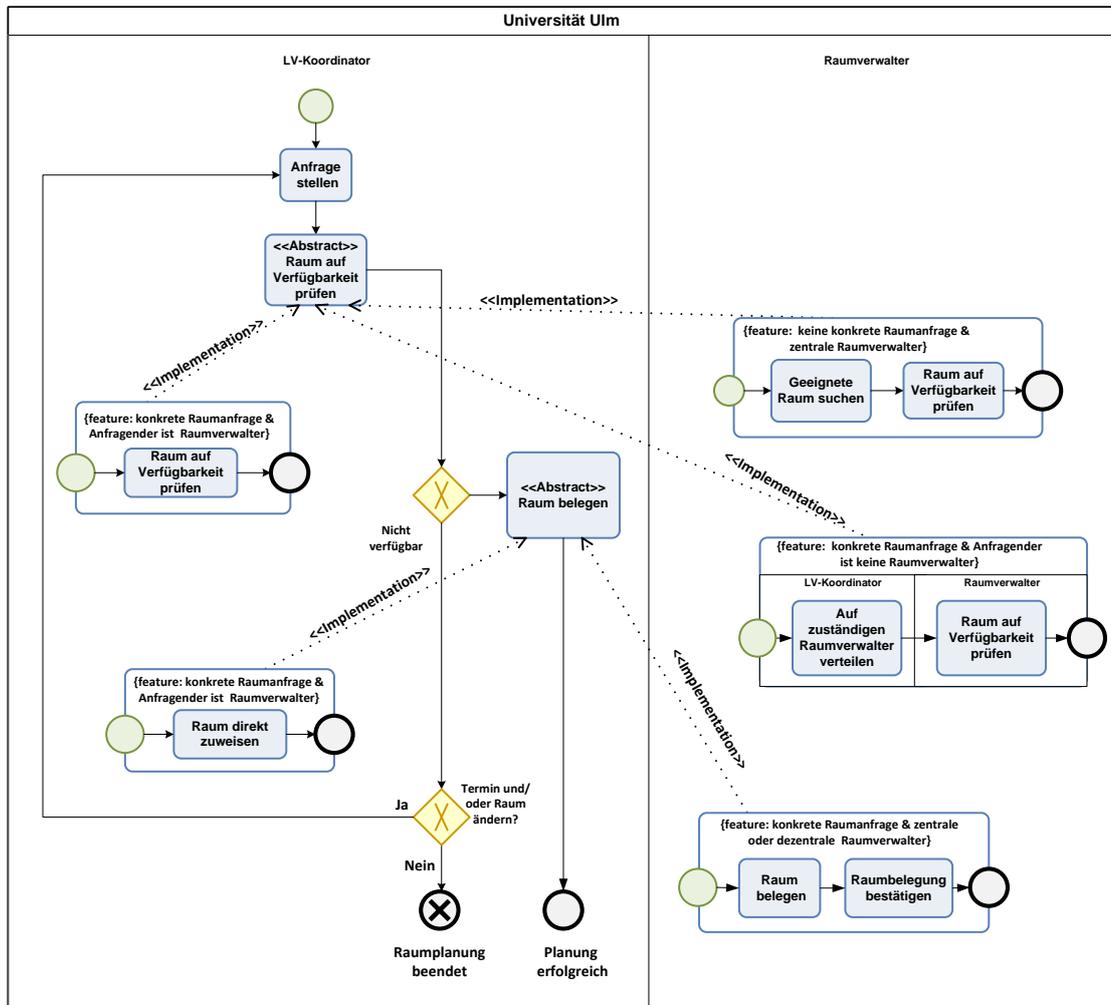


Abbildung 4.4: Zeit- und Raumplanung mit PESOA

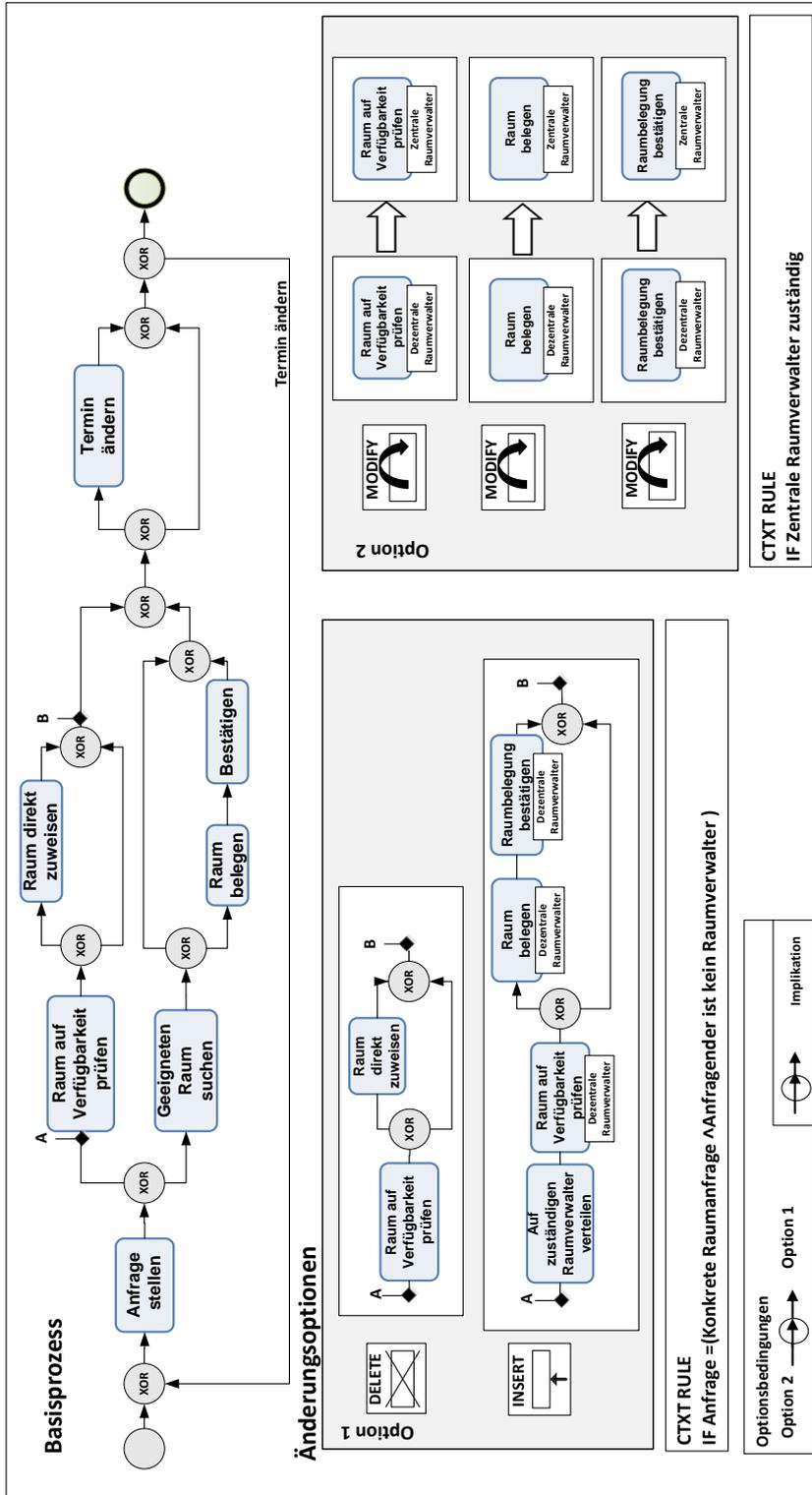


Abbildung 4.5: Zeit- und Raumplanung mit Provop

## 4.2 Prozessbeispiel 2

Der zweite betrachtete Prozess beschreibt, wie Prüfungen an der Universität Ulm durchgeführt werden. Ein Student kann sich nur zu einer Prüfung anmelden, bei der er auch für die entsprechende Lehrveranstaltung angemeldet ist. Ist er für Prüfung angemeldet, kann er diese mitschreiben. Es gibt jedoch auch Prüfungen, bei denen der Student zunächst eine Vorleistung erbringen muss, um an der Prüfung teilnehmen zu dürfen. Für solche Situationen muss der Student sich zuerst für die Vorleistung und dann erst für die Prüfung anmelden. Danach kontrolliert der Prüfer, ob die notwendige Vorleistungen erbracht wurden. Wenn ja, wird die Vorleistung von ihm als erbracht verbucht, ansonsten wird die Prüfungsanmeldung deaktiviert. Wenn der Student die Vorleistungsunterlagen nachreicht, wird die Vorleistung dennoch als erbracht angesehen und die Prüfungsanmeldung wird wieder aktiviert. Im anderen Fall darf der Student die Prüfung nicht mitschreiben. Nach der Prüfung wird die Note veröffentlicht. Bei einer nicht bestandenen Prüfung gibt es zwei Möglichkeiten, entweder der Student hat Anspruch auf eine Prüfungswiederholung, die er auch absolviert oder der Student hat bereits Wiederholt und keinen weiteren Versuch mehr und er bekommt einen Exmatrikulationsbescheid. Wenn er die Prüfung bestanden hat und ein Freiversuch für diese Prüfung möglich ist, kann er die Prüfung ebenfalls wiederholen. Bei der Prüfungswiederholung soll der Student je nach Prüfungsordnung die Anmeldung erneuern, um an der Wiederholung teilnehmen zu können. Abbildung 4.6 stellt das in BPMN modellierte originale Prozessmodell dar. Das beschriebene Beispiel beinhaltet vier Prozessvarianten. Die erste Variante ist eine Prüfung ohne Vorleistung und ohne Freiversuchsmöglichkeit. Die zweite Variante ist zwar mit Vorleistungsanmeldung aber ohne Freiversuchsmöglichkeit. Die dritte Variante ist mit Vorleistungsanmeldung und Freiversuchsmöglichkeit. Die letzte Variante ist ohne Vorleistungsanmeldung und mit Freiversuchsmöglichkeiten. Die folgenden Abbildungen stellen dar, wie dieser variantenreiche Prozess mit Hilfe der vier Ansätze realisiert werden kann, und welche Probleme dabei auftreten können.

