



ulm university

universität  
**uulm**

# Imperative vs. Deklarative Prozessmodellierung - Eine kritische Betrachtung

Masterarbeit an der Universität Ulm  
Fakultät für Ingenieurwissenschaften und Informatik  
Institut für Datenbanken und Informationssysteme

**Vorgelegt von:**

Julia Schwarz  
julia.schwarz@uni-ulm.de

**Betreuer:**

Andreas Lanz, Universität Ulm

**Gutachter:**

Prof. Dr. Manfred Reichert, Universität Ulm  
Dr. Barbara Weber, Universität Innsbruck

**Vorgelegt am:**

27.11.2014

## **Abstract**

Sowohl die Abbildung von Geschäftsprozessen in Form von Prozessmodellen, als auch die damit verbundene Verwaltung der Prozessmodelle mit Hilfe von Business Process Management, ist in den letzten Jahren zu einem wichtigen Forschungsgebiet geworden. Darüber hinaus etabliert sich dieses Thema zusehends in der Industrie. Für die Abbildung von Geschäftsprozessen existieren zwei gegensätzliche Paradigmen: die imperative und die, im Gegensatz dazu relativ junge, deklarative Prozessmodellabbildung. Hier stellt sich die Frage, in wie weit sich diese beiden Abbildungsparadigmen unterscheiden bzw. welches Abbildungsparadigma für welche Art von Geschäftsprozess besser geeignet ist. Ziel dieser Arbeit ist daher der kritische Vergleich dieser beiden Prozessmodellierungsparadigmen anhand realer Prozessbeispiele. Hierzu werden bei der Migration der Prozessmodelle von einem imperativen in ein deklaratives Schema verschiedene Aspekte aufgezeigt, die mit der deklarativen Prozessmodellierung nicht bzw. nur unzureichend erfasst werden können. Anhand der untersuchten Prozessmodelle werden anschließend die Grenzen der Abbildbarkeit durch die beiden Prozessmodellparadigmen aufgezeigt und Lösungsstrategien für die Realisierung problematischer Aspekte entwickelt. Nachfolgend werden die entstandenen Prozessmodelle hinsichtlich Vollständigkeit, Korrektheit, Verständlichkeit, Granularität sowie hinsichtlich deren Flexibilität bewertet und verglichen.

.

## Inhaltsverzeichnis

Abstract .....	II
Inhaltsverzeichnis.....	iii
1. Einleitung.....	5
1.1 Motivation.....	6
1.2 Vorgehensweise der Arbeit.....	6
2. Theoretische Grundlagen.....	8
2.1 Einführung relevanter Grundbegriffe .....	8
2.2 Imperative Prozessmodellierung .....	9
2.3 Deklarative Prozessmodellierung.....	13
2.4 Unterschiede zwischen imperativer und deklarativer Prozessmodellierung .....	17
3. Deklarative Prozessmodellierung als Forschungsgebiet .....	19
3.1 Ursprung der deklarativen Prozessmodellierung.....	19
3.2 Deklarative Prozessmodellierungssprachen .....	20
3.3 Erweiterungen.....	21
3.4 Untersuchungen zur Verbesserung deklarativer Abbildung.....	22
3.5 Zusammenfassung.....	24
4. Test Driven Modelling Suite als praktische Grundlage .....	25
5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen .....	29
5.1. Möglichkeiten bei der Abbildung von Prozessen als deklaratives Prozessmodell.....	30
5.1.1 Anzahl der Ausführungseinheiten von Aktivitäten .....	30
5.1.2 Flexible Reihenfolge von Aktivitäten.....	31
5.1.3 Granularität .....	33
5.1.4 Zusammenfassung.....	35
5.2. Schwierigkeiten bei der Abbildung von Prozessen als deklaratives Prozessmodell und mögliche Lösungswege .....	35
5.2.1 Sequenzflüsse.....	36
5.2.2 Kontextinformationen.....	38
5.2.3 Zeitliche Aspekte: Die Frist.....	42
5.2.4 Start- und Endereignisse .....	45
5.2.5 Gateways.....	55
5.2.6 Subprozesse .....	61
5.2.7 Zusammenfassung.....	63

## Inhaltsverzeichnis

5.3 Ergebnisse und Vorgehensweise der Abbildung imperativer Prozessmodelle in das deklarative Schema anhand dreier Beispielprozesse.....	64
5.3.1 Beispiel 1 .....	64
5.3.2 Beispiel 2 .....	67
5.3.3 Beispiel 3 .....	71
5.3.4 Zusammenfassung.....	74
5.4. Bewertung.....	75
6. Kritische Betrachtung der umgesetzten Prozessmodelle .....	77
6.1 Diskussion.....	77
6.1.1 Verständlichkeit .....	77
6.1.2 Vollständigkeit.....	81
6.1.3 Korrektheit .....	82
6.1.4 Granularität .....	82
6.1.5 Flexibilität .....	83
6.2 Gesamtbewertung.....	84
7. Zusammenfassung und Ausblick .....	88
7.1 Fazit .....	88
7.2 Limitationen .....	89
7.3 Ausblick .....	90
Abbildungsverzeichnis.....	91
Tabellenverzeichnis.....	92
Abkürzungsverzeichnis.....	93
Literaturverzeichnis.....	94
Anhang .....	97
Erklärung .....	115

## 1. Einleitung

Seit den 90er Jahren nimmt der Einsatz von computergestützten Informationssystemen zur Darstellung innerbetrieblicher Prozessabläufe kontinuierlich zu. Unternehmen weltweit sind bestrebt, durch die Abbildung ihrer Geschäftsprozesse deren Transparenz, Flexibilität, Effizienz und Qualität zu verbessern sowie Kosten zu reduzieren [ZPW11]. Aufgrund der starken globalen Vernetzung der Wirtschaft und dem technischen Fortschritt im Bereich der Informationstechnologie sind viele Unternehmen gezwungen, möglichst flexibel zu agieren und auf Veränderungen zu reagieren. Die Schnellebigkeit und die damit verbundenen stetig wechselnden Anforderungen, erfordern die kontinuierliche Anpassung und Überarbeitung von Geschäftsprozessen [ZPW11].

Um diesen Anforderungen gerecht zu werden, hat sich in den meisten Unternehmen der Einsatz von Business Process Management (BPM) etabliert. Der Begriff umschreibt die Verwaltung der Prozesse, die innerhalb eines Unternehmens ablaufen, erfasst diese und bildet sie mit Hilfe verschiedener Modellierungsarten sowie geeigneter Tools ab. Laut [Wes07] umfasst BPM sowohl Konzepte als auch Methoden und Techniken, die das Design, die Administration, die Konfiguration sowie die Übernahme und die Analyse von Geschäftsprozessen unterstützen.

Aus der Notwendigkeit Geschäftsprozesse zu erfassen, ergibt sich ein Bedarf an geeigneten Techniken und Werkzeugen für die Identifikation, Abbildung, Analyse und Simulation von Prozessen [ADO00]. Dafür wurden verschiedene Arten von BPM-Modellierungssprachen entwickelt. Diese kann man generell in zwei verschiedene Paradigmen unterteilen: der imperativen und deklarativen Prozessmodellabbildung. Ein erstes Annehmen des Themas erfolgte in der Entwicklung des imperativen Modellierungsparadigmas. Da diese einen sehr starren und exakten Ablauf des Prozesses vorgibt, wurde später die deklarative Abbildungsform entwickelt, um eine freiere und variabelere Abbildung der Prozesse zu ermöglichen.

Die deklarative Art der Modellierung ist eine noch relativ junge Methode der Prozessmodellabbildung, die sich stetig weiter entwickelt. Aufgrund des Mangels an wissenschaftli-

## 1. Einleitung

cher Literatur zur vergleichenden Analyse der beiden vorgestellten Abbildungsformen soll diese Arbeit einen Beitrag zu diesem Forschungsfeld leisten.

### **1.1 Motivation**

Vor Beginn dieser Arbeit wurden in einem Teamprojekt die Prozesse des Hochschuldiensportals der Universität Ulm in Form imperativer Prozessmodelle abgebildet. Diese sollen als Vorlage für die Implementierung eines neuen prozessorientierten Hochschuldiensportals dienen. Es war jedoch nicht bei allen Prozessen möglich, diese imperativ korrekt darzustellen. Es entstanden Schwierigkeiten bei dem Versuch, diese analog zum Ablauf der realen Prozesse abzubilden. Daraus resultiert die Fragestellung, inwieweit diese Prozesse in deklarativer Form besser abgebildet werden können. Auch die Frage, ob eine Migration der bereits imperativ korrekt modellierten Prozessmodelle in das deklarative Schema möglich ist war relevant. Bezogen auf die Möglichkeit der imperativen und deklarativen Prozessmodellabbildung stellt sich nun allgemein die Frage, welche Art der Abbildung für die betrachteten Prozesse besser geeignet ist und ob hier eindeutige Kriterien festgestellt werden können. Diese Arbeit soll einen Beitrag zur Beantwortung dieser Forschungsfrage leisten.

Das Forschungsfeld der vergleichenden Analyse der imperativen und deklarativen Prozessmodellabbildung ist von einem Mangel an wissenschaftlicher Literatur gekennzeichnet. Dieser Mangel an Literatur macht weitere wissenschaftliche Beiträge erforderlich. Diese Arbeit leistet einen Beitrag zur Behebung des Mangels an Literatur und zur weiteren Erschließung dieses Forschungsgebietes. Hierbei liegt der Fokus auf dem Vergleich der imperativen und deklarativen Prozessmodellabbildung anhand einer Reihe vom imperativen in das deklarative Schema migrierter Prozessmodelle.

### **1.2 Vorgehensweise der Arbeit**

Inhalt dieser Arbeit ist ein analysierender Vergleich der imperativen und deklarativen Prozessmodellabbildung anhand einer repräsentativen Reihe von Prozessen. Es folgt ein Vergleich der Möglichkeiten und Schwierigkeiten bei der Umsetzung imperativer Prozessmo-

## 1. Einleitung

delle in das deklarative Schema sowie eine anschließende Diskussion und Handlungsempfehlung.

Zunächst wird in Kapitel 2 durch theoretische Grundlagen, wie die Definition der imperativen und deklarativen Prozessmodellabbildung und einer kurzen Ausführung der theoretischen Unterschiede, die Voraussetzung für das Verständnis des Themas geschaffen. Anschließend wird in Kapitel 3 der derzeitige Forschungsstand bezüglich deklarativer Prozessmodellierung dargestellt. Kapitel 4 gibt einen Überblick über die für diese Arbeit notwendigen praktischen Grundlagen zur Modellierung der Prozesse mit Hilfe des deklarativen Modellierungstools "Test Driven Modelling Suite" [TDMS]. Die Basis für Kapitel 5 bildet eine repräsentative Auswahl von Prozessmodellen des Hochschuldienstportals, die in imperativer Form vorliegen und die in das deklarative Schema migriert werden sollen. Hierbei werden die Möglichkeiten und Schwierigkeiten bei der Umsetzung der Prozessmodelle sowie potentielle Lösungswege zur Bewältigung vorhandener Schwierigkeiten erläutert. Anschließend wird anhand dreier beispielhaft ausgewählter Prozesse die Vorgehensweise bei der Migration der Prozesse vom imperativen in das deklarative Schema und die daraus resultierenden Ergebnisse beschrieben. Zu jedem Modell wird eine Empfehlung abgegeben, die begründet, welches Abbildungsschema für welches Modell am besten geeignet scheint. Mit Hilfe der gewonnenen Ergebnisse wird in Kapitel 6 ein abschließender Vergleich der beiden Abbildungsarten durchgeführt. Hier wird auf die Grenzen und Probleme bzw. Möglichkeiten in Bezug auf Verständlichkeit, Vollständigkeit der Abbildung, Korrektheit, die Verwendung der Sprachkonstrukte sowie die Dynamik und Flexibilität der Prozessmodelle eingegangen. In Kapitel 7 wird eine Gesamtbewertung mit Handlungsempfehlung für die in dieser Arbeit betrachteten Prozesse abgegeben. Das letzte Kapitel stellt die Limitationen bezogen auf diese Arbeit dar und gibt einen kurzen Ausblick zur Weiterentwicklung des Themas.

## 2. Theoretische Grundlagen

Das folgende Kapitel bildet die theoretische Basis dieser Arbeit, definiert relevante Begriffe und ordnet diese in den Gesamtkontext ein.

Zunächst werden die Begriffe Geschäftsprozess, Prozessmodellierung und Workflow definiert. Diese Termini werden in den Kontext des BPM eingeordnet. Anschließend werden relevante Methoden vorgestellt, mit welchen Prozesse grafisch dargestellt und modelliert werden können. Danach werden die Ansätze der imperativen und deklarativen Modellierung bzw. Prozessmodellabbildung sowie deren generelle Vor- und Nachteile erläutert. Zusätzlich werden die für diese Arbeit verwendeten Notationselemente der imperativen und deklarativen Prozessmodellabbildung eingeführt. Abschließend folgt eine Gegenüberstellung der theoretisch erarbeiteten Unterschiede und Gemeinsamkeiten imperativer und deklarativer Prozessmodellabbildung.

### 2.1 Einführung relevanter Grundbegriffe

Ein Geschäftsprozess oder Business Process ist laut Hammer “a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer” [Ham90]. Somit definiert sich ein Geschäftsprozess im Gegensatz zu einem herkömmlichen Prozess als ein Prozess, der innerhalb eines Unternehmens abläuft. Dieser besteht aus Einzelaktionen, wie zum Beispiel den Tätigkeiten der Mitarbeiter, die logisch in einer Sequenz und nach bestimmten Regeln ausgeführt werden, um ein gewünschtes Ziel zu erreichen. Diese Geschäftsprozesse können beispielsweise in Form von Flussdiagrammen abgebildet werden [ADO00].

BPM umfasst Methoden, Techniken und Tools, die das Design, die Abbildung, Entwicklung und Analyse sowie das Management operativer Geschäftsprozesse unterstützen [AHW03]. Geschäftsprozesse, die Prozesse im Umfeld eines Unternehmens darstellen, können mit Hilfe von BPM verwaltet werden [AHW03].



## 2. Theoretische Grundlagen

Ein Workflow ist eine definierte Kette von Aktivitäten, die durch ein Start- und ein Endereignis begrenzt sind [Brü13]. Bezogen auf einen Geschäftsprozess sind dies z.B. bestimmte Arbeitsschritte, die hintereinander ausgeführt werden. Workflow-Management bezeichnet das Modellieren, die Steuerung, die Spezifikation, die Ausführung und die Simulation der Workflows [Brü13]. Ein Workflow-Management-System (WFMS) ist ein System, das die einzelnen Aktivitäten des Workflow-Managements koordiniert [Brü13].

Die Prozessmodellierung hat das Ziel Prozesse grafisch darzustellen [AHW03]. Es existieren verschiedene Methoden, um Prozesse und Workflows abzubilden. Diese unterscheiden sich bezüglich ihres Aufbaus, der verwendeten Prozessmodellierungssprache, Benutzerfreundlichkeit, dem Zweck bzw. Ziel der Modellierung und den nötigen Vorkenntnissen des Modellierers [ADO00, Agu04]. Eine Prozessmodellierungssprache bietet hierzu Konzepte zur grafischen Abbildung und Darstellung von allen möglichen Arten von Prozessen [FMRWWZ10]. Modellierungssprachen werden meist in Abhängigkeit vom Entwicklungsstadium des zugrunde liegenden Modells verwendet [ADO00]. Diese untergliedern sich, wie bereits in Kapitel 1 angemerkt, in die Ansätze der imperativen und deklarativen Prozessmodellabbildung, die im folgenden beschrieben werden.

### 2.2 Imperative Prozessmodellierung

Imperative Modelle richten sich nach dem "outside-to-inside" Ansatz [FMRWWZ10]: Alle möglichen Ausführungsvarianten vom Start- zum Zielpunkt werden explizit angegeben und jede gewünschte neue Alternative muss ausdrücklich hinzugefügt werden. Somit "spezifizieren [die gegebenen] Bedingungen implizit Ausführungsalternativen" [Lic12] und es sind nur die vorgegebenen Varianten und keine Abweichungen möglich.

Verdeutlicht wird dies in Abbildung 1: Das Sechseck repräsentiert ein in sich abgeschlossenes Prozessmodell mit allen möglichen Ausführungsvarianten und Aktivitäten. Die weiße Fläche innerhalb des Sechsecks repräsentiert die sog. "verbotene Zone". Der Ablauf des Prozessmodells erfolgt immer von links nach rechts, also vom Start- zum Endpunkt des Prozesses. Die grauen Linien, die sich innerhalb des Sechsecks befinden, repräsentieren dabei die explizit modellierten Lösungspfade. Folglich kann nur einer der bereits hinzugefügten Pfade ge-

## 2. Theoretische Grundlagen

wählt werden. Soll der in der Abbildung rot dargestellte Pfad, der bisher nicht definiert wurde, wählbar sein, muss dieser zuerst explizit hinzugefügt werden [Igl11].

Imperative Prozessmodelle bilden somit Prozesse ab, die eine beschränkte Anzahl an Ausführungsvarianten erlauben. Möchte man weitere Möglichkeiten realisieren, müssen diese explizit hinzugefügt und zusätzlich modelliert werden. Ganz nach der Maßgabe "Was nicht explizit erlaubt ist, ist verboten." [Igl11] sind hier ausschließlich die vom Modellierer vorgesehenen Pfade bei der Ausführung erlaubt. Hierbei wird besonderer Wert auf die Beziehung zwischen vorangegangener und nachfolgender Aktivität und den dazwischen ablaufenden Kontrollfluss gelegt [PWZPMR12].

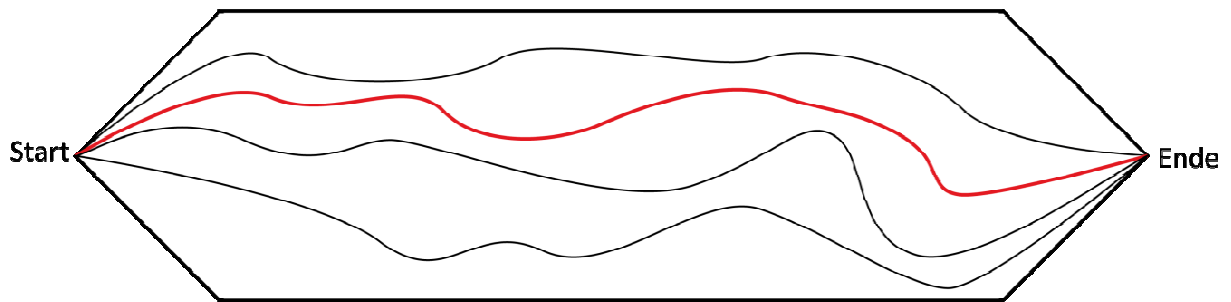





Abbildung 1: Darstellung von expliziten Lösungswegen bei der imperativen Prozessmodellierung

In Abbildung 2 ist eine Einführung in die wichtigsten BPM-Notationselemente dargestellt, die in dieser Arbeit verwendet werden. Als Grundlage der Prozesse dienen Aktivitäten. An diese Aktivitäten können Unterprozesse (zugeklappt oder aufgeklappt) angehängt werden. Konnektoren sind Pfade, die den Sequenzfluss eines Prozesses darstellen. Sie verbinden einzelne Elemente, wie z.B. Aktivitäten miteinander. Gateways dienen dazu, die Sequenzflüsse zwischen den einzelnen Aktivitäten aufzuspalten oder diese wieder zusammenzuführen. Dazu zählen z.B. datenbasierte exklusive, ereignisbasierte, parallele, inklusive und komplexe Gateways. Den Beginn eines imperativen Prozesses bildet ein Starterereignis (z.B. Nachrichten-, Zeit-, Eskalations-, Bedingungs-Fehler-, Kompensations-Starterereignis). Zwischenereignisse werden in eintretende (z.B. Nachrichtenzwischenereignis) und auslösende Zwischenereignisse (z.B. Zeitzwischenereignis) unterschieden. Endereignisse (z.B. Nachrichten- Eskalations-,

## 2. Theoretische Grundlagen

Fehler-, Abbruch- und Kompensations-Endereignisse) bilden den Abschluss eines imperativen Prozesses.



### Aktivitäten

-  Task
-  Zugeklappter Unterprozess
-  Aufgeklappter Unterprozess

### Rollen

-  Pool/Lane
-  Zugeklappter Pool
-  Zusätzlicher Beteiligter

### Artefakte

-  Textanmerkung
-  IT-System

### Konnektoren

-  Sequenzfluss
-  Ungerichtete Assoziation
-  Gerichtete Assoziation
-  Beidseitige Assoziation
-  Nachrichtenfluss








### Gateways

-  Datenbasiertes exklusives Gateway
-  Ereignisbasiertes Gateway
-  Paralleles Gateway
-  Inklusives Gateway
-  Komplexes Gateway



### Datenobjekte

-  Datenobjekt
-  Datenspeicher



### Startereignisse

-  Startereignis
-  Nachrichten-Startereignis
-  Zeit-Startereignis
-  Eskalations-Startereignis
-  Bedingungs-Startereignis
-  Fehler-Startereignis
-  Kompensations-Startereignis

### Zwischenereignisse (eintretend)

-  Nachrichten-Zwischenereignis
-  Zeit-Zwischenereignis

### Zwischenereignisse (auslösend)

-  Zwischenereignis
-  Nachrichten-Zwischenereignis

### Endereignisse

-  Endereignis
-  Nachrichten-Endereignis
-  Eskalations-Endereignis
-  Fehler-Endereignis
-  Abbruch-Endereignis
-  Kompensations-Endereignis

### Artefakte



-  Textanmerkung
-  IT-System

Abbildung 2: imperative Notationselement

## 2. Theoretische Grundlagen

Es können auch Notationselemente integriert werden, die Zusatzinformationen bereit stellen. Hierzu zählen z.B. Artefakte oder Datenobjekte. Diese Elemente werden mit Hilfe von gerichteten oder ungerichteten Assoziationen an die jeweilige Aktivität angeheftet. Durch die Verwendung von Pools und Lanes kann der Prozess zusätzlich strukturiert und untergliedert werden, indem den Aktivitäten Rollen zugewiesen werden.

Im oberen Teil von Abbildung 3 ist beispielhaft ein imperativer Prozess abgebildet. In diesem Prozess sind die drei Aktivitäten A, B und C dargestellt. Die Aktivitäten sind durch Sequenzflusspfeile miteinander zu einem Prozess verbunden. Der einzige Pfad, der von dem Prozess unterstützt wird, ist  $\langle A, B, C \rangle$ .

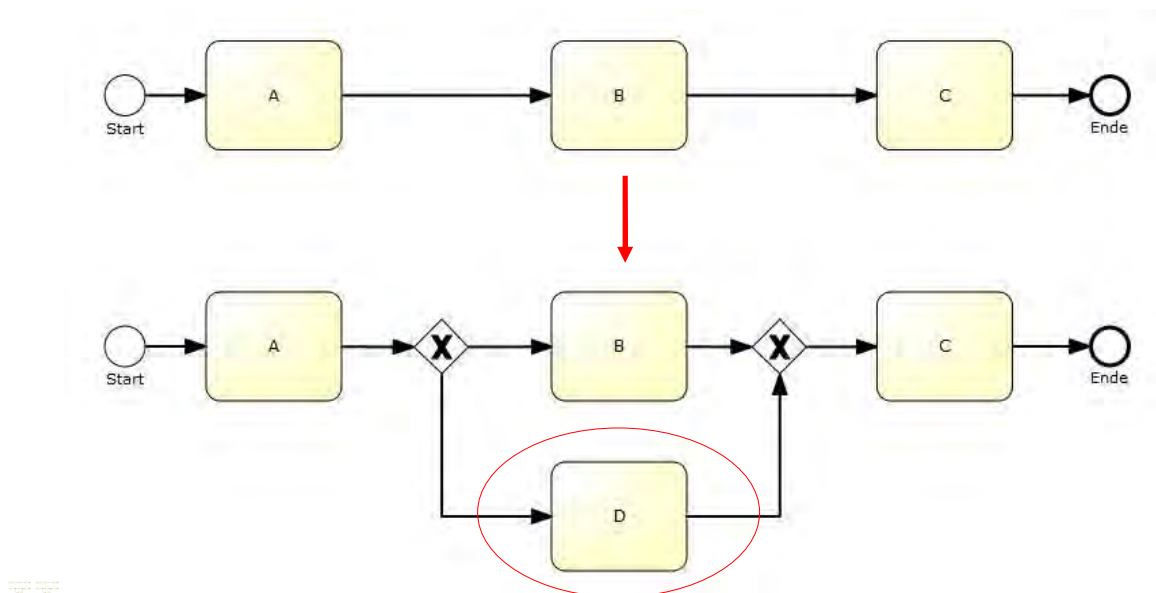


Abbildung 3: imperativer Beispielprozess

Analog zu dem bereits beschriebenen Ansatz der imperativen Prozessmodellabbildung muss jeder gewünschte zusätzliche Pfad oder jede Änderung des Prozessablaufs explizit hinzugefügt werden. Soll etwa der bereits bestehende Prozess (siehe Abbildung 3 oben), der aus den Aktivitäten A, B und C besteht, z.B. um einen neuen Pfad mit der Aktivität D erweitert werden, muss dieser explizit hinzugefügt werden (siehe Abbildung 3 unten). In diesem Fall wird das Hinzufügen des neuen Pfades mit dem datenbasierten exklusiven Gateway realisiert. Das datenbasierte exklusive Gateway stellt eine Verzweigung des Typs "entweder-oder" dar. Es

## 2. Theoretische Grundlagen

kann nach Ausführen der Aktivität A entweder der Pfad mit der Aktivität B oder der Pfad mit der Aktivität D gewählt werden. Der neue Prozess beginnt nun mit der Aktivität A; danach kann einer der Pfade mit Aktivität B oder D folgen. Der Prozess wird mit C beendet. Es wird in diesem Prozess die Ausführung der Pfade  $\langle A, B, C \rangle$  und  $\langle A, D, C \rangle$  unterstützt.

### 2.3 Deklarative Prozessmodellierung

Durch das häufig festgestellte Problem der Überspezifizierung und der zu starken Einschränkung der Ausführungsvarianten bei imperativen Prozessmodellen entwickelte sich die deklarative Prozessmodellierung [Igl11]. Diese verfolgt den "inside-to-outside-Ansatz" [FMRWWZ10]. Hier ist der Hauptaspekt die Beschreibung der Abhängigkeiten zwischen den Aktivitäten und der Einschränkungen der einzelnen Aktivitäten untereinander. Der temporale Aspekt wird dabei weitestgehend außer Acht gelassen [Jab14]. Das Prozessmodell beinhaltet zu Beginn des Prozesses alle möglichen Ausführungen der Aktivitäten, die durch bestimmte Restriktionen begrenzt werden. Grundsätzlich ist jede Ausführungsvariante erlaubt, die nicht gegen die vorgegebenen Einschränkungen verstößt [Igl11]. Im Gegensatz zum imperativen Ansatz müssen dabei nicht explizit neue Wege hinzugefügt werden. Soll eine neue Variante hinzukommen, so ist diese standardmäßig im Modell vorhanden, soweit sie nicht gegen die Regeln verstößt [Igl11].

Das in Abbildung 4 dargestellte Sechseck repräsentiert, wie auch in Abbildung 1, ein Prozessmodell mit einem Start- und Endpunkt, jedoch in deklarativer Form. Hier sind als graue Linien nur eine repräsentative Anzahl der erlaubten Pfade dargestellt. Die rot unterlegten Flächen repräsentieren die sogenannten "verbotenen Zonen" und bilden somit die Einschränkungen des Modells ab. Die weiße Fläche ist der Raum, in dem sich der Pfad frei bewegen darf. Folglich sind keine Pfade erlaubt, die die roten Flächen berühren. Neue Pfade müssen nicht explizit hinzugefügt werden, es sind vielmehr alle Pfade erlaubt, die sich auf der weißen Fläche vom Start- zum Endpunkt bewegen und die roten Zonen nicht schneiden [Igl11].

Somit basiert die deklarative Modellierung auf einer Anzahl von vorgegebenen Regeln innerhalb eines abgesteckten Gebietes. Diese bilden die Abhängigkeiten untereinander bzw. ge-

## 2. Theoretische Grundlagen

gegenseitigen Einschränkungen zwischen den einzelnen Aktivitäten ab. Hierdurch wird mehr Flexibilität als bei der sequentiell vorgegebenen Reihenfolge der imperativen Prozessmodelle gewährleistet. Außerdem ist die Anzahl der möglichen Pfade von Beginn an erheblich höher als bei imperativen Prozessmodellen.

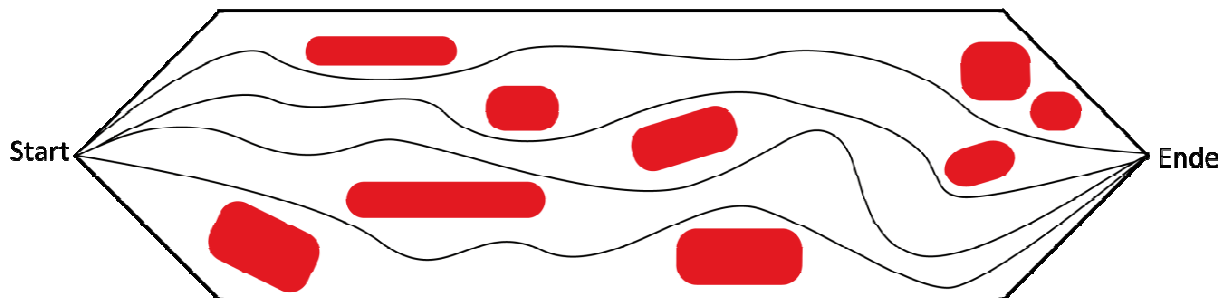
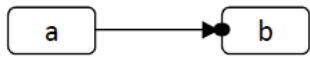






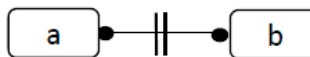
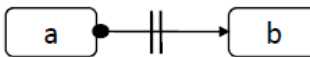
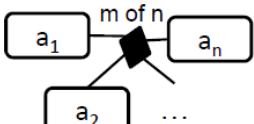


Abbildung 4: Darstellung von erlaubten Lösungswegen bei der deklarativen Prozessmodellierung

In der folgenden Tabelle 1 sind grundlegende Notationselemente, die in dieser Arbeit zur Abbildung deklarativer Prozessmodelle verwendet werden, dargestellt. In dieser Tabelle stellen a und b Aktivitäten dar. Die Erklärungen beziehen sich immer auf eine Prozessinstanz.

Bezeichnung	Abbildung	Erklärung
exactly	$n$ a	a darf exakt n mal ausgeführt werden
existence	$n..*$ a	a muss mindestens n mal ausgeführt werden
at most	$0..n$ a	a darf maximal n mal ausgeführt werden
init	init a	a muss als erstes ausgeführt werden
last	last a	a muss als letztes ausgeführt werden
response	a → b	wenn a ausgeführt wurde, muss b danach ebenfalls ausgeführt werden (aber nicht notwendigerweise direkt nach a)

## 2. Theoretische Grundlagen

<b>precedence</b>		b darf nur ausgeführt werden, wenn zuvor a ausgeführt wurde
<b>succession</b>		wenn a ausgeführt wurde, muss b danach ebenfalls ausgeführt werden (aber nicht notwendigerweise direkt nach a) und b darf nur ausgeführt werden, wenn zuvor a ausgeführt wurde
<b>chained response</b>		wenn a ausgeführt wurde, muss b direkt danach ausgeführt werden
<b>chained precedence</b>		b darf nur ausgeführt werden, wenn a direkt davor ausgeführt wurde
<b>chained succession</b>		wenn a ausgeführt wurde, muss b direkt danach ausgeführt werden und b darf nur ausgeführt werden, wenn a direkt davor ausgeführt wurde
<b>responded existence</b>		wenn a ausgeführt wurde, muss b entweder davor oder danach ausgeführt werden
<b>co-existence</b>		wenn a ausgeführt wurde, muss b entweder davor oder danach ausgeführt werden; wenn b ausgeführt wurde, muss a entweder davor oder danach ausgeführt werden
<b>not co-existence</b>		a und b dürfen nicht gemeinsam vorkommen
<b>negation-response</b>		nachdem a ausgeführt wurde, darf b anschließend nicht mehr ausgeführt werden
<b>exact choice</b>		Genau m von n Aktivitäten der Menge $\{a_1, a_2, \dots, a_n\}$ $m \leq n$ müssen ausgeführt werden

## 2. Theoretische Grundlagen

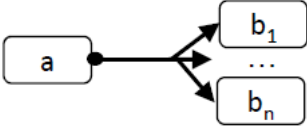
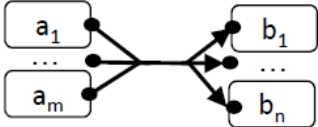
<p><b>multi re- sponse</b></p>		<p>nachdem a ausgeführt wurde, muss mindestens eine Aktivität der Menge <math>\{b_1, b_2, \dots, b_n\}</math> ausgeführt werden (aber nicht notwendigerweise direkt nach a)</p>
<p><b>multi succes- sion</b></p>		<p>wenn eine der Aktivitäten der Menge <math>\{a_1, a_2, \dots, a_n\}</math> ausgeführt wurde, muss danach mindestens eine Aktivität der Menge <math>\{b_1, b_2, \dots, b_n\}</math> ausgeführt werden; um eine Aktivität der Menge <math>\{b_1, b_2, \dots, b_n\}</math> ausführen zu können, muss zuvor mindestens eine Aktivität der Menge <math>\{a_1, a_2, \dots, a_n\}</math> ausgeführt worden sein</p>

Tabelle 1: deklarative Notationselemente

Im oberen Teil von Abbildung 5 ist ein Beispielprozess zur deklarativen Prozessmodellabbildung dargestellt. Das Prozessmodell besteht aus den Aktivitäten A, B, C und D. In diesem Prozess gibt es keinerlei Einschränkungen der Aktivitäten. Sie dürfen in jeder Reihenfolge und beliebig oft (auch mehrmals direkt hintereinander) ausgeführt werden. Folglich wird jede Art der möglichen Ausführbarkeit des Prozesses unterstützt. Beispielsweise ist hier die Ausführung der Pfade  $\langle A, D, B, C \rangle$  oder  $\langle A, A, B \rangle$  möglich.

Im unteren Teil von Abbildung 5 sind die gleichen Aktivitäten wie im Prozessbeispiel im oberen Teil, jedoch mit Einschränkungen, dargestellt. Der Prozess unterstützt nun nicht mehr jede mögliche Prozessvariante. Folgende Einschränkungen betreffen das Prozessmodell in Abbildung 10 im unteren Teil: A muss die erste Aktivität sein, die innerhalb des Pfades ausgeführt wird. C muss exakt einmal ausgeführt werden. B und D dürfen nicht beide innerhalb einer Prozessinstanz ausgeführt werden und B darf nur ausgeführt werden, wenn A zuvor (nicht notwendigerweise direkt davor) ausgeführt wurde. Es wird zum Beispiel die Ausführung der Pfade  $\langle A, B, C \rangle$ ,  $\langle A, C \rangle$  und  $\langle A, D, D, C \rangle$  unterstützt.



## 2. Theoretische Grundlagen

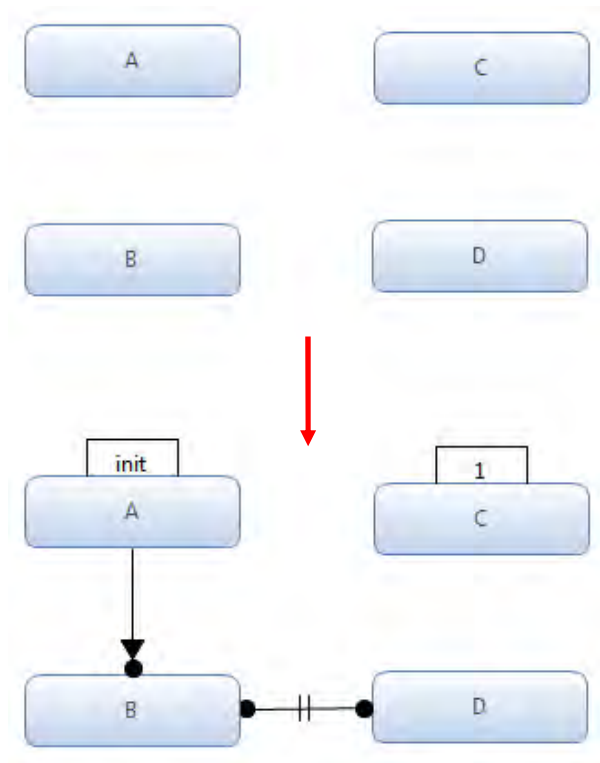


Abbildung 5: deklarativer Beispielprozess

### 2.4 Unterschiede zwischen imperativer und deklarativer Prozessmodellierung

Im Allgemeinen unterscheiden sich die imperative und deklarative Prozessmodellabbildung in ihrer Art des Aufbaus, ihrem bei der Modellierung verfolgten Ansatz, der zu Grunde liegenden Prozessmodellierungssprache und dem Zweck der Modellierung [ADO00, Agu04].

Der hauptsächliche Unterschied der imperativen und deklarativen Prozessmodellierung liegt im Ursprung der Computerprogrammierung: Die imperative Programmierung unterliegt dem Prinzip "how to do something" [RH04] und gibt genaue Angaben und Regeln, wie etwas erreicht wird. Im Gegensatz dazu basiert die deklarative Programmierung auf dem Grundsatz "say what is required and let the system determine how to achieve it" [RH04]. In imperativen Prozessmodellen wird zuvor spezifiziert, in welcher Reihenfolge und unter welchen Bedingungen die Aktivitäten ausgeführt werden sollen. Deklarative Prozessmodelle fokussieren sich darauf was getan werden muss. Durch Regeln wird hier unerwünschtes Verhalten des Prozesses eingedämmt [RH04].

## 2. Theoretische Grundlagen

An den Beispielprozessmodellen in Abbildung 4 und 10 ist zu erkennen, dass sich die imperative und deklarative Prozessmodellabbildung ebenfalls in ihrer Art der Modellierung unterscheiden. Imperative Modelle verwenden Pfade, die genau den Ablauf des Modells darstellen. Deklarative Prozessmodelle hingegen geben durch die Verwendung von Constraints Regeln vor, die unerwünschtes Verhalten des Prozesses eindämmen.

Allgemein ist festzustellen, dass die imperative und deklarative Prozessmodellabbildung Unterschiede im Ansatz und im Modellablauf aufweisen. Diese Unterschiede sollen im Laufe dieser Arbeit weiter diskutiert und anhand von Prozessbeispielen belegt werden.

### **3. Deklarative Prozessmodellierung als Forschungsgebiet**

Nachdem die theoretischen Grundlagen für diese Arbeit geschaffen worden sind, gibt dieses Kapitel einen Überblick über die deklarative Modellierung als aktuelles Forschungsgebiet. Zuerst wird auf den Ursprung der deklarativen Prozessmodellabbildung eingegangen. Anschließend wird eine Auswahl deklarativer Prozessmodellierungssprachen sowie damit verbundene Erweiterungen vorgestellt. Abschließend folgt eine Übersicht über verschiedene aktuelle Untersuchungen, die der qualitativen Verbesserung der deklarativen Abbildung dienen sollen.

#### **3.1 Ursprung der deklarativen Prozessmodellierung**

Die deklarative Abbildung von Prozessen ist ein relativ junger Ansatz im Bereich des BPM. Die imperative Prozessmodellabbildung hingegen existiert schon länger und war Vorreiter in der modellhaften Abbildung von (Geschäfts-) Prozessen [HZSHRPW13]. Allerdings ist das Interesse an deklarativen Möglichkeiten der Darstellung und somit auch der Anspruch an die Forschung in diesem Gebiet in den letzten Jahren deutlich gewachsen. Als Grund für das Interesse an der deklarativen Prozessmodellabbildung nennt die wissenschaftliche Literatur verschiedene Mängel der imperativen Prozessmodellabbildung [HZSHRPW13].