Abbildung 4.8 zeigt die Darstellung mit C-EPC. Der Prozess besitzt vier konfigurierbare XOR-Knoten und vier Requirements. Requirement 1 bestimmt welcher Pfad ausgeführt wird, wenn eine Vorleistungsanmeldung erforderlich ist. Mit Hilfe des Requirements 2 wird

## 4.2 Prozessbeispiel 2

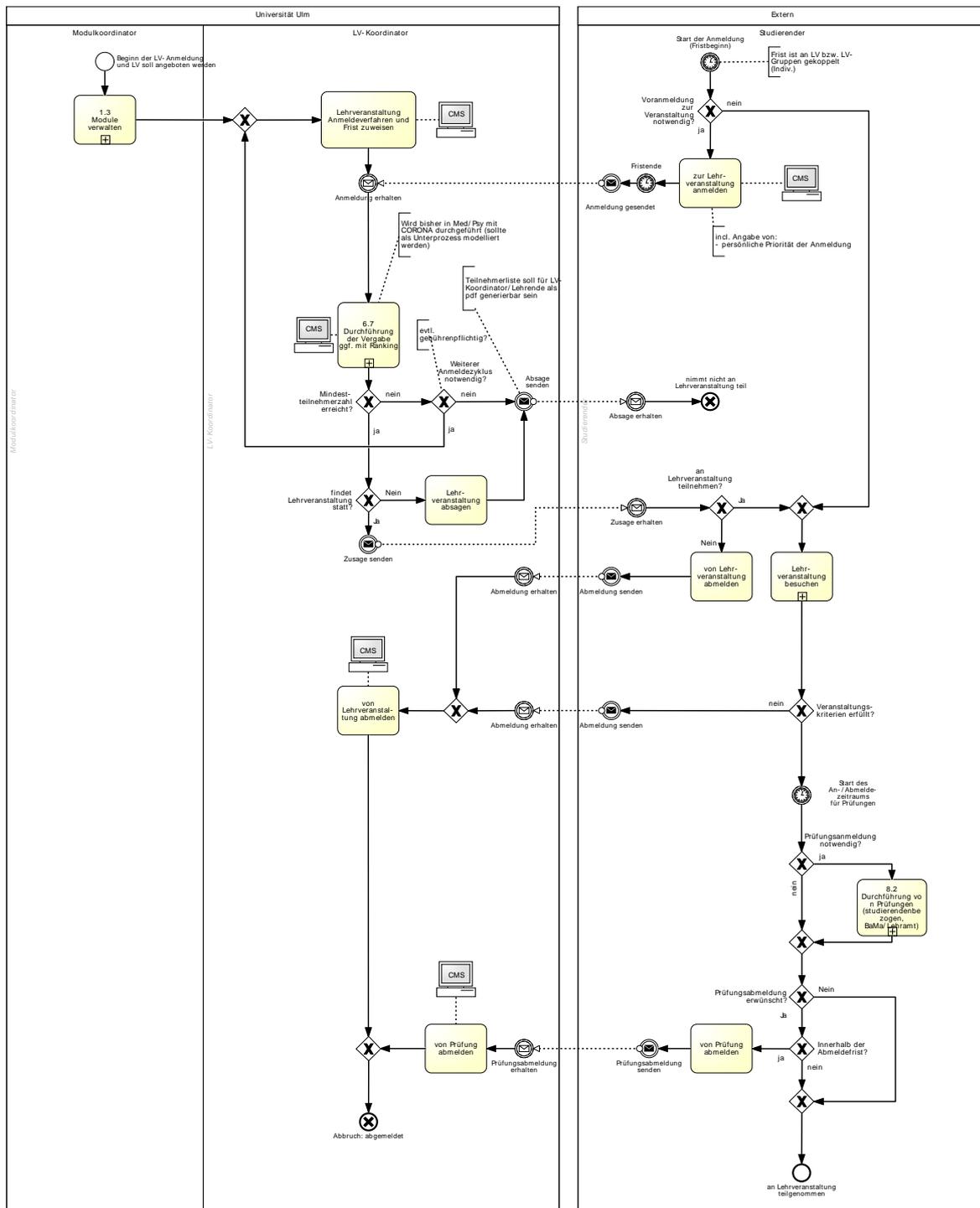


Abbildung 4.6: Durchführung von Prüfungen

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

der erste und zweite XOR-Knoten nach der Konfiguration zur gleichen Art. Requirement 1 und Requirement 3 sind voneinander abhängig aufgrund der Vorleistungsanmeldung. Wenn bei Requirement 1 der Pfad SEQ1b gewählt wurde, muss bei Requirement 3 SEQ3a ausgeführt werden. Als letztes bestimmt Requirement 4, welcher Pfad ausgeführt wird, wenn ein Freiversuch möglich ist. In dem Fall wird SEQ4a ausgeführt.

In diesem Prozess existieren Abhängigkeiten zwischen den Aufgaben. C-EPC bietet eine gute Nachvollziehbarkeit dieser Abhängigkeiten mit Hilfe der Requirements an. Wiederholungen, Sprüngen und Abbrüche sind leicht realisierbar. Weiterhin sind alle möglichen Varianten auf einen Blick sichtbar, was jedoch nicht immer vorteilhaft ist. Denn, je größer der Prozess ist, desto größer ist die Anzahl der verwendeten Elemente. Somit wird der Prozess komplexer und belastet den Benutzer. Mit einer Konfigurationstabelle zu arbeiten, wäre an dieser Stelle für den Benutzer hilfreich, denn damit kann er feststellen, für welche Variante er welche Pfade ausführen muss. Tabelle 4.1 stellt die möglichen Variante und die dazu gehörigen Pfade dar.

	XOR1	XOR2	XOR3	XOR4
ohne Vorleistung, ohne Freiversuch	SEQ1a	SEQ1a	SEQ3b	SEQ4b
Vorleistung, ohne Freiversuch	SEQ1b	SEQ1b	SEQ3a	SEQ4b
Vorleistung mit Freiversuch	SEQ1b	SEQ1b	SEQ3a	SEQ4a
ohne Vorleistung mit Freiversuch	SEQ1a	SEQ1a	SEQ3b	SEQ4a

Tabelle 4.1: Konfigurationstabelle für Prozessbeispiel 2

Abbildung 4.9 zeigt, wie dieser Prozess mit C-YAWL realisiert wird. Um die Varianten zu realisieren, werden Ports verwendet, die, je nach Prozessvariante, entweder aktiviert oder blockiert werden. Der XOR-Split „Vorleistung ist notwendig“ besitzt zwei Ports. Ist eine Vorleistung notwendig, wird C2 aktiviert und C3 blockiert. Damit wird die Aufgabe „Zur Vorleistung anmelden“ ausgeführt. Anderenfalls wird C2 blockiert und C3 aktiviert. Beide Pfade führen zur vorläufigen Prüfungsanmeldung.

Der XOR-Split „Vorleistungen erbracht“ besitzt einen Eingangsport. Die Ausführung dieses Eingangsports ist abhängig von dem vorherigen XOR-Split. Wenn keine Vorleistung erforderlich ist, wird dieser Port blockiert, und der Prozess wird über C4 zu C9 geleitet. Ansonsten wird dieser Port aktiviert und weitere Prozessentscheidungen je

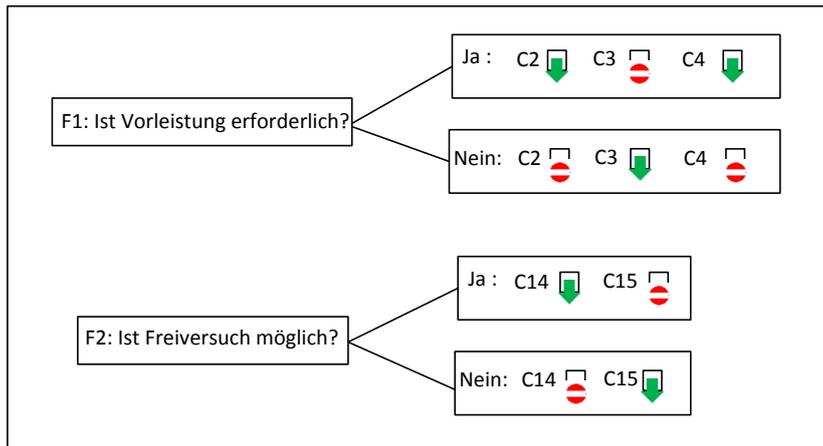


Abbildung 4.7: Fragebogen für Prozessbeispiel2

nach der Situation ausgeführt. Als Letztes besitzt der XOR-Split „Freiversuch möglich“ zwei Ausgangsports. Ist ein Freiversuch möglich, wird C14 aktiviert und nach diesem Fall weitere Aufgaben ausgeführt und C15 blockiert. Sonst wird C15 aktiviert und C14 blockiert.

Bei C-YAWL existiert keine richtige Unterstützung, um die Abhängigkeiten der Ports zu spezifizieren. Diese Situation verursacht Schwierigkeiten während der Realisierung der Prozessvarianten. Die für C-YAWL vorgesehene Lösung ist, mit einem Fragebogen zu arbeiten. Mittels dieses Fragebogens können die Abhängigkeiten der Fragen und die Zustände der Ports je nach Variante dargestellt werden. Abbildung 4.7 zeigt den möglichen Fragebogen für dieses Beispiel.

Abbildung 4.10 stellt den Prozess in PESOA dar. Der erste Variationspunkt der Prüfungsanmeldung kann mit Hilfe der Vererbungseigenschaft erweitert werden. Wenn eine Vorleistungsanmeldung erforderlich ist, wird der erweiterte Prozess ausgeführt, sonst ignoriert. Die Vorleistungsüberprüfung ist in einen Subprozess ausgegliedert und mit dem Stereotyp «Optional» markiert. Wenn die Vorleistungsanmeldung erforderlich ist, wird dieser Subprozess ausgeführt, sonst übersprungen. Diese optionale Aufgabe bietet eine kompakte Darstellung an. Jedoch verursacht die erhöhte Anzahl Subprozessen eine Erhöhung der Wartungskosten sowie der Komplexität des Prozesses.

Weiterhin bietet PESOA keine Möglichkeiten, um die Entscheidungsknoten zu konfigurieren. Somit ist es schwierig, den Knoten „Freiversuch durchführen“ als Variationspunkt

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

darzustellen. Es könnte versucht werden, „Freiversuch durchführen“ als optionale Aufgabe zu wählen und dazu „Freiversuch möglich“ als Feature zu markieren. Jedoch hat jede Aufgabe einen einkommenden und einen ausgehenden Pfad, wodurch es nötig wäre, dass Entscheidungsknoten eingebaut werden, die zu prüfen, ob die Aufgabe ausgeführt wurde oder nicht. Diese Aufgabe als optional zu kennzeichnen wäre sinnlos, da diese so oder so ausgeführt werden muss.

Abbildung 4.11 stellt dar, wie die verschiedenen Varianten dieses Prozesses mit Provop modelliert werden können. Als Basisprozess wird gewählt, dass für die Prüfungsanmeldung keine Vorleistung benötigt wird und nach dem Bestehen der Prüfung keine Freiversuche möglich sind. Es gibt somit zwei Optionen, die durch Aufsetzpunkte die Änderungen des Basisprozesses bewirken. Option 1 wird ausgeführt, wenn eine Vorleistung notwendig ist. Dazu wird die Aktivität „zur Prüfung anmelden“ gelöscht und durch die Aktivitäten „zur Vorleistung anmelden“ und „Vorläufig zur Prüfung anmelden“ ersetzt. Zusätzlich muss auch der zweite XOR-Block hinzugefügt werden, der mehrere XOR-Knoten beinhaltet. Damit wird, wenn der Student die Vorleistung nicht erbracht hat, die Vorleistung als nicht erbracht verbucht und die Prüfungsanmeldung deaktiviert. Sollte der Student die Vorleistung nachreichen, wird vom letzten XOR-Knoten zum ersten XOR-Knoten gesprungen, die Vorleistung wird als erbracht verbucht und die Prüfungsanmeldung aktiviert. Der Fall, dass ein Student die Vorleistung nicht nachreicht und die Prüfungsanmeldung abgebrochen werden muss, ist aufgrund der notwendigen Blockstruktur problematisch darzustellen. Um dies zu ermöglichen müssen vor der Prüfung und ganz am Ende zwei XOR-Knoten eingefügt werden, die es erlauben den Prozess sofort zu beenden. Diese kann jedoch für den Benutzer verwirrend sein.

## 4.2 Prozessbeispiel 2



Abbildung 4.8: Durchführung von Prüfungen mit C-EPC

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

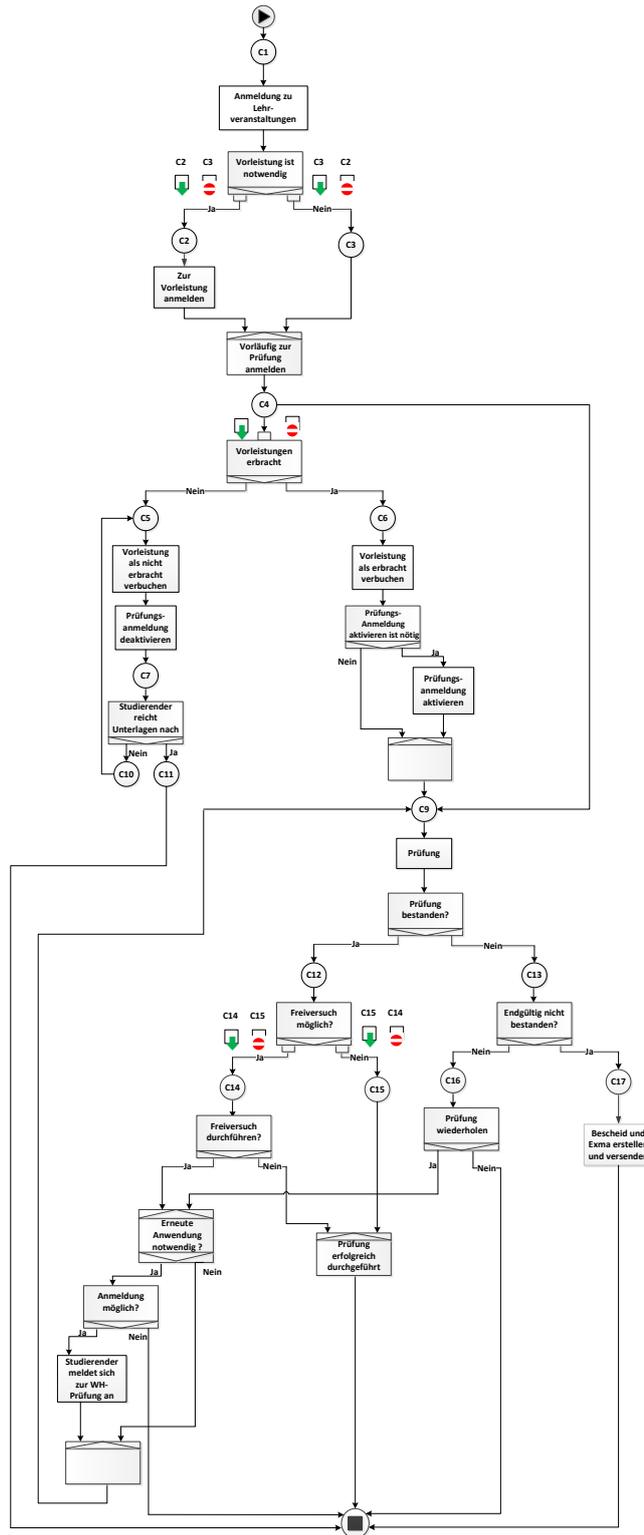


Abbildung 4.9: Durchführung von Prüfungen mit C-YAWL

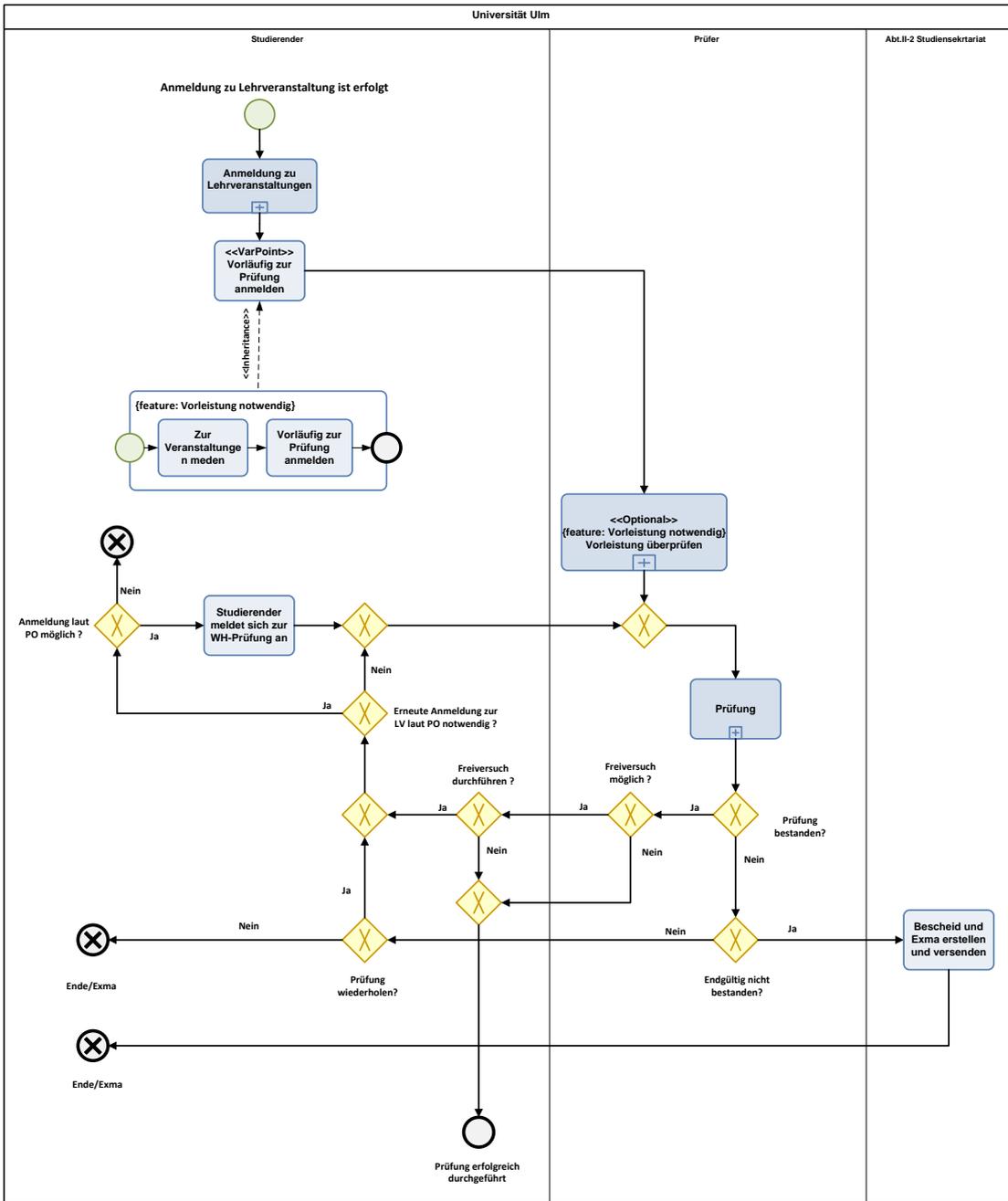


Abbildung 4.10: Durchführung von Prüfungen mit PESOA



### 4.3 Prozessbeispiel 3

Im dritten Beispielprozess geht es darum, wie eine Abschlussarbeit an der Universität Ulm erfasst wird. Zuerst muss der Student sich für die Abschlussarbeit anmelden. Abschlussarbeiten werden von Dozenten geprüft und begutachtet. Je nach Art der Abschlussarbeit und Studiengang variiert sich die Anzahl der Prüfer. Die Begutachtung kann dabei jeweils von einem internen und/oder externen Prüfer ausgeführt werden. Wenn die Begutachtung von einem internen Prüfer ausgeführt wird, reicht eine Bestätigung von ihm aus. Dies gilt auch für den zweiten internen Prüfer. Wenn die Begutachtung jedoch von einem externen Prüfer ausgeführt wird, muss dieser die Anmeldung der Abschlussarbeit schriftlich bestätigen und die Bestätigung muss vom Studiensekretariat erfasst werden. Die gleiche Arbeitsfolge gilt auch für einen zweiten externen Prüfer. Anschließend erfasst das Studiensekretariat die Anmeldung formal und versendet eine Bestätigung mit dem Abgabedatum an den Student. Der Studierende muss die Abschlussarbeit bis zum Abgabedatum bearbeiten und vollständig abgeben. Wenn er bis zu diesem Datum seine Arbeit nicht zu Ende bringen kann, kann er beim PA-Vorsitzenden einen Antrag für eine Fristverlängerung stellen. Sollte die Fristverlängerung genehmigt werden, wird das Abgabedatum vom Studiensekretariat geändert und dem Studenten wird das neue Abgabedatum mitgeteilt. Sonst muss der Student zum vorher festgelegtem Abgabedatum seine Arbeit abgeben. Falls die Abschlussarbeit bis Ende des Abgabedatums nicht im Studiensekretariat abgegeben wurde, wird sie als nicht bestanden gebucht. Nachdem der Student seine Arbeit rechtzeitig im Studiensekretariat abgegeben und als PDF-Datei ins Computersystem hochgeladen hat, wird die Abschlussarbeit vom Studiensekretariat an den oder die Prüfer weitergegeben. Wenn das Gutachten von einem externen Prüfer erstellt wird, muss dieser sein Gutachten erstellen und schriftlich an das Studiensekretariat weiterreichen. Dann wird die Note vom Sekretariat eingetragen. Analog gilt dies auch für den zweiten externen Prüfer. Bei einem internen Prüfer erstellt diese das Gutachten und gibt die Note selbst in das Computersystem ein. Ebenso der zweite interne Prüfer. Sollten die Noten von den beiden Prüfern zu weit auseinander liegen, so muss die Arbeit von einem weiteren Prüfer begutachtet werden. Die Endnote wird vom Studiensekretariat ins System eingetragen und der Student wird benachrichtigt,