Imperative Prozessmodelle können laut [Pes08] zu einer Überspezifizierung von Prozessmodellen führen und weisen einen Mangel an Flexibilität auf. Dagegen hat sich laut [Pes08] der Wunsch des Benutzers, Prozesse selbst zu steuern und somit fehlerhafte oder unerwünschte Ausführungen zu vermeiden, verstärkt. Die Prozesse sollten daher so viel Flexibilität wie möglich aufweisen und die Handlungsweise des Nutzers nicht einschränken. Um genau dieser Anforderung gerecht zu werden, wurde die deklarative Prozessmodellabbildung als Alternative zum imperativen Ansatz entwickelt [APS09].

Hull et al [HLSSDKZ99] unterstützen die Ansicht von Pesic et al und sehen den Ursprung der deklarativen Prozessmodellabbildung ebenfalls in Flexibilitätsmängeln der imperativen Prozessmodellabbildung. Auch van der Aalst et al [AWG05] stellen fest, dass die Flexibilität von

### 3. Deklarative Prozessmodellierung als Forschungsgebiet

Workflows durch die imperative Prozessmodellabbildung bisher nicht erfüllt werden konnte und führen den Vorschlag an, Prozesse auf Basis eines deklarativen Modells abzubilden.

[GHV07] ist der Ansicht, dass in den letzten Jahren viel wissenschaftliche Aktivität zum Thema Restrukturierung und Neuordnung der BPM-Modelle stattfand und nach einem prozessorientierten Ansatz zur Modellierung von Prozessen gesucht wurde [GHV07]. Daraus resultierte nach der Meinung der Autoren die deklorative Prozessmodellabbildung.

Zusammenfassend kann festgestellt werden, dass die wissenschaftliche Literatur vor allem den Mangel an Flexibilität in imperativen Prozessmodellen als Grund für die Notwendigkeit und die Entwicklung von deklarativen Prozessmodellen anführt.

### **3.2 Deklarative Prozessmodellierungssprachen**

Im Rahmen des deklarativen Prozessmodellierungsansatzes wurden verschiedene Prozessmodellierungssprachen entwickelt. Hierzu zählen z.B. Vortex [HLSSDKZ99], EmBrace [GHV07], ConDec [PA06], DecSerFlow [AP06] und Declare [PSA07].

Hull et al erreichte mit der Entwicklung von Vortex [HLSSDKZ99] einen ersten Schritt in die Richtung der flexiblen, deklarativen Prozessmodellabbildung. Vortex ist eine objektorientierte Programmiersprache, mit der Workflows deklarativ abgebildet werden können [HLSSDKZ99]. [GHV07] stellt mit der Entwicklung von EmBrace (Enterprise Modeling using Business Rules, Agents, Activities, Concepts and Events) ebenfalls einen Ansatz zur Abbildung von Workflows durch eine deklorative Prozessmodellierungssprache zur Verfügung [GHV07].

ConDec [PA06] und DecSerFlow [AP06] sind zwei an der Universität Eindhoven entwickelte, deklorative Sprachen zur Modellierung und Darstellung dynamischer Workflows. Bei diesen Sprachen steht die Flexibilität der Anwendung im Vordergrund. Die Basis bildet der deklarative Ansatz: "It specifies what should be done, and users can decide how they will do it" [PA06]. Es ist zu erwähnen, dass an der Universität Eindhoven ein erheblichen Teil zur Forschung auf dem Gebiet der deklarativen Prozessmodellabbildung stattfindet und neben ConDec und DecSerFlow auch die Prozessmodellierungssprache Declare [PSA09, PSA07, PSSA07, APS09] entwickelt wurde. Declare dient der Erstellung, Überprüfung, Durchführung

### 3. Deklarative Prozessmodellierung als Forschungsgebiet

und der dynamischen Veränderung constraint-basierter Prozessmodelle [PSA07]. Dabei unterstützt es lose strukturierte Prozesse, ohne wichtige Aspekte von Workflow-Management Systemen, wie die Veränderung eines Prozesses zur Laufzeit, die Unterstützung des Benutzers und die nötige Flexibilität außer Acht zu lassen [PSA07].

Die hier vorgestellten Ansätze sind eine Auswahl an deklarativen Prozessmodellierungssprachen, die dazu dienen sollten, Workflows im Gegensatz zur imperativen Prozessmodellierung möglichst flexibel abzubilden. Dadurch hat der Nutzer die Möglichkeit, die in den Workflows abgebildeten Prozesse selbst zu steuern. Die Autoren sind sich dabei einig, dass deklarative Prozessmodellierung dadurch definiert wird, dass das Ziel und nicht die Erreichung des Ziels entscheidend ist.

Die hier beispielhaft angeführten Prozessmodellierungssprachen dienen als Grundlage verschiedener Tools zur Unterstützung der deklarativen Prozessmodellierung, wie z.B. Declare Toolset [DTS], Alaska Simulator [ALA] oder Test Driven Modelling Suite [TDMS].

#### **3.3 Erweiterungen**

Die deklarativen Prozessmodellierungssprachen bieten in ihrer ursprünglichen Form nicht die Möglichkeit, alle Anforderungen der Nutzer wie maximale Flexibilität oder die Integration von zeitlichen Aspekten ausreichend darzustellen. Daraus resultiert die Forderung nach Erweiterungen, wie Timed Declare [Mag14] oder ConDec-R [BV12].

Laut [Mag14] erlaubt kaum eine der Prozessmodellierungssprachen die Integration der Zeitperspektive. Vor allem für Geschäftsprozesse ist es jedoch von großer Bedeutung, zeitliche Ereignisse, wie Fristen oder optimierte Antwortzeiten, abzubilden. Zu diesem Zweck beinhaltet Declare metrische zeitliche Einschränkungen. Diese garantieren die korrekte Ausführung eines Prozesses in Bezug auf Latenzzeiten sowie die Einhaltung von Fristen. Das Timed Declare stellt eine Erweiterung der Prozessmodellierungssprache Declare dar und ermöglicht die Überwachung der metrischen zeitlichen Einschränkungen während der Laufzeit [Mag14]. Dadurch ist die Integration der Zeitperspektive basierend auf der Sprache Declare möglich. Allerdings benötigt die Anwendung von Timed Declare noch weitere Forschung, da es bei-

### 3. Deklarative Prozessmodellierung als Forschungsgebiet

spielsweise nicht möglich ist, ein genaues Datum anzugeben, an dem eine Aktivität ausgeführt werden soll [MW14].

Die mehrfache Ausführbarkeit von Aktivitäten führt in der deklarativen Prozessmodellabbildung häufig zu Schwierigkeiten. [BC08] sieht einen Lösungsansatz dieses Problems in der Aufstellung eines Algorithmus, der Regeln zur Filterung enthält. Diese Regeln ermöglichen es, die Beziehungen von Aktivitäten, die mehrfach ausgeführt werden können, zu filtern. Eingeschränkt wird dieser Ansatz dadurch, dass hier untypische zeitliche Bedingungen, wie der z.B. die alternierende Ausführung von sich Wiederholenden Aktivitäten zusammen mit der variablen Anzahl von Ausführungen der Aktivitäten nicht abgebildet werden können [BLWRV12].

[BV12] schlägt aufbauend auf dem Konzept zur Aufstellung eines regelbasierten Algorithmus einen constraint-basierten Lösungsansatz zur Planung und Terminierung der Ausführung von Aktivitäten vor. Dieser Ansatz basiert auf Regeln zur Filterung der Beziehungen zwischen den sich wiederholenden Aktivitäten. Dies ermöglicht die Spezifikation des Problems durch globale Einschränkungen und erhöht gleichzeitig die Effizienz bei der Suche nach einem Lösungsansatz [BV12]. Dieser Ansatz wird von den Autoren durch die erweiterte Sprache ConDec-R [BV12] verfolgt. ConDec-R baut auf der bereits beschriebenen Prozessmodellierungssprache ConDec auf und stellt eine Ressourcenerweiterung der deklarativen Prozessmodellabbildung dar. Eingeschränkt wird die erweiterte Sprache ConDec-R dadurch, dass hier keine Lösung der Integration des Zeitproblems für alle möglichen Fälle, sondern nur für eine bestimmte Anzahl getesteter Fälle existiert und hier noch weiterer Forschungsbedarf besteht [BLWRV12].

Die hier beschriebenen Beispiele zur Erweiterung stellen eine Auswahl dar. Es kann festgestellt werden, dass auf diesem Gebiet auch weiterhin Forschungsbedarf besteht.

#### **3.4 Untersuchungen zur Verbesserung deklarativer Abbildung**

Die Deklarative Prozessmodellabbildung weist nicht nur Unzulänglichkeiten in der Integration des zeitlichen Aspekts oder in der mehrfachen Ausführbarkeit von Aktivitäten auf, sondern ebenfalls in der Abbildung bezogen auf deren Struktur und die Integration von Hier-

### 3. Deklarative Prozessmodellierung als Forschungsgebiet

archieebene von Prozessmodellen. Dies führt dazu, dass von verschiedenen Autoren Untersuchungen zur Verbesserung der deklarativen Prozessmodellabbildung diskutiert werden. Diese Verbesserungen zielen darauf ab, die Prozessmodelle übersichtlicher und für den Nutzer besser verständlich zu gestalten.

In [Zug11] wird die Frage untersucht, inwieweit die Qualität von deklarativen Prozessmodellen durch eine zielgerichtete Übertragung von kognitiven Konzepten aus der Psychologie verbessert werden kann. Dies bezieht sich auf Aspekte wie die Erstellung, Verständlichkeit, Wartbarkeit und Modularisierung eines Prozessmodells. Der Autor kommt dabei zu dem Ergebnis, dass die Qualität eines Prozessmodells mit Hilfe bestimmter Konzepte aus der kognitiven Psychologie, wie z.B. Visual Search oder Recognition verbessert und weiterentwickelt werden kann [Zug11].

[HZSHRPW13] beschäftigt sich ebenfalls mit möglichen Verbesserungen der Abbildung deklarativer Prozessmodelle und untersucht, ob eine grafische oder eine textuelle Abbildung von deklarativen Prozessmodellen für den Benutzer plausibler ist. Die Autoren stellen sich weiter die Frage, ob diese Modelle durch das Hinzufügen neuer Notationselemente in Form von grafischen Darstellungen oder Text verständlicher werden. Sie kommen zu dem Ergebnis, dass eine rein textuelle Abbildung nicht zu einer Verbesserung der Lesbarkeit führt. Desweiteren führen die Autoren in ihrer Arbeit an, dass hierarchische Konzepte in der Abbildung deklarativer Prozessmodelle eine bessere Verständlichkeit für den Leser mit sich bringen [HZSHRPW13].

Diesen Ansatzpunkt hat [ZSHPRW13] ebenfalls aufgegriffen und die Frage untersucht, ob eine automatische Strukturierung von wichtigen und unwichtigen Elementen als Subprozesse die Hierarchie in einem Prozessmodell positiv beeinflusst. Die Autoren kommen zu dem Ergebnis, dass die Verwendung von Subprozessen zum Verständnis beitragen, dieses jedoch bei übermäßiger Verwendung auch verschlechtern kann, und somit mit Vorsicht zu verwenden ist. Außerdem wird durch die Hierarchisierung nicht nur die Struktur, sondern auch die Semantik beeinträchtigt [ZSHPRW13].

In [HZSHRPW13] wird vorgeschlagen, äquivalent zu den für die imperative Modellierung bestehenden Seven Process Modelling Guidelines [MRA10] ebenfalls Richtlinien für deklarativen Prozessmodelle zu erstellen und empirisch zu belegen. Laut [HZSHRPW13] würden diese

### 3. Deklarative Prozessmodellierung als Forschungsgebiet

auch zu einer effektiveren Nutzbarkeit der Prozessmodelle und ihrer Vereinheitlichung beitragen.

Die verschiedenen Ansätze zur Verbesserung der Abbildung von deklarativen Prozessmodellen, die hier erläutert wurden, bieten noch viel Raum für weitere wissenschaftliche Arbeiten. Bestenfalls können diese, nach erfolgreicher theoretischer Forschung und Validierung in bereits bestehende bzw. in zukünftige Prozessmodellierungssprachen integriert werden.

### **3.5 Zusammenfassung**

Die theoretische Grundlage für diese Arbeit bildet die deklarative Prozessmodellierung. Ihren Ursprung hat dieses Paradigma in dem Wunsch des Benutzers, die Flexibilität der bisher üblichen imperativen Prozessmodellabbildung zu erhöhen und dadurch möglichst großen Einfluss auf die Steuerung des Prozesses ausüben zu können. Aus diesen Anforderungen haben sich verschiedene deklarative Prozessmodellierungssprachen, wie Votex, Declare, ConDec, DecSerFlow und EmBrace, entwickelt. Sie sind Grundlage für verschiedene Tools, die zur Abbildung der deklarativen Prozessmodelle verwendet werden können. Aus den stetig wachsenden Anforderungen an die deklarative Prozessmodellabbildung wurden Ressourcenerweiterungen wie Timed Declare oder ConDec-R entwickelt. In der Literatur werden weitere Ansätze zur Verbesserung der deklarativen Prozessmodellabbildung diskutiert [Zug11, HZSHRPW13, ZSHRPW13].

Allgemein besteht zur deklarativen Prozessmodellierung in der wissenschaftlichen Literatur weiterhin ein Mangel an Forschungsbeiträgen. Insbesondere im Bereich von Schnittpunkten von imperativen und deklarativen Prozessmodellen ist dieser Mangel besonders ausgeprägt. Diese Arbeit soll einen Beitrag zur Behebung dieses Mangels leisten, indem sie einen analytischen Vergleich von imperativer und deklarativer Prozessmodellabbildung anhand praktischer Beispiele aufstellt und die Abbildung imperativer Prozessmodelle in die deklarative Form diskutiert.



## 4. Test Driven Modelling Suite als praktische Grundlage

Dieses Kapitel beschäftigt sich mit dem für die Umsetzung der beschriebenen Prozessmodelle und deren Abbildung notwendigen Toolsupport. In Kapitel 3 wurde bereits die Entwicklung der deklarativen Prozessmodellabbildung, sowie das daraus resultierende imperative und deklarative Prozessmodellierungsparadigma dargestellt. Basierend auf dem deklarativen Prozessmodellierungsparadigma haben sich verschiedene Prozessmodellierungssprachen, wie z.B. Declare, entwickelt. Diese Prozessmodellierungssprachen benötigen operative Unterstützung und Werkzeuge zur Abbildung der Prozessmodelle. Diese Unterstützung bieten sogenannte Tools, wie z.B. die Test Driven Modelling Suite (TDMS) [TDMS].

Das Konzept, das der TDMS zu Grunde liegt ist das Test Driven Modelling (TDM). TDM basiert laut [ZPW11] auf der Erprobung des gewünschten Verhaltens eines Prozesses mit Hilfe von Testfällen. Dadurch kann der Benutzer feststellen, ob der von ihm modellierte Prozess korrekt bzw. nach seinen Wünschen ausgeführt wird oder wenn nötig, diesen zur Laufzeit beeinflussen. TDM besteht aus genau einem deklarativen Modell und einer beliebigen Menge dazugehöriger Prozessinstanzen. Jeder Testfall beschreibt ein eigenes Szenario, das vom Modell unterstützt werden soll. Das bedeutet, dass während des Modellierens gleichzeitig ein Testen auf die korrekte Ausführbarkeit des Modells statt findet [ZPW11]. Jeder Prozess beinhaltet somit eine automatisierte Prüfung in Echtzeit.

Die TDMS bietet beim Modellieren die notwendige operative Unterstützung in Form eines Tools [ZPW11]. Diese Toolunterstützung dient der visuellen Darstellung der Prozesse mit Hilfe eines Modellierungsprogramms. Das Prozessmodell wird mit Hilfe des Modellierungsprogramms iterativ in der TDMS erstellt. Declare bildet hier als Prozessmodellierungssprache die Basis der TDMS. Zum Zweck der Verifikation wird das Prozessmodell durch die TDMS so konvertiert, das es von Declare gelesen und in der Workengine von Declare ausgeführt werden kann [ZPW11]. Das Tool wird auf der Website der Universität Eindhoven [TDMS] kostenlos zur Verfügung gestellt.

Abbildung 6 zeigt die TDMS mit einem beispielhaften, deklarativen Prozessmodell. Der Test Case Editor, der auf der linken Seite der Abbildung zu sehen ist (grün markiert), stellt

#### 4. Test Driven Modelling Suite als praktische Grundlage

einen kalenderartigen Testfall-Simulator für das Prozessmodell zur Verfügung. Dieser ermöglicht die Überprüfung der Ausführbarkeit eines Prozessmodells und bietet zeitgleich im unteren rechten Bereich eine Testcase-Creation-and-Validation-Sektion (rot markiert). In dieser werden bei einer Nichtausführbarkeit des Prozessmodells eine detaillierte Problembeschreibung und mögliche Lösungsvorschläge angezeigt. In der Mitte von Abbildung 6 ist der Declarative Process Model Editor (blau markiert) zu sehen, der den Arbeitsbereich des Nutzers zur Modellierung der Prozessmodelle darstellt. Auf der rechten oberen Seite befinden sich alle zur Verfügung stehenden gängigen deklarativen Modellierungselemente (schwarz markiert), die zur Modellierung eines deklarativen Prozessmodells in Declare verwendet werden können.

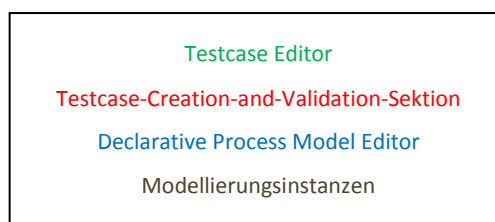
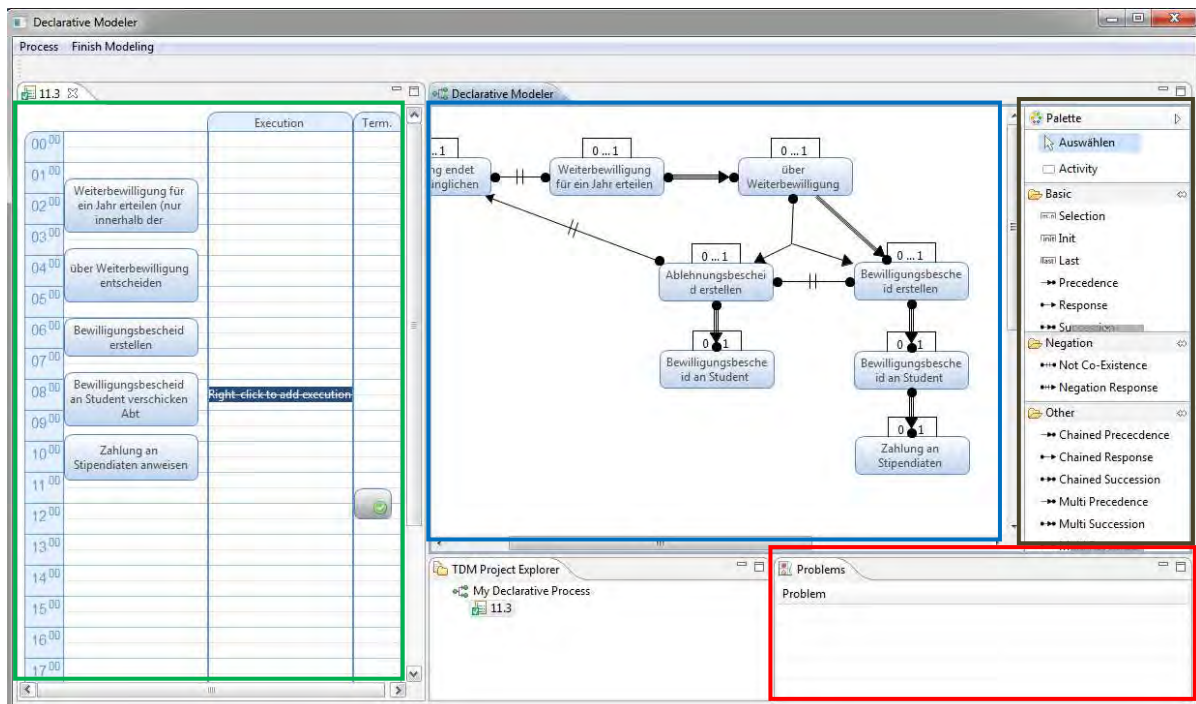


Abbildung 6: TDMS

#### 4. Test Driven Modelling Suite als praktische Grundlage

Der Test Case Editor, welcher in Abbildung 7 dargestellt ist, gibt dem Nutzer die Möglichkeit, die einzelnen Ausführungsvarianten des erstellten Prozessmodells zu überprüfen [ZPW11]. Er ermöglicht zu testen, ob eine Aktivität an einer bestimmten Stelle ausführbar ist: Die linke Spalte des Test Case Editors gibt an, ob ein Prozess in der hier dargestellten Reihenfolge, beispielsweise der Aktivitäten A, B und C, ausgeführt werden darf. Andernfalls wird hier eine Fehlermeldung in Form eines rot markierten Aktivität (siehe Aktivität C) ausgegeben. Die mittlere Spalte des Testcase Editors überprüft, ob eine Aktivität an dieser Stelle ausgeführt werden darf. Damit ist z.B. gemeint, dass die Aktivität D nicht zwischen der Aktivität A und B ausgeführt werden darf und somit einen Fehler verursacht. Die rechte Spalte zeigt an ob ein Prozessmodell beendet werden kann. Ist dies nicht der Fall, wird ebenfalls ein Fehler ausgegeben und das Feld E färbt sich rot.

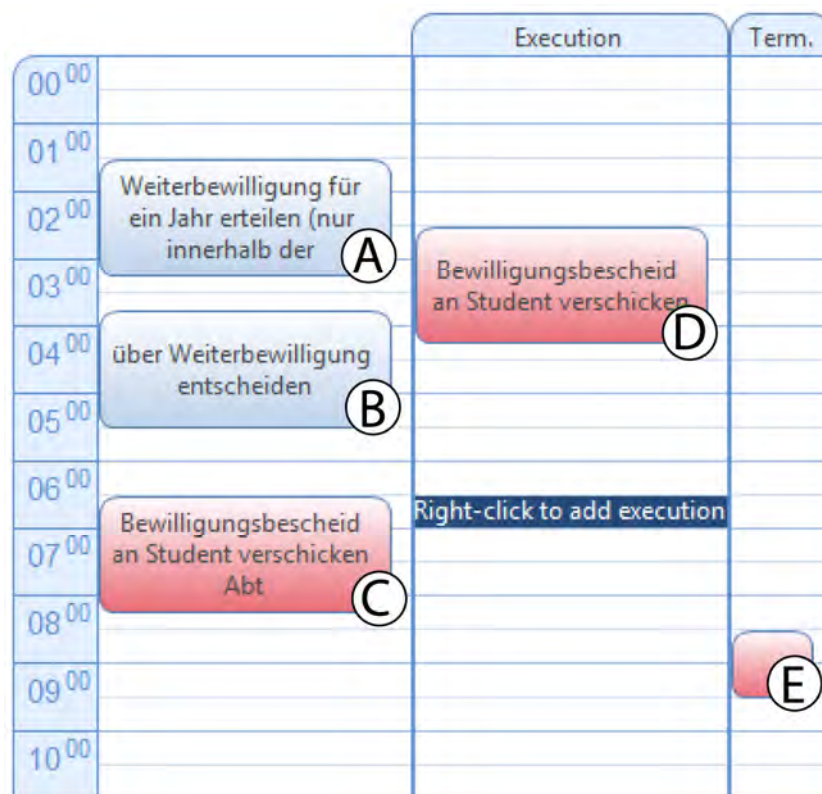


Abbildung 7: Testcase Editor der TDMS

Die TDMS hat den Vorteil, dass sie intuitiv zu bedienen und übersichtlich aufgebaut ist. Desweiteren bietet sie nicht nur die Möglichkeit des TDM, sondern auch die Ausgabe des

#### 4. Test Driven Modelling Suite als praktische Grundlage

fertigen Prozessmodells als pdf-Dokument. Eine Einschränkung ist die für den Nutzer unlogische Speicherung des erstellten Prozessmodells. Insbesondere gibt es keine Möglichkeit, den Speicherort selbst zu wählen bzw. einen selbst gewählten Dateinamen zu vergeben. Die Dateien werden unter Datum und Uhrzeit der Erstellung abgelegt. Dies erschwert ein Wiederauffinden. Weiterhin ist das gleichzeitige Öffnen mehrerer Dateien (z.B. zum Vergleich verschiedener Prozessmodelle) nicht möglich. Teilweise führt auch die nachträgliche Anordnung oder Verschiebung der Constraints zu grafisch verworrenen Ergebnissen. Die TDMS könnte durch einige kleine Erweiterungen und Verbesserungen noch bedienerfreundlicher gestaltet und optimiert werden.

Dennoch ist TDMS, vor allem im Vergleich zu anderen existierenden Tools, geeignet um deklarative Prozesse nachvollziehbar darzustellen. Deshalb wird im Folgenden dieser Arbeit die TDMS zur Abbildung der deklarativen Prozessmodelle und gleichzeitig zur Echtzeitüberprüfung auf die Korrektheit der Prozessmodelle verwendet.

## **5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen**

Nachdem bereits die Grundlage für das Verständnis von imperativen und deklarativen Prozessmodellen sowie deren Abbildung mit Hilfe von Tools geschaffen wurde, stellt dieses Kapitel den Hauptteil der Arbeit dar. Es beschäftigt sich sowohl mit den sich bei der praktischen Umsetzung der Prozessmodelle ergebenden Schwierigkeiten, als auch den positiven Aspekten bezüglich der Migration imperativer Prozessmodelle in ein deklaratives Schema.

Es werden zunächst die Möglichkeiten, die bei der Umsetzung der Prozessmodelle bestehen in Form von positiven Faktoren aufgezeigt. Anschließend folgt eine Diskussion und Charakterisierung der Schwierigkeiten, die sich bei der Migration ergeben. Es werden Lösungsalternativen, die im Rahmen dieser Arbeit entwickelt wurden, dargestellt. Abschließend wird die praktische Umsetzung und die daraus resultierenden Ergebnisse anhand von drei beispielhaft ausgewählten, repräsentativen Prozessen erläutert und eine Empfehlung gegeben, welche Art der Prozessmodellabbildung sich für welchen Prozess besser eignet.

Im Folgenden ist zu beachten, dass sämtliche Abbildungen mit zwei verschiedenen Prozessmodellen immer links bzw. oben das imperative und rechts bzw. unten auf der Abbildung das deklarative Prozessmodell zeigen. Weiterhin werden aus Platzgründen lediglich Ausschnitte der Prozessmodelle dargestellt, die kompletten Abbildungen der ursprünglich imperativen sowie der migrierten deklarativen Prozessmodelle befinden sich im Anhang dieser Arbeit.

## 5.1. Möglichkeiten bei der Abbildung von Prozessen als deklaratives Prozessmodell

Im Folgenden werden die Möglichkeiten zur Umsetzung imperativer Prozessmodelle in ein deklaratives Schema anhand passender Beispiele aufgezeigt und diskutiert. Ferner wird auf positive Aspekte bei der Migration eingegangen, die zum besseren Verständnis anhand von kurzen Beispielen erläutert werden.

### 5.1.1 Anzahl der Ausführungseinheiten von Aktivitäten

Im deklarativen Ansatz der Prozessmodellierung gibt es die Möglichkeit, die Anzahl der Ausführungseinheiten einer Aktivität anzugeben und grafisch abzubilden. Diese Angabe kann von nicht spezifiziert, was bedeutet, dass die Aktivität beliebig oft (auch direkt hintereinander) innerhalb eines Prozesses ausgeführt werden kann, bis hin zu einer bestimmten Zahl oder einem Intervall definiert werden (siehe Kapitel 2.3). Der Vorteile, den die deklarative Prozessmodellierung durch die Angabe der Ausführungseinheiten von Aktivitäten hat, ist dass Aktivitäten, die sich innerhalb eines Prozessmodells wiederholen, nicht mehrmals abgebildet werden müssen.

Die imperative Prozessmodellierung sieht nicht vor, die Anzahl der Ausführungseinheiten eines Tasks anzugeben, da hier durch den eindeutig bestimmten Ausführungspfad genau vorgegeben ist, wann, wie oft und zu welchem Zeitpunkt eine Aktivität ausgeführt wird.

Ein Beispiel hierfür ist in Abbildung 8 zu sehen. Im imperativen Prozessmodell auf der linken Seite der Abbildung müssen die Aktivitäten *Antrag in das 1. FS* (blau markiert) und *Antrag in ein höheres Fachsemester* (grün markiert) jeweils zwei mal abgebildet werden. Der Grund hierfür ist, dass beide Aktivitäten an verschiedenen Stellen des Prozesses ausgeführt werden können. Die Aktivität *Antrag in das 1. FS* zum Beispiel kann sowohl auf die Aktivität *Antrag aus 3. FS oder höher*, als auch auf die Aktivität *Antrag aus dem 1. oder 2. Fachsemester* folgen. Im deklarativen Prozessmodell auf der rechten Seite von Abbildung 8 müssen die Aktivitäten *Antrag in das 1. FS - Abt. II-2 Studiensekretariat (Uni Ulm)* und

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

*Antrag in ein höheres Fachsemester - Abt. II-2 Studiensekretariat (Uni Ulm)* jeweils nur einmal abgebildet werden.

In diesem Beispielmodell werden alle Aktivitäten im imperativen Prozessmodell von der selben Rolle ausgeführt. Einschränkend ist jedoch zu erwähnen, dass bei einer Zusammenfassung mehrerer Aktivitäten mit unterschiedlichen Rollenzuweisungen im deklarativen Prozessmodell die Kontextinformation über die ausführende Instanz der einzelnen Aktivitäten verloren geht.

Zusammenfassend ermöglicht die deklarative Prozessmodellierung durch die Angabe der Ausführungseinheiten einzelner Aktivitäten die Reduktion der Anzahl von Aktivitäten, da diese nicht mehrmals abgebildet werden müssen.

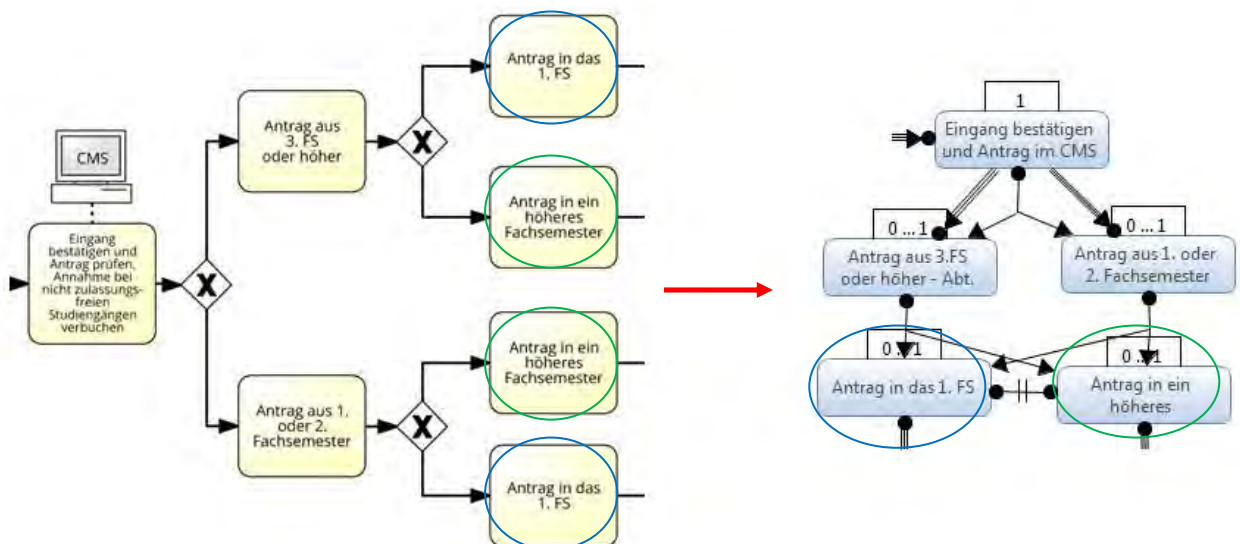


Abbildung 8: Anzahl der Ausführungseinheiten von Aktivitäten

### 5.1.2 Flexible Reihenfolge von Aktivitäten

Die deklarative Prozessmodellierung erlaubt eine flexible Reihenfolge bei der Ausführung von Aktivitäten. Die imperative Prozessmodellierung hingegen verlangt einen strengen Ablauf des Prozesses und legt genau fest, wann welche Aktivität ausgeführt werden muss.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Ein Beispiel für die Flexibilität in der deklarativen Prozessmodellierung stellt das Interleaved Parallel dar. Das Interleaved Parallel Routing besteht aus einer Reihe von Aktivitäten, die in einer willkürlichen Reihenfolge ausgeführt werden können [WADH03]. Jede dieser Aktivitäten darf nur einmal, jedoch keine der Aktivitäten gleichzeitig ausgeführt werden [WADH03].

Die Darstellung des Interleaved Parallel Routing lässt sich deklarativ sehr einfach lösen. Der Grund hierfür ist, dass in der deklarativen Prozessmodellabbildung eine "lose" Reihenfolge der Aktivitäten auf Grund der hohen Flexibilität problemlos darstellbar ist. Außerdem dürfen in der deklarativen Prozessmodellierung im Gegensatz zur imperativen Prozessmodellierung keine Aktivitäten parallel ausgeführt werden.

Ein Beispiel für die Abbildung des Interleaved Parallel Routing ist im deklarativen Prozessmodell in Abbildung 9 zu sehen. Hier werden die beiden Aktivitäten *Rückerstattung im CMS auslösen - Abt. II-2 Studiensekretariat (Uni Ulm)* und *Immatrikulation im CMS stornieren - Abt. II-2 Studiensekretariat (Uni Ulm)* jeweils genau ein mal in einer nicht festgelegten Reihenfolge ausgeführt.

In der imperativen Prozessmodellierung ist die Abbildung des Interleaved parallel Routing nicht möglich, da hier keine flexible Reihenfolge der Aktivitäten abgebildet werden kann.

Das Interleaved Parallel Routing ist ein Beispiel dafür, dass die deklarative Prozessmodellierung im Gegensatz zur imperativen Prozessmodellierung eine erhöhte Flexibilität ermöglicht. Die erhöhte Flexibilität führt dazu, dass die Abbildung von allen Aktivitäten, die keine bestimmte Reihenfolge einhalten müssen im deklarativen Prozessmodell problemlos dargestellt werden können.



Abbildung 9: Interleaved Parallel Routing



### 5.1.3 Granularität

Die Granularität eines Prozessmodells beschreibt den Detailgrad, den ein Prozessmodell abbildet [GHV07]. Es gibt verschiedene Granularitätsstufen, von grobgranular bis feingranular. Ein feingranulares Prozessmodell enthält einen hohen Detailgrad, ein grob granulares Prozessmodell verschafft eher einen Überblick über den Prozess [GHV07].

Deklarative Prozessmodelle können im Gegensatz zu imperativen Prozessmodellen jegliche Art von Granularitätsstufe abbilden. Auf Grund der Flexibilität der deklarativen Prozessmodellierung ist hier z.B. die Abbildung verschiedener Eltern- und Subprozesse innerhalb eines Prozessmodells möglich. Dadurch wird eine sehr feingranulare Abbildung des Prozessmodells ermöglicht. Imperative Prozessmodelle können im Vergleich zu deklarativen Prozessmodellen als grobgranular eingeordnet werden. Hier werden beispielsweise verschiedene Hierarchieebenen separat und als eigenständiger Prozess behandelt.

Ein Beispiel hierfür ist in Abbildung 10 im oberen Teil auf der linken Seite zu sehen. Das imperative Prozessmodell integriert den Subprozess *2.9.1 NR durchführen*. Der Subprozess *2.9.1 NR durchführen* ist ebenfalls in der Abbildung in der rechten oberen Hälfte abgebildet und besteht aus drei Aktivitäten. Der Subprozess wird im imperativen Prozessmodell als aufklappbarer Subprozess dargestellt, der sich eine Hierarchieebene tiefer befindet als der Hauptprozess. Der Subprozess wird im Hauptprozess in einer Aktivität verlinkt, jedoch nicht im Elternprozess mit abgebildet.

Ein Beispiel für die hohe Granularität eines deklarativen Prozessmodells ist in Abbildung 10 im unteren Teil zu sehen. Hier ist das zuvor beschriebene Prozessmodell inklusive des Subprozesses im deklarativen Schema abgebildet. Die Granularität des deklarativen Prozessmodells ist somit höher.

Im Vergleich des imperativen Prozessmodells samt dazugehörigem Subprozess mit dem deklarativen Prozessmodell fällt auf, dass hier die Flexibilität der Abbildung des deklarativen Modells großen Einfluss auf die Granularität hat. Fein granulare Prozessmodelle mit einem hohen Detailgrad und vielen Hierarchieebenen können deklarativ besser abgebildet werden als imperativ. Die Integration von Subprozessen ist deklarativ auf Grund der erhöhten Flexibilität meist problemlos möglich. Folglich können jegliche Ebenen des Pro-

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

zessmodells innerhalb eines Prozessmodells dargestellt werden. Die imperative Prozessmodellierung verlangt hier eine Verwendung von Subprozessen. Die Subprozesse werden als eigenständiger Unterprozess in das Prozessmodell integriert und bilden somit eine eigene Hierarchieebene. Die Information des gesamten Prozesses wird nicht innerhalb eines Prozessmodells dargestellt, sondern durch Unterprozesse eingebunden. Folglich ist die Granularität des Hauptprozessmodells hier grob.