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

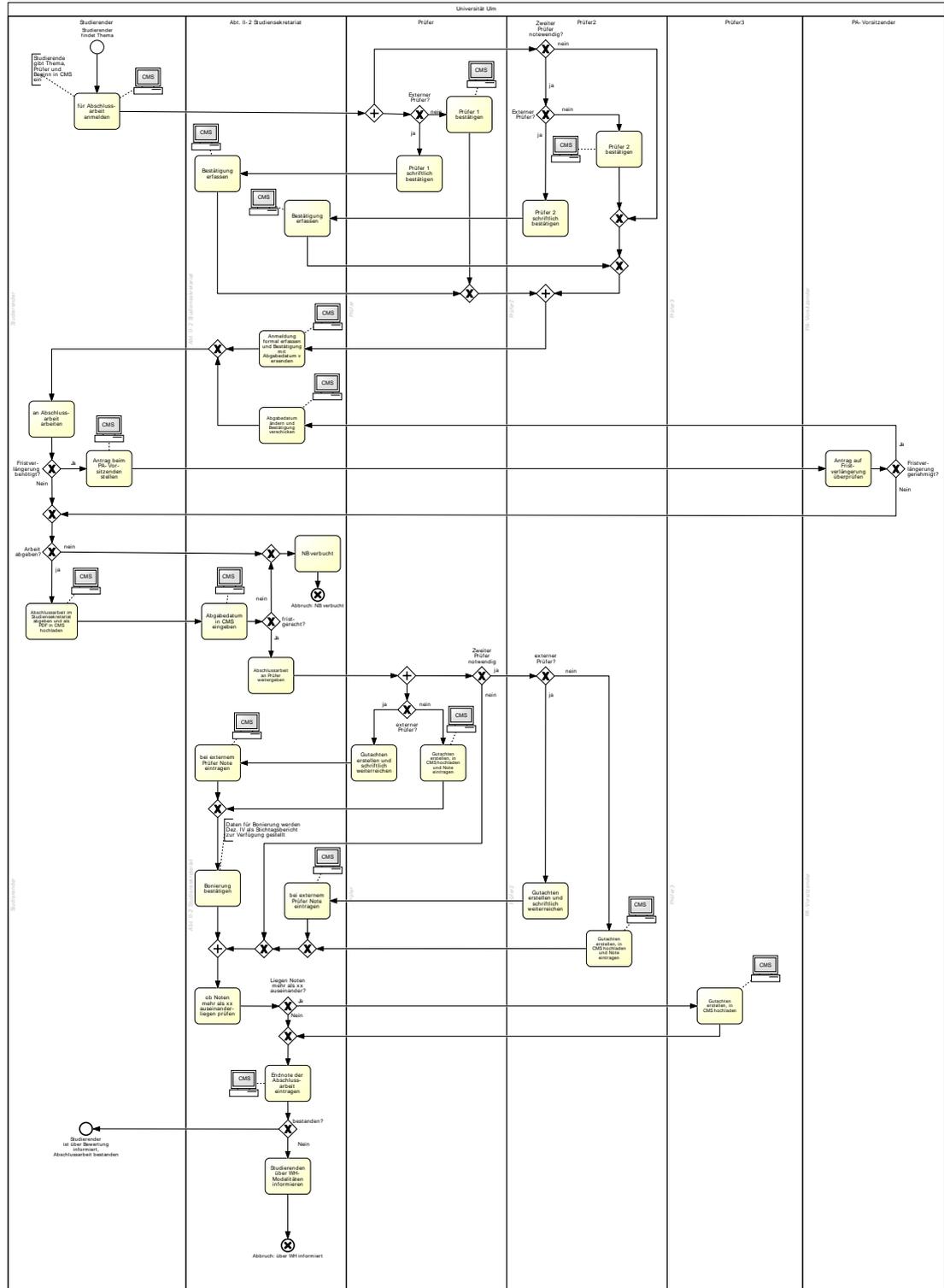


Abbildung 4.12: Bachelor- und Masterarbeit erfassen

ob er die Abschlussarbeit bestanden hat oder sie wiederholen muss. Abbildung 4.12 stellt das in BPMN modellierte originale Prozessmodell dar.

Der erste Variationspunkt in diesem Prozess ist, ob ein Prüfer intern oder extern ist. Der Nächste entscheidet die Notwendigkeit eines zweiten Prüfers und wenn ja, ob dieser zweite Prüfer wiederum extern oder intern ist. Die Bewertung der Abschlussarbeit ist abhängig davon welcher Prüfer diese betreut hat. Deswegen muss dieser Variationspunkte beim Gutachten berücksichtigt und Prozessvarianten anhand dieser Angaben realisiert werden.

Der mittels C-EPC modellierte Prozess ist in den Abbildungen 4.14 sowie 4.15 zusehen. Um den Prozess aufzuteilen, werden die sogenannten Prozesswegweiser „Prozess Teil 1“ und „Prozess Teil 2“ verwendet. Ein Prozesswegweiser ist ein Element von EPC und hilft die Prozessschnittstellen darzustellen.

Der in den Abbildungen 4.14 und 4.15 dargestellte Prozess zeigt, wie die Prozessvarianten für diesen Prozess in C-EPC dargestellt werden können. Er besitzt 11 konfigurierbare XOR-Knoten und 11 Requirements. Mit Hilfe der Requirements sind die Abhängigkeiten der Aufgaben leicht zu implementieren. Hat sich der Studierende beispielsweise mit einem internen Prüfer angemeldet, wird Requirement 1 ausgeführt. Abhängig davon muss Requirement 6 beim Gutachten ausgeführt werden. Die gleiche Abhängigkeiten gelten zwischen Requirement 2 und Requirement 7 bzw. zwischen Requirement 3 und Requirement 8. Trotz aller Variationen bzw. obwohl die Abhängigkeiten der Aufgaben gut realisierbar sind, besitzt dieser Prozess sehr viele Elemente. Je mehr Elemente der Prozess beinhaltet, desto schwieriger ist die Nachvollziehbarkeit. Deswegen sollte hier, gleich wie in dem vorherigen Beispiel, mit einer Konfigurationsliste gearbeitet werden. Somit kann der Benutzer die Abhängigkeiten, sowie die dazugehörige Varianten, besser nachvollziehen und mögliche Fehler können vermieden werden. Tabelle 4.2 zeigt die möglichen Prozessvarianten und die dazugehörige Pfade.

Abbildungen 4.16 und 4.17 zeigen, wie dieser Prozess in C-YAWL modelliert werden kann. Zur Darstellung der Varianten werden wieder Ports verwendet. Abhängig von der Art des Prüfers werden diese entweder blockiert oder aktiviert. Die möglichen Ports-Blockierungen und –Aktivierungen sind in den Abbildungen 4.16 und 4.17 ver-

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

	XOR1	XOR2	XOR3	XOR6	XOR7	XOR8
Prüfer 1 intern	SEQ1b	SEQ2b	-	SEQ6b	SEQ7a	-
Prüfer 1 extern	SEQ1a	SEQ2b	-	SEQ6a	SEQ7a	-
Prüfer 1 intern Prüfer 2 intern	SEQ1b	SEQ2a	SEQ3b	SEQ6b	SEQ7b	SEQ8b
Prüfer 1 intern Prüfer 2 extern	SEQ1b	SEQ2a	SEQ3a	SEQ6b	SEQ7b	SEQ8a
Prüfer 1 extern Prüfer 2 extern	SEQ1a	SEQ2a	SEQ3a	SEQ6a	SEQ7b	SEQ8a
Prüfer 1 extern Prüfer 2 intern	SEQ1a	SEQ2a	SEQ3b	SEQ6a	SEQ7b	SEQ8b

Tabelle 4.2: Konfigurationstabelle für Prozessbeispiel 3

anschaulicht. Zum Beispiel besitzt der XOR-Split „Externer Prüfer“ zwei Ports. Wenn der erste Prüfer extern ist, wird Port C1 aktiviert und C2 blockiert. Im anderen Fall wird C2 aktiviert und C1 blockiert. Genau dieselben Prinzipien gelten für den zweiten Prüfer. Der XOR-Split „zweiter Prüfer notwendig“ besitzt ebenfalls zwei Ports. Wenn der zweite Prüfer nicht benötigt wird, wird C4 aktiviert. Dann springt der Prozess von C4 zu C7. Sonst wird der erste Port ausgeführt, C4 blockiert und C3 aktiviert. Ein davon abhängige Variationspunkt ist, ob der zweite Prüfer extern oder intern ist. Wenn er extern ist, wird C5 aktiviert und C6 blockiert, sonst wird C6 aktiviert und C5 blockiert.

Im Anschluss müssen weitere Prozessteile an diese Angaben angepasst werden. Das bedeutet beispielsweise, dass der Prüfer, der die Abschlussarbeit betreut hat, sie auch begutachtet. Wie in der Abbildung veranschaulicht ist, werden die Ports C17 und C18 dementsprechend entweder aktiviert oder blockiert.

In diesem Prozess treten ähnliche Probleme wie bei Prozessbeispiel 2 auf. Insbesondere bietet C-YAWL keine durchgängige Möglichkeit, um die Abhängigkeiten zwischen den Ports zu realisieren. Hier sollte der Benutzer wieder durch einen entsprechenden Fragebogen unterstützt werden, um diese Abhängigkeiten zusätzlich realisieren. Die Abbildung 4.13 stellt den möglichen Fragebogen dar.

Abbildung 4.18 zeigt die Darstellung des Prozesses mit PESOA. Um die Prozessvarianten darstellen zu können, werden hier die Erweiterungseigenschaften und optionale Aktivitäten verwendet. Die Aktivität „Prüfer 1 bestätigen“ ist mit dem Stereotyp «Abstract» markiert. Ist der Prüfer extern, wird diese Aktivität erweitert und die Aktivität „Prüfer 1 schriftlich bestätigen“ ausgeführt. Die Aktivität „Prüfer 2 bestätigen“ wird zusätzlich als «Optional» markiert. Wenn der zweite Prüfer erforderlich ist, wird die Aktivität ausgeführt, ansonsten wird sie übersprungen. Falls Prüfer 2 extern ist, wird die Aktivität zusätzlich

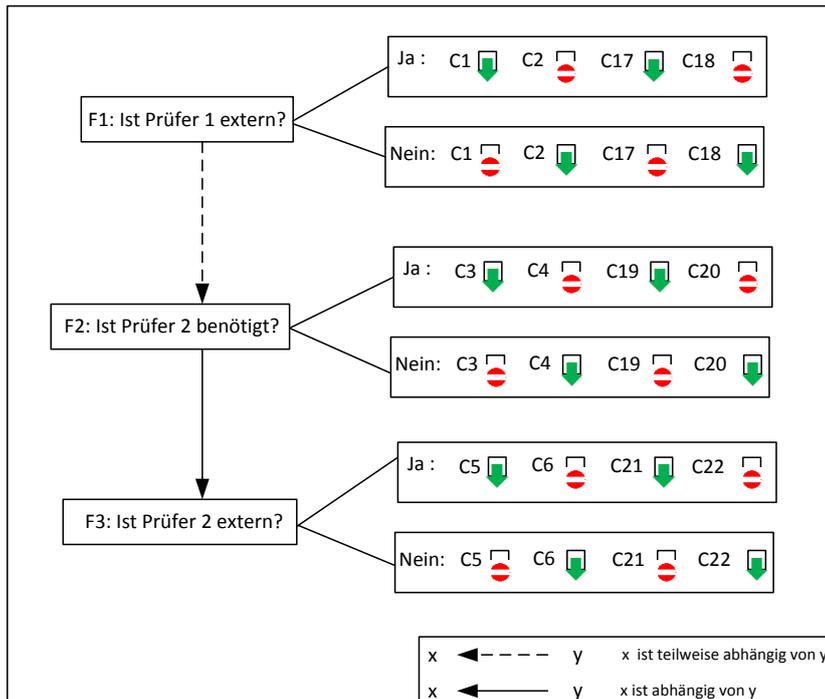


Abbildung 4.13: Fragebogen für Prozessbeispiel 3

erweitert und Aktivität „Prüfer 2 schriftlich bestätigen“ ausgeführt. Genau dieselben Prinzipien gelten beim Gutachten der Abschlussarbeit auch. Der Prüfer, der diese Arbeit betreut hat, führt auch das Gutachten aus. Die Aktivitäten „Gutachten ausführen“ und „2. Gutachten ausführen“ sind als «Abstract» markiert. Diese Aktivitäten werden abhängig davon erweitert, ob der Prüfer extern oder intern ist. PESOA bietet für diese Prozessvarianten eine gute Struktur. Insbesondere können unnötige XOR-Gateways eingespart werden. Die Entscheidungen werden mittels der Features und Erweiterungen realisiert. Optionale Aufgaben ermöglichen eine hohe Flexibilität.