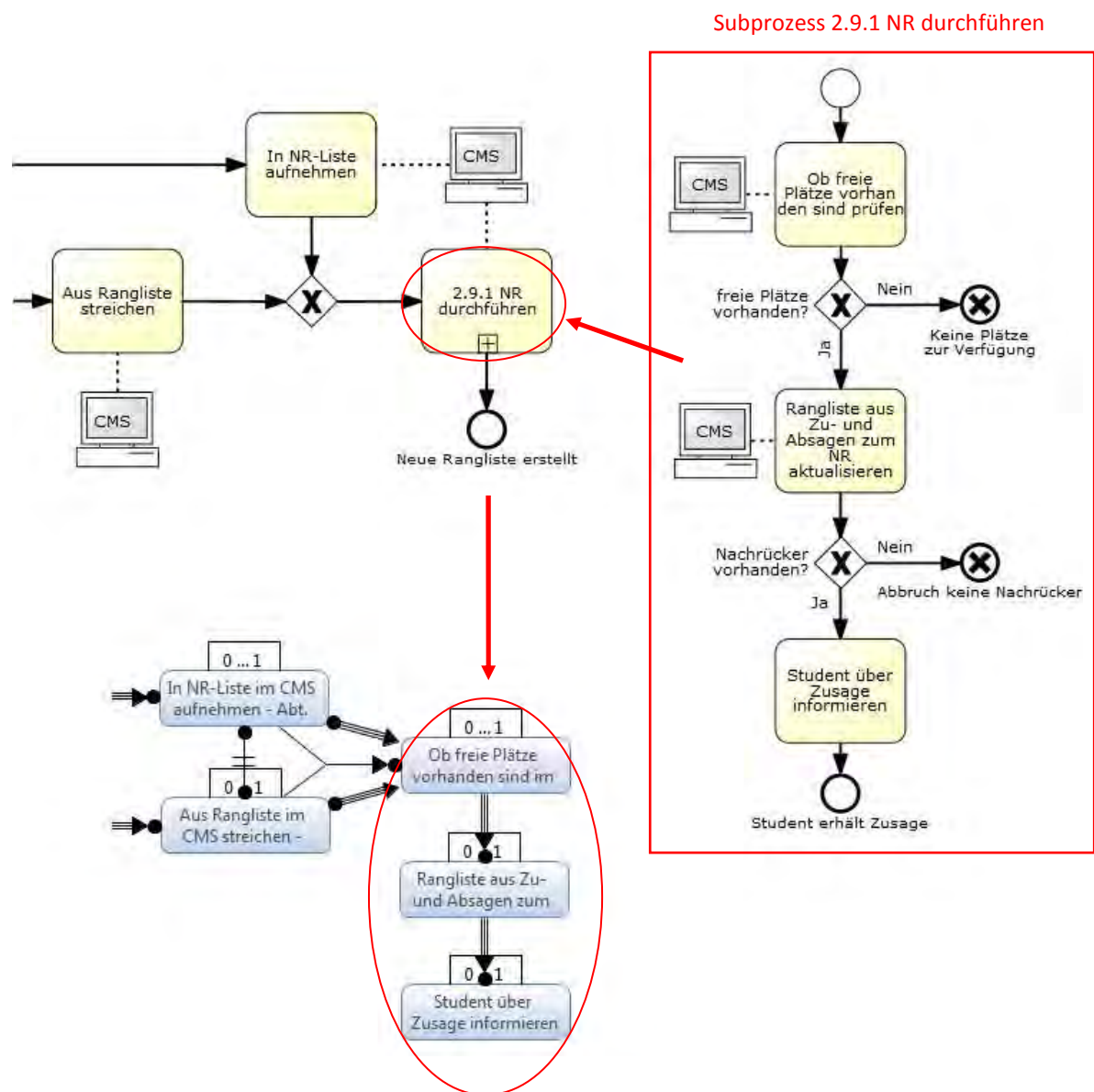


Abbildung 10: Granularität im deklarativen Prozessmodell

#### **5.1.4 Zusammenfassung**

In diesem Abschnitt wurden Möglichkeiten der Migration imperativer Prozessmodelle in das deklarative Schema anhand verschiedener Beispiele diskutiert und die Vorteile der deklarativen Abbildung aufgezeigt

Zusammenfassend ermöglicht die deklarative Prozessmodellierung durch die Angabe der Ausführungseinheiten einzelner Aktivitäten die Reduktion der Anzahl von Aktivitäten, da diese nicht mehrmals abgebildet werden müssen.

Des Weiteren ist die Abbildung einer losen Reihenfolge von Aktivitäten, wie z.B. beim Interleaved Parallel Routing auf Grund der benötigten Flexibilität deklarativ eindeutig besser darstellbar und kann imperativ nicht korrekt abgebildet werden. Imperative Prozessmodelle basieren auf strengen Verfahrensvorgaben und sind nicht dafür geeignet Prozessmodelle mit hohen Flexibilitätsanforderungen abzubilden.

Zusätzlich zeichnen sich deklarative Prozessmodelle dadurch aus, das sie jegliche Art von Granularität abbilden können. Imperative Prozessmodelle sind im Gegensatz dazu als eher grobgranular einzuordnen.

Die deklarative Prozessmodellierung hat also im Vergleich zur imperativen Prozessmodellierung den Vorteil, flexiblere Darstellung zu ermöglichen, Ausführungseinheiten für Aktivitäten anzugeben und Modelle jeglicher Granularität besser abzubilden.

#### **5.2. Schwierigkeiten bei der Abbildung von Prozessen als deklaratives Prozessmodell und mögliche Lösungswege**

Nachdem bereits Möglichkeiten der Migration imperativer Prozessmodelle in die deklarative Form aufgezeigt wurden, werden im Folgenden Prozessmodellaspekte diskutiert, bei denen eine Migration nicht ohne weiteres möglich und mit Schwierigkeiten verbunden ist. Es wird hierbei auch auf Auswirkungen der Schwierigkeiten auf das Prozessmodell eingegangen. Des Weiteren werden die Grenzen bei der korrekten und vollständigen Umsetzung der deklarativen Prozessmodelle aufgezeigt.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Diese Grenzen sind unter anderem auf den bereits erläuterten grundlegend unterschiedlichen Ansatz der beiden Abbildungsvarianten sowie die damit verbundenen begrenzten Darstellungsmöglichkeiten der beiden Modellierungsparadigmen zurückzuführen. Um dennoch eine korrekte Übertragung der imperativen Prozessmodelle in das deklarative Schema zu gewährleisten, werden hier mögliche Lösungswege aufgezeigt. Sie basieren auf den durch die Sprache Declare zur Verfügung stehenden Mitteln und werden auch damit realisiert. Die einzelnen Ergebnisse werden wie im vorherigen Abschnitt anhand von Beispielen aus Prozessmodellen erläutert.

### 5.2.1 Sequenzflüsse

Ein Sequenzfluss beschreibt eine Abfolge von Aktivitäten, die innerhalb eines Prozessmodells in einer bestimmten Reihenfolge ablaufen.

Grundsätzlich lassen sich Aktivitäten in der deklarativen Prozessmodellierung genauso gut abbilden wie in einem imperativen Prozessmodell. In beiden Prozessmodellierungsparadigmen werden diese als beschriftetes Kästchen dargestellt und können somit vom Nutzer ohne Probleme interpretiert werden. In der deklarativen Prozessmodellierung ist die Ausführung jeder Aktivität zu jeder Zeit möglich, so lange keine Einschränkungen gegeben sind. Constraints stellen die Einschränkungen, wie z.B. Beziehungen zwischen Aktivitäten, dar und werden nach der Art der Einschränkung und deren Funktion unterschieden (siehe Kapitel 2.3). Soll ein bestimmter Verlauf eines Prozesses dargestellt werden, muss dieser durch das Hinzufügen von Constraints realisiert werden. Constraints stellen allerdings keinen Richtungsverlauf des Prozessmodells in Form von Pfeilen dar, sondern geben die Beziehungen der Aktivitäten untereinander an. In der imperativen Prozessmodellierung wird der Sequenzfluss in Form von Pfaden dargestellt, die den Verlauf des Prozesses durch Richtungspfeile abbilden.

Ein Beispiel für die Verwendung von Pfaden im imperativen Prozessmodell ist in Abbildung 11 zu sehen. Es ist erkennbar, dass durch die Pfade der Verlauf des Prozesses grafisch veranschaulicht wird. Der Prozessausschnitt beginnt mit der Aktivität *6.5 Anmeldung*

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

zu *Lehrveranstaltungen*. Anschließend kann die Aktivität *zur Vorleistung anmelden* folgen oder es wird sofort die Aktivität *vorläufig zur Prüfung anmelden* ausgeführt.

Das migrierte deklarative Prozessmodell ist in Abbildung 11 auf der rechten Seite zu sehen. Hier stellen Constraints nicht wie im imperativen Prozessmodell der Verlauf des Prozesses bildlich dar, sondern geben die Beziehungen bzw. Einschränkungen der Aktivitäten untereinander an.

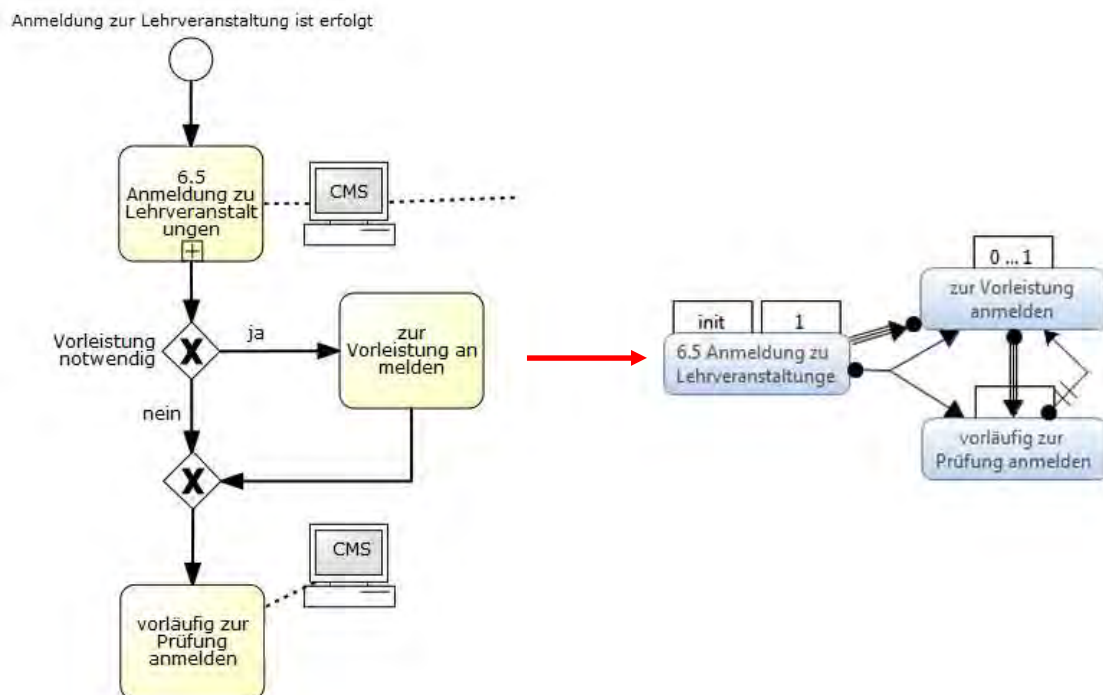


Abbildung 11: Abbildung von Sequenzflüssen

Für den Betrachter ist die deklarative Darstellung des Prozessmodells auf Grund der Verwendung von Constraints weniger gut verständlich. Ein eindeutiger Prozessablauf ist meist nicht erkennbar. Mit ansteigendem Umfang der Sequenzflüsse des Prozessmodells steigt auch die Komplexität der Constraints an. Im imperativen Prozessmodell hingegen kann der Nutzer dem Verlauf des Prozesses anhand der Pfade folgen, der hier bildlich, ähnlich einer Landkarte, dargestellt wird.

### 5.2.2 Kontextinformationen

Im Folgenden wird auf Kontextinformationen in Form von Rollenzuweisungen durch Pools und Lanes, sowie Artefakte und Kommunikationsereignisse eingegangen. Es ist eine Unterscheidung zwischen einem fachlichen und einem technischen Prozess notwendig, um einen Lösungsweg zur korrekten Abbildung des Prozessmodells zu gewährleisten. In einem technischen Prozess kann die Prozessmodellierungssprache Declare den Inhalt nicht vollumfänglich darstellen. Meist ist hier die Integration z.B. von Kontextinformationen, wie Rollen, Artefakte und Kommunikationsereignisse nur durch die Erweiterung der Sprache möglich. Diese Erweiterung der Sprache ist jedoch nicht Thema dieser Arbeit. Deshalb werden die Prozessmodelle bei den folgenden Lösungswegen als fachliche Prozesse betrachtet. Um die korrekte Abbildung der Kontextinformation bei Rollenzuweisungen, Artefakten und Kommunikationsereignisse zu gewährleisten, wird hier auf sog. Workarounds zurück gegriffen. Sie ermöglichen die Integration der Kontextinformation ohne Erweiterung der Prozessmodellierungssprache und bieten einen Lösungsweg mit den bereits vorhandenen Möglichkeiten von Declare. Die Workarounds werden in den folgenden Abschnitten für die Verwendung von Rollenzuweisungen, Artefakten und Kommunikationsereignissen in deklarativen Prozessmodellen beschrieben.

- **Rollen**

Im Gegensatz zur imperativen Prozessmodellierung sind in der deklarativen Prozessmodellierung keine Rollenzuweisungen durch Pools und Lanes vorgesehen. Die Zuweisung von Rollen ist jedoch von großer Bedeutung für das Verständnis eines Prozesses, da sie angeben wer welche Aktivität ausführt. Sie gewährleisten einen eindeutigen Ablauf des Prozesses und geben dem Nutzer wichtige Kontextinformationen.

Um die korrekte Migration des Prozessmodells in das deklarative Schema und somit auch die Ausführung der Aktivitäten durch die richtige Person oder Organisationseinheit zu gewährleisten, ist es notwendig, die deklarativ abgebildeten Aktivitäten durch ihre Rollenbezeichnungen zu ergänzen. Die einzelnen Aktivitäten werden bei der Migration nicht nur mit ihrer ursprünglichen Bezeichnung übernommen, sondern durch

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

die Beschriftung der Aktivität der ausführenden Instanz ergänzt. Zur Vereinheitlichung der für diese Arbeit migrierten Prozessmodelle wird nach der Übernahme der Beschriftung der Aktivitäten die Bezeichnung der Lane nach einem Spiegelstrich und die Bezeichnung des Pools in Klammern hinzugefügt.

Ein Beispiel hierfür ist in Abbildung 12 dargestellt: auf der linken Seite ist die im imperativen Prozessmodell abgebildete Aktivität *Lehrendendaten aktualisieren*, die sich in der Lane *Abt. III-1 Personalservice* im Pool *Universität Ulm* befindet, zu sehen. Bei der Übertragung in das deklarative Prozessmodell wird die Beschriftung in *Lehrendendaten aktualisieren - Abt. III-1 Personalservice (Universität Ulm)* umbenannt.

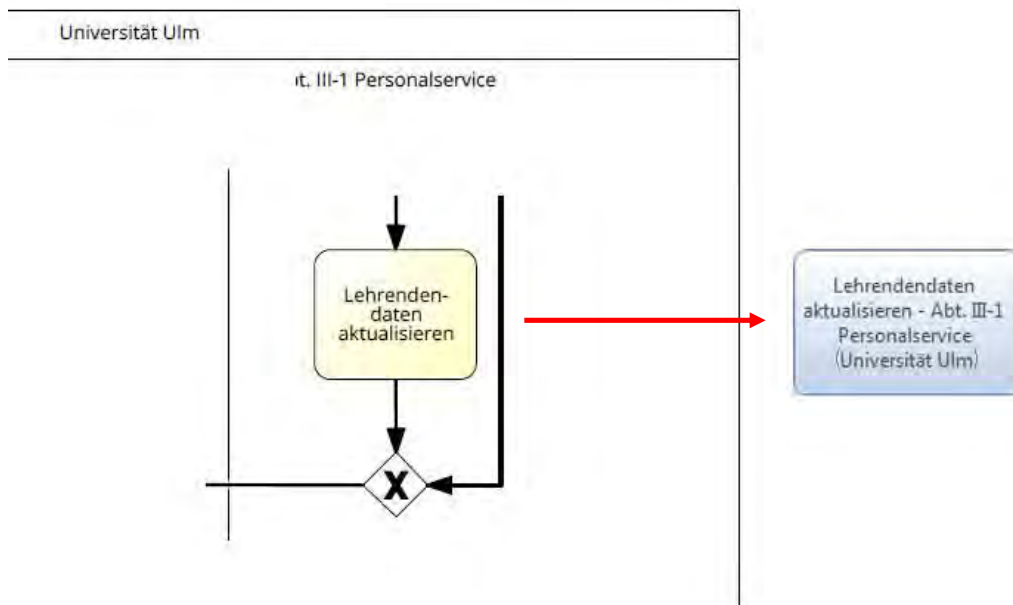


Abbildung 12: Integration von Rollen im deklarativen Prozessmodell

Durch diesen Workaround wird die korrekte Migration des Prozessmodells in das deklarative Schema und somit auch die Ausführung der Aktivitäten durch die richtige Person oder Organisationseinheit gewährleistet. Betrachtet man den Prozess in technischem Sinn, kann die Integration der Rollen nicht mit einer Ergänzung der Aktivitäten gelöst werden und verlangt nach einer Erweiterung der Sprache.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

- **Artefakte**

Artefakte, wie z.B. die Zuweisung eines IT-Systems zu einer Aktivität, werden in den vorliegenden deklarativen Prozessmodellen häufig verwendet. Sie dienen ähnlich wie die Rollenzuweisungen dazu, Kontextinformationen für den Nutzer bereit zu stellen. Artefakte spezifizieren, mit Hilfe welches Systems (z.B. CMS, uni-assist, Hochschulstart, SVA) eine Aktivität ausgeführt wird bzw. geben die Beteiligung des Systems an der Aktivität an.

In der deklarativen Abbildung steht jedoch keine Möglichkeit der Zuweisung von Artefakten zur Verfügung. Auch hier muss wie bei der Integration der Rollen zwischen dem technischen und dem fachlichen Prozess unterschieden werden.

Wird der Prozess aus der fachlichen Sicht betrachtet, wird auch hier auf einen sog. Workaround zurück gegriffen: Die Bezeichnung der Aktivität wird um das jeweilige Artefakt ergänzt und dieses schriftlich hinzugefügt.

Ein Beispiel dazu ist in Abbildung 13 zu sehen: Die im imperativen Prozessmodell abgebildete Aktivität *Bewilligungsbescheid erstellen* auf der linken Seite der Abbildung wird um das angehängte IT-System ergänzt. Folglich wird die Aktivität nach Ergänzung der Artefakte und Rollen in *Bewilligungsbescheid im CMS erstellen - Abt. I-2 Recht und Organisation (Uni Ulm)* umbenannt.



Abbildung 13: Integration von Artefakten im deklarativen Prozessmodell

Ohne diesen Workaround, der eine Lösung mit den durch die Prozessmodellierungssprache Declare zur Verfügung stehenden Mitteln darstellt, müsste die Sprache entsprechend erweitert werden. Somit ist auch in diesem Fall, analog zur Ergänzung der



## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Rollenzuweisungen, eine Integration der Artefakte notwendig. Sie ist Voraussetzung, um eine korrekte Ausführung des Prozesses zu gewährleisten und dem Nutzer die nötigen Hintergrundinformationen zur Verfügung zu stellen.

- **Kommunikationsereignisse**

Eine weitere Schwierigkeit bei der Migration der Prozessmodelle stellt die Abbildung von Kommunikationsereignissen dar. Diese werden imperativ durch eintretende oder auslösende Zwischenereignisse in Form eines Nachrichtensymbols dargestellt. Kommunikationsereignisse repräsentieren die Übermittlung bzw. den Empfang einer Nachricht. Ähnlich wie bei der bereits erwähnten Integration von Rollen und Artefakten sind Kommunikationsereignisse jedoch nicht in der deklarativen Prozessmodellabbildung vorgesehen.

Auch in diesem Fall muss zwischen fachlichem und dem technischem Prozess unterschieden werden. Aus technischer Sicht ist eine Erweiterung der Prozessmodellierungssprache Declare, um die Kommunikationsereignisse vollständig integrieren zu können, notwendig. Die Lösung aus fachlicher Sicht erfolgt intuitiv in Form eines Workarounds. Es muss zwischen einem Send- bzw. Receive-Nachrichtenereignis und einem Nachrichtenfluss unterschieden werden, der von einer Aktivität ausgeht. Handelt es sich um ein Send- und Receive-Nachrichtenereignis, wird das Nachrichtenereignis als eigenständige Aktivität in das deklarative Prozessmodell migriert. Hier ist es wichtig auf die korrekte Migration der Reihenfolge der einzelnen Aktivitäten zu achten damit die Sequenzbeziehung erhalten bleibt. Geht der Nachrichtenfluss von einer Aktivität aus, wird dieser ähnlich wie bei der Integration von Rollen und Artefakten in die bestehende Aktivität integriert und textuell ergänzt.

Ein Beispiel für die Migration eines Send- und Receive-Nachrichtenereignisses ist in Abbildung 14 zu sehen. Hier werden das auslösende und das eintretende Nachrichtenereignis *Nominierung senden* und *Nominierung empfangen* in die gleichnamigen und im deklarativen Prozessmodell abgebildeten Aktivitäten migriert. Diese Aktivitäten werden durch ein Chained-Succession-Constraint mit ihrer jeweiligen vorange-

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

henden bzw. nachfolgenden Aktivität verbunden. Somit sind diese beiden Aktivitäten bezüglich der Ausführung und der Reihenfolge eindeutig bestimmt und die Sequenzbeziehungen werden korrekt migriert.

Zur Vereinfachung der hier abgebildeten Prozessmodelle wird in dieser Arbeit auf die Migration der Kommunikationsereignisse in die deklarativen Prozessmodelle verzichtet.

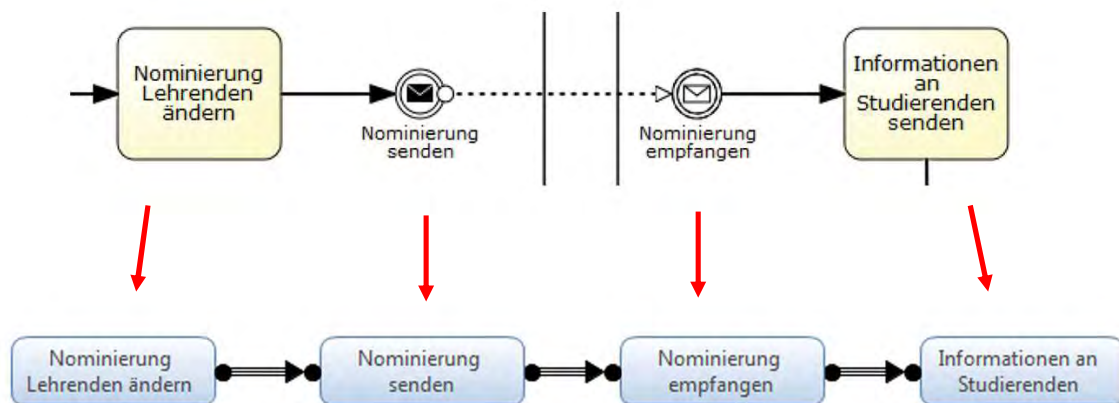


Abbildung 14: Integration von Nachrichtenergebnisse im deklarativen Prozessmodell

### 5.2.3 Zeitliche Aspekte: Die Frist

Zu zeitlichen Aspekten zählen in der imperativen Prozessmodellierung z.B. die Gesamtdauer eines Prozesses, die Dauer einer Aktivität, der zeitliche Abstand zwischen zwei aufeinander folgenden Aktivitäten oder die Frist. Die Frist beschreibt eine eingeschränkte Dauer und gibt den Zeitpunkt für das Eintreten eines Ereignisses an [Lan08]. Dies kann der frühestmögliche Zeitpunkt (Start), aber auch der spätestmögliche Zeitpunkt (Ende) sein, zu dem eine Aktivität begonnen werden darf bzw. abgeschlossen sein muss [Lan08]. Ähnlich wie bei den bereits thematisierten Aspekten der Rollen und Artefakte, ist in der deklarative Modellierung keine Verwendung zeitlicher Aspekte möglich, weder als Symbole noch in anderer Form.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Allgemein stellt die Abbildung zeitlicher Aspekte an sich ein sehr umfangreiches und komplexes Thema dar [LWR14] und würde deshalb den gesteckten Rahmen dieser Arbeit überschreiten. Es wird daher nur in geringem Umfang auf mögliche Lösungen für die Integration von zeitlichen Aspekten bei der deklarativen Prozessmodellabbildung eingegangen. Eine häufig auftretende zeitliche Einschränkung, vor allem innerhalb der im Rahmen dieser Arbeit betrachteten Prozessmodelle, ist der zeitliche Aspekt in Form von Fristen. Auf sie wird im Folgenden auch das Hauptaugenmerk gelegt.

Im imperativen Prozessmodell führt eine Frist nach deren Ablauf zur Beendigung einer Aktivität, an die sie angehängt ist. Meist folgt darauf eine weitere Aktivität, die an den Ablauf der Frist geknüpft ist. Die Frist ist symbolisch durch eine Uhr dargestellt, die durch einen schriftlichen Zusatz die zeitliche Begrenzung angibt (siehe Abbildung 15 oberer Teil). Deklarativ kann eine Frist als zusätzlicher Weg dargestellt und somit in den Prozess integriert werden. Sie kann jedoch nicht explizit in Form eines Symbols abgebildet werden.

Ein Beispiel für die Migration einer Frist von einem imperativen Prozess in einen deklarativen ist in Abbildung 15 zu sehen. Hier bestehen nach der Ausführung der Aktivität *Informationen überprüfen* prinzipiell drei verschiedene Möglichkeiten: bei Einhalten der Frist die Ausführung der Aktivität *zeugnisrelevante Informationen bearbeiten* und bei nicht Einhalten der Frist die Ausführung der Aktivität *Information erneut versenden* oder der Aktivität *Zeugnis, Urkunde und Diploma Supplement erstellen*.

Bei der Migration des imperativen Prozessmodells in das deklarative Schema wird die Frist, wie im unteren Teil von Abbildung 15 zu sehen, nicht explizit dargestellt. Die symbolhafte Darstellung der Frist, wie in Form einer Uhr im imperativen Prozessmodell, entfällt hier. Nach der Aktivität *Informationen im CMS überprüfen* folgt auch in diesem Prozessmodell eine der drei bereits genannten Aktivitäten. Dies wird durch die in der Abbildung im deklarativen Prozessmodell rot eingekreisten Constraints realisiert. Sie geben an, dass eine der drei Aktivitäten folgen muss.

Diese Lösung stellt für den Betrachter zwar keine bildliche Frist z.B. in Form eines Symbols dar, allerdings bleibt analog zum imperativen Prozessmodell die Semantik weitestgehend erhalten.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

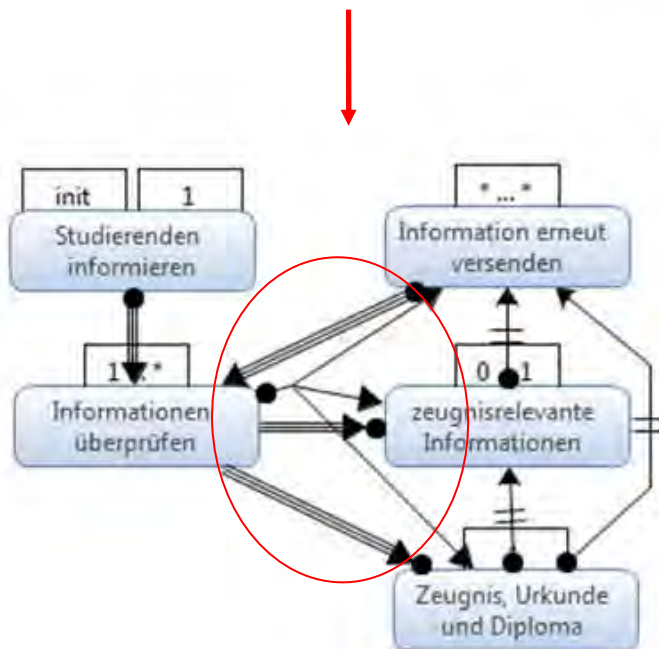
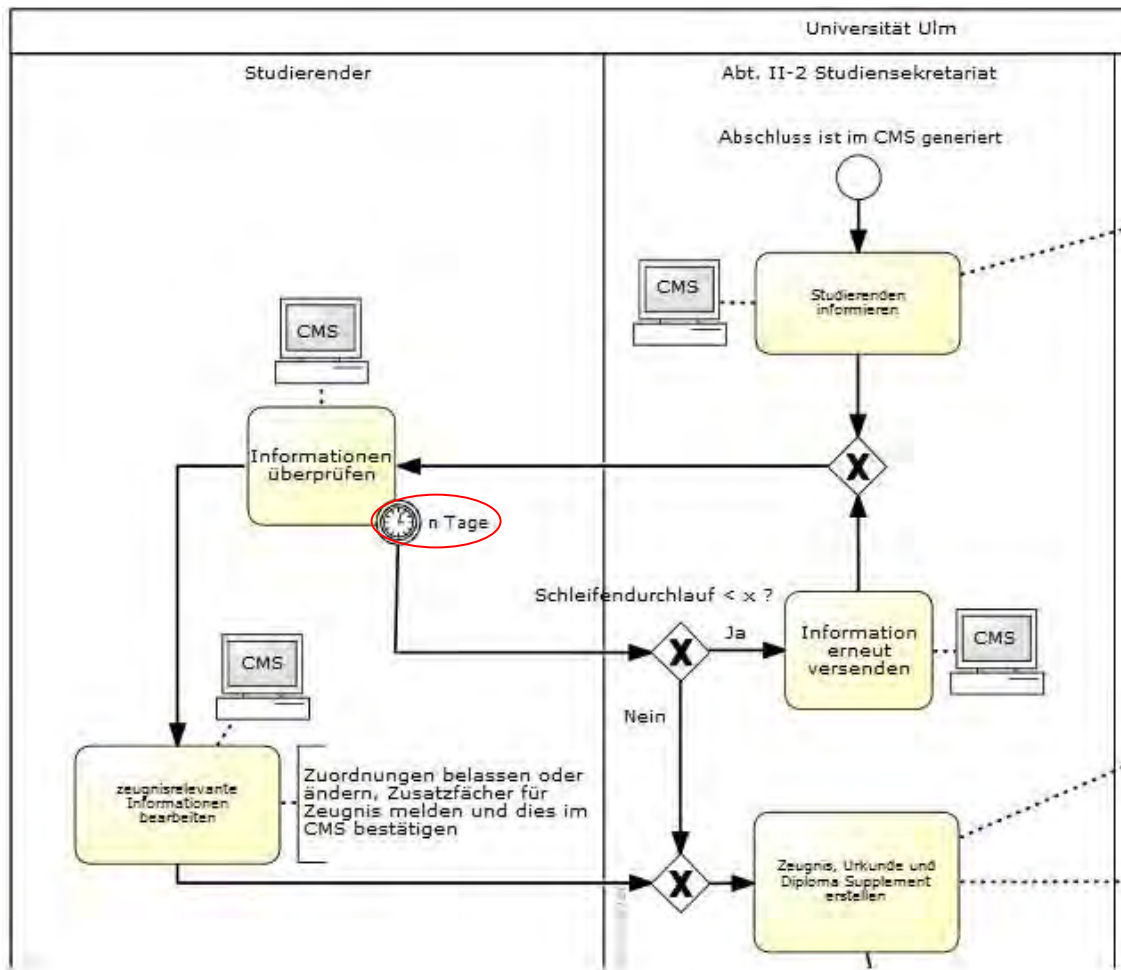


Abbildung 15: Integration zeitlicher Aspekte im deklarativen Prozessmodell: Die Frist

#### 5.2.4 Start- und Endereignisse

Einer der größten Unterschiede zwischen der imperativen und der deklarativen Prozessmodellierung ist die Verwendung von Start- und Endereignissen. Im Folgenden wird die Verwendung von Start- und Endereignissen sowie deren Unterscheidung nach Art und Funktion anhand von konkreten Beispielen veranschaulicht, dabei auftretende Schwierigkeiten diskutiert und mögliche Lösungswege vorgeschlagen.

- **Verwendung von mehreren Start- oder Endereignissen**

Imperative Prozessmodelle bilden Start- und Endereignisse in Form von Symbolen ab, die mehrfach innerhalb eines Prozessmodells verwendet werden können. Beispielhaft hierfür zeigt das imperative Prozessmodell im oberen Teil von Abbildung 14 die Verwendung der vier verschiedenen Startereignisse *Hat Beratungsbedarf* durch den *Studierenden*, *Doktoranden*, *Bewerber* und *Schüler*. Jeder dieser Startpunkte kann den Prozess anstoßen.

Die deklarative Prozessmodellierung sieht im Gegensatz zur imperativen Prozessmodellierung keine Start- und Endereignisse in Form von Symbolen vor [HZSHRPW13]. In der deklarativen Prozessmodellabbildung wird die Aktivität, die zuerst ausgeführt werden soll, mit dem Zusatz *Init* gekennzeichnet. Die Aktivität, die zuletzt ausgeführt werden soll erhält den Zusatz *Last*. Die Zusätze dürfen innerhalb eines Prozessmodells jedoch nur einmal verwendet werden und ermöglichen somit keine Kennzeichnung mehrerer Aktivitäten als Start- oder Endereignis. Ein deklarativer Prozess kann genauso wie ein imperativer mit einer oder mit mehreren Aktivitäten beginnen oder enden, die jedoch nicht eindeutig gekennzeichnet werden können. Solange keine Aktivität den Zusatz *Init* oder *Last* erhält oder durch Einschränkungen der eindeutige Pfad des Prozesses bestimmt wird, kann der Prozess mit jeder beliebigen Aktivität beginnen oder enden. Soll jedoch nur eine bestimmte Anzahl von Aktivitäten innerhalb eines Prozessmodells als Start- (siehe Beispielmmodell in Abbildung 16) oder Endpunkt fungieren, stellt dies eine große Herausforderung für die deklarative Modellierung dar. Die Notwendigkeit diese Aktivitäten durch Einschränkungen, wie Constraints, zu cha-

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

rakterisieren, führt schnell dazu, dass die einzelnen Pfadmöglichkeiten kaum mehr erkennbar sind. Das Prozessmodell wird für den Betrachter unübersichtlich und schlecht lesbar.

Ein Beispiel für die Migration eines imperativen Prozessmodells mit mehreren Start- oder Endpunkten in das deklarative Schema ist in Abbildung 16 zu sehen. Das imperative Prozessmodell beginnt mit einem der vier Startpunkte *Hat Beratungsbedarf* mit der Rolle *Studierender, Bewerber, Doktorand* oder *Schüler*. Jeder dieser Startpunkte wird von der Aktivität *Beratungsbedarf übermitteln* (mit der entsprechenden Rolle) gefolgt. Anschließend wird in allen Fällen die Aktivität *Beratung planen und abstimmen - Abt. II-4 Zentrale* mit der Rollenzuweisung *Studienberatung (Universität Ulm)* ausgeführt. Auch im deklarativen Prozessmodell, das im unteren Teil von Abbildung 16 zu sehen ist, beginnt der Prozess mit einer der Aktivitäten *Beratungsbedarf übermitteln* durch den *Studierenden, Doktoranden, Bewerber* und *Schüler*. Im Gegensatz zum imperativen Prozessmodell sind hier die Aktivitäten, die den Prozess anstoßen können, nicht gekennzeichnet und können daher vom Betrachter nicht eindeutig identifiziert werden. Es ist jedoch deutlich zu erkennen, dass dieser bedingt durch die große Anzahl von notwendigen Constraints weniger übersichtlich ist.

Der in Abbildung 16 dargestellte Ausschnitt eines Prozessmodells bildet verschiedene Varianten des Prozessmodells ab. Im Rahmen der imperativen Abbildung von Prozessvarianten wurde festgestellt, dass die Abbildung mit hohem Anpassungs- und Wartungsaufwand verbunden ist [Hal09]. Da im imperativen Beispielmodell nur eine geringe Anzahl an Prozessvarianten abgebildet ist, kann das Prozessmodell übersichtlich dargestellt werden. Die Modellierung dieser Prozessvarianten stellt im deklarativen Prozessmodell dagegen eine große Herausforderung dar. Bei dem Beispielprozessmodell in Abbildung 16 ist kaum Flexibilität gefragt, sondern die Ausführung des Prozessmodells nach einer begrenzten Anzahl genau festgelegter Pfade vorgegeben. Bei Modellen, die eine noch größere Anzahl von verschiedenen Startereignissen enthalten, kann es sinnvoll sein, diese in verschiedene Einzelmodelle oder Subprozesse aufzuspalten, um die Übersichtlichkeit zu erhöhen.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

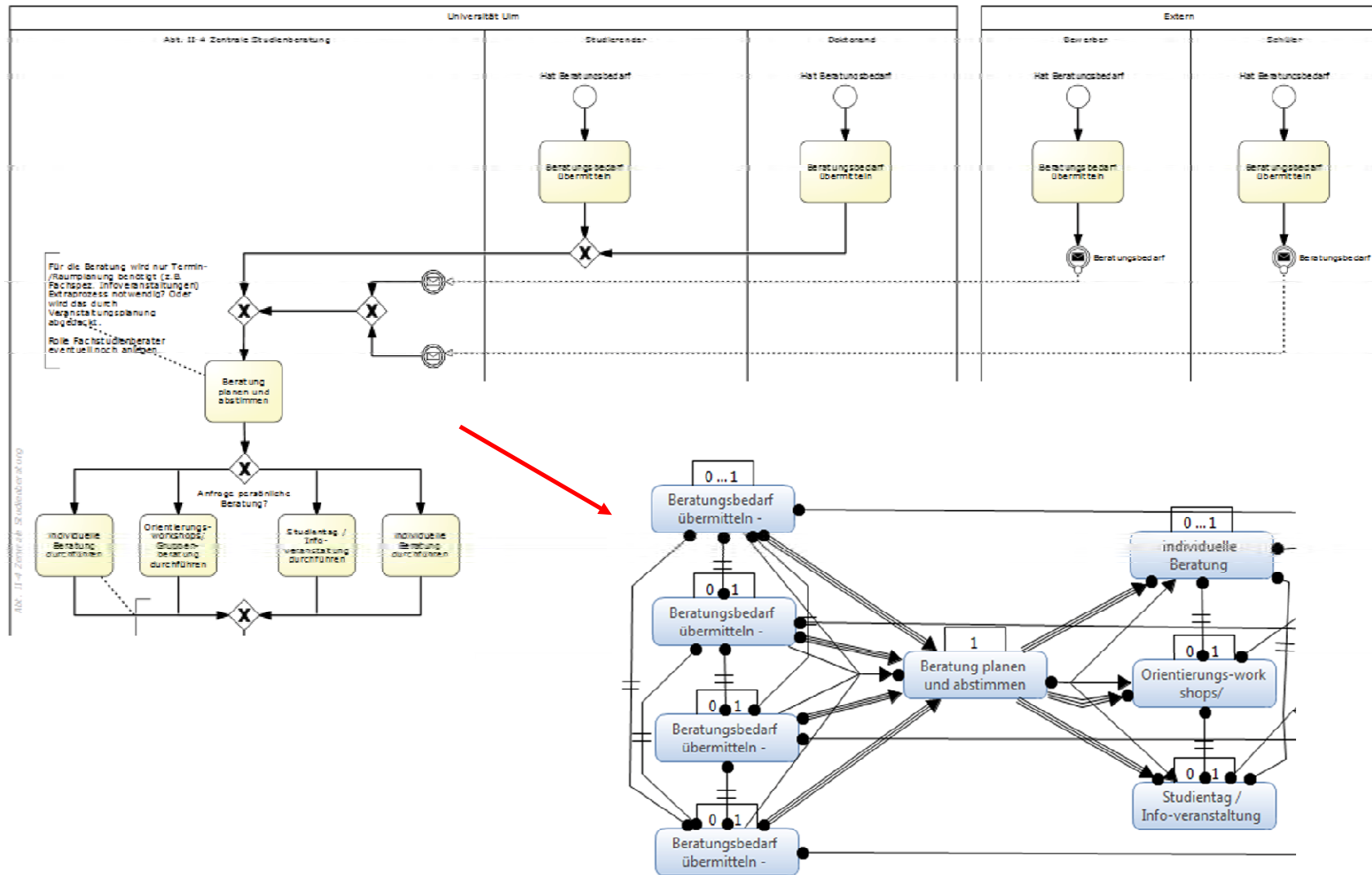


Abbildung 16: verschiedene Startpunkte im deklarativen Prozessmodell

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Zusammenfassend kann festgestellt werden, dass die Kennzeichnung mehrerer Start- oder Endereignisse innerhalb eines deklarativen Prozessmodells nicht möglich ist. Dies führt zu einer schlechteren Lesbarkeit für den Betrachter.

- **Unterschiedliche Arten von Start- und Endereignissen**

Die imperative Prozessmodellabbildung ermöglicht nicht nur die eindeutige Deklaration von Start- und Endereignissen, sondern ebenfalls deren Unterscheidung nach Art und Funktion. Beispielsweise können Fehler-, Abbruch- oder "normale" Endereignisse und Nachrichten-, Eskalations- oder Fehlerstartereignisse durch ein spezifisches Symbol abgebildet werden (siehe Kapitel 2.2). In der deklarativen Prozessmodellabbildung ist im Gegensatz dazu keine bildliche Darstellung in Form von Symbolen oder die Unterscheidung von Start- und Endereignissen nach Art oder Funktion möglich. Somit ist es für den Betrachter nicht ersichtlich, aus welchem Grund ein Prozess beendet oder begonnen wird.