In der Abbildung 4.19 wird die Modellierung des Prozesses mit dem Provop-Ansatz dargestellt. Als Basisprozess wurde die Variante gewählt, bei der ein interner Prüfer ausreichend ist. Weitere Prozessvarianten werden mittels Optionen dargestellt. Dazu gibt es sechs Optionen. Die Änderungen werden über die Aufsetzpunkte durchgeführt, die als A, B, C, D, E markiert sind. Je nach Kontextregel werden Aktivitäten entweder vom Prozess gelöscht oder zum Prozess hinzugefügt. Zum Beispiel, falls der Prüfer 1 extern ist, wird Option 1 angewendet und die Aktivität „Prüfer 1 bestätigen“ gelöscht

#### *4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse*

und stattdessen werden die Aktivitäten „Prüfer 1 schriftlich bestätigen“ und „Bestätigung erfassen“ hinzugefügt. Ebenfalls wird Aktivität „Gutachten erstellen, in CMS hochladen und Note eintragen“ gelöscht, stattdessen werden die Aktivitäten „Gutachten erstellen und schriftlich weiterreichen“ und „Bei externem Prüfer Note eintragen“ hinzugefügt. Die Optionsbedingungen zeigen, welche Optionen voneinander abhängig sind und welche nicht zusammen ausgeführt werden können. Beispielsweise sind hier die Option 2 und Option 3 als wechselseitiger Ausschluss markiert. Das heißt, wenn Option 2 ausgeführt wird, darf Option 3 nicht ausgeführt werden. Der Grund dafür sind die unterschiedlichen Kontextregeln. Genauer gesagt, dass Option 2 gilt, wenn der zweite Prüfer notwendig und intern ist, aber Option 3 gilt, wenn der zweite Prüfer notwendig und extern ist. Es ist unmöglich, beide zusammen auszuführen.

Mit den Kontextregeln und Optionsbedingungen lassen sich mit dem Provop-Ansatz die Prozessvarianten einfach und fehlerfrei abbilden. Provop ermöglicht mit seiner strukturierten Art eine gute Nachvollziehbarkeit und eine kompakte Darstellung.

### 4.3 Prozessbeispiel 3

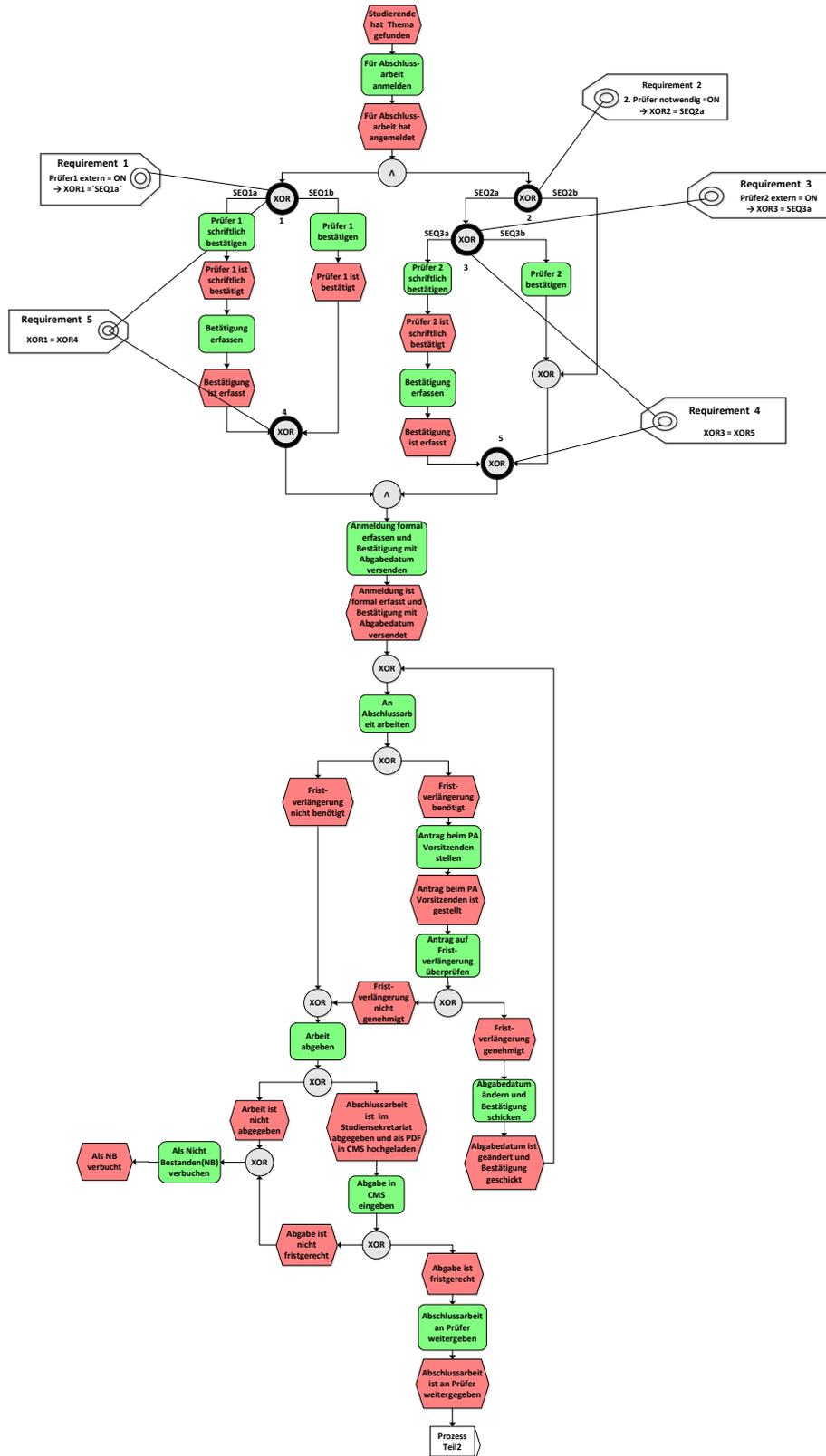


Abbildung 4.14: C-EPC Bachelor- und Masterarbeit erfassen Teil 1

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

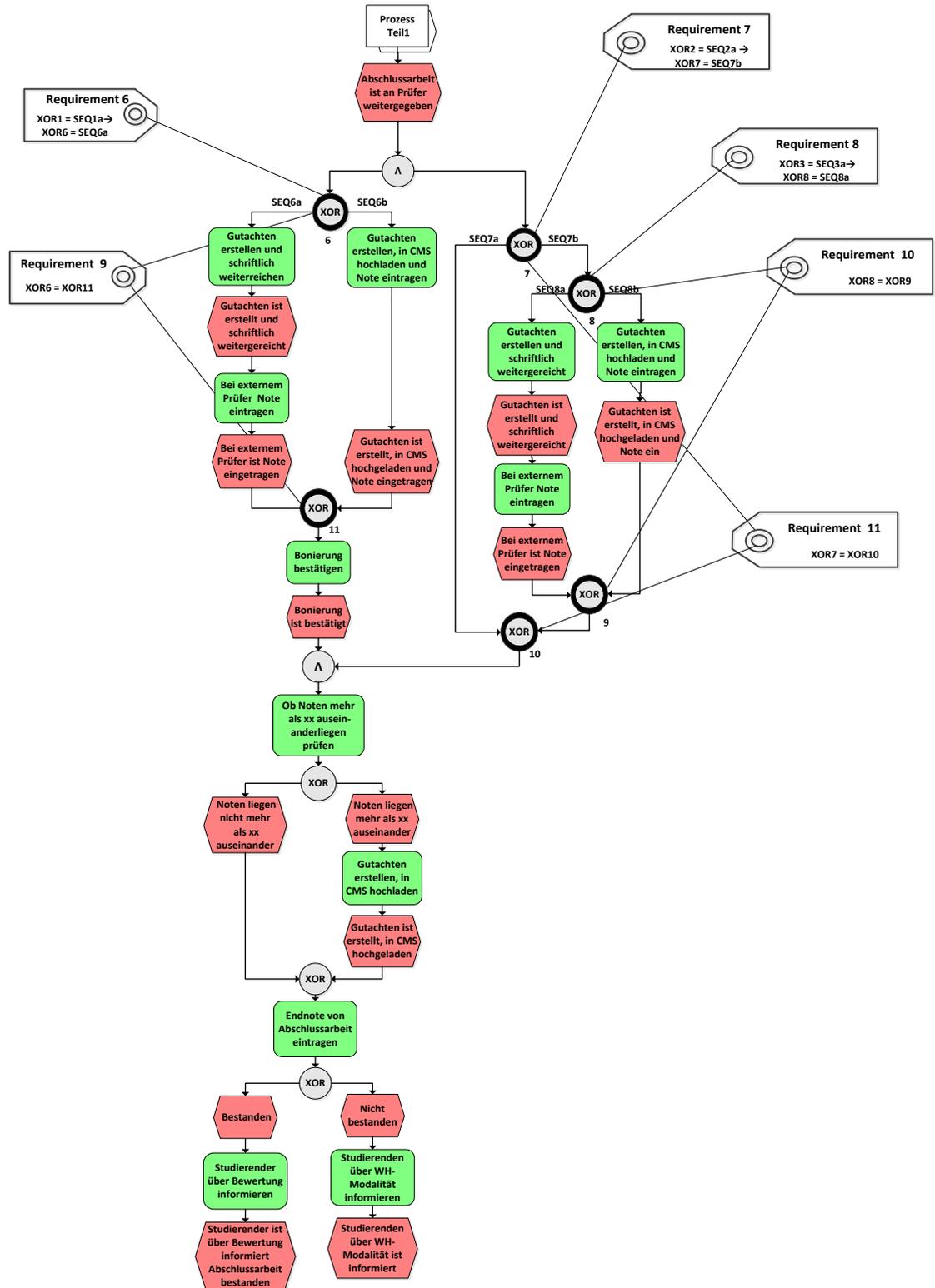


Abbildung 4.15: C-EPC Bachelor- und Masterarbeit erfassen Teil 2

### 4.3 Prozessbeispiel 3

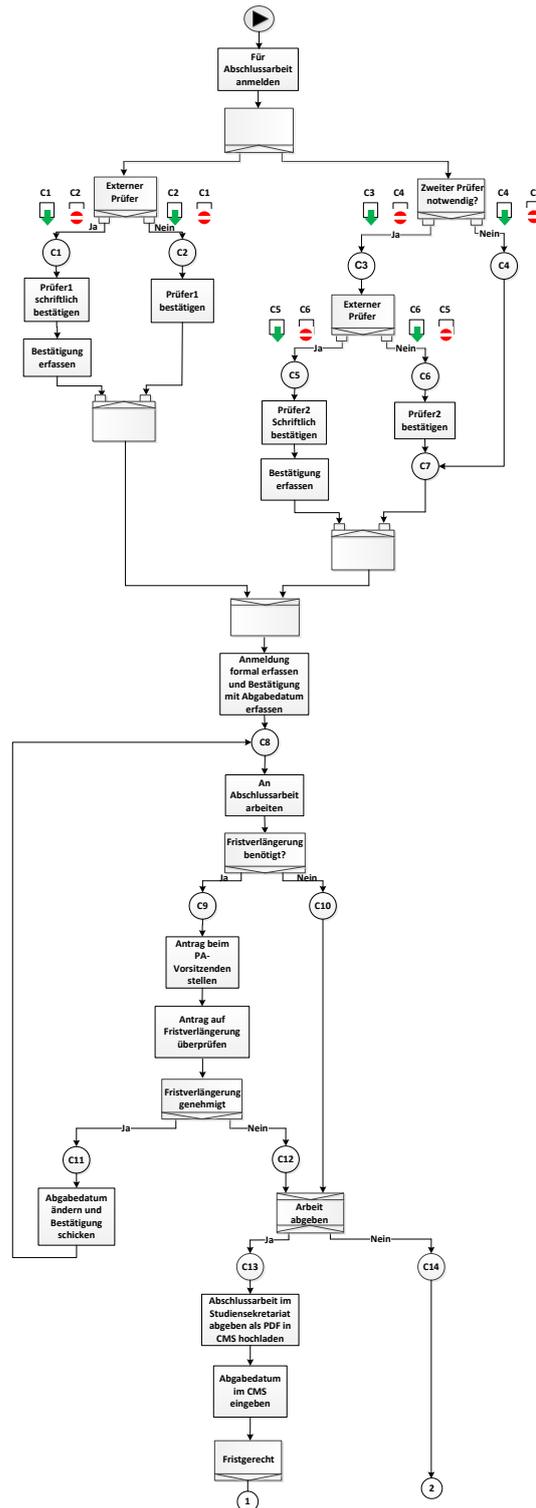


Abbildung 4.16: C-YAWL Bachelor- und Masterarbeit erfassen Teil 1

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

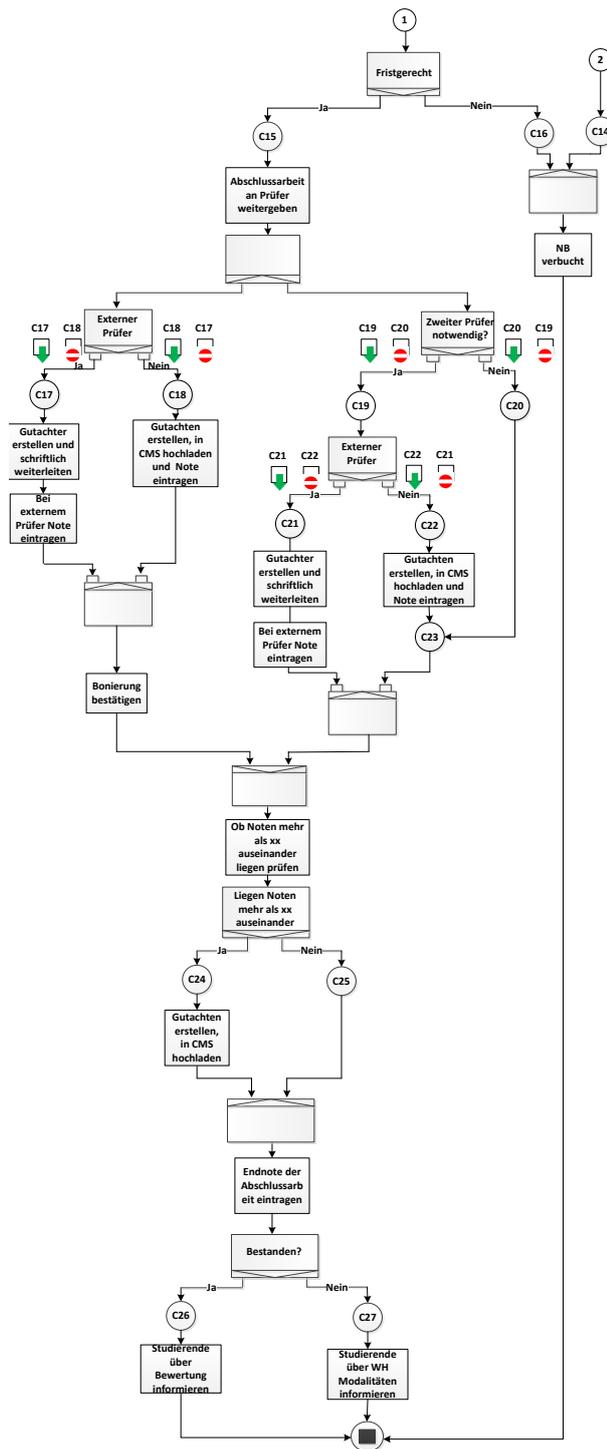


Abbildung 4.17: C-YAWL Bachelor- und Masterarbeit erfassen Teil 2

### 4.3 Prozessbeispiel 3

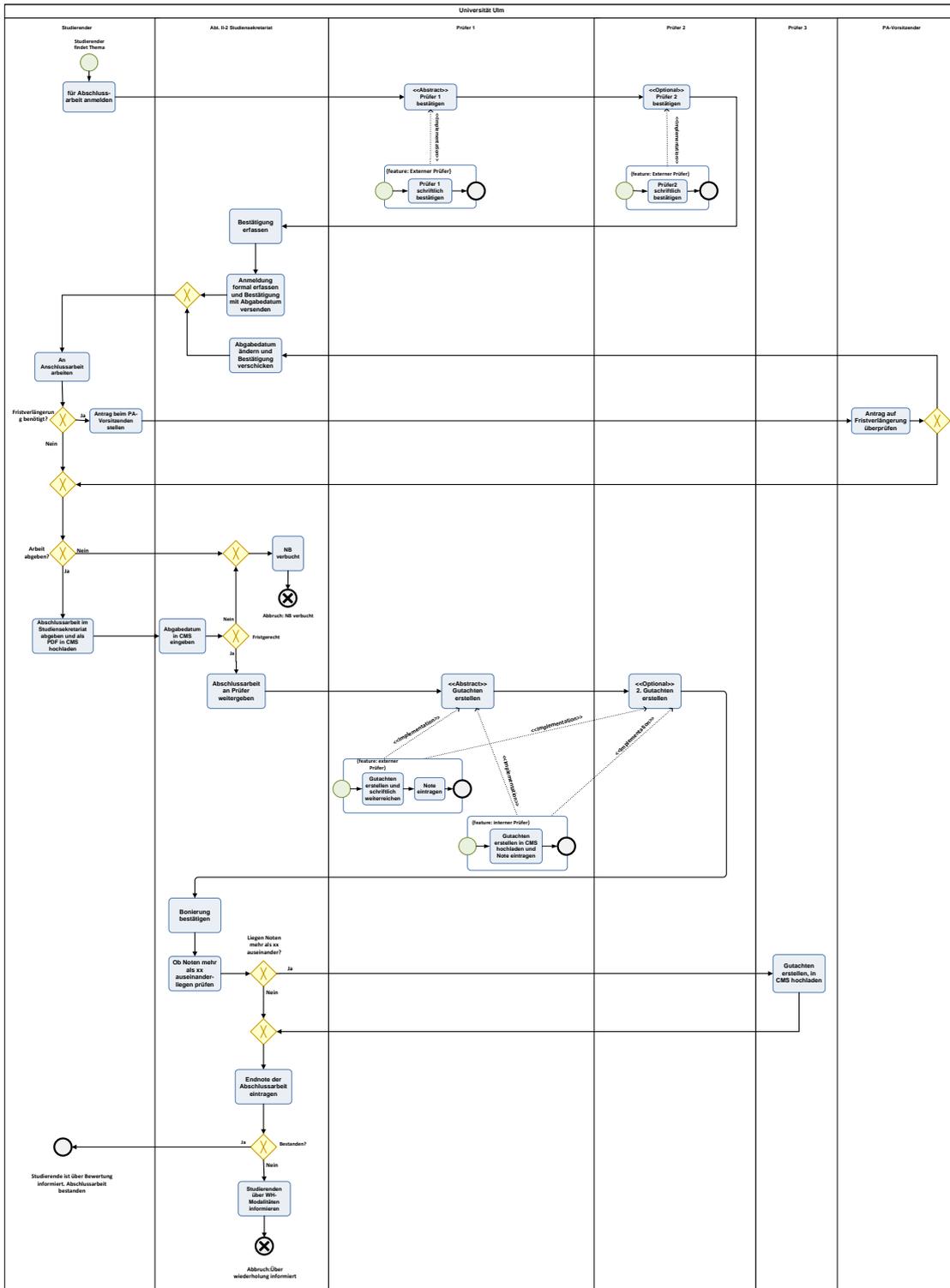


Abbildung 4.18: Bachelor- und Masterarbeit erfassen mit PESOA



## 4.4 Zusammenfassung der drei Prozessbeispiele

In diesem Kapitel wurden die Eigenschaften der verschiedenen Ansätze zur Modellierung variantenreiche Prozesse anhand von drei Beispielen untersucht. Die daraus resultierenden Ergebnisse werden nun in diesem Abschnitt zusammengefasst.

C-EPC ermöglicht es mit konfigurierbaren Konnektoren die Variationspunkte festzulegen und Prozessvariante zu modellieren. Dabei helfen die Requirements die Beschränkungen zwischen den Alternativen von konfigurierbaren Knoten zu beschreiben. Zusätzlich bieten Requirements die Möglichkeit Abhängigkeiten zwischen den Prozessteilen zu beschreiben. Aber je nach Prozessumfang werden die mittels C-EPC modellierten Prozesse schnell sehr groß. Beispielsweise besitzen Prozessbeispiel 2 und 3 sehr viele Elemente, Beispiel 3 musste zur sinnvoller Darstellung sogar in zwei Teile geteilt werden. In diesem Fall es ist für den Benutzer schwierig, den gesamten Prozess nachzuvollziehen. Deswegen wurde in diesen zwei Beispielen mit Konfigurationstabellen gearbeitet, damit die Benutzer unterstützt werden können. Ein weiteres Problem ist, dass C-EPC keine Unterstützung dafür bietet, um variable Rollenzuteilungen für Aufgaben zu realisieren. Ein Problem, welches beispielsweise Beispielprozess 1 aufgetaucht ist. Der für die Zeit- und Raumplanung zuständige Raumverwalter ist, ist hier ein Variationspunkt. Der Prozess muss somit entsprechend dem zuständigen Raumverwalter konfiguriert werden. Aber an dieser Stelle führen der zentrale Raumverwalter und der dezentrale Raumverwalter die gleichen Aufgaben aus. Dies kann mit C-EPC nicht adäquat abgebildet werden. Zusätzliche konfigurierbare Knoten in den Prozess einzufügen und gleiche Aufgaben als Variation zweimal zu zeigen macht hier wenig Sinn.