Ein Beispiel für ein Prozessmodell mit mehreren Start- bzw. Endereignissen, die nach ihrer Art und Funktion unterschieden werden, ist in Abbildung 17 zu sehen. Hier wird im imperativen Prozessmodell im oberen Teil einerseits das "normale" Endereignis *STG-Wechsel erfolgt* und andererseits das Fehlerendereignis *STG-Wechsel nicht erfolgt* verwendet.

Bei der Migration des Prozessmodells in ein deklaratives Schema wird auf die Verwendung spezifischer Endereignisse verzichtet. Wie in Abbildung 17 im unteren Teil zu sehen, werden lediglich die Aktivitäten übernommen. Die Aktivitäten werden so lange anhand der Bedingungen ausgeführt, bis der Prozess beendet ist und sich keine Aktivität mehr im Zustand *waiting* oder *running* befindet. Die korrekte Abbildung des Prozesses bleibt also in jedem Fall gewährleistet, da dieses äquivalent zu seinem Vorlagemodell in imperativer Form, auch bei der Migration in das deklarative Schema entweder mit der Aktivität *Studiengangwechsel im CMS - Abt. II.2 Studiensekretariat (Uni Ulm)* durchführen oder *Annahme löschen (nicht zulassungsfreie Studiengänge) - Abt. II.2 Studiensekretariat (Uni Ulm)* beendet wird. Falls nötig, kann somit überprüft werden, durch welche Aktivität der Prozess beendet oder begonnen wurde, um zu entscheiden ob eine Kompensation not-



## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

wendig ist. Jedoch ist es für den Nutzer des Prozessmodells nicht ersichtlich, ob im ursprünglichen Prozess nach dieser Aktivität ein "normales" Ende oder ein Abbruch stattfindet.

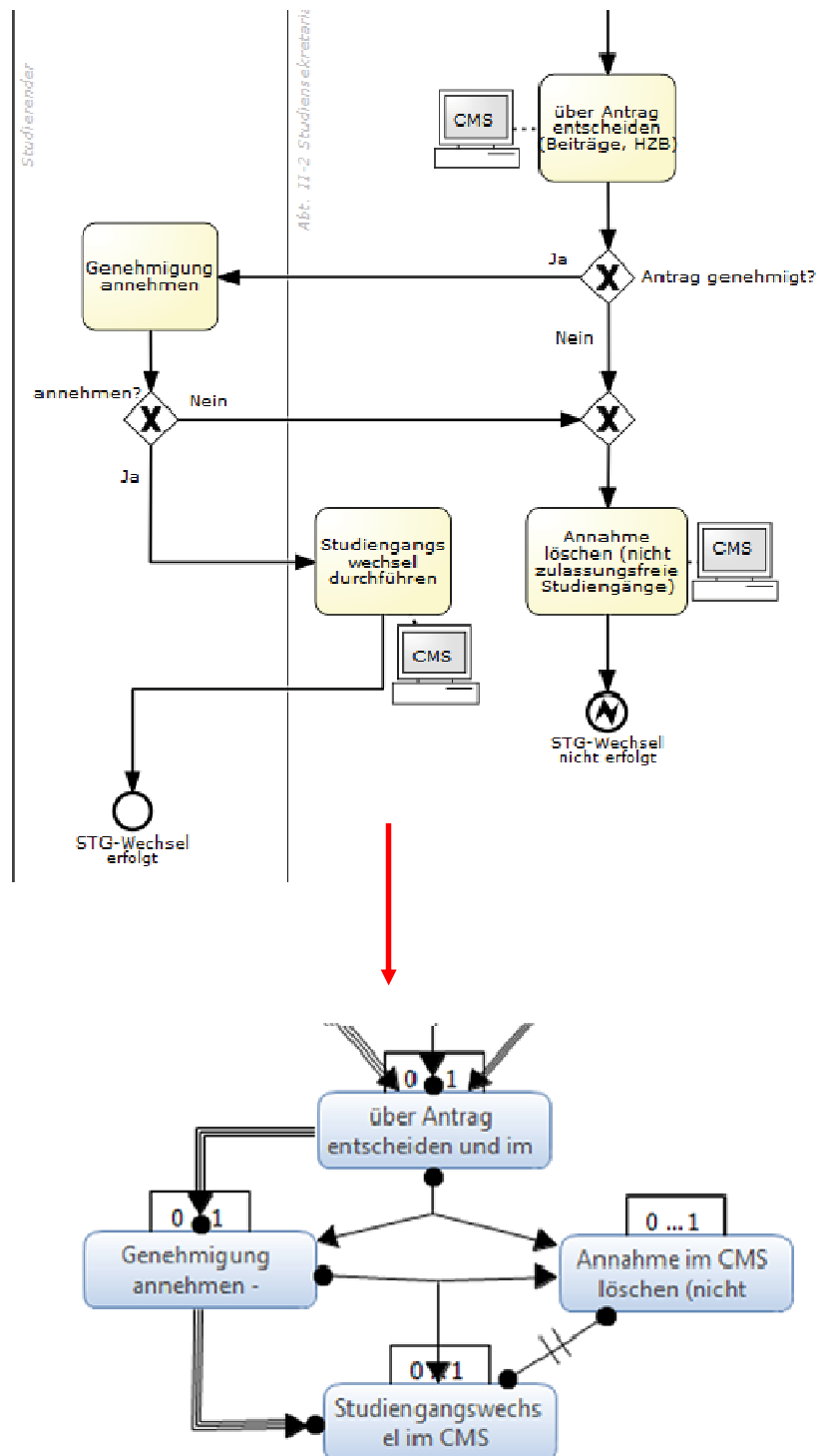


Abbildung 17: Art der Endereignisse wird im deklarativen Prozessmodell nicht erfasst

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Da die Art des Startes oder Endes in vielen Fällen nicht von vorrangiger Bedeutung ist, kann häufig auf die Abbildung und Unterscheidung des einzelnen Anfangs- bzw. Endereignisses verzichtet werden. Bei der deklarativen Modellierung ist es nur wichtig, dass der Anfang oder das Ende des Prozesses durch die richtige Aktivität gewährleistet werden kann.

Eine Ausnahme bilden Endereignisse in einem Prozess, die eine Kompensation durch einen Subprozess erfordern. Kompensationsendereignisse dienen der Fehlerbehandlung nach einem Abbruch in einem imperativen Prozessmodell. Dieses Kompensationsendereignis und die Fehlerbehandlung durch einen Subprozess gehen bei der Migration in das deklarative Schema verloren. Es ist daher notwendig, dass die Aktivität, die den Prozess beendet, explizit abgegriffen wird. Weiterhin ist es sinnvoll, diese Aktivität entsprechend zu kennzeichnen. Damit wird gewährleistet, dass das Kompensationsendereignis korrekt ausgeführt und der Fehler entsprechend durch den Subprozess behandelt wird.

Zusammenfassend kann festgehalten werden, dass die Unterscheidung der Start- und Endereignisse nach ihrer Art in der deklarativen Prozessmodellabbildung nicht möglich ist.

- **Startpunkte ohne folgende Aktivitäten**

Existieren in einem imperativen Prozessmodell zwei oder mehr verschiedene Startpunkte, die durch ein Gateway synchronisiert werden, wobei zwischen dem Startereignis und dem Gateway keine Aktivität vorkommt, so werden diese im deklarativen Schema nicht abgebildet. Der Grund hierfür ist, dass wie zuvor beschrieben, im deklarativen Schema keine expliziten Start- oder Endereignisse für die Modellierung vorgesehen sind. Hier muss unterschieden werden, ob es sich um eine technische oder fachliche Betrachtung des Prozesses handelt. Im technisch betrachteten Prozessmodell sind die explizit dargestellten Startereignisse irrelevant. Betrachtet man das Prozessmodell hingegen von der fachlichen Seite, ist deren Unterscheidung relevant.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

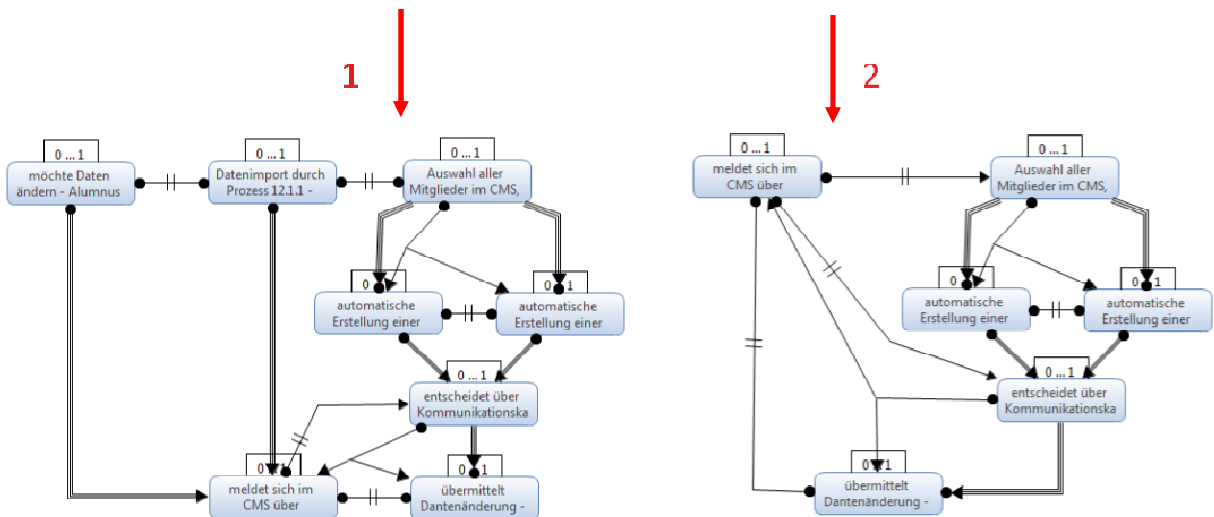
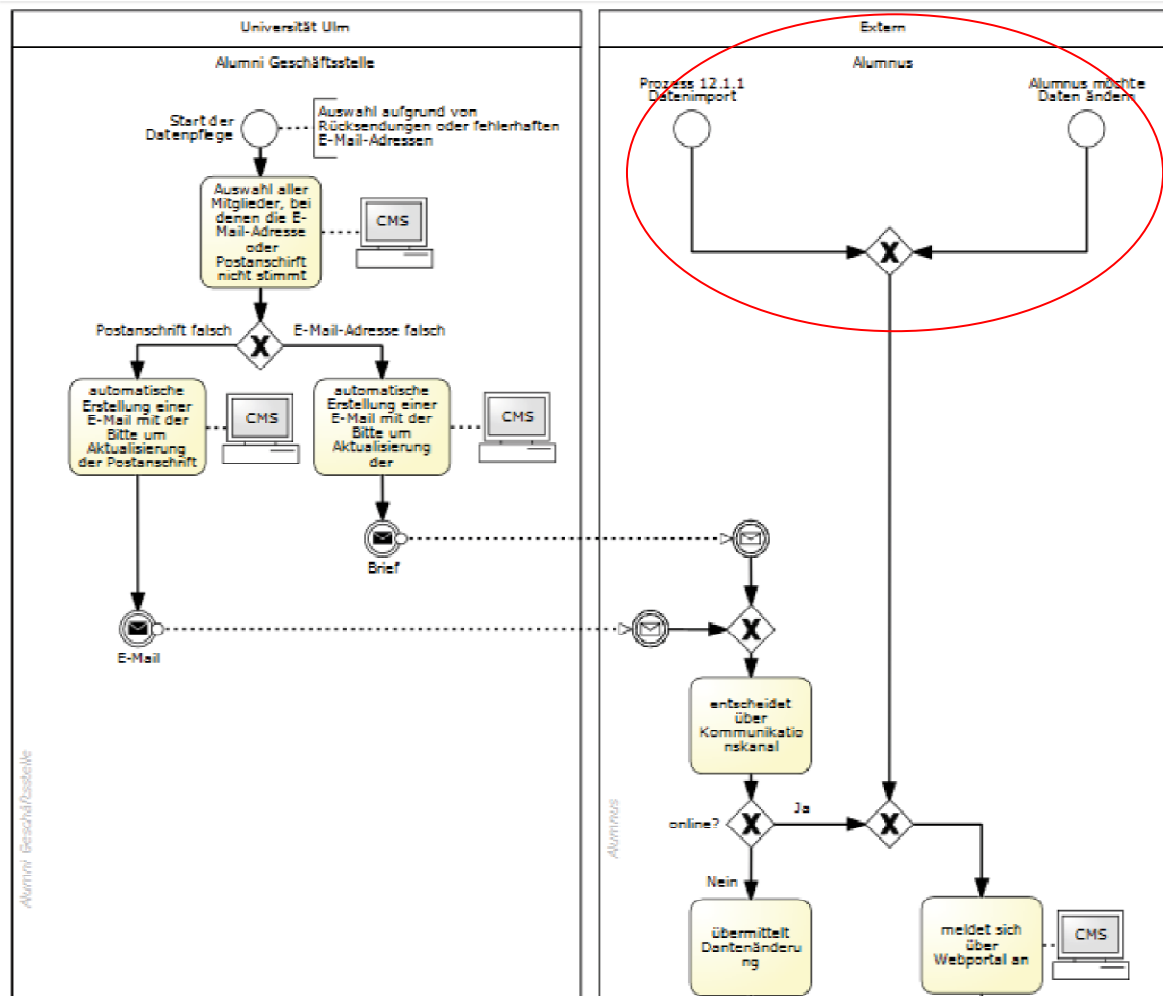


Abbildung 18: Startpunkte ohne nachfolgende Aktivitäten im deklarativen Prozessmodell

Für einen solchen Fall gibt es zwei Möglichkeiten, die im folgenden beschrieben werden:  
 Ein Beispiel für ein solches Szenario, ist auf Abbildung 18 im oberen Teil im imperativen

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Prozessmodell dargestellt: Die beiden parallelen Startpunkte *Prozess 12.1.1 Datenimport* und *Alumnus möchte Daten ändern*, welche die möglichen Ausgangsbedingungen darstellen, werden ohne vorherige Aktivität durch ein datenbasiertes exklusives Gateway zusammengeführt.

Die erste Möglichkeit, um einen solchen Fall im deklarativen Schema abzubilden ist im linken Teil von Abbildung 18 unten unter Variante 1 zu sehen. Hier wurden die beiden Startereignisse nicht explizit in das deklarative Prozessmodell übertragen. Der Prozess wird dennoch korrekt abgebildet und beginnt wie der imperative Prozess entweder mit der Aktivität *Auswahl aller Mitglieder, bei denen die E-Mail-Adresse oder Postanschrift nicht stimmt* oder mit der Aktivität *meldet sich über Webportal an*. Es ist jedoch nicht ersichtlich, dass die Aktivität *meldet sich über Webportal an* durch zwei verschiedene Startereignisse ausgelöst werden kann.

Die explizite Abbildung der beiden Startereignisse im deklarativen Prozessmodell wird in Variante 2 im rechten Teil von Abbildung 18 unten abgebildet. Die beiden Startereignisse können als zwei zusätzliche Aktivitäten ohne Funktion modelliert werden. In diesem Fall beginnt der Prozess mit einer der drei Aktivitäten *Auswahl aller Mitglieder aus dem CMS, bei denen die E-Mail-Adresse Alumni Geschäftsstelle (Uni Ulm) oder Postanschrift nicht stimmt, möchte Daten ändern - Alumnus (extern) oder Datenimport durch Prozesse 12.1.1 - Alumnus (extern)*. Auch hier wird das Prozessmodell deklarativ korrekt abgebildet, mit dem Unterschied, dass die zwei Startpunkte als Aktivitäten integriert wurden.

Zusammenfassend ist es je nach Bedeutung der Startereignisse möglich, diese als Aktivitäten einzufügen und somit für den Betrachter deklarativ explizit darzustellen. Beginnt man jedoch wie üblich mit den darauffolgenden Aktivitäten des Prozesses, wird dieser ebenfalls korrekt abgebildet. Hier kann der Modellierer abwägen, welche Variante den Prozess korrekt wieder gibt und ob die Kontextinformation relevant ist.

- **Abbruch als zusätzliche Aktivität**

Wie aus den bereits beschriebenen Schwierigkeiten bezogen auf die Abbildung von Start- und Endereignissen im deklarativen Schema hervorgeht, ist eine eindeutige Lösung hier nicht immer möglich. Um jedoch die fehlerfreie Abbildung der Prozesse zu gewährleis-

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

ten, ist in der deklarativen Modellabbildung in einzelnen Fällen das Einfügen einer Aktivität zur expliziten Abbildung eines Endereignisses notwendig. Hier handelt es sich um Endereignisse, die sich innerhalb eines Pfades des Prozessmodells befinden und frühzeitig zu einem Abbruch des Prozesses führen. Hier muss je nach Fall unterschieden werden, ob eine zusätzliche Aktivität zur korrekten Abbildung des Prozesses notwendig ist oder nicht.

Ein Beispiel hierfür ist in Abbildung 19 zu sehen. Nach dem Gateway mit der Bezeichnung *Kann nachrücken?* folgen die zwei Abbruch-Endereignisse *vorläufig abgelehnt* und *abgelehnt*. Da sich zwischen dem Gateway und den Endereignissen keine Aktivitäten befinden, wird die Unterscheidung der beiden Endereignisse nicht in das deklarative Schema übernommen.

Hier gibt es zwei Lösungsmöglichkeiten, die im Folgenden erläutert werden:

Im ersten Fall werden die bestehenden Aktivitäten aus dem imperativen Schema in das deklarative übernommen und die Abbruch-Ereignisse nicht explizit abgebildet. Diese Variante der Migration ist auf Abbildung 19 unten links unter Variante 1 zu sehen. Der Prozess endet entweder nach der Aktivität *Brief mit Ergebnis erstellen und versenden* oder es folgt die Aktivität *Bankdaten und Einwilligungen bzügl. Informationen an Förderer formlos zusammenfassen und übermitteln*. Das Modell kann deklarativ korrekt ausgeführt werden und wird ebenfalls an den Stellen beendet, wo sich imperativ die Abbruchereignisse befinden. Allerdings ist im deklarativen Prozessmodell für den Betrachter weder erkennbar, dass es sich um ein Abbruchereignis handelt, noch um welche Art von Abbruch.

Der andere Fall, der im unteren Teil von Abbildung 19 auf der rechten Seite unter Variante 2 dargestellt ist, sieht die Integration der Abbruch-Endereignisse in Form von zusätzlichen Aktivitäten vor. Hier endet das Modell entweder mit den Aktivitäten *vorläufig abgelehnt - Studierender (Uni Ulm)*, *abgelehnt - Studierender (Uni Ulm)*, oder fährt mit der Aktivität *Bankdaten und Einwilligungen bezügl. Informationen an Förderer formlos zusammenfassen und übermitteln* fort. Hier wurden die Abbruch-Endereignisse vom imperativen Prozess als Aktivitäten in das deklarative Schema übernommen. Es ist deutlich zu er-

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

kennen, dass bei zunehmender Komplexität des Prozesses, die Übersichtlichkeit abnimmt und selbst bei einfachen Sachverhalten schnell ansteigt.

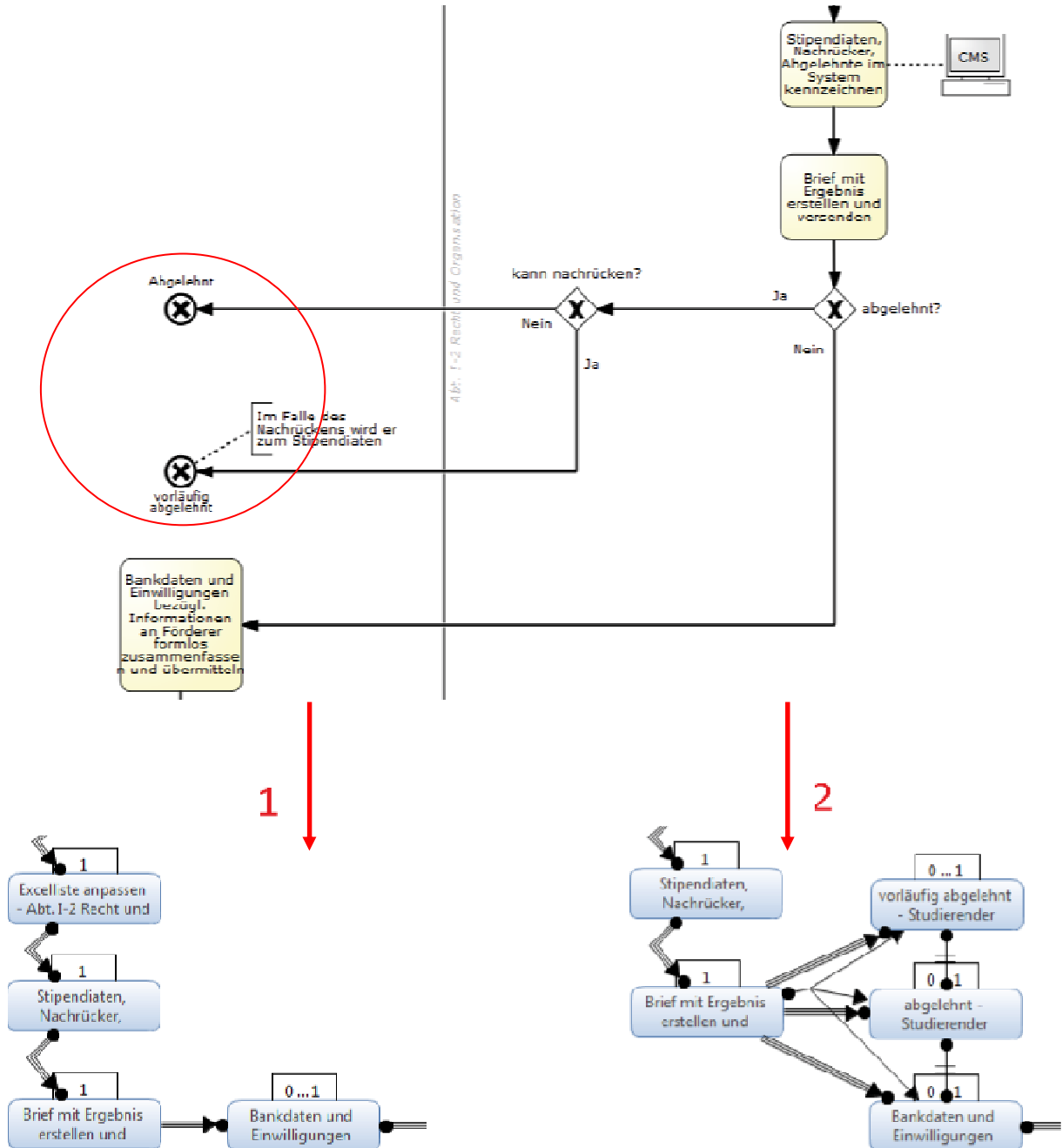


Abbildung 19: Endpunkte ohne voran gehende Aktivitäten im deklarativen Prozessmodell

Ist die Unterscheidung der Endpunkte für den Prozess von Bedeutung, können diese durch Hinzufügen von Aktivitäten berücksichtigt und in das deklarative Modell integriert

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

werden. Andernfalls endet der Prozess mit der letzten auszuführenden Aktivität. Auch hier kann der Modellierer abwägen, mit welcher Darstellung das Modell korrekt abgebildet wird.

### 5.2.5 Gateways

- **datenbasiertes exklusives Gateway**

Datenbasierte exklusive Gateways dienen dazu, eine "entweder-oder-Entscheidung" zwischen einer bestimmten Anzahl von möglichen Pfaden abzubilden. Die Abbildung datenbasierter exklusiver Gateways im deklarativen Schema ist mit Einschränkungen möglich, da hier die Entscheidungsregel nicht übertragen werden kann.

Abbildung 20 stellt beispielhaft die Migration eines Gateways vom imperativen in das deklarative Schema dar. Im deklarativen Prozessmodell in Abbildung 20 auf der rechten Seite folgt auf die Aktivität *Antrag aus 3. FS oder höher - Abt. II-2 Studiensekretariat (Uni Ulm)* entweder die Aktivität *Antrag in das 1. FS - Abt. II-2 Studiensekretariat (Uni Ulm)* oder *Antrag in ein höheres Fachsemester - Abt. II-2 Studiensekretariat (Uni Ulm)*. Dass nur eine der beiden Aktivitäten ausgeführt werden darf, wird deklarativ durch das Constraint *Not Co-Existence* sicher gestellt, welches in Abbildung 20 rot eingrahmt ist. Es ist zu sehen, dass durch die Aufspaltung der Pfeile eine modellgetreue und einfache Modellierung sowie eine für den Betrachter bildhafte Darstellung gegeben ist, die sich leicht interpretieren lässt und für den Nutzer verständlich abgebildet ist.

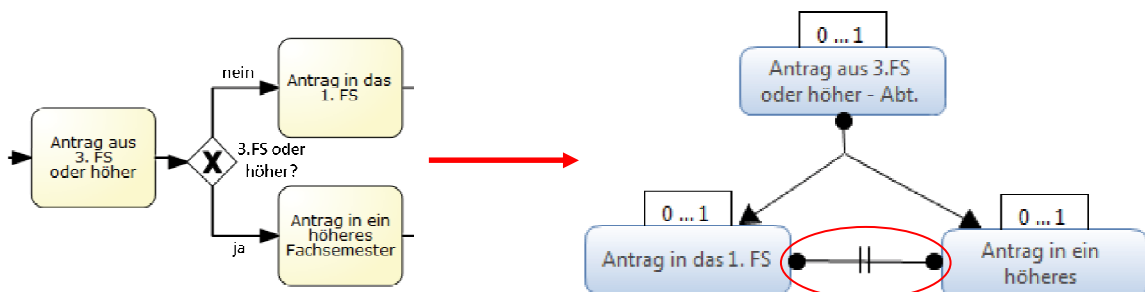


Abbildung 20: datenbasiertes exklusives Gateways im deklarativen Prozessmodell

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Die Einschränkung der korrekten Migration datenbasierter exklusiver Gateways stellt jedoch die Entscheidungsregel dar. Das datenbasierte exklusive Gateway des in Abbildung 20 auf der linken Seite imperativ dargestellten Prozessmodells beinhaltet die Entscheidungsregel *3. FS oder höher?*, die mit *ja* oder *nein* beantwortet werden kann. Folglich wird einer der beiden Pfade mit der Aktivität *Antrag in das 1. FS - Abt. II-2 Studiensekretariat (Uni Ulm)* oder *Antrag in ein höheres Fachsemester - Abt. II-2 Studiensekretariat (Uni Ulm)* ausgeführt. Diese Entscheidungsregel kann im deklarativen Prozessmodell nicht wieder gegeben werden. Es wird zwar einer der beiden Pfade gewählt, allerdings nicht basierend auf der explizit dargestellten Entscheidungsregel.

- **Paralleles Gateway**

In der imperativen Prozessmodellierung kann ein paralleles Gateway verwendet werden, um einen Pfad in mehrere parallel ablaufende Pfade zu untergliedern. Diese laufen zeitgleich ab und werden wiederum durch ein paralleles Gateway synchronisiert. Voraussetzung hierfür ist, dass alle Aktivitäten bei der Wiederzusammenführung der Pfade durch das schließende Gateway ausgeführt worden sein müssen, bevor der Prozesspfad fortgesetzt werden kann. Im Gegensatz zur imperativen Prozessmodellierung erlaubt die deklarative Prozessmodellierung keine gleichzeitige Ausführung parallel ablaufender Aktivitäten.

Im imperativen Prozessmodell in Abbildung 21 auf der linken Seite wird der Pfad nach der Aktivität *Bewilligungsbescheid erstellen* in die beiden parallel ablaufenden Pfade mit den Aktivitäten *Bewilligungsbescheid an Student verschicken oder Bewilligungsbescheid an Abt. IV-3 Kasse verschicken* und *Zahlung an Stipendiaten anweisen* untergliedert. Diese Aktivitäten laufen gleichzeitig ab und werden anschließend wieder durch ein paralleles Gateway synchronisiert. In Abbildung 21 auf der rechten Seite ist die Übertragung des Prozessausschnitts in ein deklaratives Schema zu sehen. Hier wird vom Prozess vorgegeben, dass die beiden Aktivitäten *Bewilligungsbescheid an Abt. IV-3 Kasse verschicken - Abt. I-2 Recht und Organisation (Uni Ulm)* und *Zahlung an Stipendiaten anweisen - Abt. IV-3 Kasse (Uni Ulm)* aufeinander folgen müssen. Der Ablauf der Aktivitäten, die auf die Aktivität *Bewilligungsbescheid erstellen - Abt. I-2 Recht und Organisation (Uni Ulm)* ist



## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

nicht weiter eingeschränkt, allerdings können diese im deklarativen Prozessmodell nicht parallel, sondern ausschließlich nacheinander ausgeführt werden.

Da parallele Gateways in der deklarativen Prozessmodellabbildung nicht zeitgleich ablaufen dürfen, dies jedoch in manchen Fällen gefordert ist, wäre hier eine Erweiterung der Prozessmodellierungssprache sinnvoll. Das in Abbildung 21 vorgestellte Prozessmodellbeispiel stellt in der deklarativen Form eine nicht zufriedenstellende Lösung dar, da die eigentlich parallel ablaufenden Aktivitäten hintereinander ausgeführt werden müssen.

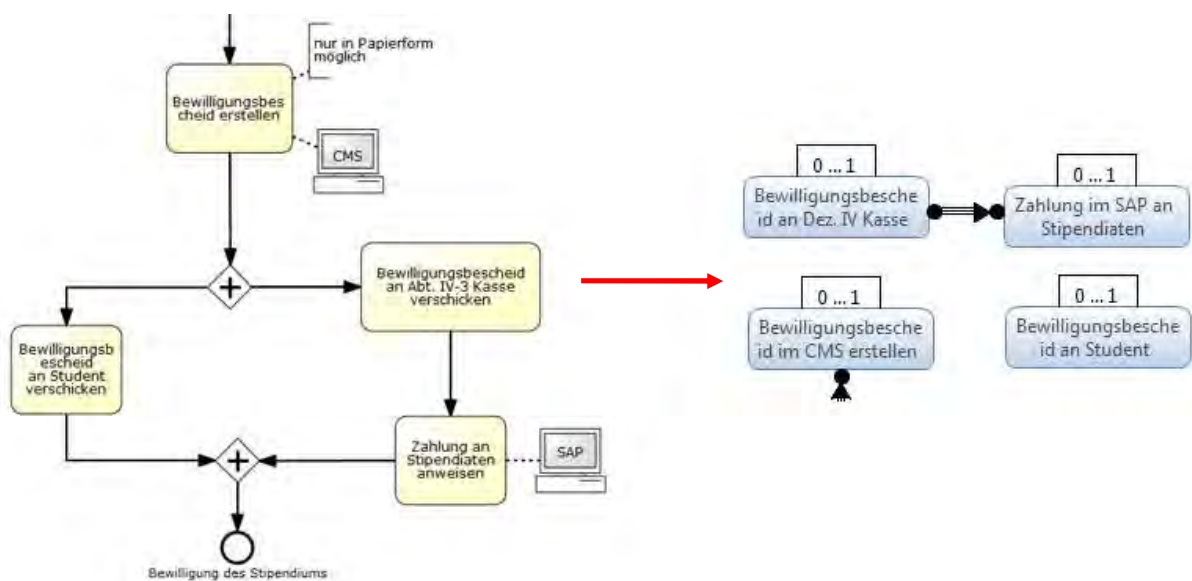


Abbildung 21: paralleles Gateways im deklarativen Prozessmodell

- **Viele aufeinander folgende Gateways ohne Aktivitäten**

Ähnlich wie bei Endereignissen ohne vorangehende Tasks nach einem Gateway, können mehrere Gateways, die im imperativen Prozessmodell ohne dazwischenliegende Aktivitäten aufeinander folgen, im deklarativen Schema nicht explizit abgebildet werden. Bei der Migration eines imperativen Prozessmodells mit vielen aufeinander folgenden Verzweigungen, jedoch wenigen oder keinen Tasks, in das deklarative Schema, ist das Prozessmodell zwar korrekt übertragbar, jedoch werden die Gateways nicht übernommen.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Im Vordergrund steht hier die Fragestellung: *Wann ist der Prozess zu Ende?*, lässt jedoch das *Wie?* außen vor. Wie bereits thematisiert, werden im deklarativen Prozessmodell keine Entscheidungsregeln der Gateways abgebildet, wodurch ein großer Teil des Kontextes verloren geht.

In dem Fall des imperativen Prozessmodells in Abbildung 22 im oberen Teil folgen z.B. auf die letzte Aktivität *3.2.1. Prüfung* mehrere Gateways, die zu den verschiedenen Endereignissen *Ende/Exma* und *Prüfung erfolgreich durchgeführt* führen. Werden hier bei der Migration ausschließlich die Aktivitäten übernommen und in das deklarative Schema übertragen, befinden sich nach der Aktivität *3.2.1. Prüfung - Prüfer (Uni Ulm)* keine Verzweigungen oder Endereignisse, sondern bei erfolgreicher Ausführung endet der Prozess mit dieser Aktivität. Es wird deutlich, dass hier z.B. die Entscheidungsregel *Prüfung bestanden?* nicht übernommen wird. Der Prozess würde, wie in Abbildung 22 im unteren Teil deklarativ dargestellt, beispielsweise mit der Aktivität *3.2.1. Prüfung - Prüfer (Uni Ulm)* enden, ohne dass der Nutzer wüsste, ob die Prüfung bestanden oder ein Freiversuch möglich ist. Folglich geht ein großer Teil der für das Prozessmodell notwendigen Kontextinformation verloren. Der Nutzer erfährt hier nicht wie und wieso das Prozessmodell beendet wird, sondern erhält nur die Information, durch welche Aktivität der Prozess beendet wird.

Um das Prozessmodell inhaltlich korrekt und für den Nutzer verständlich abbilden zu können, muss der Modellierer abwägen, welche Verzweigungen und vor allem Entscheidungsregeln für das Modell notwendig sind und diese in Form von zusätzliche Aktivitäten einfügen. Es ist jedoch anzumerken, dass das Hinzufügen von zusätzlichen Aktivitäten zur Abbildung von Entscheidungsregeln lediglich durch den Bezug zum Kontext begründet wird. Das Einfügen der Aktivitäten ist hier nicht für die korrekte Abbildung des Prozesses notwendig.

Ohne das Hinzufügen zusätzlicher Aktivitäten, um eventuell für das Verständnis notwendige Gateways abzubilden, bleibt das Modell übersichtlicher. Die Aktivitäten bleiben hier auf ein Minimum beschränkt und lenken nicht durch zusätzlichen Aktivitäten vom Hauptprozess ab

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

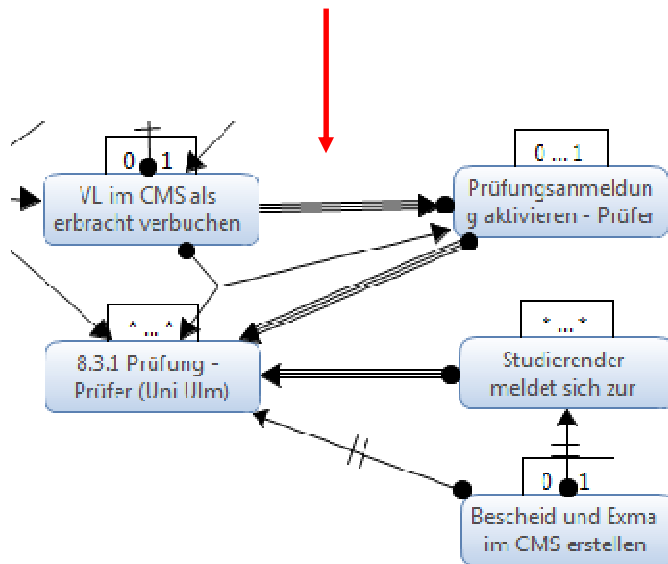
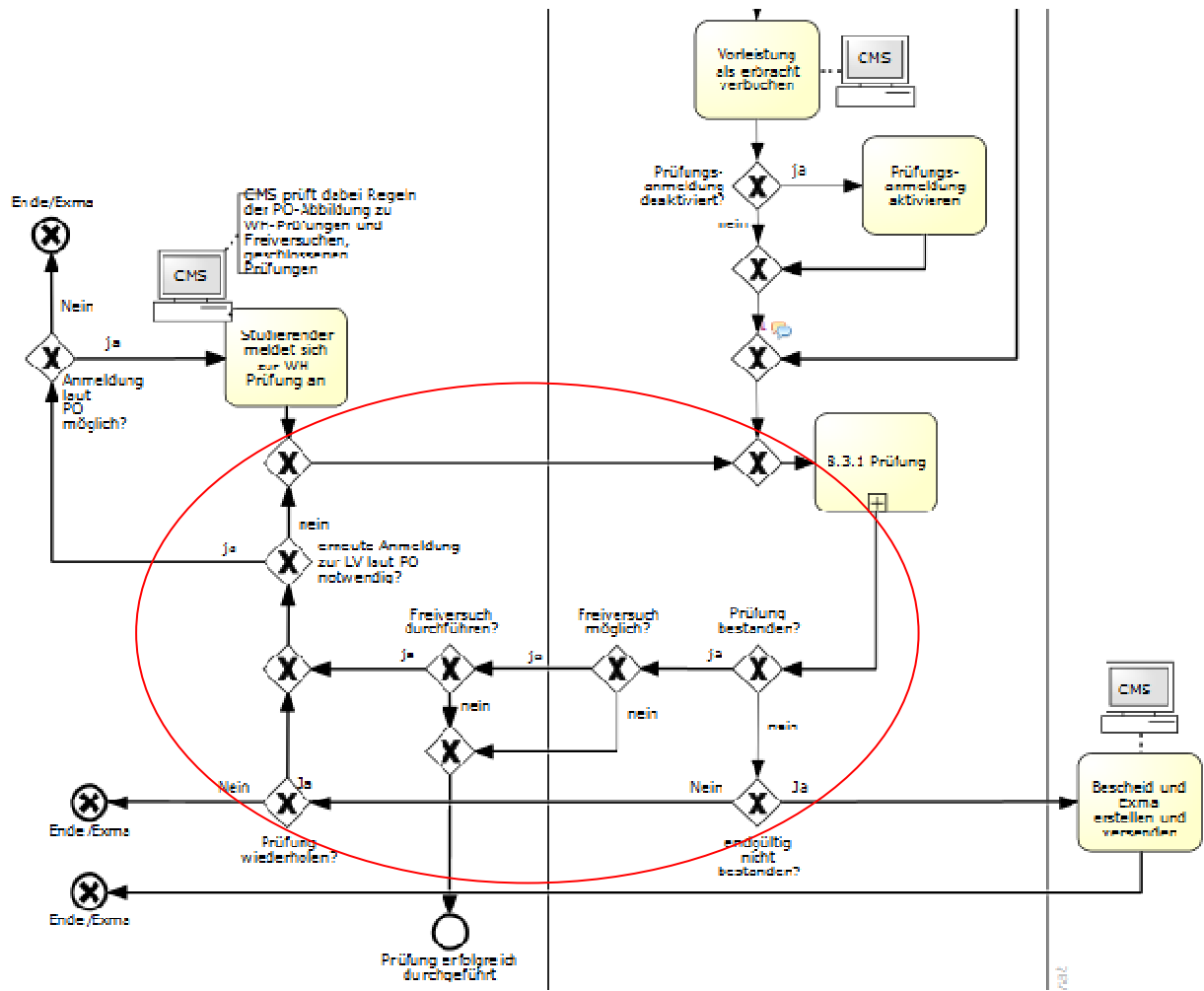


Abbildung 22: viele aufeinander folgenden Gateways ohne Aktivitäten im deklarativen Prozessmodell

- **Schleifen, die durch Gateways erzeugt werden**

In der imperativen Modellabbildung kommen häufig sog. Schleifen vor, die sich dadurch auszeichnen, dass bestimmte Aktivitäten wiederkehrend hintereinander in einem Prozessmodell ausgeführt werden können. Diese bilden einen Kreislauf in Form von Pfaden, der bei Nichterfüllung der Bedingung immer wieder wiederholt wird.