Ähnlichen Probleme treten auch bei C-YAWL auf. C-YAWL bietet keine Möglichkeit, die gleichen Aktivitäten für unterschiedliche Personen zu zuweisen. C-YAWL benötigt weniger Modellierungselemente als C-EPC, allerdings wird das Prozessmodell dennoch schnell groß. Ferner bietet C-YAWL keine Möglichkeit, um Abhängigkeiten zwischen Prozessteilen und Prozessvarianten darzustellen. Aus diesen Grund wurden bei den Prozessbeispielen 2 und 3 mit Fragebögen bearbeitet, umso die Abhängigkeiten zwischen den Fragen sowie den Prozessteilen darzustellen.

PESOA bietet verschiedene Möglichkeiten wie Vererbung und Erweiterung an, um

#### 4 Vergleich der Ansätze zur Modellierung variantenreicher Prozesse

Prozessvarianten zu modellieren. Die Variationspunkte sind mit Stereotypen markiert und je nach Kontext werden die möglichen Vererbung oder Erweiterungen von markierten Aufgaben ausgeführt. Kontextregeln werden als Features an die Aktivitäten bzw. Subprozesse angeheftet. Im Prozessbeispiel 1 sollten der zentrale und dezentrale Raumverwalter zusammengefasst und als Raumverwalter in einer Lane dargestellt werden. Wenn sich die Prozessteile, die bei der Variationsdarstellung notwendig sind, in mehreren Lanes befinden, ist es bei PESOA schwierig eine semantisch korrekte Modellierung zu schaffen. Mittels Features könnten teilweise die gleichen Aufgaben von unterschiedlichen Personen ausgeführt werden. Aber trotzdem ist es an manchen Stellen immer noch nicht ganz klar, wer welche Aufgabe ausführen soll. Darüber hinaus bietet PESOA keine Möglichkeit Konnektoren zu konfigurieren. Obwohl der Knoten „Freiversuch möglich“ in Prozessbeispiel 2 ein Variationspunkt ist, konnte diese Variation nicht dargestellt werden. Der Stereotyp «Optional » bietet eine flexible Darstellung. Beispielsweise konnte die Teilaufgabe „Vorleistung notwendig“ in Prozessbeispiel 2, die von einem Prüfer ausgeführt wird, in einen Subprozess gepackt und als «Optional» markiert werden. Diese Art der Modellierung bietet eine kompakte Darstellung, sie erhöht aber die Anzahl der Prozesse. Infolge dessen treten erhöhte Wartungskosten der Prozesse als Nachteil auf. Die Erweiterungs- und Vererbungseigenschaften ermöglichen es wiederum XOR-Konnektoren einzusparen, wie im Prozessbeispiel 3, veranschaulicht wurde. Ein Hauptproblem bei PESOA ist, aber dass es keine vorgegebene Regeln bietet, wie die Variationspunkte gestaltet werden sollen. Jeder Benutzer muss je nach seiner Vorstellung mit diesem Ansatz arbeiten. Führt in die Praxis dazu, dass keine einheitlichen Prozessdarstellungen erfolgt.

Anders als die vorherigen Ansätze bietet Provop eine Reihe von Änderungsoperationen. Mit diesen Operationen können die erforderlichen Änderungen im Basisprozess ausgeführt werden. Für jedes Beispiel wurde ein Basisprozess als eine mögliche Prozessvariante als Basisprozess ausgewählt. Die Verwendung der Kontextregeln, und der Abhängigkeiten zwischen den Optionen, helfen dabei eine saubere und fehlerfreie Darstellung der Prozessvarianten zu erreichen und ermöglichen eine bessere Nachvollziehbarkeit des Prozesses. Um die kontextbezogene Rollenzuweisung für eine Aktivität darzustellen, bietet Provop die Änderungsoperation MODIFY. Damit werden die Aufga-

#### 4.4 Zusammenfassung der drei Prozessbeispiele

ben je nach zuständiger Person markiert bzw. modifiziert. Dies ist in Prozessbeispiel 1 sichtbar. Im Gegensatz dazu ist es schwierig, mögliche Abbrüche des Prozesses darzustellen. Dies ist insbesondere dann der Fall dieser Abbruch innerhalb einer bestimmten Variante erfolgt. Beispielsweise ist dieses Problem bei Prozessbeispiel 2 sichtbar. Um hier einen Abbruch zu realisieren, mussten zusätzliche XOR-Konnektoren mittels Insert-Operation in den Prozess eingefügt werden. Es ist für den Benutzer aber nicht eindeutig ersichtlich, welche Bedeutung diese XOR-Knoten haben soll.

Zum Schluss kann gesagt werden, dass jeder Ansatz je nach dem Prozessumfang unterschiedliche Vor- und Nachteile hat. Deswegen muss jedes Unternehmen selbst entscheiden, welcher Ansatz seiner Anforderungen entspricht.



# 5

## Zusammenfassung

Im Rahmen dieser Arbeit wurden verschiedener Ansätze zur Modellierung variantenreicher Prozesse vergleichend betrachtet. Dieses Kapitel befasst sich mit den in dieser Arbeit behandelten Themen und stellt sie in einen Gesamtzusammenhang.

In der Praxis hat sich gezeigt, dass Geschäftsprozesse häufig, je nach dem Anwendungskontext, unterschiedliche Varianten besitzen. Mittels klassischer Modellierungssprachen wie EPC, BPMN oder YAWL können diese Prozessvarianten lediglich entweder in einem Single-Modell oder in einem Multi-Modell Ansatz dargestellt werden. Beiden Ansätze verursachen jedoch erhebliche Nachteile wie hohe Redundanz und Wartungskosten.

Um diese Nachteile zu überwinden und mögliche Prozessvarianten darzustellen, wurden in den letzten Jahren der verhaltensbasierte Ansatz und der strukturbasierte Ansatz vorgeschlagen. Der verhaltensbasierte Ansatz repräsentiert eine Obermenge der Prozessvarianten in einem Referenzmodell. Dieses Modell erfasst sowohl die Gemeinsamkeiten als auch die Unterschiede der möglichen Varianten. Im Gegensatz dazu verwendet der

## 5 Zusammenfassung

strukturbasierter Ansatz eine Basisvariante, sowie Änderungsoperationen wie Einfügen, Löschen oder Verschieben, um Prozessvarianten abzuleiten. C-EPC, C-YAWL und PESOA sind verhaltensbasierte Ansätze. Provop ist strukturbasiert.

C-EPC ist eine Erweiterung der Modellierungssprache EPC. Mit Hilfe konfigurierbare Funktionen und Konnektoren werden die möglichen Varianten realisiert. Requirements sind an konfigurierbare Knoten gebunden und stellen die Einschränkungen dar.

C-YAWL ist ebenfalls eine Spracherweiterung. Sie ist von der Modellierungssprache YAWL abgeleitet. Bei C-YAWL werden die Konfigurationen mittels der sogenannten Ports dargestellt. Durch das Verstecken und Blockieren dieser Ports werden die möglichen Prozessvarianten realisiert.

PESOA bietet verschiedene Möglichkeiten wie Vererbung, Erweiterung und Parametrisierung an, um Prozessvarianten zu modellieren. Variationspunkte werden mit Stereotypen markiert. Je nach Kontextregel werden diese Variationspunkte erweitert und Prozessvarianten realisiert.

Provop ist ein strukturbasierter Ansatz. Er bietet eine Reihe von Änderungsoperationen wie INSERT, DELETE, MODIFY und MOVE. Bei Provop wird zuerst einen Basisprozess festgestellt und mögliche Variationspunkte in diesem Prozess markiert. Optionen kombinieren verschiedene Änderungsoperationen je nach Kontext. Mittels dieser Optionen und den Variationspunkten werden die Prozessvarianten dargestellt.

In dieser Arbeit wurden diese Ansätze anhand von drei Beispielen miteinander verglichen und es wurde festgestellt, dass jeder Ansatz, je nach zu modellierendem Prozess, unterschiedliche Vor- und Nachteile besitzt. Obwohl C-EPC mittels konfigurierbaren Konnektoren und Requirements eine leichte Variantendarstellungen ermöglicht, bietet es keine Möglichkeit, die gleichen Aktivitäten kontextbezogene unterschiedlichen Personen zuzuweisen. Darüber hinaus werden Prozesse, die mittels C-EPC modelliert sind, schnell sehr groß. Die Requirements reichen nicht immer aus, um die Abhängigkeiten zwischen den Prozessteilen vollständig nachvollziehen zu können. In diesem Fall sollte mit Konfigurationstabellen gearbeitet werden. Ähnliche Probleme treten beim C-YAWL auf. C-YAWL bietet ebenfalls keine Möglichkeit, die kontextbezogenen Bearbeiterzuweisungen sowie Abhängigkeiten zwischen den Prozessteilen darzustellen. Um die Nachvollziehbarkeit zu erhöhen, sollte daher mit Fragebögen gearbeitet werden. Damit

können die Prozessvarianten anhand dieser Fragebögen modelliert werden. PESOA ermöglicht mit seinen Eigenschaften ein flexible und kompakte Darstellungen variantenreicher Prozesse. Es ist aber schwierig Lane übergreifende Prozessteilen in einer Prozessvariante zusammen darzustellen. Mittels Vererbungs- und Erweiterungseigenschaften können XOR-Knoten eingespart werden. Als Nachteil kann erwähnt werden, dass PESOA keine Möglichkeit anbietet Konnektoren zu konfigurieren. Es gibt auch keine eindeutigen Regeln, wie die von PESOA angebotenen Möglichkeiten verwendet werden können. Jeder Benutzer muss daher, je nach seiner Vorstellung, selbst entscheiden. Provop bietet eine saubere und kompakte Darstellung. Die Abhängigkeiten zwischen den Prozessteilen können mittels Optionen und Optionsbedingungen einfach dargestellt werden. Die Änderungsoperation MODIFY ermöglicht es eine kontextbezogene Bearbeiterzuordnung zu realisieren. Als ein Nachteil von Provop kann erwähnt werden, dass er schwierig sein kann Abbrüche darzustellen, die innerhalb eine bestimmte Variante auftreten.

Diese Arbeit hat gezeigt, dass jeder der untersuchten Ansätze unterschiedliche Vor- und Nachteile hat. Je nach Prozess und vorhandenen Anforderungen muss einer von diesen Ansätzen ausgewählt werden, um die Prozessvarianten zu modellieren. In Zukunft werden weitere Ansätze entwickelt, welche die in dieser Arbeit erwähnten Nachteile zu überwinden helfen.