Im imperativen Beispielprozessausschnitt auf Abbildung 23 im oberen Teil befinden sich die Aktivitäten *Studienplan anpassen* und *Änderung prüfen* innerhalb einer solchen Schleife. Nach Ausführung der beiden Aktivitäten, wird an dem darauffolgenden XOR-Gateway abgefragt, ob die Änderungen *ok* waren, wenn dies nicht der Fall ist, werden die beiden Tasks erneut ausgeführt. Ansonsten läuft der Prozess seinen vorgesehenen Pfad weiter. Die Schleife wiederholt sich so lange, bis die Frage *Änderung ok?* am XOR-Gateway mit *Ja* beantwortet wird und der Prozess fortgeführt werden kann.

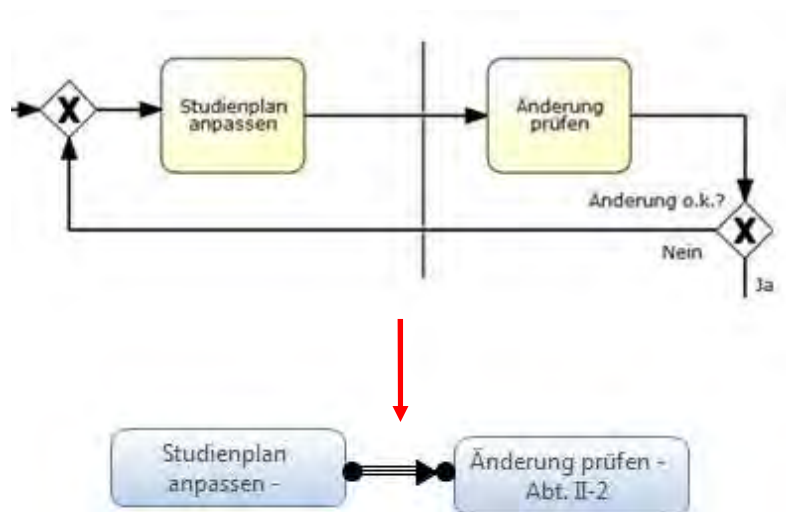


Abbildung 23: Schleifen im deklarativen Prozessmodell

Deklarativ ist die Abbildung einer solchen einfachen Schleife gut zu realisieren. Die einzige Einschränkung ist auch hier, dass die Entscheidungsregel, die zum erneuten Ausführen der Schleife führt, nicht abgebildet wird. Da die beiden Aktivitäten *Studienplan anpassen - Fakultät/ Studiengang (Uni Ulm)* und *Änderung prüfen - Abt. II-2 Studiensekretariat*

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

(*Universität Ulm*) direkt aufeinander folgen, ist dies die einzige Einschränkung die im deklarativen Prozessmodell abgebildet wird.

Einschränkend ist zu sagen, dass jede Schleife im Kontext des Prozessmodells betrachtet werden muss, um eine genaue Aussage über die Komplexität der deklarativen Abbildung treffen zu können. Handelt es sich beispielsweise um eine verschachtelte Schleife, nimmt hier die Komplexität bei der Abbildung durch die verschiedenen Beziehungen der Aktivitäten untereinander zu.

### 5.2.6 Subprozesse

Mit wachsender Größe eines Prozessmodells sinkt auch dessen Übersichtlichkeit. Die imperative Prozessmodellierung erlaubt eine hierarchische Abbildung von Prozessmodellen, was bei komplexen Prozessen die Übersichtlichkeit erhöht. Subprozesse können als eigenständige Prozessmodelle abgebildet und in den Elternprozess integriert werden. Für deklarative Prozessmodelle schlagen Haisjackl und Zugal ebenfalls eine hierarchische Abbildung z.B. in Form von Subprozessen vor (siehe Kapitel 3.4). Die Verwendung von Subprozessen ist bei relativ umfangreichen und großen Modellen sinnvoll, da die Übersichtlichkeit erhöht und zur Strukturierung des Prozesses beigetragen wird [HZSHRPW13, ZSHRPW12]. Dadurch ist das Modell für den Nutzer verständlicher.

Im Rahmen der Migration der Prozessmodelle in die deklarative Form wurde auch die Verwendung von Subprozessen in der deklarativen Modellabbildung untersucht. Diese ist jedoch im Gegensatz zur imperativen Prozessmodellierung in den meisten deklarativen Prozessmodellierungssprachen nicht möglich. Hier kann der Subprozess nicht, wie z.B. im imperativen Prozessmodell direkt in die Aktivität als Unterprozess verankert werden.

Im Beispiel in Abbildung 24 auf der linken Seite befindet sich vor Beginn des Subprozesses die Aktivität *Antrag stellen und Unterlagen beilegen - Studierender (Uni Ulm)* mit genau einem Constraint als Verbindung zum folgenden Subprozess. Dieser (Abbildung 24 auf der rechten Seite) wird anschließend wie ein normaler Prozess ausgeführt, bis sich keine der Aktivitäten mehr im Zustand *running* oder *waiting* befinden. Anschließend läuft der Hauptprozess mit der Aktivität *über Antrag entscheiden und im CMS vermerken (Beiträge, HZB)* -

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Abt. II-2 Studiensekretariat (Uni Ulm) weiter, bis auch hier alle Aktivitäten ausgeführt wurden.

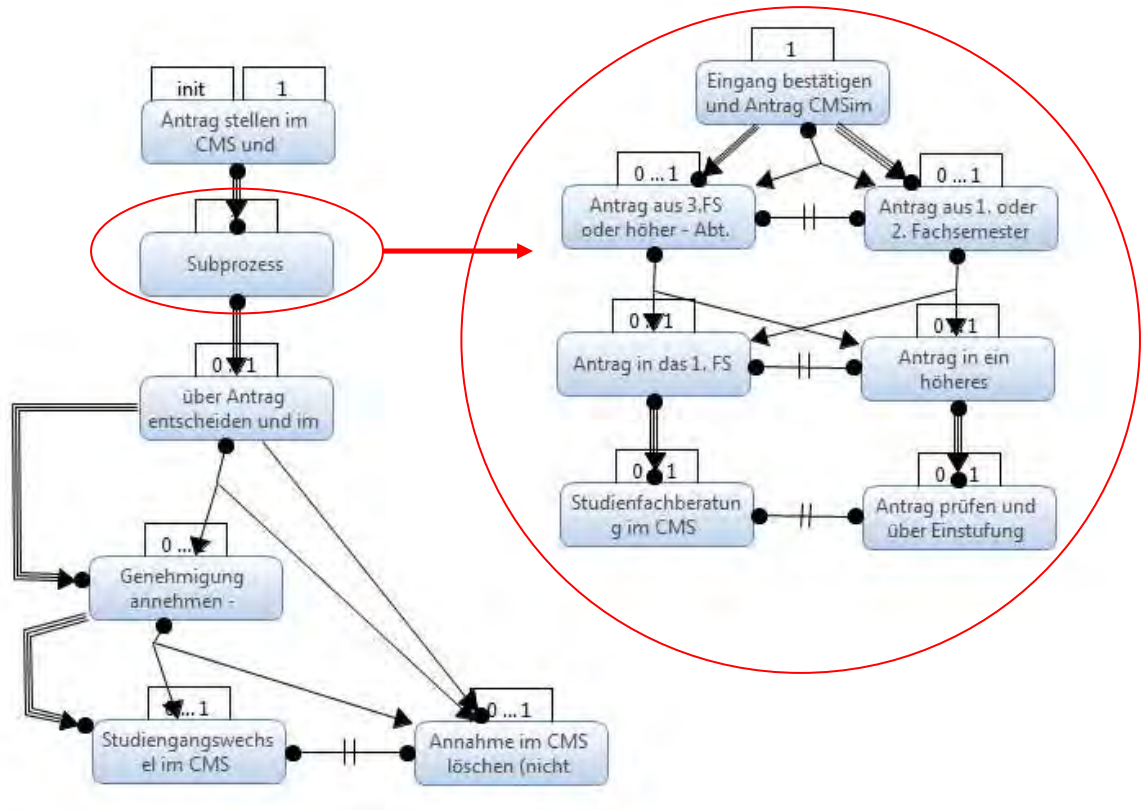


Abbildung 24: Subprozesse im deklarativen Prozessmodell

Subprozesse können jedoch in der deklarativen Modellabbildung nur unter bestimmten Voraussetzungen verwendet werden. Die Einschränkung zur Verwendung eines Subprozesses ergibt sich daraus, dass der Subprozess genauso wie in der imperativen Prozessmodellierung als abgeschlossener Prozess gesehen werden muss. Dieser Prozess wird eigenständig an einer bestimmten Stelle des Elternprozesses ausgeführt. Dies ist in Abbildung 24 auf der rechten Seite dargestellt. Hier ist der Subprozess als eigenständiger Prozess dargestellt. Auf Grund der fehlenden Abbildung zeitlicher Aspekte ergibt sich bei der Integration von Subprozessen in der deklarativen Prozessmodellabbildung das Problem, dass nicht klar ist, wann der Subprozess beendet ist und der Elternprozess weiter ausgeführt werden kann. Hier wäre eine Erweiterung der Sprache sinnvoll, um eine korrekte Integration des Subprozesses zu gewährleisten.

### 5.2.7 Zusammenfassung

Zusammenfassend treten bei der Abbildung imperativer Prozessmodelle im deklarativen Schema eine Reihe von Schwierigkeiten auf, die durch alternative Lösungsansätze kompensiert werden müssen.

Die deklarative Prozessmodellierung bietet keine Möglichkeit der Integration von Kontextinformationen, wie Rollen, Artefakte, Nachrichtenergebnisse oder zeitliche Aspekte. Der vollständige Verlust dieser Informationen führt dazu, dass der Kontext eines Prozessmodells kaum dargestellt wird und für den Nutzer weniger verständlich ist.

Gateways mit Entscheidungsregeln, wie z.B. das datenbasierte exklusive Gateway können im deklarativen Prozessmodell nicht korrekt abgebildet werden. Die Entscheidungsregel zur Wahl des Pfades wird nicht berücksichtigt und fällt bei der Abbildung weg. Auch die Abbildung paralleler Gateways können im deklarativen Prozessmodell nicht korrekt wieder gegeben werden, da eine zeitgleiche und somit parallele Ausführung von Aktivitäten in der deklarative Prozessmodellierung nicht möglich ist.

Deklarative Prozessmodelle verwenden im Gegensatz zu imperativen Prozessmodellen keine Start- oder Endereignisse, wodurch die Information über das Ende und den Anfang des Prozessmodells für den Nutzer verloren geht. Des Weiteren können diese im deklarativen Prozessmodell nicht nach ihrer Art und Funktion unterschieden.

Die Integration von Subprozessen ist in der deklarativen Prozessmodellierung ebenfalls nicht standardmäßig möglich und erfordert eine Erweiterung der Sprache.

Die Möglichkeit, deklarativ die Ausführungseinheiten der einzelnen Aktivitäten anzugeben, führt in einigen Fällen zu Abbildungsschwierigkeiten. Imperativ wird dies bei der Modellabbildung nicht verwendet, da sich hier die Ausführungsanzahl aus dem Ablauf des Prozesspfades ergibt.

Die imperative Prozessmodellierung hat also im Vergleich zur deklarativen Prozessmodellierung den Vorteil, Prozessmodelle verständlicher und vollständiger abzubilden. In vielen Fällen der deklarativen Prozessmodellabbildung ist die Verwendung eines Workarounds oder die Erweiterung der Sprache notwendig, um die Prozessmodelle korrekt, verständlich und vollständig abbilden zu können.

### 5.3 Ergebnisse und Vorgehensweise der Abbildung imperativer Prozessmodelle in das deklarative Schema anhand dreier Beispielprozesse

Im Folgenden wird an drei beispielhaft ausgewählten Prozessmodellen repräsentativ die Migration von Prozessmodellen vom imperativen in das deklarative Schema erläutert und dabei auf die genaue Vorgehensweise bei der Übertragung eingegangen. Es werden hier einige der bereits beschriebenen Aspekte der deklarativen Prozessmodellierung aufgegriffen und in den Kontext der Beispielprozessmodelle eingeordnet. Es folgt eine Art Leitfaden zur Migration imperativer Prozessmodelle in das deklarative Schema, der in dieser Form für fast jedes hier migrierte Modell angewendet werden konnte. Zu jedem Beispielprozessmodell wird eine Empfehlung gegeben, welche Art der Prozessmodellierung für welches Prozessmodell am besten geeignet ist.

#### 5.3.1 Beispiel 1

In Abbildung 25 und 26 ist die Migration eines beispielhaft ausgewählten Prozessmodells dargestellt, das vom imperativen in ein deklaratives Schema übertragen wird.

Bei der Migration des in Abbildung 25 imperativ abgebildeten Beispielprozessmodells in ein deklaratives Schema, müssen als erstes alle bestehenden Aktivitäten inklusive des Textes in das deklarative Modell übertragen werden. Das Beibehalten der imperativ bereits bestehenden Bezeichnungen erleichtert den Vergleich der Modelle, ist jedoch nicht zwingend notwendig. Anschließend werden hier die Aktivitäten textuell um die Bezeichnung ihrer Rollen (Pools und Lanes) ergänzt. Es wird nach dem in Kapitel 5.2.2 bereits erläuterten Schema verfahren.

Ein Beispiel hierfür ist die Aktivität *Studienplan anpassen* im imperativen Prozessmodell in Abbildung 25, die sich in der Lane *Fakultät/Studiengang* und im Pool *Uni Ulm* befindet. Diese wird analog zu allen anderen Aktivitäten um ihre Rolle basierend auf der Bezeichnung des Pools und der Lane ergänzt und bei der Migration in *Studienplan anpassen - Fakultät/Studiengang (Uni Ulm)* umbenannt. Da in diesem Prozessmodell keine Artefakte abgebildet sind, müssen diese nicht zusätzlich ergänzt werden.



## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Durch die Ergänzung des Textes durch die Rollen wird, wie bereits beschrieben (siehe Kapitel 5.2.2) keine technisch äquivalente Abbildung erreicht. Da die deklarative Prozessmodellierungssprache Declare keine Möglichkeit zur vollständigen Abbildung ermöglicht. Auch hier stellt die Ergänzung des Textes nur einen Workaround dar.

Bei der deklarativen Prozessmodellabbildung spielt die Angabe der möglichen Ausführungseinheiten eine große Rolle. Das bedeutet, wie bereits in Kapitel 5.1.2 beschrieben, dass bei jeder einzelnen Aktivität, wie im deklarativen Prozessmodell in Abbildung 26 zu sehen, in einem kleinen Kasten die Anzahl der möglichen Ausführungseinheiten einer Aktivität innerhalb einer Prozessinstanz angegeben werden muss. Die beiden Aktivitäten *Studienplan anpassen - Fakultät/Studiengang (Uni Ulm)* und *Änderung prüfen - Abt. II-2 Studiensekretariat (Uni Ulm)*, die im deklarativen Prozessmodell in Abbildung 26 dargestellt sind, können unendlich oft innerhalb einer Prozessinstanz wiederholt werden, da sie sich innerhalb einer Schleife befinden. Deshalb wird bei diesen beiden Aktivitäten keine Anzahl der Ausführungswahrscheinlichkeit angegeben. Die drei anderen Aktivitäten *1.3 Module verwalten - Fakultät/Studiengang (Uni Ulm)*, *Studiengangsbeschreibung, -ziele, -... , anpassen - Fakultät/Studiengang (Uni Ulm)* und *DS aktualisieren - Abt. II-2 Studiensekretariat (Uni Ulm)* dürfen maximal ein mal pro Instanz ausgeführt werden und werden deshalb mit dem Intervall *0...1* gekennzeichnet.

Als nächstes werden die Beziehungen der Aktivitäten in Abbildung 25 untereinander betrachtet. Diese sind imperativ als Pfade abgebildet und werden deklarativ durch Constraints dargestellt. Betrachtet man die einzelnen Pfade, die sich aus dem inklusiven Gateway ergeben, wird deutlich, dass hier immer die Aktivitäten *Studienplan anpassen* und *Änderung prüfen*, sowie die beiden Aktivitäten *Studiengangsbeschreibung, -ziele, -..., anpassen* und *DS aktualisieren* gemeinsam und direkt aufeinander folgend ausgeführt werden müssen. Um diese Bedingung zu gewährleisten, werden diese Aktivitäten bei der Migration in ein deklaratives Schema jeweils mit dem Chained Succession Constraint verbunden (siehe Abbildung 26).

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

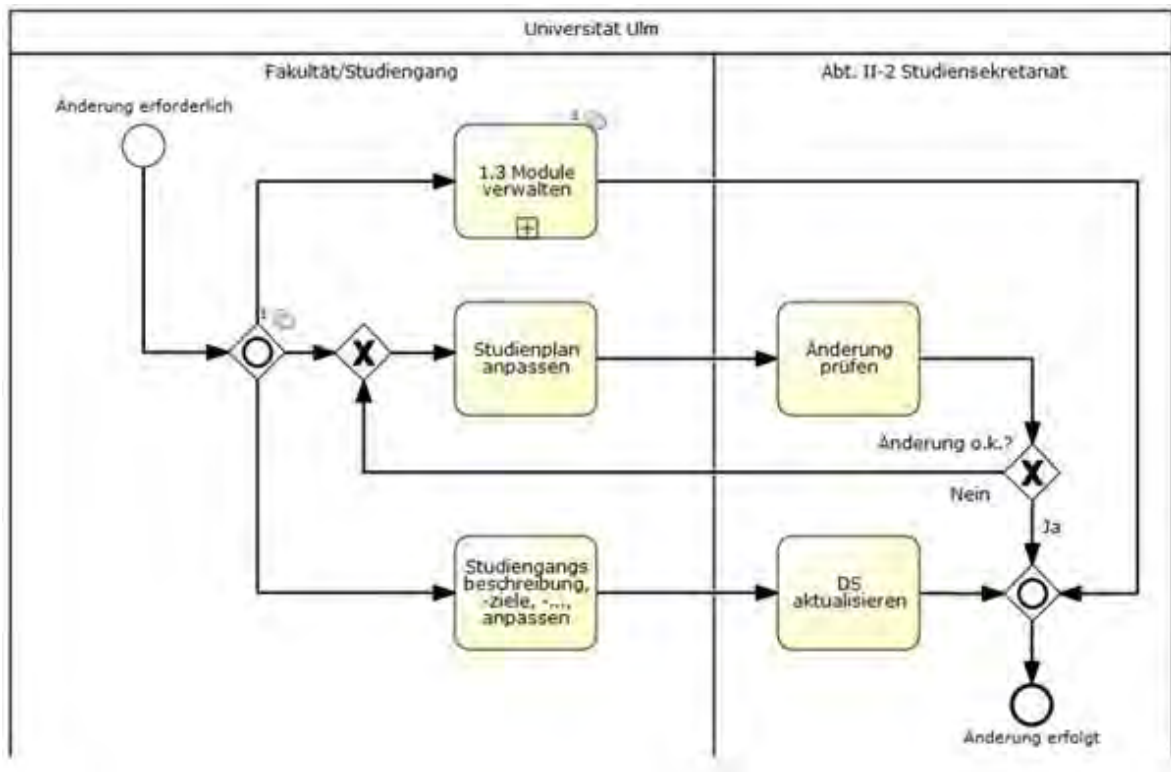


Abbildung 25: Beispiel 1, imperatives Prozessmodell

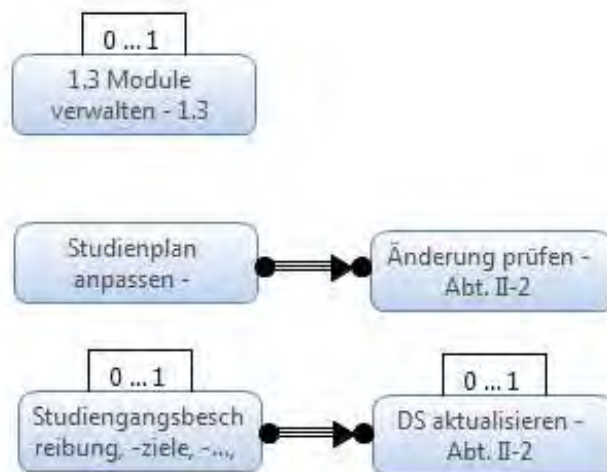


Abbildung 26: Beispiel 1, deklaratives Prozessmodell

Das inklusive Gateway am Anfang und am Ende des Prozessmodells stellt eine Oder-Entscheidung dar. Es gibt hier einen oder mehrere mögliche Pfade, die ausgeführt werden können. Da hier nur die bereits dargestellte Reihenfolge der Aktivitäten auf den einzelnen

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Pfaden eingehalten werden muss, jedoch nicht spezifiziert wird, wie viele Pfade und in welcher Reihenfolge diese ausgeführt werden müssen, ist hier eine hohe Flexibilität gefragt. In dem deklarativen Prozessmodell in Abbildung 26 werden keinerlei weitere Einschränkungen vorgenommen, da die Aktivitäten flexibel ablaufen können.

Der Ablauf des deklarativen Prozessmodells gibt zusammengefasst nur die folgende Einschränkung vor: Die Aktivitäten *Studienplan anpassen - Fakultät/Studiengang (Uni Ulm)* und *Änderung prüfen - Abt. II-2 Studiensekretariat (Uni Ulm)* müssen direkt nacheinander ausgeführt werden. Das gleiche gilt ebenfalls für die beiden Aktivitäten *Studiengangsbeschreibung, -ziele, -..., anpassen - Fakultät/Studiengang (Uni Ulm)* und *DS aktualisieren - Abt. II-2 Studiensekretariat (Uni Ulm)*. Alle anderen Ausführungszeitpunkte der Aktivitäten sind nicht spezifiziert und ermöglichen dem Prozessmodell eine flexible Ausführung.

Bei diesem ersten Beispielmodell ist deutlich zu erkennen, dass auf Grund der geforderten Flexibilität innerhalb des Prozessmodells die deklarative Abbildung besser geeignet ist. Dies ist Zurückzuführen auf das im imperativen Prozessmodell verwendete inklusive Gateway. Dieses verlangt die Ausführung eines Pfades oder mehrerer parallel ablaufender Pfade. Deklarativ kann diese Flexibilität der Ausführung der Pfade sehr einfach und übersichtlich dargestellt werden.

### 5.3.2 Beispiel 2

Analog zu der Vorgehensweise im ersten Beispielprozessmodell wird auch hier das in Abbildung 27 dargestellte imperative Prozessmodell schrittweise in ein deklaratives Schema migriert. Zunächst werden wieder alle Aktivitäten samt der Beschriftung vom imperativen in das deklarative Prozessmodell übertragen. Anschließend werden die Aktivitäten um die Rollenzuweisungen der Lanes *Abt. II-4 Zentrale Studienberatung*, *Studierender*, *Doktorand*, *Bewerber* und *Schüler* sowie die hier verwendeten Pools *Uni Ulm* und *Extern* textuell ergänzt. Da dieses Prozessmodell keine Artefakte enthält, müssen diese auch hier nicht zusätzlich hinzugefügt werden. Weiterhin werden, wie im ersten Beispielmodell, die Ausführungshäufigkeiten zu jeder Aktivität hinzugefügt (siehe Abbildung 28). Hier gibt es genau eine Aktivität, die genau ein mal in jeder Instanz vorkommen muss: *Beratung planen und abstimmen - Abt. II-4 Zentrale Studienberatung (Uni Ulm)*. Die Aktivität *an Beratung teilnehmen* muss vier mal im

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

deklarativen Schema abgebildet werden, da er sich durch die jeweiligen Rollenzuweisungen unterscheidet und somit nicht auf eine Aktivität reduziert werden kann, ohne entsprechend an Information zu verlieren.

Dadurch, dass das Prozessmodell, wie in Abbildung 27 zu erkennen, nur mit einem der vier verschiedenen Startereignisse *Beratungsbedarf übermitteln - Studierender (Uni Ulm)*, *Beratungsbedarf übermitteln - Doktorand (Uni Ulm)*, *Beratungsbedarf übermitteln - Bewerber (Extern)* und *Beratungsbedarf übermitteln - Schüler (Extern)* beginnen darf, muss im deklarativen Prozessmodell in Abbildung 28 eine große Anzahl an Not-Co-Existence-Constraints verwendet werden, um die Ausführung der jeweils anderen Aktivitäten auszuschließen. Die Integration mehrerer Startereignisse wurde bereits in Kapitel 5.2.4 erläutert und wird auch auf dieses Beispielprozessmodell übertragen. Die starke Einschränkung des Pfades wird durch die Auswahl der Startaktivität bedingt. Es darf, abhängig vom Startpunkt nur ein bestimmter Folge- bzw. Endpunkt ausgeführt werden. Wird zum Beispiel der Startpunkt *Beratungsbedarf übermitteln - Studierender (Uni Ulm)* gewählt, muss auch der dazugehörige Endpunkt *an Beratung teilnehmen - Studierender (Uni Ulm)* ausgeführt werden.

Weiterhin werden im deklarativen Prozessmodell durch Constraints die Folgebeziehungen und Einschränkungen der Aktivitäten untereinander dargestellt.

Auf die Migration der im imperativen Prozessmodell verwendeten Nachrichtenergebnisse wird hier zur Vereinfachung des Prozessmodells verzichtet

Das in Abbildung 27 vorgestellte Beispielmodell bildet verschiedene Varianten eines Prozessmodells ab. Hier stellt sich die Frage, ob alle Aktivitäten, die hier verwendet werden, tatsächlich notwendig sind. Es wäre sicherlich sinnvoll, bei der Abbildung im deklarativen Schema über eine Reduktion der Aktivitäten nachzudenken. Beispielsweise könnten die verschiedenen Startaktivitäten *Beratungsbedarf übermitteln (Studierender, Doktorand, Bewerber, Schüler)* und *an Beratung teilnehmen (Studierender, Doktorand, Bewerber, Schüler)* jeweils zu einer Aktivität zusammengefasst werden, ohne dabei die Rollen anzugeben. Diese Vorgehensweise ist in Abbildung 29 zu sehen. Hier wurden die Rollen bei der Abbildung des Prozesses vollständig weggelassen. Es ist zu sehen, dass der Prozess dadurch erheblich vereinfacht wird. Allerdings ist ebenfalls anzumerken, dass hier wichtige Kontextinformationen, z.B. beim Ausführen der Aktivität *weitere (externer) Gesprächspartner hinzuziehen* verloren

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

geht. Diese darf nur durch die Rolle *Studierender* ausgeführt werden. Zusammengefasst ist bei der deklarativen Abbildung des Prozessmodells in Abbildung 29 zu sehen, dass dieses durch das Zusammenfassen der Aktivitäten erheblich vereinfacht werden konnte. Hier werden die verschiedenen Varianten des Prozessmodells in einem Prozess zusammengefasst. Allerdings gehen auch hier wichtige Kontextinformationen verloren.

Vergleicht man das in Abbildung 27 dargestellte imperative Prozessmodell mit dem in Abbildung 28 abgebildeten deklarativen Prozessmodell, ist hier eine eindeutigere Tendenz im Bezug auf die richtige Wahl der Abbildung für diesen Prozess zu erkennen. Insbesondere ist deutlich erkennbar, dass das imperative Prozessmodell eine übersichtlichere Darstellung bietet. Dies liegt daran, dass sich die Ausführungsvarianten des Prozessmodells auf einige wenige mögliche Pfade reduzieren und somit die deklarative Abbildung nur bedingt geeignet ist. Es ist ein strenger Ablauf des Prozesses und keine Flexibilität verlangt. Je nach Wahl des Startereignisses ist immer nur ein Prozess mit dem zur Rolle passenden Endereignis möglich. Die deklarative Prozessmodellierung ist eindeutig besser zur Abbildung von Prozessmodellen geeignet, die eine höhere Flexibilität und keine strenge Vorgabe bei der Ausführung der Aktivitäten erfordern.

Vergleicht man das in Abbildung 27 dargestellte imperative Prozessmodell mit dem in Abbildung 29 dargestellten deklarativen Prozessmodell, ist zu erkennen, dass hier das deklarative Prozessmodell durch die Reduktion auf einige wenige Aktivitäten, sowie die übersichtliche Darstellung des Prozessmodells besser zur Abbildung geeignet ist. Hier werden mehrere Prozessvarianten innerhalb eines Prozessmodells abgebildet. Es ist in diesem Fall vom Modellierer abzuwägen, inwieweit der Verlust der Kontextinformation die Zusammenfassung der Aktivitäten rechtfertigt.

Zusammenfassend wurde in dieser Arbeit bei der Umsetzung der Prozessmodelle jedoch großen Wert auf die Integration der Rollen gelegt und auf eine einheitliche Umsetzung aller Prozessmodelle geachtet. Um die Kontextinformation in Form von Rollenzuweisungen hier zu erhalten wird die Umsetzung des deklarativen Prozessmodells in Abbildung 28 empfohlen.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

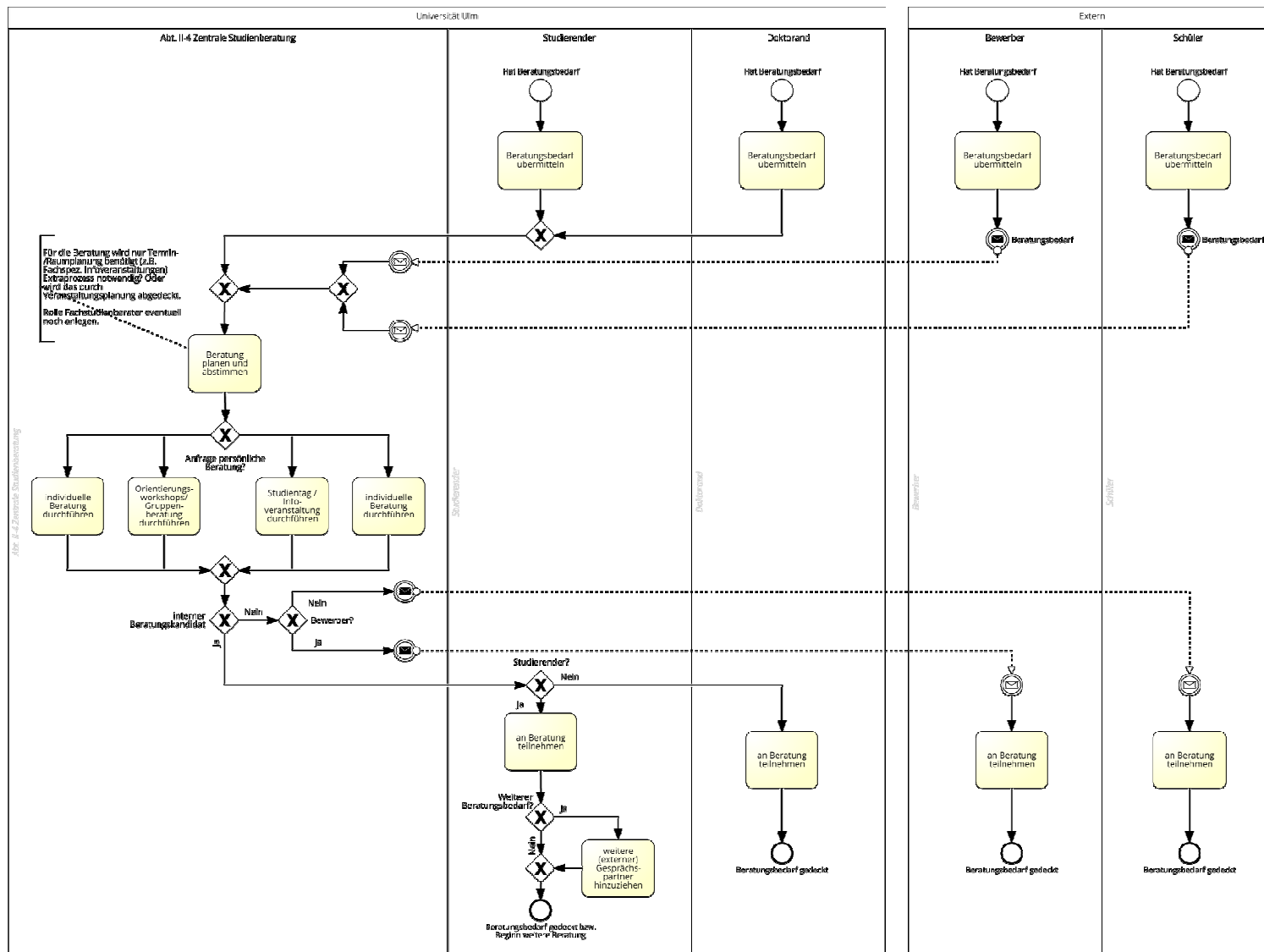


Abbildung 27: Beispiel 2, imperatives Prozessmodell

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

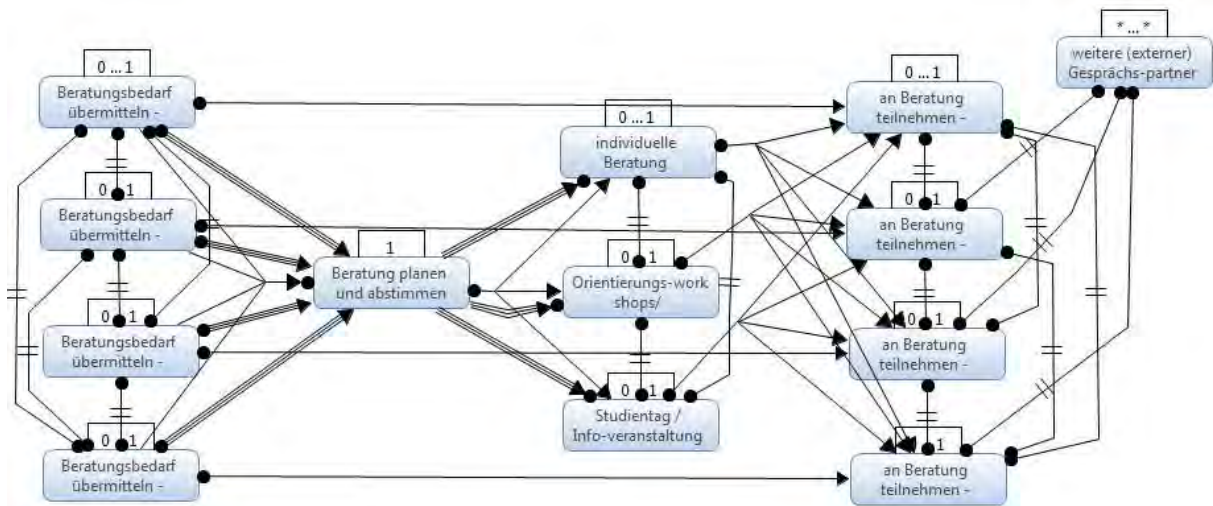


Abbildung 28: Beispiel 2, deklaratives Prozessmodell 1

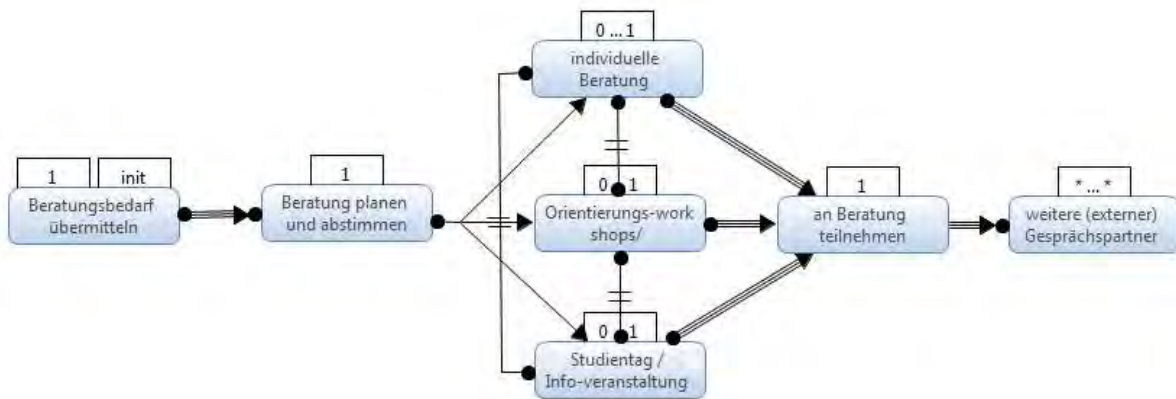


Abbildung 29: Beispiel 2, deklaratives Prozessmodell 2

### 5.3.3 Beispiel 3

Auch bei diesem Prozessmodellbeispiel (siehe Abbildung 30) wird bei der Migration des imperativen Prozessmodells in das deklarative Schema nach dem bereits im ersten und zweiten Beispiel beschriebenen Schema vorgegangen. Es werden zunächst die Aktivitäten vom imperativen Prozessmodell in das deklarative übernommen. Diese werden textuell um ihre jeweilige Rollenzuweisung und die Artefakte, wie in diesem Fall das *CMS* und das *Regisafe*, er-

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

gänzt. Anschließend werden die eigentlichen Beziehungen der Aktivitäten untereinander in Form von Constraints dargestellt.

Eine große Herausforderung bei der Migration des imperativen Prozessmodells in ein deklaratives Schema stellt die Verwendung vieler datenbasierter exklusiver und paralleler Gateways dar (siehe Abbildung 30). Der in der Abbildung blau eingekreiste Bereich beinhaltet ein paralleles Gateway, das sich wiederum in zwei datenbasierte exklusive Gateways aufspaltet. Das parallele Gateway führt dazu, dass auf jeden Fall beide nachfolgenden Pfade (in Abbildung 30 grün und rot markiert) ausgeführt werden. Danach folgt jedoch auf jeden der beiden parallelen Pfade ein datenbasierte exklusive Gateways. Es darf hier jeweils nur einer der Pfade ausgewählt werden. Nach der Aktivität *Beglaubigungen und Anschreiben erstellen und Durchschrift in Regisafe* müssen dementsprechend zwei der vier verfügbaren Folgeaktivitäten *Dokumente zusenden lassen*, *Dokumente abholen*, *3.5 Studiengang-wechsel* und *3.3 Exmatrikulation* ausgeführt werden. Allerdings dürfen die beiden Aktivitäten *3.5 Studiengang-wechsel* und *3.3 Exmatrikulation* sowie *Dokumente zusenden lassen* und *Dokumente abholen* jeweils nie gemeinsam innerhalb einer Prozessinstanz ausgeführt werden. Dies kann, wie in Abbildung 31 zu sehen, deklarativ durch ein Exclusive-Choice-Constraint dargestellt werden. Hier ist es möglich, dass beliebig viele Aktivitäten aus einer bestimmten Menge ausgewählt und diese gemeinsam innerhalb einer Prozessinstanz ausgeführt werden. Das Not-Co-Existence-Constraint verhindert, dass dabei zwei Aktivitäten innerhalb eines Prozessmodells ausgeführt werden, die nicht gemeinsam ausgeführt werden dürfen.

Eine weitere Besonderheit an diesem Prozess ist, dass im imperativen Prozessmodell, wie in Abbildung 30 zu sehen, eine Frist in Form einer symbolischen Uhr verwendet wird, welche an die Aktivität *Informationen überprüfen* angehängt ist. Wie bereits in Kapitel 5.2.3 erläutert wurde, ist die Abbildung zeitlicher Aspekte, wie in diesem Fall einer Frist, in der deklarativen Prozessmodellabbildung nicht vorgesehen. Wie in Abbildung 31 des bereits in die deklarative Form migrierten Prozessmodells zu sehen ist, muss die Frist bzw. der Fristablauf als Aktivität *Abbruch durch Fristablauf - Studierender (Uni Ulm)* hinzugefügt werden. Die Benennung des Abbruchs wird äquivalent zur Benennung der Aktivitäten aus dem imperativen Prozessmodell in das deklarative übernommen, um einen direkten Vergleich der beiden Modelle zu gewährleisten. Somit ist die Frist bzw. der Abbruch des Modells durch deren Nichteinhalten entsprechend in das deklarative Prozessmodell integriert.



## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Vergleicht man das Ergebnis des deklarativen Prozessmodells mit dem imperativen, ist an diesem Beispiel deutlich erkennbar, dass hier auf Grund der Vielzahl von Ausführungsvarianten, die sich durch die diversen Gateways und Schleifen des Modells ergeben, die deklarative Abbildung besser geeignet scheint, da die verschiedenen möglichen Pfade kompakter, übersichtlicher und unverschleieter abbildet.

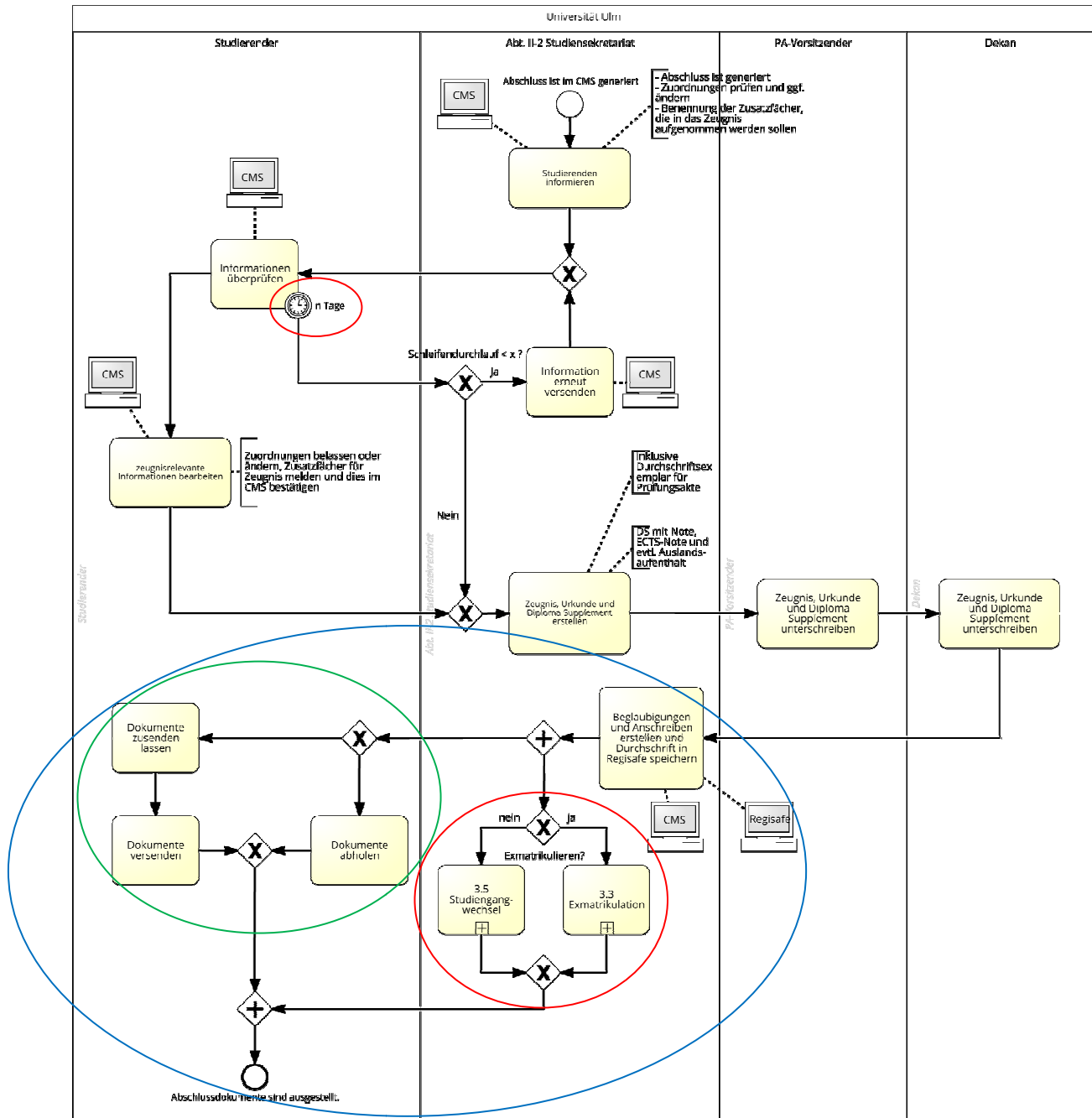


Abbildung 30: Beispiel 3, imperatives Prozessmodell

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

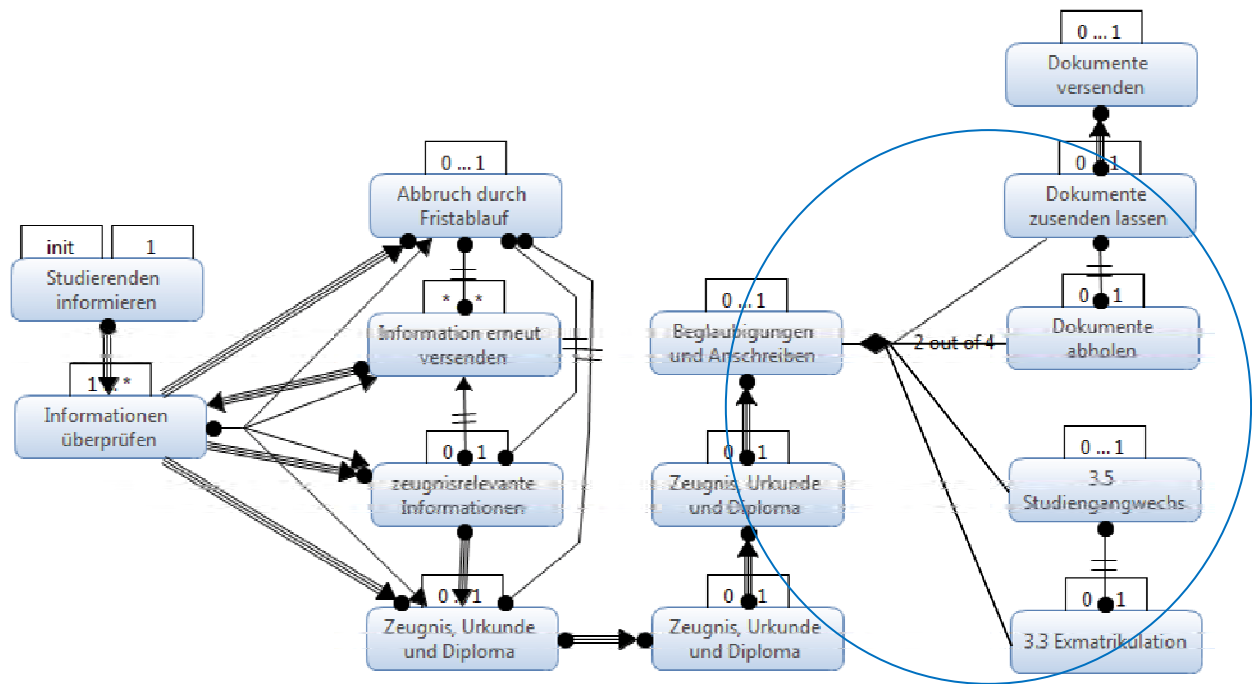


Abbildung 31: Beispiel 3, deklaratives Prozessmodell

### 5.3.4 Zusammenfassung

Zusammenfassend ist zu bemerken, dass bei der Migration der Prozessmodelle vom imperativen in das deklarative Schema immer nach dem gleichen Schema verfahren werden kann: Zuerst werden die Aktivitäten inklusive ihrer Benennungen in das deklarative Prozessmodell übernommen. Anschließend werden diese soweit nötig durch weitere Kontextinformationen, wie Rollenzuweisungen und Artefakte in Form von textuellen Ergänzungen der Aktivitäten, erweitert. Die textuelle Erweiterung der Aktivitäten ist jedoch nur als Workaround zu betrachten. Im deklarativen Prozessmodell werden alle Aktivitäten um die Anzahl der Ausführungen ergänzt und anschließend die Constraints hinzugefügt.

Dies ist lediglich eine Art Schema, das genutzt werden kann, um ein imperatives Prozessmodell in ein deklaratives zu übertragen. Wie man an den drei Beispielprozessmodellen sehen konnte, existieren oft Abweichungen und Besonderheiten bei den Modellen, die je nach Fall individuell gehandhabt werden müssen.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Des Weiteren konnte festgestellt werden, dass bei der Migration der Prozessmodelle und vor allem bei der Modellierung der deklarativen Prozesse nicht immer einheitlich vorgegangen werden kann und hier oft der Modellierer entscheiden muss, ob eine Information relevant ist oder nicht. Bei vielen Abbildungskonstruktionen muss eine variable Herangehensweise verwendet werden, da nicht in jedem Prozess z.B. ein Endereignis auf die gleiche Art und Weise abgebildet werden kann. Dies resultiert daraus, dass immer die Korrekte Abbildung des Prozesses im Vordergrund steht und sich die Art der Modellierung danach richtet.

Bei der Frage danach, welche Abbildungsart sich am besten für ein gegebenes Prozessmodell bzw. einen Prozess eignet, ist ebenfalls nicht immer eine eindeutige Entscheidung möglich. Zusammenfassend ist jedoch festzuhalten, dass bei Prozessen mit vielen verschiedenen Ausführungsvarianten und möglichen Pfaden die deklarative Abbildung besser geeignet ist. Die imperative Modellabbildung ist, wie bereits zu Anfang dieser Arbeit diskutiert, für starre Prozesse vorgesehen, die kaum Abweichungen und nur eine geringe Anzahl an Pfaden erlauben.

### 5.4. Bewertung

Überraschenderweise war die Umsetzung der imperativen Prozessmodelle in das deklarative Schema in vielen Fällen relativ problemlos möglich. Die Größe der meisten imperativen Prozessmodelle stellte jedoch eine Herausforderung bei der Umsetzung und der Integration der Constraints dar. Mit Hilfe der Testfallanalyse der Test Driven Modelling Suite war jedoch eine schrittweise Überprüfung auf die Richtigkeit und Ausführbarkeit der deklarativen Prozessmodelle möglich.

Eine Erschwernis bei der Lösung der bei der Migration auftretenden Schwierigkeiten stellte die Limitierung der zur Verfügung stehenden Konstrukte dar. In diesem Kapitel wurden Lösungen anhand der bereits vorhandenen Abbildungsmöglichkeiten vorgeschlagen. Da sich diese jedoch weitestgehend auf die Abbildung von Aktivitäten und Constraints beschränkt, besteht häufig nur die Möglichkeit, Änderungen in Form von schriftlichen Zusätzen zu realisieren. Diese trifft beispielsweise auf Artefakte, zeitliche Ereignisse und auch Start bzw. Endereignisse zu.

## 5. Vergleich imperativer und deklarativer Prozessmodelle anhand von Prozessbeispielen

Insgesamt wäre es sicherlich sinnvoll, über eine Erweiterung der Modellierungselemente in der deklarativen Modellierung nachzudenken. Diese würden die Lesbarkeit und auch die Abbildbarkeit in vielerlei Hinsicht erleichtern. Zukünftig wäre Forschung bzw. Überlegungen in diese Richtung sicherlich von großem Nutzen und würden die deklarative Abbildung positiv unterstützen und erleichtern.

Es fällt ebenfalls auf, dass bei vielen Fällen keine einheitliche Lösung des Problems möglich ist, sondern diese je nach Problemstellung gewählt werden muss. Dies tritt zum Beispiel häufig bei der Migration von Endereignissen vom imperativen in das deklarative Schema auf.

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

### 6.1 Diskussion

Dieses Kapitel diskutiert die in Kapitel 5 erarbeiteten Möglichkeiten und Schwierigkeiten der Umsetzung von imperativen Prozessmodellen in ein deklaratives Schema und betrachtet die Ergebnisse bezogen auf deren Verständlichkeit, Vollständigkeit, Korrektheit, Granularität und deren Flexibilität. Dazu wird anhand dieser Aspekte ein Vergleich der imperativen und deklarativen Prozessmodellabbildung erarbeitet.

#### 6.1.1 Verständlichkeit

Ein wichtiger Aspekt der imperativen und deklarativen Prozessmodellabbildung ist die Verständlichkeit eines Prozessmodells. Hierunter versteht man, ob das Prozessmodell für den Benutzer klar und unmissverständlich dargestellt ist [RW12].

Wie bereits zu Beginn des 5. Kapitels beschrieben, bietet die deklarative Prozessmodellabbildung im Gegensatz zur imperativen Prozessmodellabbildung bisher keine Möglichkeit, Kontextinformationen, wie z.B. Artefakte, Kommunikationsereignisse oder Rollen in das Prozessmodell zu integrieren und symbolhaft darzustellen [GV07]. Der Vorteil ist hierbei, dass weniger verschiedene Abbildungssymbole, die auf Aktivitäten und deren Interaktion in Form von Constraints beschränkt sind, verwendet werden. Es findet somit durch die Reduktion auf das Wesentliche keine Überladung der Prozesse durch zu viele verschiedene Symbole statt und der Nutzer wird nicht vom Hauptprozess abgelenkt. Ein Nachteil ist jedoch, dass dem Betrachter durch das Fehlen der (symbolhaft abgebildeten) Kontextinformationen Information zum Verständnis des Prozessmodells verloren geht. Dies kann zu Fehlern bei der Lesbarkeit und Ineffizienz führen [VGHJZ13]. Folglich ist die Verständlichkeit eines Prozessmodells in der imperativen Prozessmodellabbildung auf Grund fehlender Kontextinformation eher gering.

Die imperative Prozessmodellabbildung hingegen kann zum Verständnis notwendige Kontextinformationen in Form von Symbolen darstellen und somit die Lesbarkeit erleichtern [FLMRWWZ09]. Durch die Verwendung von Symbolen und der damit verbundenen Integra-

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

tion von Kontextinformationen ist die Verständlichkeit des Prozessmodells für den Nutzer hoch. Es ist jedoch anzumerken, dass in der imperativen Prozessmodellabbildung eine große Anzahl an verschiedenen Symbolen zur Verfügung steht, die sicherlich nicht alle Verwendung finden. Somit wäre in der imperativen Prozessmodellabbildung eine Reduktion der zur Verfügung stehenden Symbole auf eine tatsächlich notwendige Anzahl Symbole sinnvoll, um die Verständlichkeit weiter zu erhöhen.

Die in der imperativen Prozessmodellierung verwendeten Symbole helfen dem Betrachter im Vergleich zur deklarativen Modellierung die Prozesse leichter zu erfassen. Die Hintergrundinformation wird bildlich dargestellt, wodurch der Nutzer auch ohne das Lesen des Textes den Ablauf des Modells erfassen kann. Die in der deklarativen Abbildung fehlenden Symbole werden in dieser Arbeit durch einen Workaround kompensiert, indem beispielsweise die Beschriftung der Aktivitäten durch zusätzlichen Text ergänzt werden. Eine Folge davon ist, dass die Bezeichnung der Aktivitäten teilweise sehr lang wird und sich die Verständlichkeit des Modells verschlechtert. Dies ist z.B. anhand des Beispiels in Abbildung 32 zu erkennen ist. Hier wird die imperativ abgebildete Aktivität *Bewilligungsbescheid erstellen* durch Rollen und Organisationseinheiten sowie dem angehängten Artefakt ergänzt und in *Bewilligungsbescheid im CMS erstellen - Abt. I-2 Recht und Organisation (Uni Ulm)* benannt.

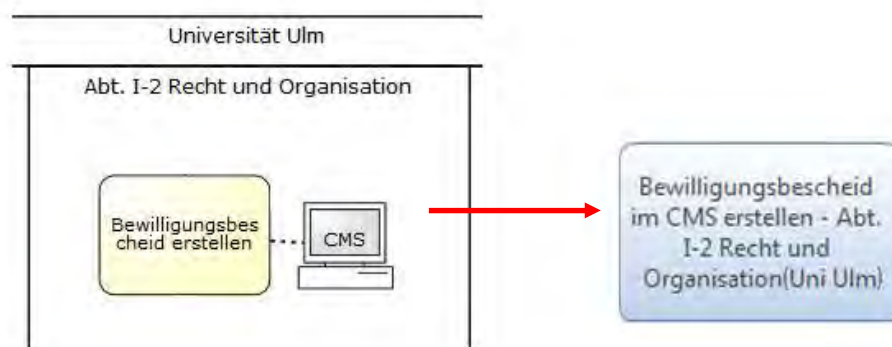


Abbildung 32: Beschriftung einer Aktivität im deklarativen Prozessmodell

Die imperative Prozessmodellabbildung stellt wichtige Informationen zum sequentiellen Ablauf des Prozessmodells zur Verfügung, indem sie die Pfade bildlich darstellt - ähnlich einer

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

Landkarte. Die Richtung der Pfeile gibt den Pfadverlauf des Prozessmodells wieder. Deklarative Prozessmodelle liefern die Information zum sequentiellen Ablauf weniger verständlich als imperative Prozessmodelle. Die in der deklarativen Prozessmodellierung verwendeten Constraints stellen nicht den Verlauf des Prozesspfades dar, sondern vielmehr die Beziehungen der Aktivitäten untereinander [WZP11]. Diese sind für den Betrachter meist schwer verständlich und lassen den Nutzer meist keinen eindeutigen Ablauf des Prozesses erkennen [WZP11].

Ein Beispiel hierfür ist in Abbildung 33 zu sehen. Das imperative Prozessmodell im oberen Teil liefert dem Betrachter Informationen über den sequentiellen Ablauf des Prozessmodells. Dies geschieht mit Hilfe von Pfaden, die die Aktivitäten miteinander verbinden. Hier werden die Informationen zum Ablauf des Prozesses für den Nutzer verständlich abgebildet. Das deklarative Prozessmodell im unteren Teil in Abbildung 33 hingegen besteht aus einer großen Anzahl verschiedener Constraints, die dem Nutzer kaum Informationen über den sequentiellen Ablauf des Prozessmodells zur Verfügung stellen. Die Pfade im imperativen Prozessmodell sind für den Betrachter leichter verständlich und übersichtlicher dargestellt als die Constraints im deklarativen Prozessmodell.

Ebenfalls wichtig für die verständliche Abbildung eines Prozessmodells sind Start- und Endereignisse. Die imperative Prozessmodellabbildung kann Start- und Endereignisse in Form von Symbolen darstellen. Für den Betrachter erhöht sich die Verständlichkeit des Prozessmodells, da Beginn und Ende des Prozessmodells klar erkennbar sind. Die deklarative Prozessmodellabbildung ermöglicht dagegen die Kennzeichnung von nur einer Start- und Endereignisse innerhalb eines Prozessmodells (siehe Kapitel 5.2.4). Folglich ist die Kennzeichnung mehrerer Start- und Endaktivitäten nicht möglich. In solchen Fällen kann der Betrachter Start- und Endereignisse nicht identifizieren [HZSHRPW13]. Sofern ein deklaratives Prozessmodell nur eine Start- bzw. Endaktivität enthält, die eindeutig gekennzeichnet werden kann, ist der Beginn bzw. das Ende des Prozessmodells verständlich dargestellt.

Die imperative Prozessmodellabbildung bildet im Gegensatz zur deklarativen Prozessmodellabbildung keine Entscheidungsregeln bei Gateways, insbesondere beim datenbasierten exklusiven Gateway, ab. Hier geht eine wichtige Kontextinformation für den Nutzer verloren, die zum Verständnis des Prozesses beiträgt.

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

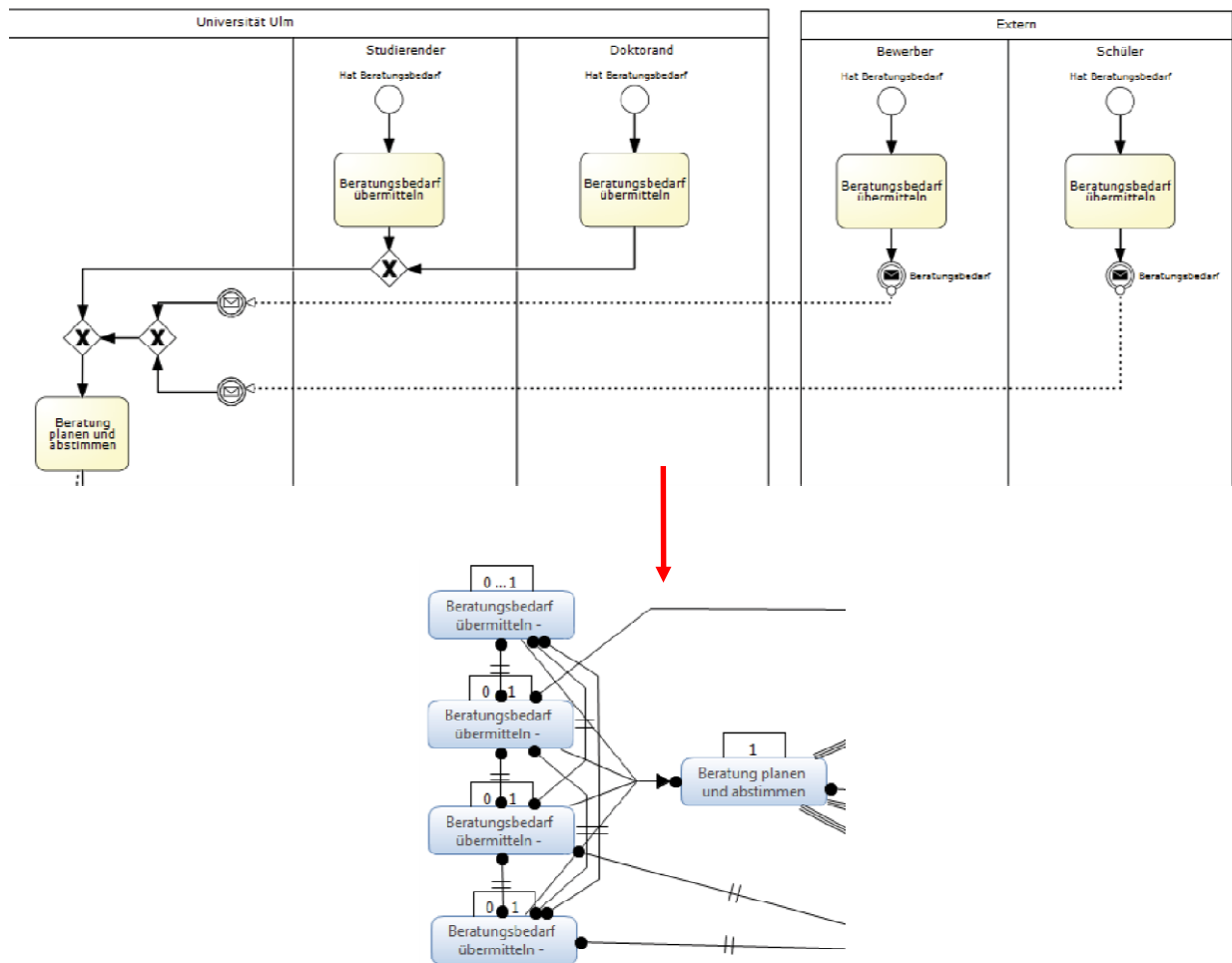


Abbildung 33: Sequenzfluss im deklarativen Prozessmodell

Ein weiterer wichtiger Aspekt ist die Verwendung von Subprozessen. Die deklarative Prozessmodellabbildung sieht im Gegensatz zur imperativen Prozessmodellabbildung keine Verwendung von Subprozessen vor. Die Verständlichkeit des deklarativen Prozessmodells ist bei wachsender Größe gering, da hier keine Strukturierung durch verschiedene Hierarchieebenen möglich ist.

Zusammenfassend führt das Fehlen von Abbildungsmöglichkeiten für Kontextinformationen, Start- und Endereignisse sowie die Unterscheidung nach deren Art und Funktion bei deklarativen Prozessmodellen zu einer vergleichsweise geringen Verständlichkeit für den Betrachter. Des Weiteren bedingen die Verwendung von Constraints, das Wegfallen der Entscheidungsregeln bei Gateways und die nicht vorhandene Möglichkeit der Integration von Subprozessen eine schlechteren Verständlichkeit des Prozessmodells als bei der deklarativen Prozessmodellierung.



### 6.1.2 Vollständigkeit

Für die Abbildung eines Prozessmodells ist nicht nur eine verständliche Darstellung wichtig, sondern auch eine vollständige Wiedergabe. Dies bedeutet, dass das Modell inhaltlich ausreichend Informationen enthalten muss, um es für den Nutzer verständlich zu machen [RW12].

In dieser Arbeit wurde bereits thematisiert, dass die deklarative Prozessmodellabbildung im Gegensatz zur imperativen keine Integration von Kontextinformation erlaubt. Dazu zählt das Fehlen von Abbildungsmöglichkeiten für Rollen, Artefakte, Kommunikationsereignisse sowie die symbolhafte Unterscheidung von Start- und Endereignisse nach ihrer Art. Das Fehlen dieser Abbildungsmöglichkeiten führt dazu, dass das Prozessmodell nicht vollständig abgebildet werden kann und wichtige Kontextinformationen verloren gehen.

Ein weiterer Aspekt der (im Gegensatz zur imperativen Prozessmodellabbildung) unvollständigen Abbildung deklarativer Prozessmodellierung ist der temporale Aspekt, wie z.B. der Frist (siehe Kapitel 5.2.3). Die imperative Prozessmodellierung integriert den zeitlichen Aspekt, die deklarative lässt diesen jedoch vollständig außen vor. Auch dadurch ist keine vollständige Abbildung des Prozessmodells möglich.

Zur vollständigen Abbildung des Prozessmodells würde des Weiteren die Möglichkeit, Entscheidungsregeln bei Gateways abzubilden beitragen. Die deklarative Prozessmodellierung ermöglicht die Abbildung von Entscheidungsregeln, die imperative Prozessmodellierung lässt diese jedoch weg.

Abschließend kann festgehalten werden, dass imperative Prozessmodelle in den meisten Fällen eine vollständige Darstellung der Prozesse gewährleisten. Grund sind die umfangreichen Möglichkeiten, die bei der Abbildung zur Verfügung stehen. Deklarative Prozessmodelle dagegen erfassen weder den temporalen Aspekt, noch ermöglichen sie die Integration von Kontextinformation oder die Abbildung von Entscheidungsregeln bei Gateways, was dazu führt, dass hier meist keine Vollständige Abbildung möglich ist.

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

### 6.1.3 Korrektheit

Die inhaltliche Korrektheit eines Prozessmodells definiert sich durch die Übereinstimmung eines Prozessmodells mit der Realität [RW12]. Es ist wichtig, dass ein Modell den real existierenden Prozess originalgetreu wiedergibt.

Sowohl die imperative als auch die deklarative Prozessmodellabbildung ermöglichen nicht immer eine korrekte Darstellung eines Prozesses. Die imperative Prozessmodellabbildung bietet eine Vielzahl von Abbildungsmöglichkeiten, die zu einer korrekten Abbildung des Prozesses beitragen. Sie stellen dazu möglichst viele Informationen zur Verfügung. Beispielsweise können Start- und Endereignisse nach ihrer Art unterschieden werden. Es ist möglich, abzubilden, ob es sich um ein normales Ende, um einen Abbruch des Prozesses oder um ein Ende handelt, das Kompensation erfordert. In der deklarativen Prozessmodellabbildung ist diese Unterscheidung nach der Art der Start- bzw. Endereignisse nicht möglich.

Die Zuweisung von Rollen, die angeben durch welche Instanz eine Aktivität ausgeführt wird, tragen ebenfalls zur Korrektheit eines Prozessmodells bei. Die Verwendung von Rollenzuweisungen wird jedoch in der deklarativen Prozessmodellabbildung vollständig vernachlässigt. Auch das Fehlen des zeitlichen Aspektes, führt dazu, dass ein Prozessmodell nicht korrekt wiedergegeben werden kann.

Insgesamt weist die deklarative Prozessmodellabbildung im Gegensatz zur imperativen mehr Unvollständigkeiten in der Abbildung auf. Dies führt dazu, dass nicht jeder Prozess korrekt abgebildet werden kann.

### 6.1.4 Granularität

Die Granularität eines Prozessmodells gibt an, inwiefern das Prozessmodell die Ansprüche bezüglich der Detailtiefe erfüllen kann. Ein feingranulares Prozessmodell enthält einen hohen Detailgrad, ein grob granulares Prozessmodell verschafft eher einen Überblick über den Prozess [GHV07].

In deklarativen Prozessmodellen müssen Subprozesse nicht wie in der imperativen Prozessmodellabbildung künstlich abgegrenzt werden. Durch die Abbildung mehrerer Prozesse

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

innerhalb eines Prozessmodells muss eine Aktivität, die in mehreren Prozessen verwendet wird, nur einmal abgebildet werden. Dies ermöglicht eine feingranulare Abbildung des Prozessmodells [GHV07].

Bei der Integration von Subprozessen (siehe Kapitel 5.1.3) wurde bereits thematisiert, dass imperative Prozessmodelle Subprozesse direkt in den Elternprozess integrieren können. Viele Aktivitäten werden in mehreren Prozessvarianten verwendet und können keinem bestimmten Prozessmodell eindeutig zugeordnet werden. Damit ist die imperative Prozessmodellabbildung im Vergleich zur deklarativen Prozessmodellabbildung als grobgranular einzuordnen [GHV07].

Die Abbildung der Granularität ist zusammengefasst im deklarativen Prozessmodell eindeutig besser abbildbar, da die imperative Prozessmodellabbildung die Integration mehrere Hierarchieebenen innerhalb eines Prozessmodells und somit eine fein granulare Abbildung ermöglicht.

### 6.1.5 Flexibilität

Die Flexibilität von Prozessmodellen beschreibt die Reaktion des Prozessmodells auf mögliche Änderungen der Rahmenbedingungen im Umfeld einer Prozessinstanz. Speziell wird betrachtet, wie sich der Prozess innerhalb einer Ausnahmesituation, beispielsweise beim Auftreten von Fehlern oder unvorhersehbaren Ereignissen, verhält [SAMS01]. In solchen Situationen ist eine flexible Reaktion des Prozesses gefordert.

In einer dynamischen Umgebung ergibt die imperative Prozessmodellabbildung großen Anzahl an Prozessvarianten, die eine sehr komplexe und umfangreiche Abbildung zur Folge haben und somit nur schwer abgebildet werden können [VGHJZ13]. Grund dafür ist die strenge Vorgabe der Aktivitätsabfolgen. Der deklarative Ansatz hingegen erlaubt auf Grund seiner Flexibilität bei der Abfolge von Aktivitäten eine große Anzahl an Ausführungsmöglichkeiten in nur einem Modell [Brü13].

Dazu ist die Darstellung von Prozessmodellen bei denen die Reihenfolge der Aktivitäten nicht relevant ist im imperativen Ansatz problematisch. Die sequentielle Abfolge der Aktivitäten

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

steht hier im Vordergrund und wird durch Pfade repräsentiert. Deklarativ lassen sich Prozesse mit einer losen Reihenfolge von Aktivitäten ohne Schwierigkeiten realisieren. Damit sind deklarative Prozessmodelle vor allem für Prozesse mit relativ losen Verfahrensvorgaben und ohne explizit vorgesehene Reihenfolge geeignet. Ein Beispiel hierfür ist das Interleaved Parallel Routing (siehe Kapitel 5.1.2). Auch hier ist Flexibilität bei der Abfolge der Aktivitäten gefragt, die sich deklarativ sehr gut abbilden lässt

Die Wartbarkeit und das Realisieren von Änderungen ist bei beiden Modellierungsansätzen sehr zeitaufwändig und schwierig, da diese das komplette Modell beeinflussen können [FMRWWZ10, PWZPMR12]. Aufgrund der komplexen Zusammenhänge von Aktivitäten die Änderung eines imperativen oder deklarativen Prozessmodells zur weiterhin korrekten Ausführung mit hohem Aufwand verbunden. Die deklarative Prozessmodellabbildung bietet allerdings die Möglichkeit der Laufzeit-Flexibilität [GV07]. Diese Funktion ermöglicht es, den Prozess während der Laufzeit zu beeinflussen und auf die Richtigkeit zu überprüfen.

Die Verwendung von Ausführungseinheiten in der deklarativen Prozessmodellierung ermöglichen eine Reduktion der Aktivitäten auf ein Minimum und trägt zu einer hohen Flexibilität des Prozessmodells bei.

Zusammenfassend kann festgehalten werden, dass die deklarative Prozessmodellabbildung im Gegensatz zur imperativen Abbildung ein höheres Maß an Flexibilität aufweist. Dies zeigt sich in der Möglichkeit, eine große Anzahl von Prozessvarianten innerhalb einer Prozessmodells abzubilden sowie der Möglichkeit der Abbildung einer losen Abfolge von Aktivitäten. Auch die Laufzeit-Flexibilität und die Möglichkeit, die Ausführungseinheiten eines Aktivität anzugeben tragen zur Flexibilität des deklarativen Prozessmodell bei

### **6.2 Gesamtbewertung**

Zusammenfassend hält diese Arbeit fest, dass sich die imperative und die deklarative Prozessmodellabbildung in vielen Bereichen deutlich unterscheiden. In Tabelle 2 sind die im Laufe dieser Arbeit erarbeiteten Unterschiede zusammengefasst.

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

	<b>imperativ</b>	<b>deklarativ</b>
<b>Ansatz</b>	inside-to-outside	outside-to-inside
<b>Integration von Kontextinformation</b>	ja	nein
<b>symbolhafte Abbildung</b>	ja	nein
<b>Verwendung von Start- und Endereignissen</b>	ja	teilweise
<b>Integration des zeitlichen Aspekts</b>	ja	nein
<b>Unterscheidung von Start und Endereignissen nach Art und Funktion</b>	ja	nein
<b>Integration von Subprozessen</b>	ja	nein
<b>Wartbarkeit</b>	zeitaufwändig	zeitaufwändig
<b>Information zum sequentiellen Ablauf des Prozessmodells</b>	leicht verständlich	schwer verständlich
<b>zeitgleiche Ausführung von Aktivitäten</b>	ja	nein
<b>flexible Reihenfolge von Aktivitäten</b>	nein	ja
<b>Abbildung der Entscheidungsregel bei Gateways</b>	ja	nein
<b>Angabe der Ausführungseinheiten bei Aktivitäten</b>	nein	ja
<b>Dynamik des Prozessmodells</b>	unflexibel	flexibel

Tabelle 2: Unterschiede der imperativen und deklarativen Prozessmodellabbildung

Die in Tabelle 2 dargestellten Unterschiede von imperativer und deklarativer Prozessmodellabbildung wurden im weiteren Verlauf der Arbeit in Bezug auf ihre Verständlichkeit, Vollständigkeit, Korrektheit, Granularität und Flexibilität analysiert. Eine Übersicht über die Ergebnisse dieser Analyse sind in Tabelle 3 dargestellt. Es wurde festgestellt, dass die imperative Prozessmodellierung im Vergleich zur deklarativen Prozessmodellierung Prozessmodelle eindeutig verständlicher, vollständiger und korrekter abbildet. Die deklarativen Prozessmodellabbildung weist dagegen im Vergleich zur imperativen Prozessmodellabbildung ein hohes Maß an Granularität und Flexibilität auf.

Die in Tabelle 3 abgebildeten Eigenschaften haben zur Folge, dass sich die imperative und die deklarativen Prozessmodellierung für unterschiedliche Arten von Prozessmodellen eignen, die im Folgenden charakterisiert werden.

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

	<b>imperativ</b>	<b>deklarativ</b>
<b>Verständlichkeit</b>	hoch	gering
<b>Vollständigkeit</b>	hoch	gering
<b>Korrektheit</b>	hoch	gering
<b>Granularität</b>	gering	hoch
<b>Flexibilität</b>	gering	hoch

Tabelle 3: Vergleich der imperativen und deklarativen Prozessmodellabbildung

Die imperative Prozessmodellabbildung ist für sich rasch ändernde und schnelllebige Geschäftsprozesse eine ineffiziente Lösung, da die dynamische Umgebung eine große Anzahl verschiedener Prozessvarianten zur Folge hat [VGHZ13]. Sie eignet sich vor allem für Prozesse mit einer klar definierten sequentiellen Reihenfolge von Aktivitäten, die eine strenge Verfahrensvorgabe, aber wenig Flexibilität erfordern. Wird innerhalb des Prozessmodells Wert auf die Bereitstellung von Kontext- und Hintergrundinformationen wie Rollen, Artefakten, Kommunikationsereignisse o.ä. gelegt, so bietet die imperative Prozessmodellierung eine übersichtliche Lösung. Imperative Prozessmodelle eignen sich ebenfalls zur Abbildung von Modellen, die für den Nutzer leicht lesbar und verständlich sein sollen. Imperative Prozessmodelle sind z.B. für Workflows im IT-Bereich geeignet, die als Vorlage zur Erstellung eines Programms oder einer Software dienen.

Die deklarative Prozessmodellierung ist ideal zur Abbildung von sehr flexiblen und dynamischen Prozessmodellen, die eine große Anzahl von Varianten enthalten. Es ist außerdem möglich, Aktivitäten ohne festgelegte Reihenfolge darzustellen. Dabei wird auf strenge Verfahrensvorgaben verzichtet. Des Weiteren sind deklarative Prozessmodelle für die Abbildung von Prozessen geeignet, die keine oder wenig Kontext- und Hintergrundinformationen benötigen. Die deklarative Prozessmodellabbildung ermöglicht eine sofortige Überprüfung der Ausführbarkeit durch die Laufzeit-Flexibilität. Deklarative Prozessmodelle sind beispielsweise für flexible Workflows, wie z.B. im Gesundheits- oder Bildungswesen, gut geeignet [Mal09].

Die für diese Arbeit verwendeten Prozessmodelle stellen den Ablauf von Prozessen innerhalb des Hochschuldienstportals der Universität Ulm dar. Diese Prozesse haben einen klar definierten Ablauf von Aktivitäten, die sich kaum ändern und somit wenig Dynamik und Fle-

## 6. Kritische Betrachtung der umgesetzten Prozessmodelle

xibilität der Prozessmodellierung erfordern. Damit ist die imperative Prozessmodellierung gut für diese Prozesse geeignet. Die Prozesse sind weiterhin stark auf die Angabe von Kontextinformationen angewiesen, was ebenfalls die Verwendung der imperativen Prozessmodellierung erfordert. Die Kontextinformationen wurden in Form von Rollenzuweisungen, zeitlichen Aspekten wie Fristen (die für den Hochschulbetrieb unerlässlich sind) und Kommunikationsereignisse in die Prozessmodelle integriert. Da diese Modelle hauptsächlich zur Erstellung eines neuen Hochschuldienstportals genutzt werden, ist hier auch eindeutig die imperative Prozessmodellabbildung geeignet. Dies gilt auch insofern, da die Prozessmodelle von weiteren Personen analysiert werden müssen und als Vorlage für eine spätere Implementierung dienen. Hierfür wird eine übersichtliche Darstellung und eine schnelle Erfassung des Prozesses benötigt, die ebenfalls durch die imperative Prozessmodellabbildung gegeben ist.