# Abbildungsverzeichnis

2.1	EPC Basiselementen [HKS93] . . . . .	8
2.2	EPC Beispielprozess - Schadensdatenbearbeitung [Wie] . . . . .	10
2.3	Modellierungselemente von YAWL [tHvdA05] . . . . .	12
2.4	Kreditantragsprozess mit YAWL [YAWb] . . . . .	13
2.5	BPMN Basiselementen [FR10] . . . . .	15
2.6	BPMN Ereignisse [OMGa] . . . . .	16
2.7	BPMN Gateways [OMGa] . . . . .	16
2.8	BPMN Beispielprozess [OMGb] . . . . .	18
3.1	Konfigurationen einer Funktion [RvdA07] . . . . .	23
3.2	C-EPC Prozess vor und nach der Konfiguration . . . . .	25
3.3	Konfiguration durch Blockieren und Verstecken [YAWa] . . . . .	26
3.4	Beispielprozess mit C-YAWL [ATP11, WSR09] . . . . .	28
3.5	Beispielprozess mit PESOA [PSWW05] . . . . .	30
3.6	Symbole der Änderungsoperationen [Hal09] . . . . .	33
3.7	Beispielprozess mit Provop [Wor] . . . . .	35
4.1	Zeit- und Raumplanung . . . . .	39
4.2	Zeit- und Raumplanung mit C-EPC . . . . .	42
4.3	Zeit- und Raumplanung mit C-YAWL . . . . .	43
4.4	Zeit- und Raumplanung mit PESOA . . . . .	44
4.5	Zeit- und Raumplanung mit Provop . . . . .	45
4.6	Durchführung von Prüfungen . . . . .	47

## *Abbildungsverzeichnis*

4.7 Fragebogen für Prozessbeispiel2 . . . . .	49
4.8 Durchführung von Prüfungen mit C-EPC . . . . .	51
4.9 Durchführung von Prüfungen mit C-YAWL . . . . .	52
4.10 Durchführung von Prüfungen mit PESOA . . . . .	53
4.11 Durchführung von Prüfungen mit Provop . . . . .	54
4.12 Bachelor- und Masterarbeit erfassen . . . . .	56
4.13 Fragebogen für Prozessbeispiel 3 . . . . .	59
4.14 C-EPC Bachelor- und Masterarbeit erfassen Teil 1 . . . . .	61
4.15 C-EPC Bachelor- und Masterarbeit erfassen Teil 2 . . . . .	62
4.16 C-YAWL Bachelor- und Masterarbeit erfassen Teil 1 . . . . .	63
4.17 C-YAWL Bachelor- und Masterarbeit erfassen Teil 2 . . . . .	64
4.18 Bachelor- und Masterarbeit erfassen mit PESOA . . . . .	65
4.19 Bachelor- und Masterarbeit erfassen mit Provop . . . . .	66

# Tabellenverzeichnis

3.1	Beschränkungen für die Konfiguration der Konnektoren . . . . .	24
3.2	Konfigurationsoptionen für einen Konnektor mit n eingehenden/ausgehenden Pfaden [tHvdAAR09] . . . . .	27
4.1	Konfigurationstabelle für Prozessbeispiel 2 . . . . .	48
4.2	Konfigurationstabelle für Prozessbeispiel 3 . . . . .	58



# Literaturverzeichnis

- [All05] T. Allweyer. *Geschäftsprozessmanagement Strategie, Entwurf, Implementierung, Controlling*. Hanser-Verlag, 2005.
- [ATP11] C. Ayora, V. Torres, and V. Pelechano. BP Variability Case Studies Development using different Modeling Approaches. Tech.Rep. ProS-TR-2011-03, Universidad Politécnica de Valencia, 2011.
- [ATW<sup>+</sup>14] C. Ayora, V. Torres, B. Weber, M. Reichert, and V. Pelechano. VIVACE: A framework for the systematic evaluation of variability support in process-aware information systems, Information and Software Technology. *Information and Software Technology*, 57:248–276, Mai 2014.
- [Bau90] B. Baumgarten. *Petri-Netze: Grundlagen und Anwendungen*. BI-Wissenschafts-Verlag, 1990.
- [Com] BPMN Community. BPMN-Community Tutorial. <http://de.bpmn-community.org/tutorials/26/> (letzter Zugriff 13. 11. 2014).
- [FR10] J. Freund and B. Rücker. *Praxishandbuch BPMN2.0*. Hanser, 2010.
- [Gie98] O. Gierhake. *Integriertes Geschäftsprozessmanagement*. Vieweg-Verlag, 1998.
- [Hal09] A. Hallerbach. *Management von Prozessvarianten*. PhD thesis, Fakultät für Ingenieurwissenschaften und Informatik der Universität Ulm, 2009.
- [HBR08a] A. Hallerbach, T. Bauer, and M. Reichert. Anforderungen an die Modellierung und Ausführung von Prozessvarianten. *Datenbank Spektrum*, 24:48–58, February 2008.

## Literaturverzeichnis

- [HBR08b] A. Hallerbach, T. Bauer, and M. Reichert. Modellierung und Darstellung von Prozessvarianten in Provop. In *Modellierung'08 Conference*, S.41–56. Koellen-Verlag, March 2008.
- [HBR10] A. Hallerbach, T. Bauer, and M. Reichert. Configuration and Management of Process Variants. In *International Handbook on Business Process Management*. Springer, S.237–255, August 2010.
- [HC95] M Hammer and J. Champy. *Business Reengineering*. Campus-Verlag, 1995.
- [HKS93] W. Hoffmann, J. Kirsch, and A.-W. Scheer. *Modellierung mit Ereignisgesteuerten Prozessketten*. Veröffentlichungen des Instituts für Wirtschaftsinformatik-Heft 101, Universität des Saarlandes, Januar 1993.
- [HM08] W. Hesse and H. C. Mayr. Modellierung in der Softwaretechnik: eine Bestandsaufnahme. *Informatik-Spektrum*, 31(5):377–393, 2008.
- [HN09] F. Hoglebe and M. Nüttgens. Rahmenkonzept zur Messung und Bewertung der Gebrauchstauglichkeit von Modellierungssprachen. Arbeitsberichte zur Wirtschaftsinformatik Nr.7, Universität Hamburg, April 2009.
- [ISO] Prozessmanagement / Qualitätsmanagement DIN EN ISO 9000 / 9001:2008. [http://www.qmti.de/prozm/prozessmanagement\\_e.htm](http://www.qmti.de/prozm/prozessmanagement_e.htm) (letzter Zugriff 08. 12. 2014).
- [Kö12] O. Königs. *Open Source und Workflow im Unternehmen: Eine Untersuchung von Processmaker, Joget, Bonita Open Solution, uEngine und Activiti*. Diplomica Verlag, 2012.
- [KKS04] R. Klein, F. Kupsch, and A.-W. Scheer. *Modellierung interorganisationaler Prozesse mit Ereignisgesteuerten Prozessketten*. Veröffentlichungen des Instituts für Wirtschaftsinformatik-Heft 178, Universität des Saarlandes, November 2004.
- [KNS92] G. Keller, M. Nüttgens, and A.-W. Scheer. *Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)"*.

Veröffentlichungen des Instituts für Wirtschaftsinformatik-Heft 89, Universität des Saarlandes, Januar 1992.

- [LR09] M. La Rosa. *Managing Variability in Process-Aware Information Systems*. PhD thesis, Faculty of Science and Technology Queensland University of Technology Brisbane, Australia, 2009.
- [LRDtH08] M. La Rosa, M. Dumas, and A. ter Hofstede. Modelling business process variability. Technical Report 13358, Queensland University of Technology, 2008.
- [LRGDvdA08] M. La Rosa, F. Gottschalk, M. Dumas, and W. M. van der Aalst. Linking domain models and process models for reference model configuration. In *Business Process Management Workshops, 4928:417-430*. Springer, 2008.
- [NR02] M. Nüttgens and F. J. Rump. Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In *Promise, 2:64-77*, 2002.
- [OMGa] Business Process Model and Notation (BPMN) Version 2.0 Januar 2011. <http://www.omg.org/spec/BPMN/2.0/> (letzter Zugriff 02.12.2014).
- [OMGb] OMG. BPMN 2.0 by Example. <http://www.omg.org/spec/BPMN/20100601/10-06-02.pdf> (letzter Zugriff 26.10.2014).
- [Pet62] C. A. Petri. *Kommunikation mit Automaten*. PhD thesis, Technische Universität Darmstadt, 1962.
- [PSWW05] F. Puhlmann, A. Schnieders, J. Weiland, and M. Weske. Variability mechanisms for process models. PESOA-Report, TR 17, BMBF-Project, Hasso Plattner Institut, Postdam, Germany, 2005.
- [RvdA07] M. Rosemann and W. M. van der Aalst. A configurable reference modeling language. *Information Systems*, 32(1):1–23, 2007.
- [RW12] M. Reichert and B. Weber. *Enabling flexibility in process-aware information systems: challenges, methods, technologies*. Springer, 2012.

## Literaturverzeichnis

- [SM97] A.-W. Scheer and Nüttgens M. *Objektorientierte Ereignisgesteuerte Prozesskette (oEPK)- Methode und Anwendung*. Veröffentlichungen des Instituts für Wirtschaftsinformatik-Heft 141, Universität des Saarlandes, Mai 1997.
- [SS08] H. Schmelzer and W. Sesselmann. *Geschäftsprozessmanagement in der Praxis- Kunden zufrieden stellen, Produktivität steigern, Wert erhöhen*. Hanser-Verlag, 2008.
- [tHvdA05] A. H. ter Hofstede and W. M. van der Aalst. YAWL: yet another workflow language. *Information Systems*, 30(4):245–275, 2005.
- [tHvdAAR09] A. H. ter Hofstede, W. M. van der Aalst, M. Adams, and N. Russell. *Modern Business Process Automation: YAWL and its support environment*. Springer-Verlag, 2009.
- [tHvdAD05] A. H. ter Hofstede, W. M. van der Aalst, and M. Dumas. *Process-Aware Information Systems*. Wiley-Interscience, 2005.
- [TZW<sup>+</sup>12] V. Torres, S. Zugal, B. Weber, M. Reichert, C. Ayora, and V. Pelechano. A qualitative comparison of approaches supporting business process variability. In *3rd International Workshop on Reuse in Business Process Management*, 2012.
- [Whi04] S. A. White. *Introduction to BPMN*. IBM Cooperation 2.0, 2004.
- [Wie] M. Wiecek. Konzeption eines softwaregestützten Managements flexibler Prozesse des Versicherungswesens auf Basis von BPMN . <http://www.wieczekit.de/2012/09/konzeption-eines-softwaregestuetzten-managements-flexibler-prozesse-des-versicherungswesens-auf-basis-von-bpmn/> (letzter Zugriff 26. 10. 2014).
- [Wor] Managing Process Model Complexity via Abstract Syntax Modifications. <http://www.workflowpatterns.com/patterns/presentation/abstractsyntax/images/fig6.jpg> (letzter Zugriff 08. 12. 2014).

- [WSR09] B. Weber, S. Sadiq, and M. Reichert. Beyond rigidity – dynamic process lifecycle support. *Computer Science-Research and Development*, 23(2):47–65, 2009.
- [YAWa] YAWL Foundation. YAWL Foundation Downloads. <http://www.yawlfoundation.org/yawlbook/slides/chapter18.ppt> (letzter Zugriff 04. 11. 2014).
- [YAWb] YAWL Foundation. YAWL Tutorial. <http://www.yawlfoundation.org/pages/resources/creditcardexample.html> (letzter Zugriff 26. 10. 2014).

Name: Gözde Ipek Bayrak

Matrikelnummer: 726218

### **Erklärung**

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den .....

Gözde Ipek Bayrak