Zusammenfassend betrachtet ist der Großteil der hier verwendeten Prozessmodelle des Hochschuldienstportals durch die imperative Prozessmodellabbildung besser verständlich abgebildet als durch die deklarative Prozessmodellabbildung. Nur ein kleiner Teil dieser Prozessmodelle erfordert eine erhöhte Flexibilität und Granularität, die eine deklarative Prozessmodellabbildung rechtfertigen würde. Im Vergleich mit dem Nutzen durch eine imperative Modellabbildung kann dieser Teil aber vernachlässigt werden.

## **7. Zusammenfassung und Ausblick**

### **7.1 Fazit**

Das Ziel dieser Arbeit war es, anhand von realen Prozessbeispielen eine vergleichende Analyse zwischen der imperativen und der deklarativen Prozessmodellierung zu erarbeiten und die Ergebnisse zu diskutieren. Die Aufgabenstellung erforderte auch die kritische Betrachtung der Migration imperativer Prozessmodelle in ein deklaratives Schema. Ein weiteres Ziel war das Aufzeigen von Aspekten, die in der deklarativen Prozessmodellierung nicht bzw. nur unzureichend erfasst werden können. Für diese Schwierigkeiten der Erfassung sollten alternative Lösungen aufgezeigt werden.

Die Ergebnisse der Arbeit zeigen, dass sich die imperative und die deklarative Prozessmodellabbildung nicht nur in der Art der Modellierung von Grund auf unterscheiden, sondern auch in der Eignung der Anwendung für verschieden Arten von Prozessmodellen. Die hier verwendeten Prozessmodelle aus dem Kontext des Hochschuldiensteportals der Universität Ulm weisen in der deklarativen Prozessmodellabbildung eine höhere Flexibilität und Granularität auf. In der imperativen Prozessmodellierung lassen sich die Prozessmodelle im Vergleich verständlicher, vollständiger und korrekter abbilden. Um den Anforderungen an die Prozesse des Hochschuldiensteportals zu genügen, empfiehlt diese Arbeit die Prozessmodelle in dieser Form fortzuführen.

Die Migration der imperativen Prozessmodelle in ein deklaratives Schema ergab diverse Schwierigkeiten. Die erste Schwierigkeit bestand darin, einen ersten Ansatz zur Übertragung zu finden und einen korrekt ablaufenden Prozess zu erstellen. Weitere Probleme ergaben sich aus Limitierungen der deklarativen Prozessmodellierungssprache. Hier wurden alternative Lösungsvorschläge anhand der zur Verfügung stehenden Konstrukte erarbeitet. Beispielsweise lassen sich mit Workarounds viele Darstellungsprobleme, wie z.B. die Integration von Rollen oder Artefakten im deklarativen Ansatz umgehen. Aber auch mit diesen alternativen Lösungen bietet die deklarative Darstellung in den meisten Fällen eine im Vergleich zur imperativen Abbildung unverständlichere und unvollständigere Darstellung. Die Handlungsempfehlung, die Beispielprozessmodelle in imperativer Form abzubilden, wird durch die Analyse der Migration also bestätigt und verstärkt.



## 7. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde der Mangel an wissenschaftlicher Literatur im Forschungsfeld der vergleichenden Analyse der imperativen und deklarativen Prozessmodellabbildung bereits thematisiert. Der Grund hierfür ist das noch junge Alter der deklarativen Prozessmodellierung im Vergleich zur imperativen. Diese Arbeit liefert einen Forschungsbeitrag zur Erschließung dieses Gebietes. Vor allem zu Prozessen aus dem universitären Umfeld liefert diese Arbeit einen Erkenntnisgewinn. Zukünftig kann die Abwägung zur Verwendung imperativer und deklarativer Prozessmodellierung in diesem Umfeld anhand der Ergebnisse dieser Arbeit vereinfacht werden. Des Weiteren hat diese Arbeit durch die grundlegende vergleichende Analyse imperativer und deklarativer Prozessmodellierung das Forschungsfeld durch theoretische Erkenntnisse ergänzt.

### 7.2 Limitationen

Die Ergebnisse dieser Arbeit müssen mit einigen Einschränkungen betrachtet werden. Zunächst wurden für diese Arbeit ausschließlich Prozessmodelle verwendet, die aus einem Teamprojekt zur Abbildung des Hochschuldienstportals der Universität Ulm stammen. Die Ergebnisse dieser Arbeit treffen damit auch nur auf die zu Grunde liegenden Prozessmodelle zu und können nicht in eine allgemeingültige Aussage zur Prozessmodellierung übertragen werden. Des Weiteren sind die Ergebnisse dieser Arbeit auch nur für die hier verwendete Prozessmodellierungssprache Declare gültig. Andere Sprachen bieten andere Möglichkeiten der Darstellung und könnten somit zu abweichenden Ergebnissen führen.

Beim Vergleich der imperativen und deklarativen Prozessmodellierung wurde der zeitliche Aspekt bis auf die Frist vollständig außer Acht gelassen. Dazu wurde auf die Migration der Kommunikationsereignisse vom imperativen in das deklarative Schema verzichtet, um die Übersichtlichkeit der deklarativen Prozessmodelle soweit wie möglich zu erhalten. Deshalb stellt auch die Nichtbetrachtung der zeitlichen Aspekte und der Migration von Kommunikationsereignisse eine Einschränkung dieser Arbeit dar - die Ergebnisse der Arbeit sind nur für den betrachteten Bereich gültig.

### 7.3 Ausblick

Um eine verständlichere, vollständigere und korrektere Abbildung der deklarativen Prozessmodelle zu erreichen, wäre weitere Forschung im Bereich der Erweiterungen der Prozessmodellierungssprache Declare sinnvoll. So wäre beispielsweise eine Ergänzung zur Integration von Kontextinformationen empfehlenswert.

Eine weitere Möglichkeit zu weiterer Forschung auf Grundlage dieser Arbeit wäre die Migration imperativer Prozessmodelle in ein deklaratives Schema auf Basis einer anderen Prozessmodellierungssprache, wie z.B. ConDec, DecSerFlow oder EmBrace. Damit könnte die Frage beantwortet werden, ob mit anderen Prozessmodellierungssprachen ähnliche Ergebnisse erzielt werden.

Außerdem bietet die Arbeit die Grundlage für die Validierung ihrer Ergebnisse. Nach Vorbild der Struktur dieser Arbeit könnte die Analyse sowie die Migration von imperativen und deklarativen Prozessmodellen für Prozesse aus anderen Bereichen, wie zum Beispiel für Geschäftsprozesse aus dem Unternehmensumfeld, verwendet werden. Dadurch könnte überprüft werden, ob die Ergebnisse der Arbeit sich auf andere Anwendungsfelder übertragen lassen.

Die Ergebnisse dieser Arbeit können aber auch praktisch genutzt werden. Dies trifft vor allem auf das Anwendungsfeld der Hochschulen und Universitäten zu. Zum einen ist die Universität Ulm in der Lage, mit Hilfe dieser Arbeit eine Entscheidung über die zukünftige Verwendung einer Prozessmodellierung für ihr Hochschuldienstportal zu treffen. Dazu können aber auch andere Institutionen im universitären Umfeld die Arbeit nutzen, um für ähnliche Prozesse Entscheidungen zur Verwendung einer Prozessmodellierung zu treffen. Dabei sollte darauf geachtet werden, dass die zu Grunde liegenden Prozesse ähnliche Anforderungen an Vollständigkeit (hoch), Verständlichkeit (hoch), Korrektheit (hoch), Flexibilität (niedrig) und Granularität (niedrig) stellen wie die Prozesse des Hochschuldienstportals.

## Abbildungsverzeichnis

Abbildung 1: Darstellung von expliziten Lösungswegen bei der imperativen Prozessmodellierung....	10
Abbildung 2: imperative Notationselement.....	11
Abbildung 3: imperativer Beispielprozess.....	12
Abbildung 4: Darstellung von erlaubten Lösungswegen bei der deklarativen Prozessmodellierung...	14
Abbildung 5: deklarativer Beispielprozess .....	17
Abbildung 6: TDMS.....	26
Abbildung 7: Testcase Editor der TDMS.....	27
Abbildung 8: Anzahl der Ausführungseinheiten von Aktivitäten .....	31
Abbildung 9: Interleaved Parallel Routing.....	32
Abbildung 10: Granularität im deklarativen Prozessmodell .....	34
Abbildung 11: Abbildung von Sequenzflüssen .....	37
Abbildung 12: Integration von Rollen im deklarativen Prozessmodell .....	39
Abbildung 13: Integration von Artefakten im deklarativen Prozessmodell .....	40
Abbildung 14: Integration von Nachrichtenergebnisse im deklarativen Prozessmodell .....	42
Abbildung 15: Integration zeitlicher Aspekte im deklarativen Prozessmodell: Die Frist .....	44
Abbildung 16: verschiedene Startpunkte im deklarativen Prozessmodell .....	47
Abbildung 17: Art der Endereignisse wird im deklarativen Prozessmodell nicht erfasst .....	49
Abbildung 18: Startpunkte ohne nachfolgende Aktivitäten im deklarativen Prozessmodell .....	51
Abbildung 19: Endpunkte ohne voran gehende Aktivitäten im deklarativen Prozessmodell .....	54
Abbildung 20: datenbasiertes exklusives Gateways im deklarativen Prozessmodell .....	55
Abbildung 21: paralleles Gateways im deklarativen Prozessmodell.....	57
Abbildung 22: viele aufeinander folgenden Gateways ohne Aktivitäten im deklarativen Prozessmodell .....	59
Abbildung 23: Schleifen im deklarativen Prozessmodell .....	60
Abbildung 24: Subprozesse im deklarativen Prozessmodell.....	62
Abbildung 25: Beispiel 1, imperatives Prozessmodell.....	66
Abbildung 26: Beispiel 1, deklaratives Prozessmodell .....	66
Abbildung 27: Beispiel 2, imperatives Prozessmodell.....	70
Abbildung 28: Beispiel 2, deklaratives Prozessmodell 1 .....	71
Abbildung 29: Beispiel 2, deklaratives Prozessmodell 2 .....	71
Abbildung 30: Beispiel 3, imperatives Prosezmodell.....	73
Abbildung 31: Beispiel 3, deklaratives Prozessmodell .....	74
Abbildung 32: Beschriftung einer Aktivität im deklarativen Prozessmodell.....	78
Abbildung 33: Sequenzfluss im deklarativen Prozessmodell .....	80

## **Tabellenverzeichnis**

Tabelle 1: deklarative Notationselemente.....	16
Tabelle 2: Unterschiede der imperativen und deklarativen Prozessmodellabbildung .....	85
Tabelle 3: Vergleich der imperativen und deklarativen Prozessmodellabbildung.....	86

## Abkürzungsverzeichnis

BPM	Business Process Management
BPMN	Business Process Model and Notation
bzgl.	bezüglich
bzw.	beziehungsweise
et al	et alia
etc.	et cetera
IT	Informations Systeme
o.ä.	oder ähnliches
sog.	sogenannte/r/n
TDMS	Test Driven Modelling Suite
TDM	Test Driven Modelling
WFM	Workflow Management
WFMS	Work Flow Management System
z.B.	zum Beispiel

## Literaturverzeichnis

- [ADO00] W. M. P. van der Aalst, J. Desel, A. Oberweis: Business Process Management Models, In: Techniques and Empirical Studies. Springer-Verlag, 2000
- [Agu04] R. S. Aguilar-Savén: Business process modelling: Review and framework. International Journal of Production Economics, Volume 90, 2004
- [AHW03] W. M. P. van der Aalst, A.H.M. Ter Hofstede, M. Weske: Business Process Management: A survey. In: Business Process Management, Springer-Verlag, 2003
- [ALA] Alaska Simulatoer, [http://bpm.q-e.at/?page\\_id=669](http://bpm.q-e.at/?page_id=669), zuletzt aufgerufen am 04.11.2014
- [AP06] W. M. P. van der Aalst, M. Pesic: DecSerFlow: Towards a truly declarative service flow language. In: Web Services and Formal Methods, Springer-Verlag, 2006
- [APS09] W. M. P. van der Aalst, M. Pesic, H. Schonenberg: Declarative workflows: Balancing between flexibility and support. Computer Science + Research and Development, Volume 23, 2009
- [AWG05] W. M. P. van der Aalst, M. Weske, D. Grünbauer: Case handling: a new paradigm for business process support. Data & Knowledge Engineering, Volume 53, 2005
- [BLWRV 12] I. Barba, A. Lanz, B. Weber, M. Reichert, C. Del Valle: Optimized Time Management for Declarative Workflows. In: Enterprise, Business-Process and Information Systems Modeling, Springer-Verlag, 2012
- [BC08] R. Bartak, O. Cepek: Incremental Filtering algorithms for precedence and dependency constraints. International Journal of Artificial Intelligence Tools, Volume 17, 2008
- [Brü13] J. Brüning: Metamodellbasierte und hierarchieorientierte Workflowmodellierung. Dissertation, Universität Rostock, 2013
- [BV12] I. Barba, C. Del Vale: Filtering Rules for ConDec Templates Pseudocode and Complexity. Technischer Bericht, Universität Sevilla, Spanien, 2012
- [DTS] Declare Tool Set: <http://www.win.tue.nl/declare/>, zuletzt aufgerufen am: 01.10.2014
- [FLMRWWZ09] D. Fahland, D. Lübke, J. Mendngi, H. Reijers, B. Weber, M. Weidlich, S. Zugal: Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In: Enterprise, Business-Process and Information Systems Modeling, Springer-Verlag, 2009
- [FMRWWZ10] D. Fahland, J. Mendling, H. A. Reijers, B. Weber, M. Weidlich, S. Zugal: Declarative versus Imperative Process Modeling Languages: The Issue of Maintainability. In: Business Process Management Workshops, Springer-Verlag, 2010
- [GHV07] S. Goedertier, R. Haesen, J. Vanthienen: EM-BrA2CE v0.1: A vocabulary and execution model for declarative business process modeling. Katholieke Universiteit Leuven - Faculty of Economics and Applied Economics, 2007

## Literaturverzeichnis

- [GV07]** S. Goedertier, J. Vanthienen: Declarative Process Modeling with Business Vocabulary and Business Rules. In: *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, Springer-Verlag, 2007
- [Hal09]** A. J. Hallerbach: Management von Prozessvarianten. Dissertation, Universität Ulm, 2009
- [Ham90]** M. Hammer: Reengineering work: Don't automate. *Harvard Business Review*, 1990
- [HLSSDKZ99]** R. Hull, F. Llirbat, E. Siman, J. Su, G. Dong, B. Kumar, G. Zhou: Declarative Workflows that Support Easy Modification and Dynamic Browsing. *ACM SIGSOFT Software Engineering Notes*, Volume 24, 1999
- [HZSHRPW13]** C. Haisjackl, S. Zugal, P. Soer, I. Hadar, M. Reichert, J. Pinggera B. Weber: Making Sense of Declarative Process Models: Common Strategies and Typical Pitfalls?, In: *Enterprise, Business-Process and Information Systems Modeling*, Springer-Verlag, 2013
- [Igl11]** M. Igl: ESProNa Eine Constraintsprache zur multimodalen Prozessmodellierung und navigationsgestützten Ausführung. Dissertation, Universität Bayreuth, 2011
- [Lan08]** A. Lanz: Realisierung einer Zeitmanagementkomponente eines adaptiven Prozess-Management-Systems. Diplomarbeit, Universität Ulm, 2008
- [Lic12]** W. Lichtenegger: Methoden zur teilautomatischen Konstruktion von Ist-Prozessmodellen mittels Process-Mining sowie zur Integration manuell konstruierter und automatisch generierter Ist-Prozessmodelle. Logos Verlag Berlin GmbH, 2012
- [LWR14]** A. Lanz, B. Weber, M. Reichert: Time Patterns for Process-aware Information Systems. *Requirements Engineering*, Volume 19, 2014
- [Mag14]** F. M. Maggi: Discovering Metric Temporal Business Constraints from Event Logs. In: *Perspectives in Business Informatics Research*, Springer-Verlag, 2014
- [Mal09]** P. Malets: Modellierung von Einschränkungen auf Geschäftsprozesse. Diplomarbeit Universität Stuttgart, 2009
- [MRA10]** J. Mendling, H. A. Reijers, W.M.P. van der Aalst: Seven Process Modeling Guidelines (7PMG). *Information and Software Technology*, Volume 52, 2010
- [MW14]** F. M. Maggi, M. Westergaard: Using Timed Automata for a Priori Warnings and Planning for Timed Declarative Process Models. *International Journal of Cooperative Information Systems*, Volume 23, 2014
- [PA06]** M. Pesic, W. M. P. van der Aalst: A Declarative Approach for Flexible Business Processes Management. In: *Business Process Management Workshops*, Springer-Verlag, 2006
- [Pes08]** M. Pesic: Constraint-Based Workow Management Systems: Shifting Control to Users. Dissertation, Technische Universiteit Eindhoven, 2008
- [PSA07]** M. Pesic, H. M. Schonenberg, W. M. P. van der Aalst: DECLARE: Full Support for Loosely- Structured Processes. In: *Proceedings Enterprise Distributed Object Compting Conference*, 2007

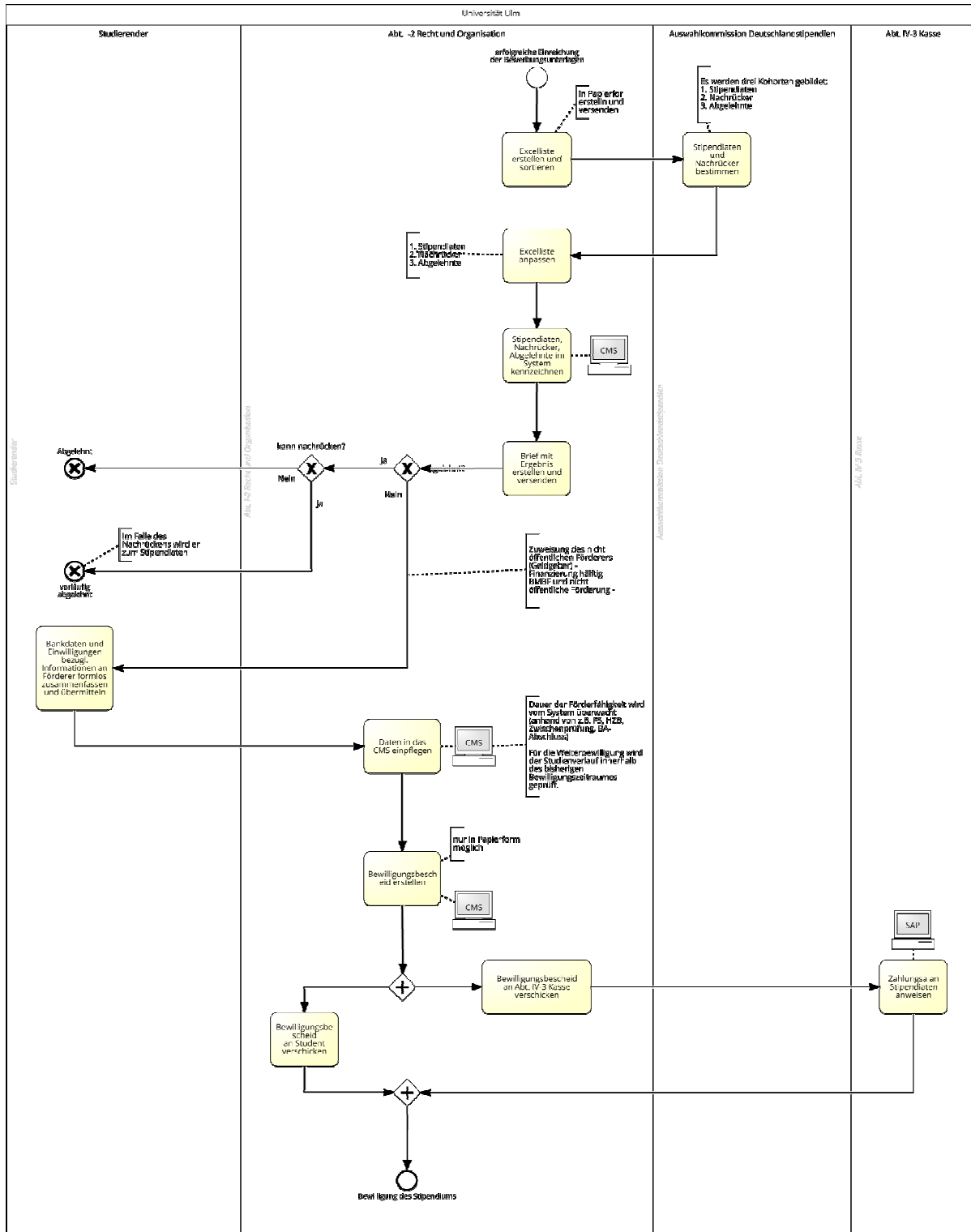
## Literaturverzeichnis

- [PSA09]** M. Pesic, H. M. Schonenberg, W. M. P. van der Aalst: DECLARE Demo: A Constraint based Workflow Management System. In: Proceedings of the BPM2009 Demonstration Track, 2009
- [PSSA07]** M. Pesic, M.H. Schonenberg, N. Sidorova, W. M. P. van der Aalst: Constraint-Based Workflow Models: Change Made Easy. In: On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, Springer-Verlag, 2007
- [PWZPMR12]** P. Pichler, B. Weber, S. Zugal, J. Pinggera, J. Mendling, H. A. Reijers: Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. In: Business Process Management Workshops, Springer-Verlag, 2012
- [RH04]** P. V. Roy, S. Haridi: Concepts, Techniques and Models of Computer Programming. The MIT Press, 2004
- [RW12]** M. Reichert, B. Weber: Enabling Flexibility in Process-Aware Information Systems. Springer-Verlag, 2012
- [SAMS01]** S. Schwarz, A. Abecker, H. Maus, M. Sintek: Anforderungen an die Workflow Unterstützung für wissensintensive Geschäftsprozesse. In: Workshop "Geschäftsprozessorientiertes Wissensmanagement", 2001
- [TDMS]** Test Driven Modelling Suite: [http://bpm.q-e.at/stefan\\_zugal?page\\_id=343](http://bpm.q-e.at/stefan_zugal?page_id=343), zuletzt aufgerufen am 01.10.2014
- [VGHJZ13]** F. Vermeer, J. van Grondelle, E. de Haan, S. Jansen, M. Zoet: Transforming Existing Procedural Business Processes into a Constraint-Based Formalism. In: Proceedings of ACIS 2013. Sidney, Australia, 2013
- [WADH03]** P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede: Analysis of Web Services Composition Languages: The Case of BPEL4WS. In Conceptual Modeling - ER 2003, Springer-Verlag, 2003
- [Wes07]** M. Weske: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag, 2007
- [WZP11]** B. Weber, S. Zugal, J. Pinggera: The Impact of Testcases on the Maintainability of Declarative Process Models. In: Enterprise, Business-Process and Information Systems Modeling, Springer-Verlag, 2011
- [ZPW11]** S. Zugal, J. Pinggera, B. Weber: Creating Declarative Process Models Using Test Driven Modeling Suite. In: IS Olympics: Information Systems in a Diverse World, Springer-Verlag, 2011
- [ZSHPRW13]** S. Zugal, P. Soffer, C. Haisjackl, J. Pinggera, M. Reichert, B. Weber: Investigating expressiveness and understandability of hierarchy in declarative business process models. In: Software & Systems Modeling, Springer-Verlag, 2013
- [Zug11]** S. Zugal: Applying Cognitive Psychology for Improving the Creation, Understanding and Maintenance of Business Process Models. Dissertation, Universität Innsbruck, 2011

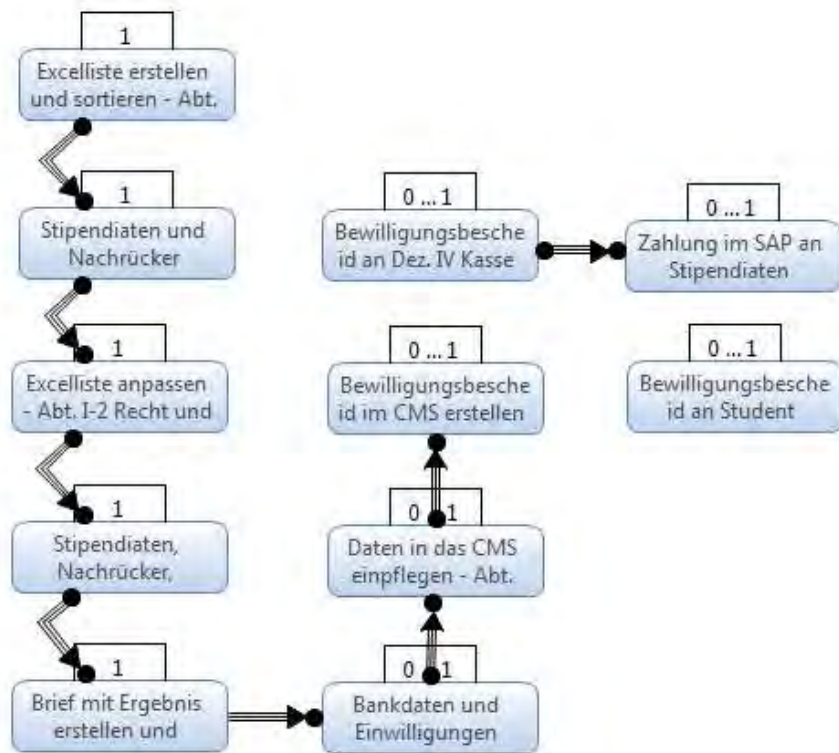


# Anhang

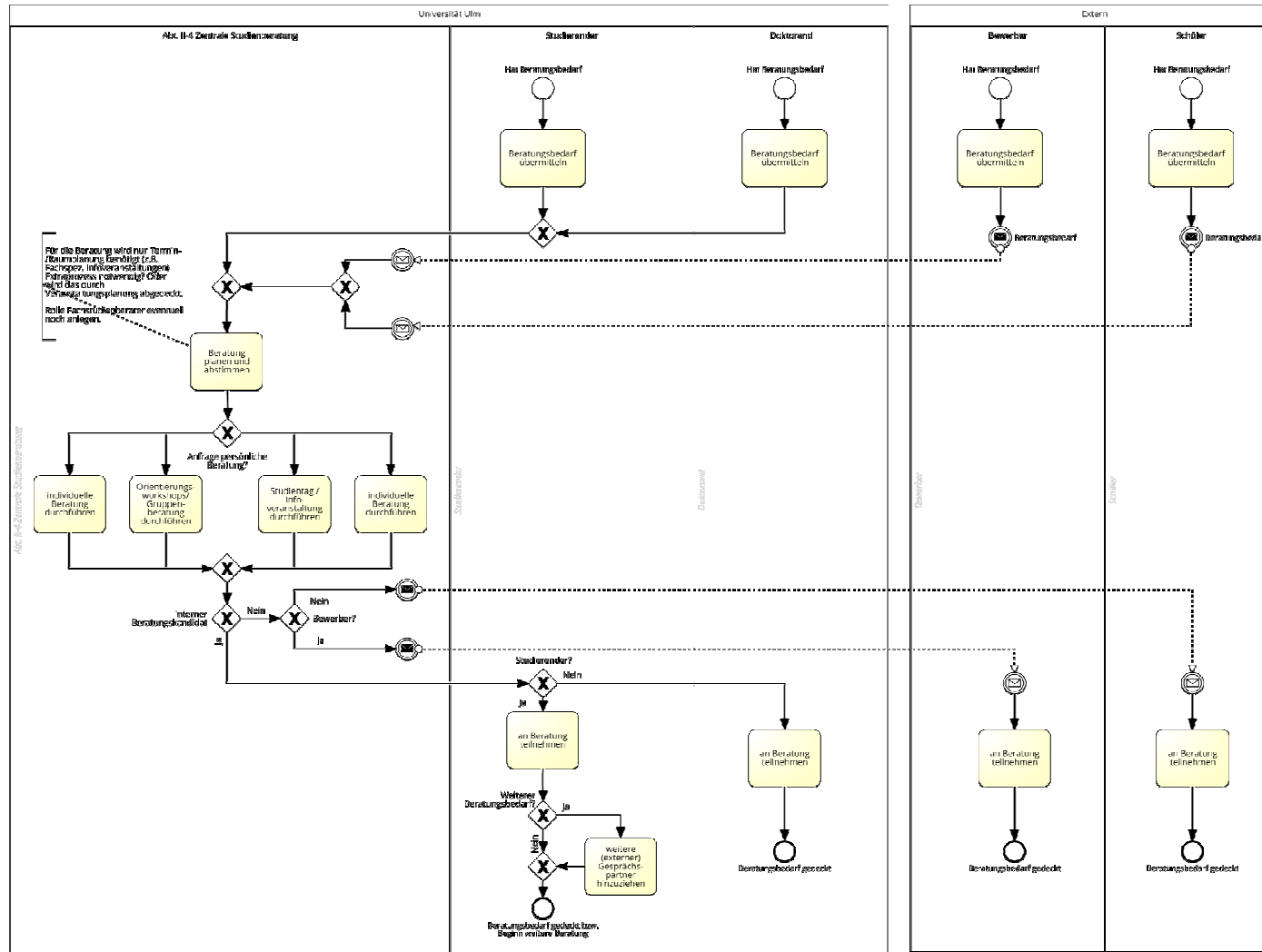
## Beispiel 1: Ranking und Zulassung zum Deutschlandstipendium - imperativ



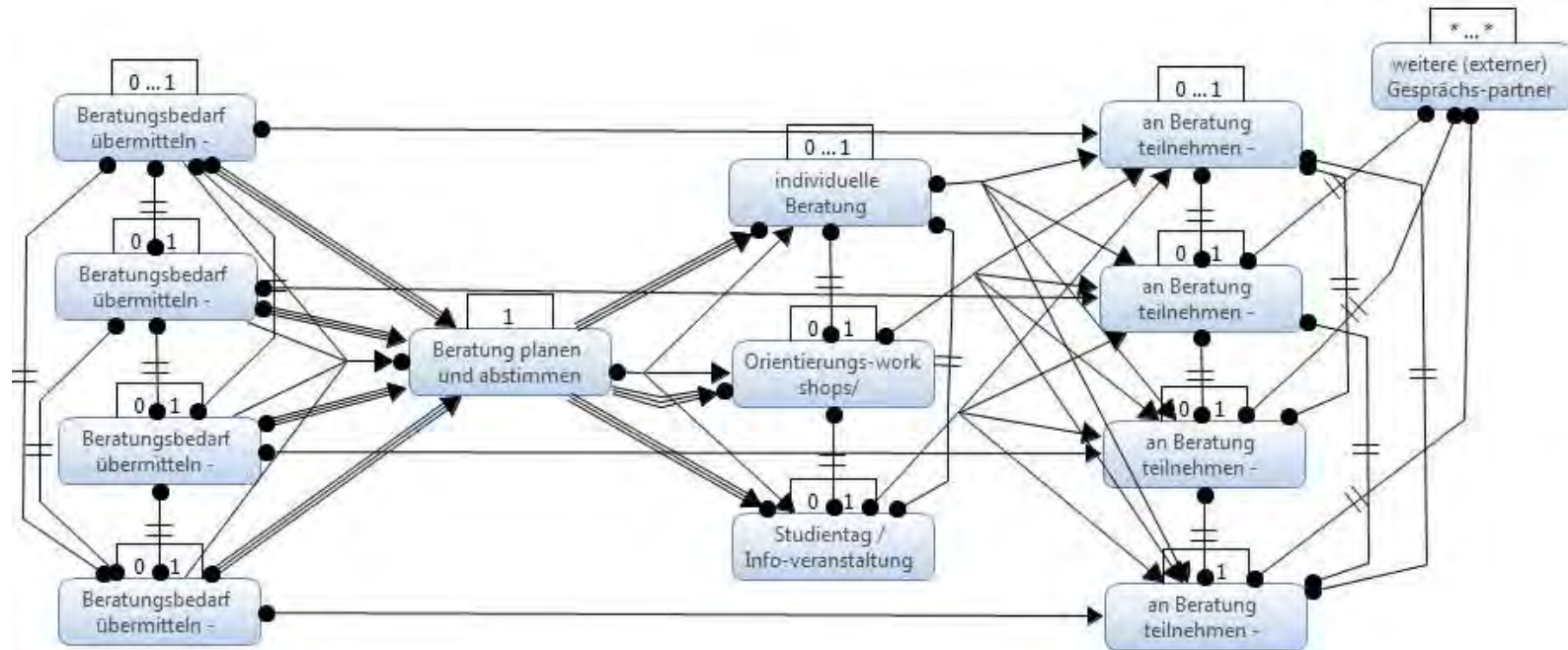
Beispiel 1: Ranking und Zulassung zum Deutschlandstipendium- deklarativ



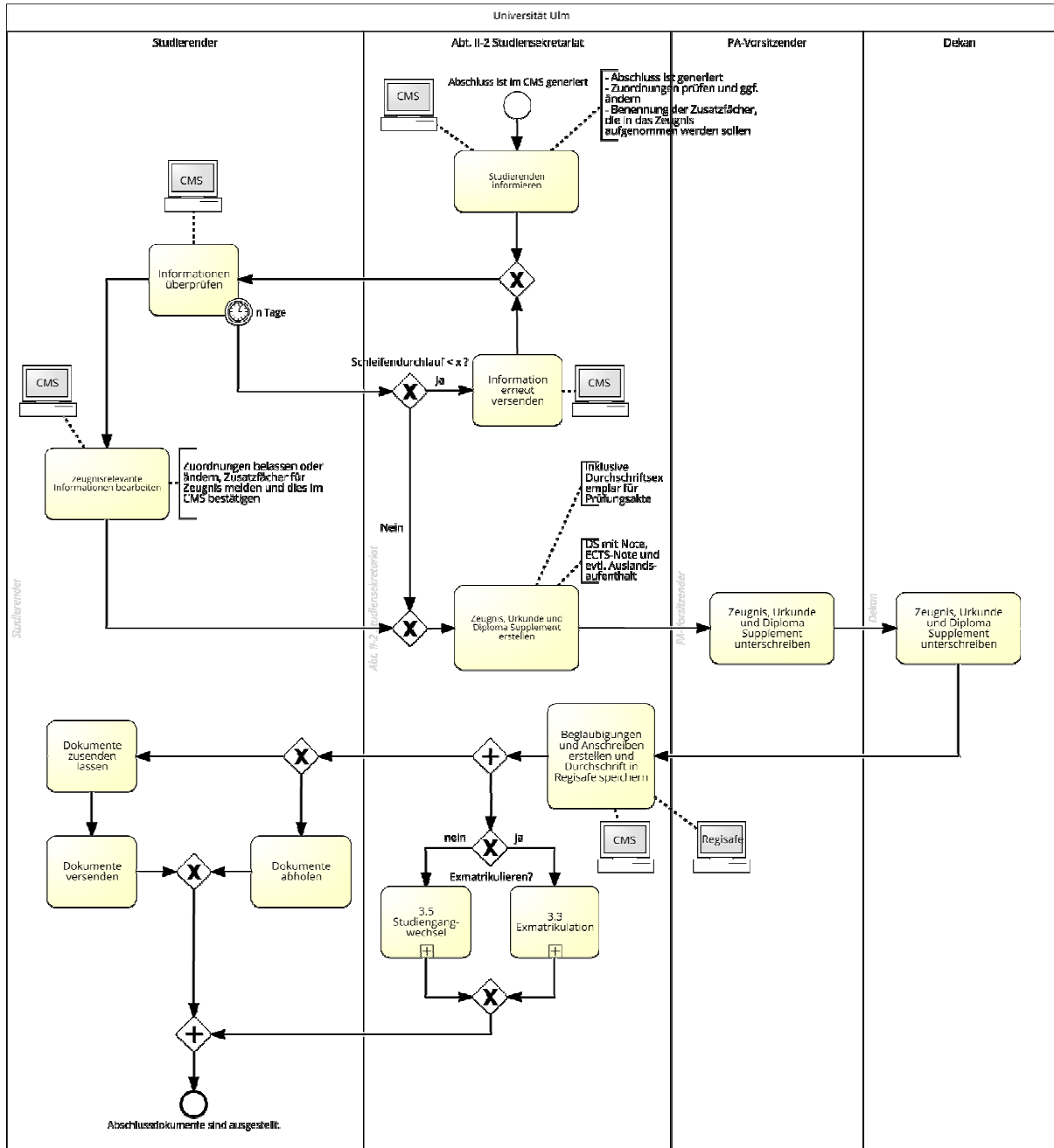
Beispiel 2: Beratung - imperativ



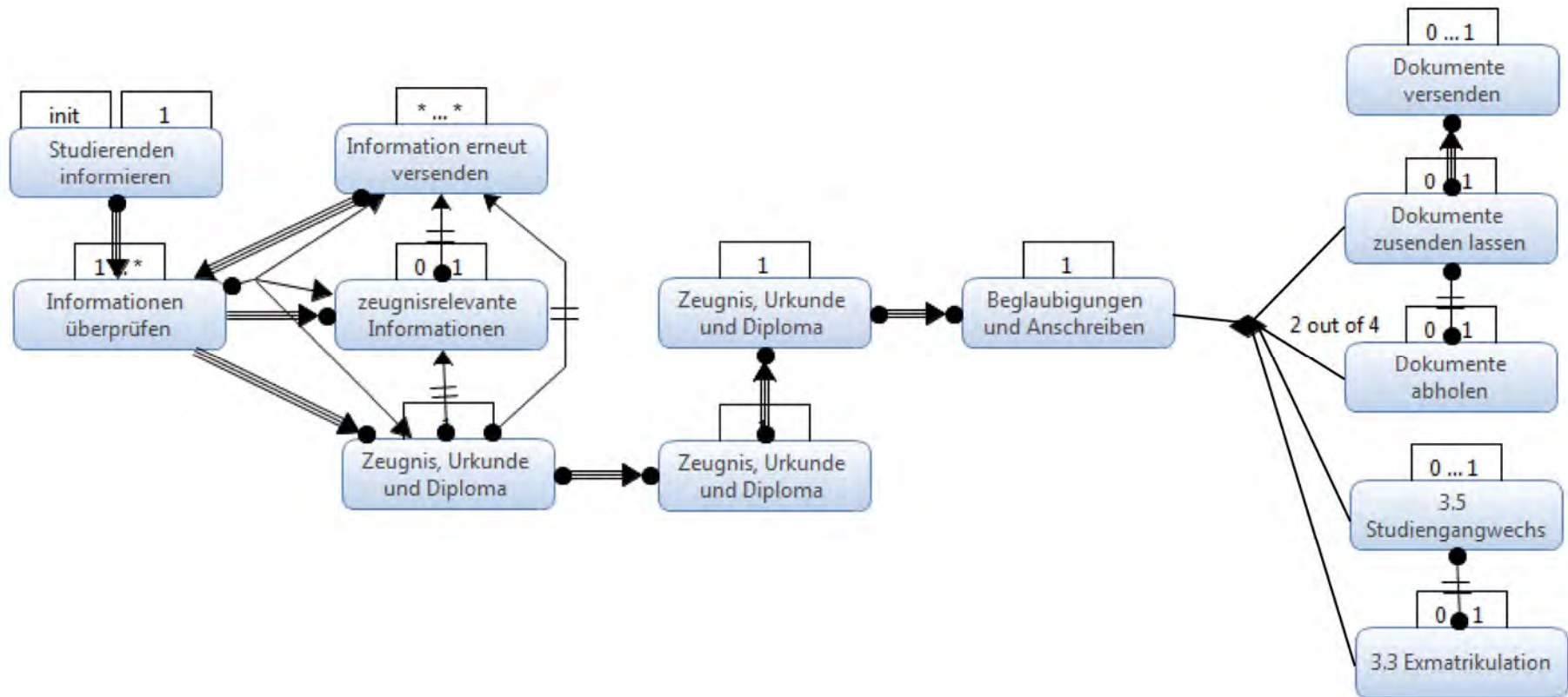
Beispiel 2 Beratung: - deklarativ



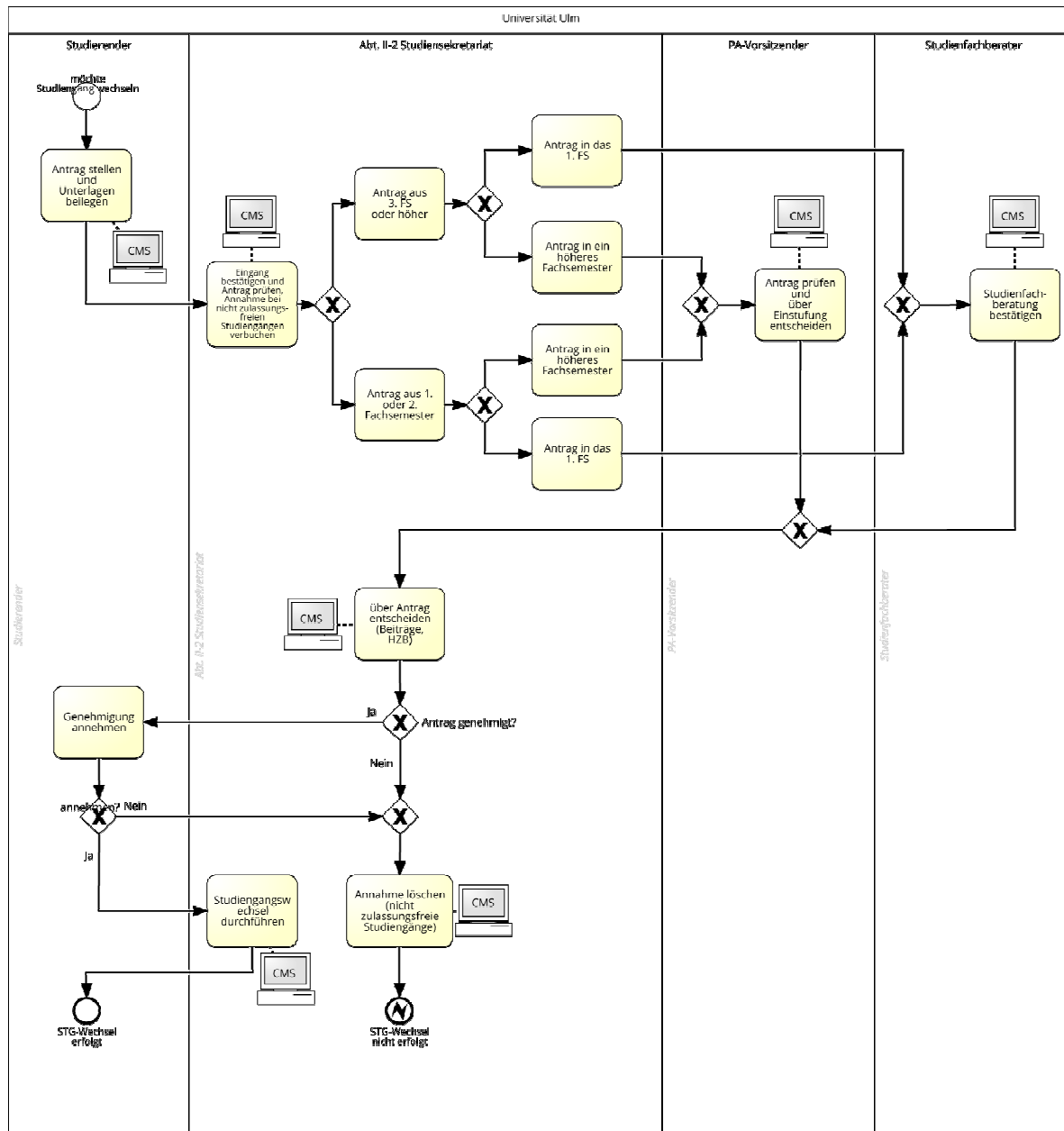
Beispiel 3: Abschlussdokumente ausstellen - imperativ



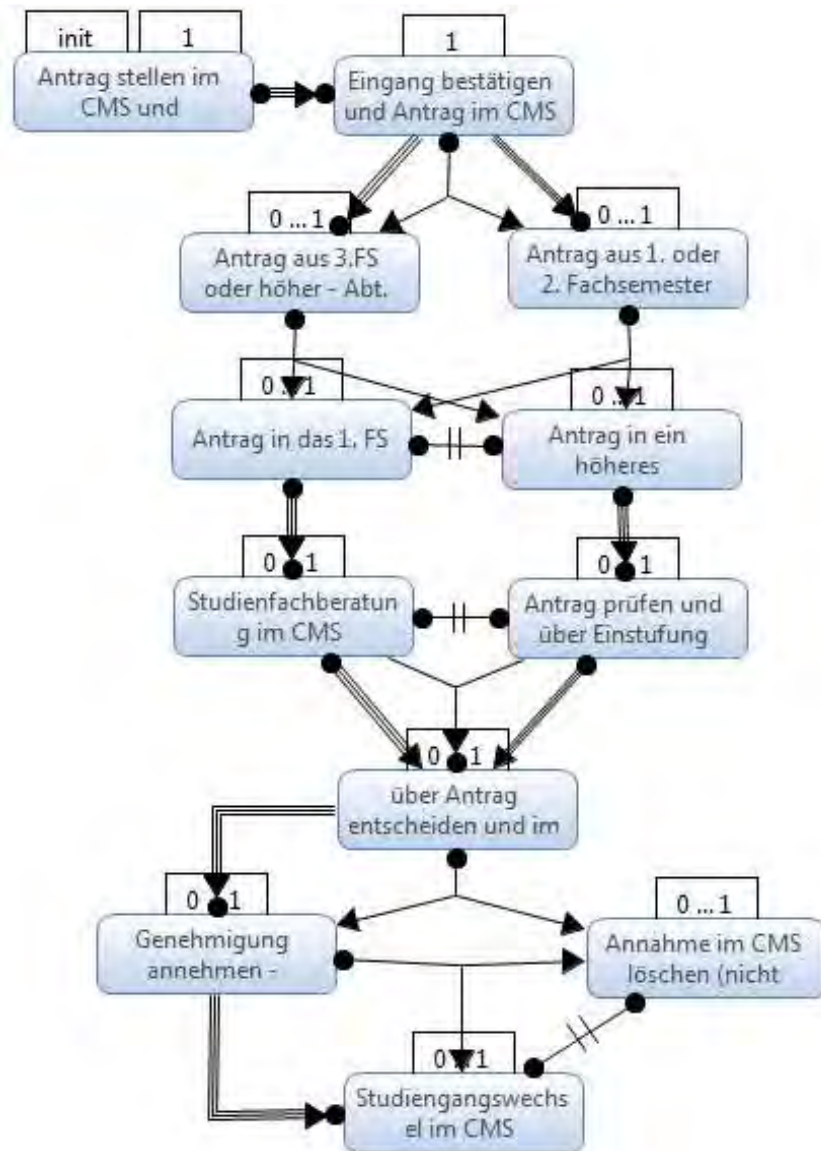
Beispiel 3: Abschlussdokumente ausstellen - deklarativ



Beispiel 4: Studiengangswechsel - imperativ

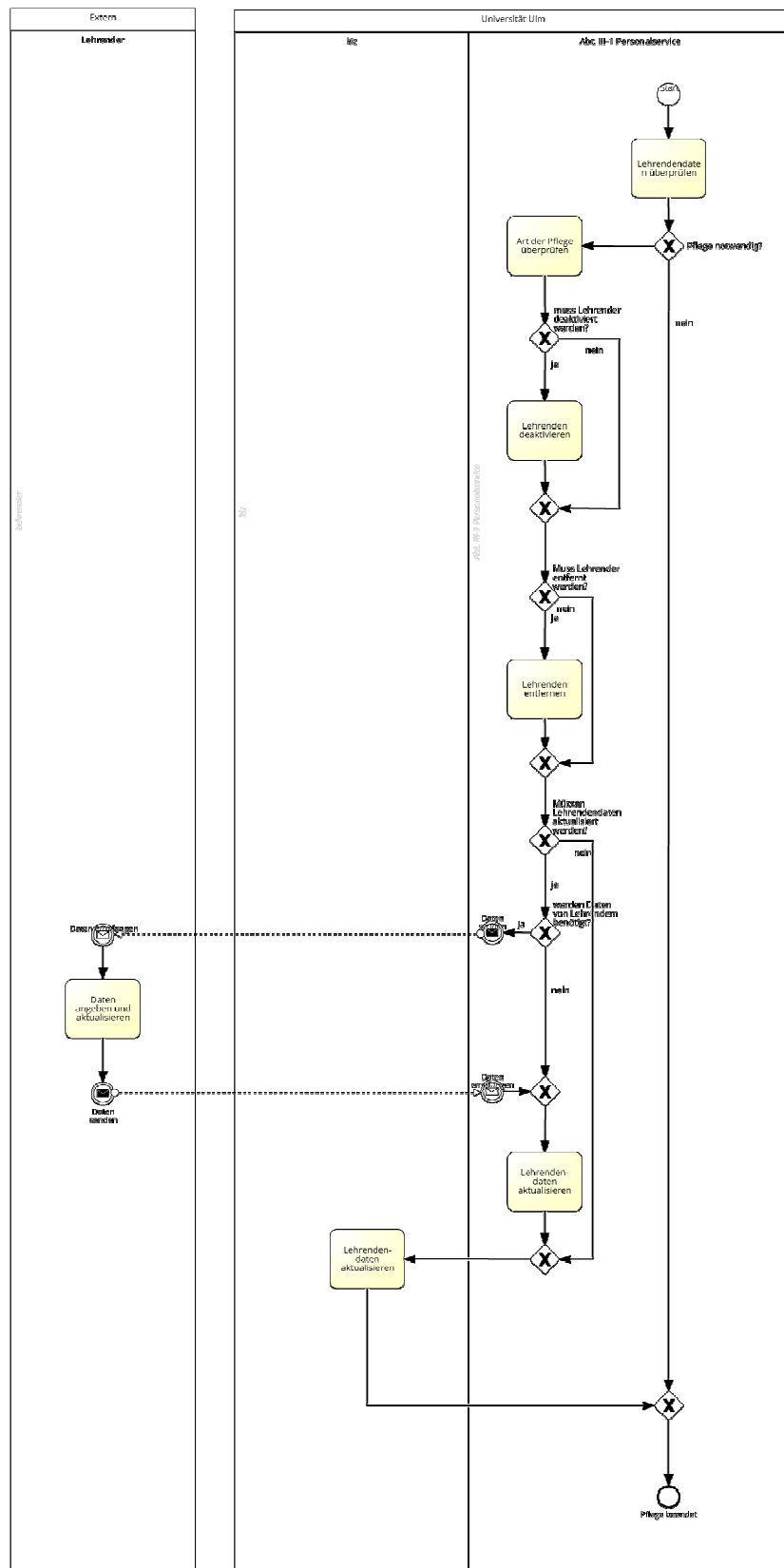


Beispiel 4: Studiengangswechsel - deklarativ

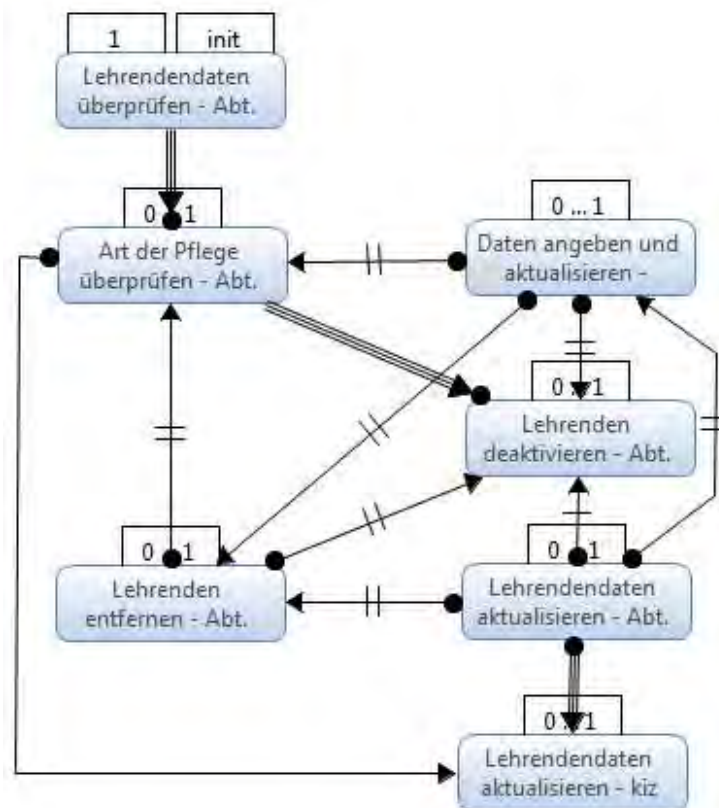




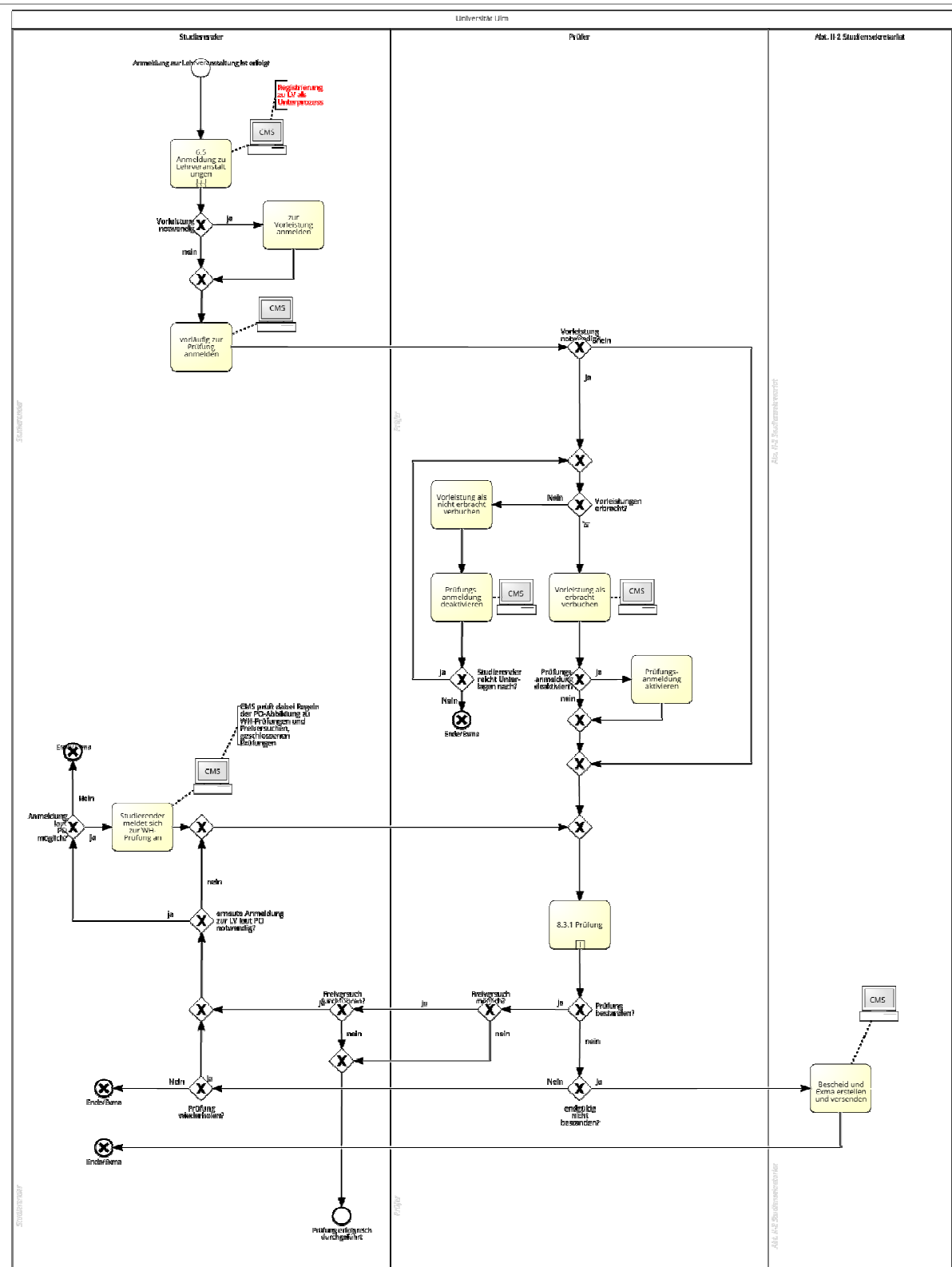
Beispiel 5: Pflege der Lehrendendaten - imperativ



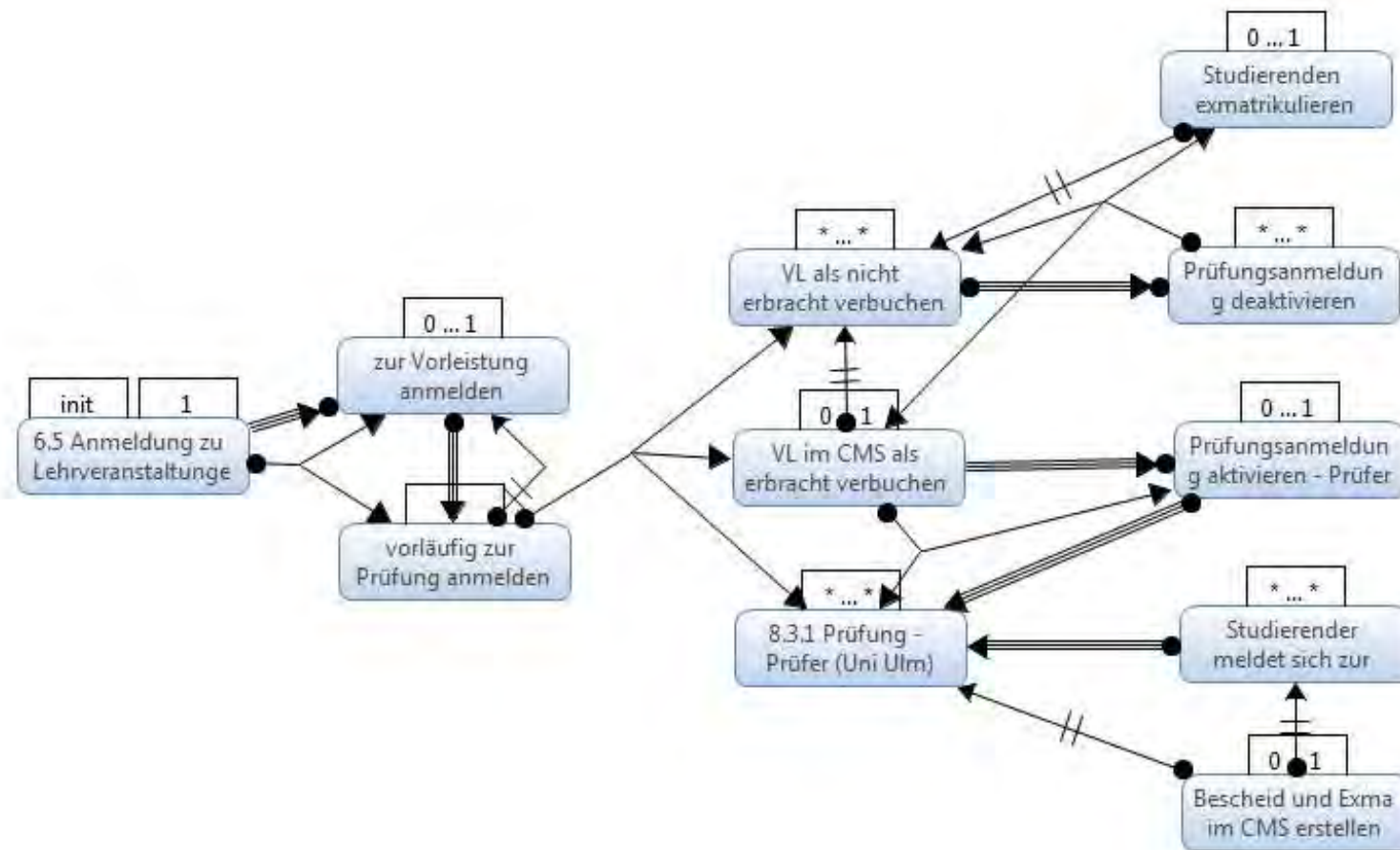
Beispiel 5: Pflege der Lehrendendaten- deklarativ



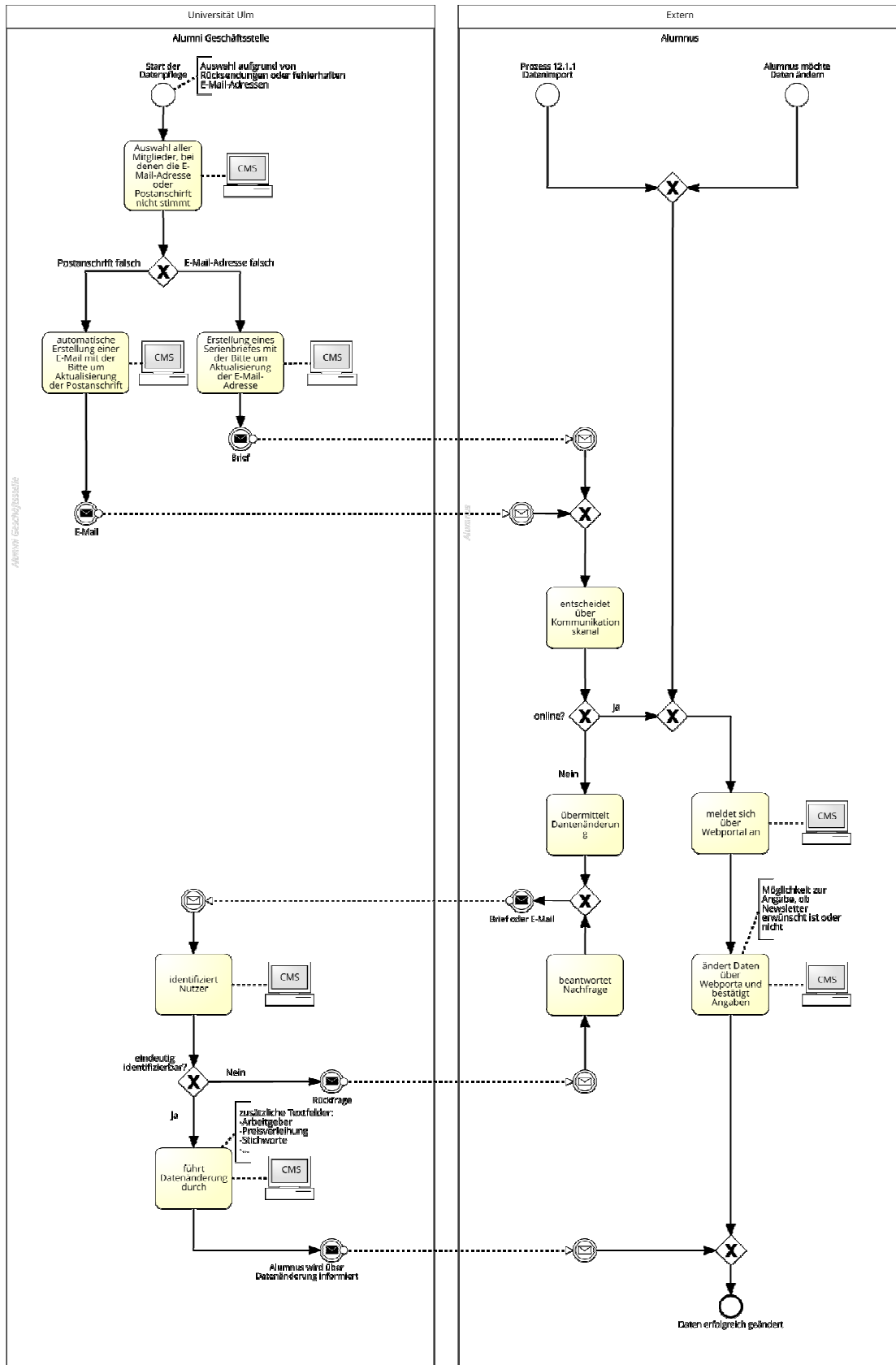
Beispiel 6: Durchführen von Prüfungen - imperativ



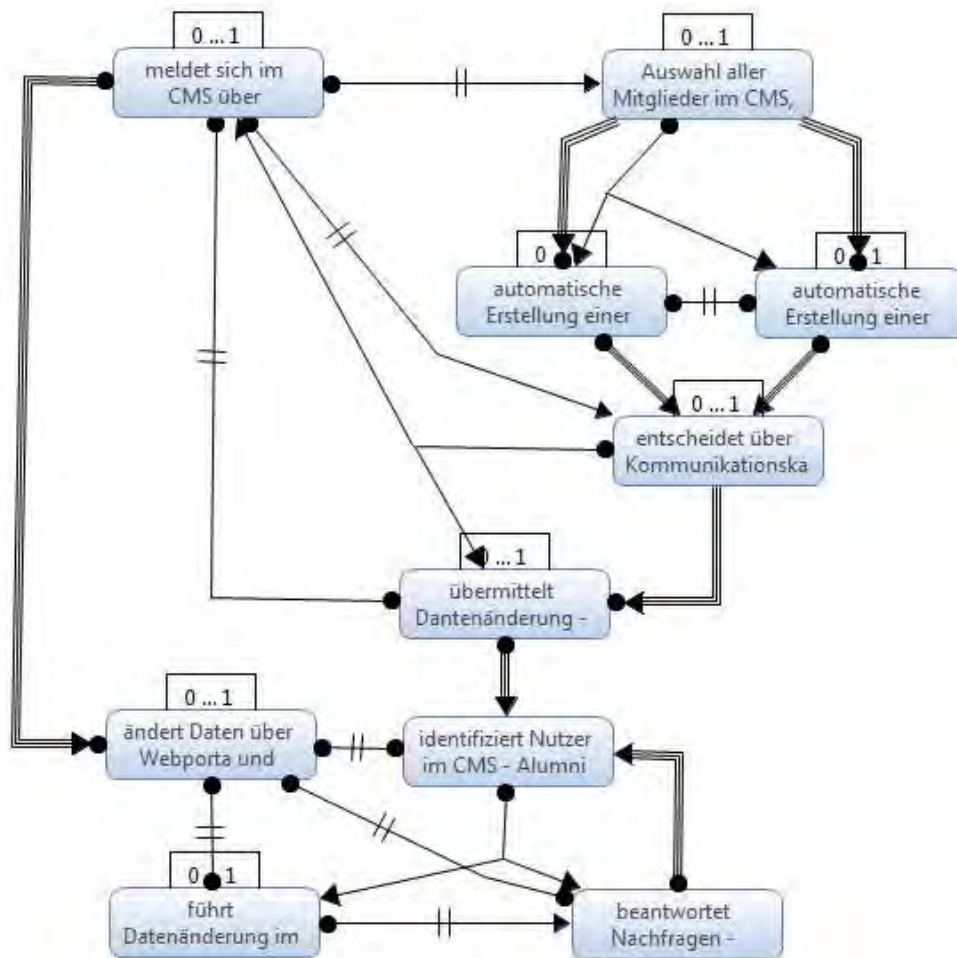
Beispiel 6: Durchführen von Prüfungen- deklarativ



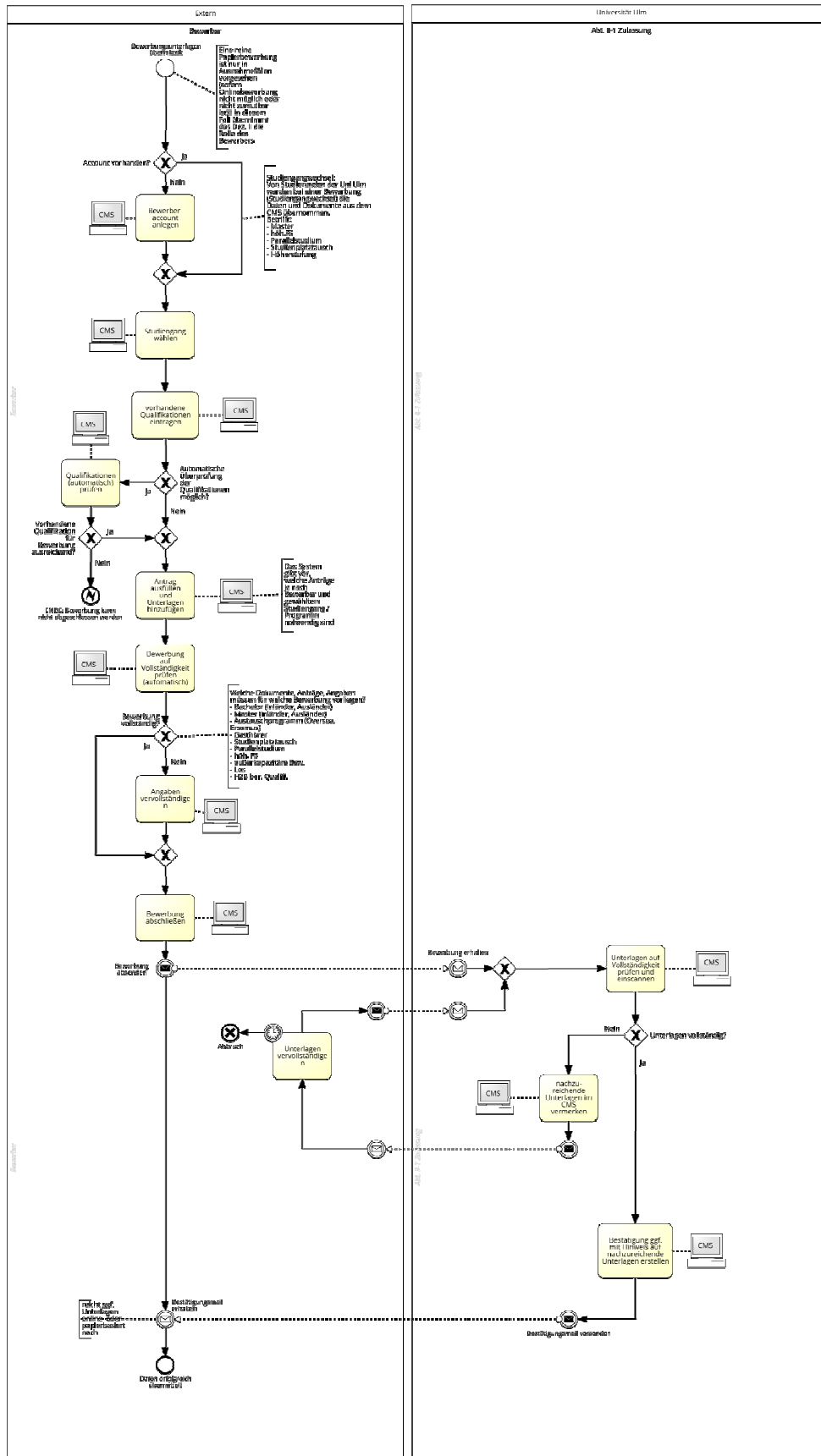
Beispiel 7 : Datenpflege - imperativ



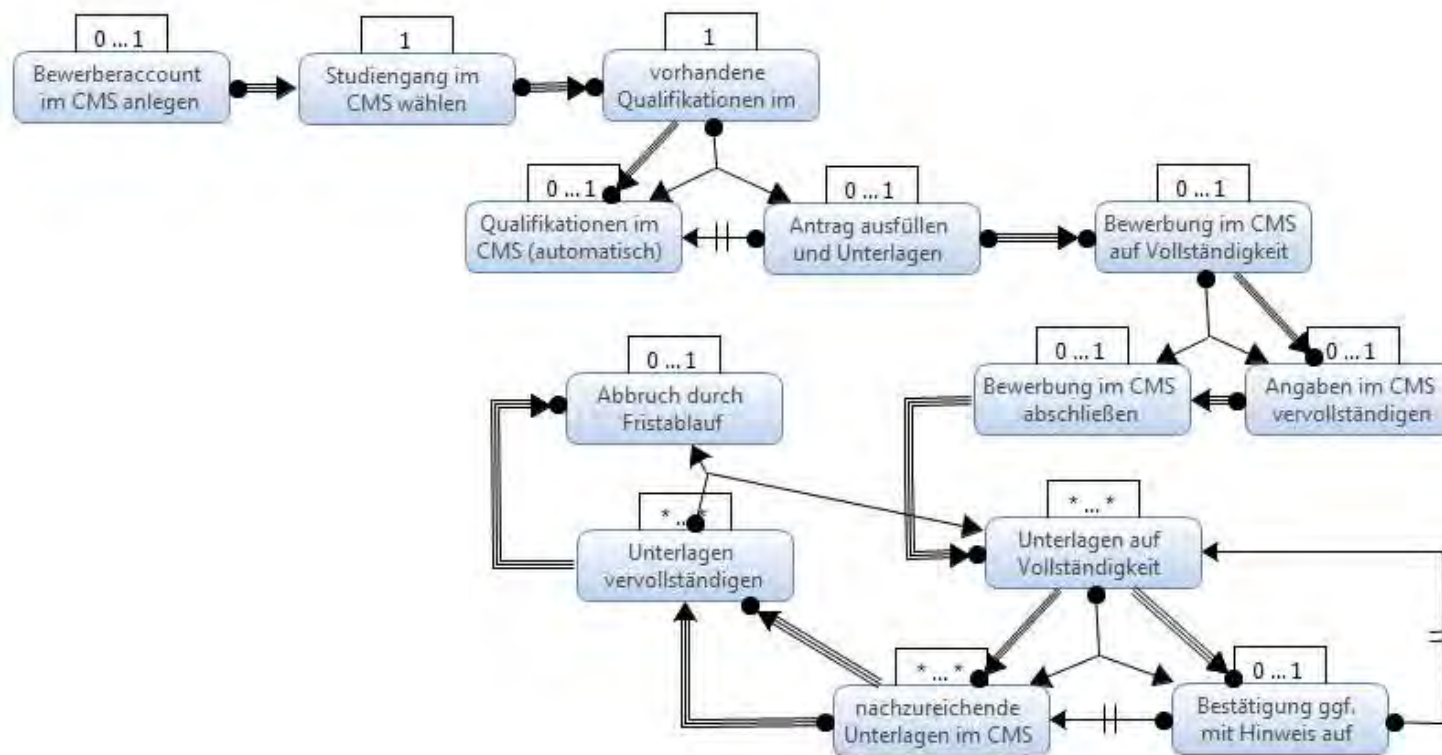
Beispiel 7 : Datenpflege - deklarativ



Beispiel 8: Datenübermittlung Bewerber - imperativ

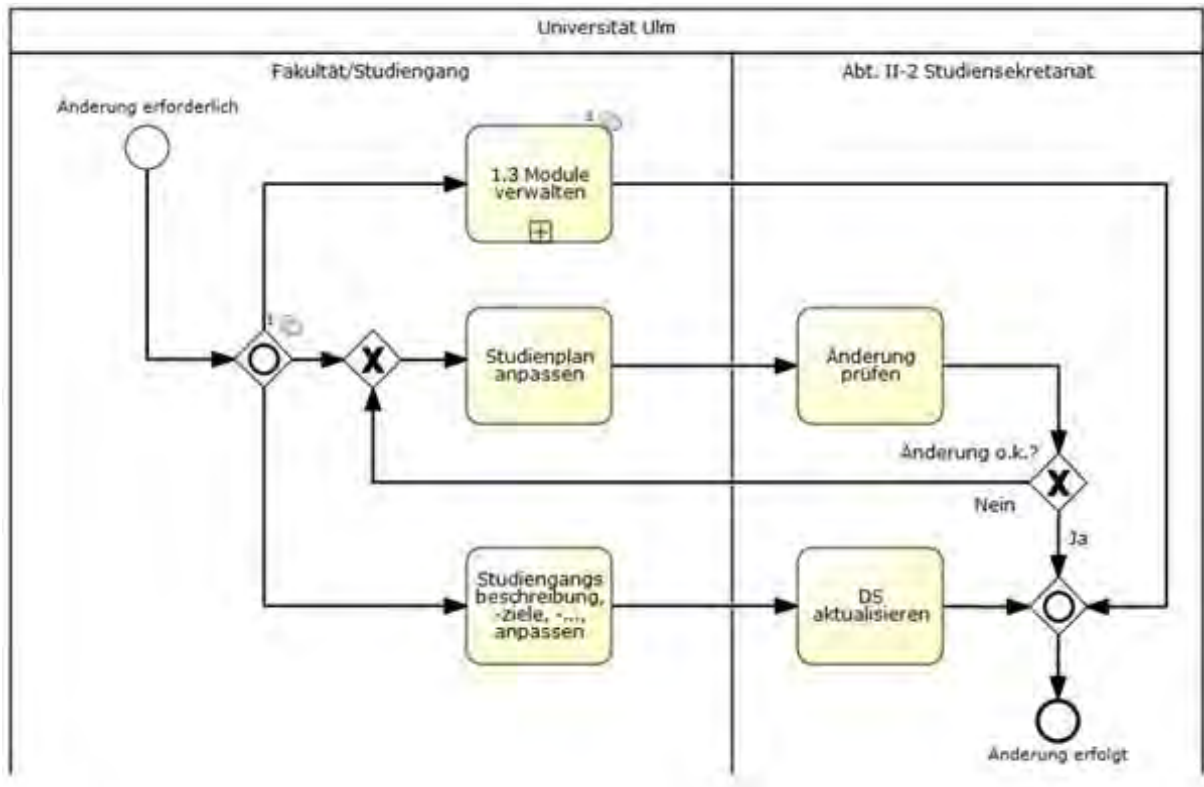


Beispiel 8: Datenübermittlung Bewerber - deklarativ

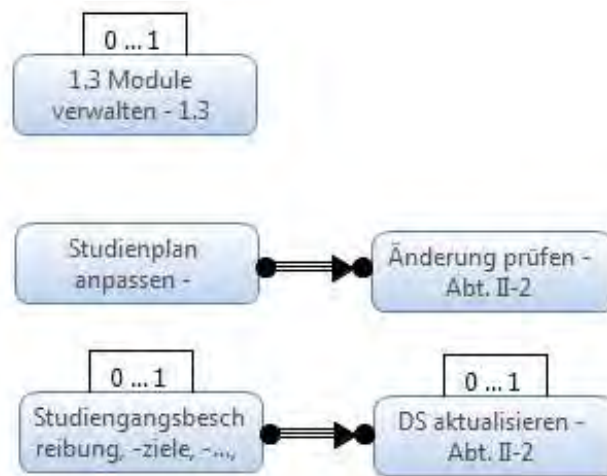




Beispiel 9: Modulhandbuch verwalten - imperativ



Beispiel 9: Modulhandbuch verwalten - deklarativ



Erklärung

Name: Julia Schwarz

Matrikelnummer: 806270

## **Erklärung**

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

Ulm, den 27.11.2014

---

Julia Schwarz

Dateiname: Masterarbeit\_23\_11  
Verzeichnis: C:\Users\Julia\Desktop  
Vorlage: C:\Users\Julia\AppData\Roaming\Microsoft\Templates\Normal.dot  
m  
Titel:  
Thema:  
Autor: Julia  
Stichwörter:  
Kommentar:  
Erstelldatum: 17.11.2014 09:05:00  
Änderung Nummer: 21  
Letztes Speicherdatum: 26.11.2014 21:40:00  
Zuletzt gespeichert von: Julia  
Letztes Druckdatum: 27.11.2014 13:32:00  
Nach letztem vollständigen Druck  
Anzahl Seiten: 115  
Anzahl Wörter: 22.613 (ca.)  
Anzahl Zeichen: 142.464 (ca.)